AUERBACH

BEST PRACTICES SERIES

Architectures for E-Business Systems

Building the Foundation for Tomorrow's Success

Editor

SANJIV PURBA

BEST PRACTICES SERIES

Architectures for E-Business Systems

Building the Foundation for Tomorrow's Success

BEST PRACTICES SERIES

Architectures for E-Business Systems

Building the Foundation for Tomorrow's Success

Editor

Sanjiv Purba

AUERBACH PUBLICATIONS

A CRC Press Company

Boca Raton London New York Washington, D.C.

Library of Congress Cataloging-in-Publication Data

Architectures for e-Business systems : building the foundation for tomorrow's success / edited by Sanjiv Purba.
p. cm. -- (Best practices series (Boca Raton, Fla.))
Includes bibliographical references and index.
ISBN 0-8493-1161-6 (alk. paper)
1. Electronic commerce. 2. Computer architecture. 3. Success in business. I. Purba, Sanjiv. II. Series

HF5548.32 .A73 2001
658.8'4--dc21 2001046416

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Direct all inquiries to CRC Press LLC, 2000 N.W. Corporate Blvd., Boca Raton, Florida 33431.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the Auerbach Web site at www.auerbach-publications.com

No claim to original U.S. Government works
International Standard Book Number 0-8493-1161-6
Library of Congress Card Number 2001046416
Printed in the United States of America 1 2 3 4 5 6 7 8 9 0
Printed on acid-free paper

Contributors

Ed Bailey, *Chairman, TN3270 Work Group, Internet Engineering Task Force, Research Triangle Park, NC*

John Care, *Director, Technical Services, Yardley, PA*

Jim Q. Chen, *Assistant Professor, St. Cloud State University, St. Cloud, MN*

Chao-Min Chiu, *Assistant Professor, Department of Information Management, National Kaohsiung First University, Yenchao, Kaohsiung, Taiwan*

Trevor Clarke, *Management Consultant, Deloitte Consulting, Toronto, Ontario, Canada*

Carlson Colomb, *Director of Marketing, Aviva Business Unit, Elcon Technology, Montreal, Quebec, Canada*

Douglas G. Conorich, *Global Solutions Manager, Managed Security Services, IBM, Fort Worth, TX*

Michelle Cook, *Founding Partner, Global Trade Solutions, Ottawa, Ontario, Canada*

Curtis Cook, *CITP, Global Trade Solutions, Ottawa, Ontario, Canada*

John Davies, *Vice President and Chief Information Officer, QLogitek, Toronto, Ontario, Canada*

Lawrence D. Dietz, *Director, Market Intelligence, Symantec Corporation, Cupertino, CA*

Charles Dow, *Vice President, Product Engineering — Banking Systems, SLMsoft.com Inc., Toronto, Ontario, Canada*

Sylvain Duford, *Chief Architect, Cactus Internet, Hull, Quebec, Canada*

Adam Fadlalla, *Assistant Professor, Department of Computer and Information Science, Cleveland State University, Cleveland, OH*

John Fiske, *Independent Writer, Prides Crossing, MA*

Chris Forsythe, *Senior Member, Statistics and Human Factors Technical Staff, Sandia National Laboratories, Albuquerque, NM*

Stephen Fried, *Senior Manager, Global Risk Assessment and Secure Business Solutions, Lucent Technologies, Warren, NJ*

Monica J. Garfield, *Doctoral Student, MIS, University of Georgia, Athens, GA*

James E. Gaskin, *Consultant, Mesquite, TX*

Ido Gileadi, *Senior Manager, Deloitte Consulting, Toronto, Ontario, Canada*

Donald Golden, *Assistant Professor, Department of Computer and Information Science, Cleveland State University, Cleveland, OH*

Anura Gurugé, *Consultant, Gurutech, Meredith, NH*

Linda G. Hayes, B.B.A., M.S., C.P.A., J.D., *Founder, President, and Chief Executive Officer, AutoTester, Inc., Dallas, TX*

Richard D. Heath, *President and Chief Executive Officer, Royal Oaks Information Systems, St. Cloud, MN*

Gilbert Held, *Director, 4-Degree Consulting, Macon, GA*

David K. Holthaus, *Software Specialist, Nationwide Insurance Enterprise, Columbus, OH*

Mark G. Irving, *CISA IS Auditor, Minnesota Power, Duluth, MN*

Paul J. Jalics, *Assistant Professor, Department of Computer and Information Science, Cleveland State University, Cleveland, OH*

Marie Karakanian, *Senior Manager, Deloitte Consulting, Toronto, Ontario, Canada*

Jim Kates, *Chief Technology Officer, Security Experts, Inc., Largo, FL*

Bryan T. Koch, CISSP, *Principal Security Architect, Guardent, Inc., St. Paul, MN*

Rahul Kumar, *Consultant, Deloitte Consulting, Toronto, Ontario, Canada*

Polly Perryman Kuver, *Consultant, Boston, MA*

Dennis Seymour Lee, President, Digital Solutions and Video, Forest Hills, NY

Alex Lee, *Co-Founder and President, QUEUE Systems Inc., Toronto, Ontario, Canada*

David Litwack, *President, dml Associates, Fairfax, VA*

Jeffery J. Lowder, *Chief of Network Security Element, United States Air Force Academy, Colorado Springs, CO*

Phillip Q. Maier, *Secure Network Initiative, Lockheed Martin, Sunnyvale, CA*

Victor Matos, *Professor, Department of Computer and Information Science, Cleveland State University, Cleveland, OH*

Lynda L. McGhie, *Director, Information Security, Lockheed Martin, Bethesda, MD*

Patrick G. McKeown, *Professor of Management, Terry College of Business, University of Georgia, Athens, GA*

Nathan J. Muller, *Senior Technical Consultant, e.spire Communications, Herndon, VA*

Ed Norris, CISSP, *Senior Security Consultant, Digital Equipment Corporation, Lancaster, MA*

Srinivas Padmanabharao, *Consultant, Deloitte Consulting, Toronto, Ontario, Canada*

Dean Peasley, M.B.A., M.S., *Senior Business Analyst, Liberty Mutual.*

T.M. Rajkumar, *Associate Professor, Decision Sciences and MIS, Miami University, Oxford, OH*

David Russo, *Senior Architect, Open Connect Systems, Dallas, TX*

Duane E. Sharp, *Director, SharpTech Associates, Mississauga, Ontario, Canada*

Carol A. Siegel, *Chief Security Officer, American International Group, New York, NY*

Carol A. Siegel, *Chief Security Officer, American International Groups, New York, NY*

Michael Simonyi, *Vice President, Research and Development, Stonewall, Toronto, Ontario, Canada*

Nancy Stonelake, *Senior Manager, Deloitte Consulting, Toronto, Ontario, Canada*

Michael J.D. Sutton, Adm.A., CMC, ISP, MIT, *Ottawa, Ontario, Canada*

Christine B. Tayntor, *Manager, Corporate Staff Applications, AlliedSignal, Inc., Morristown, NJ*

Marie Tumolo, *Assistant Professor, California State University, Fullerton, CA*

John R. Vacca, *Consultant, Pomeroy, OH*

John van den Hoven

Ramesh Venkataraman, *Assistant Professor, Department of Accounting and Information Systems, Indiana University, Bloomfield, IN*

David Wadsworth, *Java Evangelist, Sun Microsystems, Markham, Ontario, Canada*

Contents

Introduction xv

SECTION I STRATEGY AND PLANNING 1

1 Business-to-Business Integration Using E-Commerce 5
Ido Gileadi

2 Business-to-Business Exchanges 13
Marie Tumolo

3 The ASP Model: Evolution and Future Challenges 29
John Davies

4 Enclaves: The Enterprise as an Extranet 39
Bryan T. Koch

5 The Legal and Regulatory Environment of the Internet 53
Lawrence D. Dietz

6 Writing and Implementing Internet Acceptable Use Policies 65
James E. Gaskin

7 Designing Equitable Chargeback Systems 73
Christine B. Tayntor

SECTION II E-ENABLED BUSINESS SOLUTIONS 87

8 Building an E-Business Solution 91
Michael Simonyi

9 Customer Relationship Management: New Technology, Same Rules 105
Curtis Cook

10 e-CRM Is Not eASY 111
Alex Lee

11 **Electronic Bill Presentment and Payment** 119
Rahul Kumar

12 **Are Human Resource Departments Ready for E-HR?** 133
Marie Karakanian

13 **Call Management and the Internet** 141
John Fiske

SECTION III WIRELESS AND MOBILE BUSINESS SOLUTIONS 151

14 **Living in a Wireless World: Wireless Technology 101** 155
Michelle Cook

15 **Wireless: A Business and Technology Perspective** 163
Mark Barnhart, Ritu Joshi, Dean Peasley, and Jeffrey Staw

16 **Building a Wireless Web Solution: Tools and Justification for Building Wireless Web Solutions** 189
Sylvain Duford

17 **Putting Data in the Palm of Your Hand** 209
Alex Lee

18 **Programming Wireless Applications** 217
Gilbert Held

19 **Wireless Communications Data Center: Part I** 225
John R. Vacca

20 **Wireless Communications Data Center: Part II** 243
John R. Vacca

21 **Wireless Internet Security** 263
Dennis Seymour Lee

SECTION IV PROJECT APPROACHES AND LIFE CYCLE 287

22 **Prototyping Methods for Building Web Applications** 291
Jim Q. Chen and Richard D. Heath

23 **Component-Based Development** 311
Nancy Stonelake

24 **User Interface Rejuvenation Methodologies Available with Web-to-Host Integration Solutions** 321
Carlson Colomb and Anura Gurugé

25 **Determining Whether To Buy or Build an E-Commerce Infrastructure** 333
Carol A. Siegel

26 An Implementor's Guide to E-Commerce 353
John Care

SECTION V E-ENABLED ARCHITECTURE AND DESIGN 363

27 Usability Design for Web-Based Applications 367
Chris Forsythe

28 Component Architectures: COM and CORBA 377
T.M. Rajkumar and David K. Holthaus

29 Distributed Computing: Objects and Components in the Enterprise 389
David Russo

30 Developing Internet Solutions with SOAP 401
Charles Dow

31 Elements of a Well-Designed Corporate Web Site 411
Mark G. Irving

32 Developing a Trusted Infrastructure for Electronic Commerce Services 419
David Litwack

33 Evaluating and Selecting E-Commerce Software Solutions 431
Duane E. Sharp

SECTION VI THE E-ENABLED TOOLKIT 441

34 Selecting Hardware and Operating System Software for E-Commerce 445
Duane E. Sharp

35 XML — Rosetta Stone for Data 453
John van den Hoven

36 XML: A Common Data Exchange Language for Businesses 459
Charles Dow

37 Reengineering Information Systems with XML 469
Chao-Min Chiu

38 Linux and the Web 493
Srinivas Padmanabharao

39 Java and Its Supporting Technologies 501
Ramesh Venkataraman

40 Java and C++: Similarities, Differences, and Performance 513
Adam Fadlalla, Paul J. Jalics, and Victor Matos

41 **JavaBeans and Java Enterprise Server Platform** 533
David Wadsworth

SECTION VII SOLUTION TESTING. 545

42 **Integrated Automated Software Testing: A Life Cycle Approach** 549
Linda G. Hayes

43 **Web-Based Testing and Capacity Planning**. 561
Trevor Clarke

44 **Evaluating and Selecting Automated Testing Tools** 569
Polly Perryman Kuver

45 **Testing Object-Based Applications**. 579
Polly Perryman Kuver

46 **Security Testing Is Not All the Same: A Reference Taxonomy** 589
Jim Kates

47 **Introduction to Penetration Testing** 599
Stephen Fried

48 **Performing Penetration Testing**. 609
Stephen Fried

SECTION VIII SOLUTION PERFORMANCE AND SECURITY 621

49 **Framework for Internet Security Planning** 625
Monica J. Garfield and Patrick G. McKeown

50 **Internet Security: Securing the Perimeter** 637
Douglas G. Conorich

51 **Security Management for the World Wide Web** 653
Lynda L. McGhie and Phillip Q. Maier

52 **Firewall Management and Internet Attacks**. 681
Jeffery J. Lowder

53 **Protecting against Hacker Attacks**. 699
Ed Norris

54 **Improving Performance in New Programming Environments: Java** 711
Paul J. Jalics and Donald Golden

55 **Optimizing Web Design and Database Performance** 725
Srinivas Padmanabharao

SECTION IX ADVANCED TOPICS 735

56 Cellular Digital Packet Data: An Emerging Mobile Network Service 739
Nathan J. Muller

57 TN3270 and TN5250 Internet Standards 753
Ed Bailey

58 Knowledge Portal Classification Technology: Mapping Knowledge Landscapes 765
Michael J.D. Sutton

About the Editor 777

Index 779

Introduction

Proponents of the Internet have experienced an exciting ride over the past few years, often with unpredictable results. Is there a reasonable explanation for the unprecedented success and downturn of this medium? In some ways, the Internet took on the fervor of rock stardom. Previous IT trends such as client/server architecture, ERP, CASE Technology, and Object Orientation were popular within the industry. What made the Internet different from anything that came before it was the medium's popularity with the mainstream media and the public-at-large. Some of this might have been due to an after effect of the Y2K bug. The millennium bug caught the attention of the public and made IT popular everywhere — albeit out of fear. After the millennium bug virtually became a nonevent, the public may have been ready to accept another situation that potentially brought an overwhelming IT impact on society. The Internet, too, was thought to be all reaching, and although it promised a lot, it also threatened the way things were done by brick-and-mortar companies.

Before too long, it became clear that the panic around the Internet was unfounded. It did not fundamentally change the way business was conducted. Making money was as important as ever, and giving things away for free clearly had its limitations. This has led to a period of extreme detachment from this technology space. As dot.com companies are grappling with rigid market conditions and we keep hearing how the big technology players are being punished on Wall Street, it becomes very easy to think that the Internet was perhaps only a fad.

The Internet has transformed many things over the years, including how consumers and businesses interact. e-mail has become a killer application that is used everyday. Some surveys suggest that an average knowledge based business worker sends/receives up to 100 e-mail messages a day. Similarly, Web portals and news organizations dispense up-to-the-minute instant news over the Internet with access to background stories and information. Business hubs are used by groups of businesses to automate sets of recurring transactions that include office supply purchases, service requests, and other routine requisitions. These offer the benefits of simplification, fewer errors, and reduced costs. These and other common uses of the Internet demonstrate its ability to last. The expectations of the Internet were too great a few years ago, but this does not change the possibilities it offers for the future.

The Internet will continue to impact individuals and organizations — in how they conduct business and how they manage themselves. With the continued proliferation of Personal Device Assistants (PDAs) and other computing devices, the Internet is going to emerge as a more comprehensive and distributed technology than ever before. The Internet's influence will continue to grow as we learn how to leverage it effectively and properly.

PURPOSE OF THIS BOOK

The purpose of this book is to focus on E-enabled business solutions. These are emerging in the industry as convenient ways to package Web services. The future of the Internet could be determined by who pays for it going forward. In the last expansion, venture capitalists paid a significant portion of the capital for the Internet. In the next iteration, business users will be asked to pay for services based on business value. Business solutions are perhaps the strongest way to articulate this value.

SCOPE OF THIS BOOK

This book focuses on building business solutions for the Internet. This includes strategy and planning, E-enabled business solutions, wireless and mobile business solutions, project development approaches, E-enabled architecture and design, toolkits, testing, performance, and security.

INTENDED AUDIENCE

This book is intended for IT practitioners and technically inclined business users, including executives, managers, business analysts, data analysts, architects, developers, consultants, methodologists, Web masters, and testers.

GETTING THE MOST FROM THIS BOOK

Section I, "Strategy and Planning," explores defining strategy and plans in the current Internet environment to build an infrastructure that will support organizations into the next IT phase. This includes a review of business-to-business integration, the ASP model, and legal issues. This section also focuses on how revenue can be generated from Web-based services.

Section II, "E-Enabled Business Solutions," examines a cross section of business solutions that are being leveraged in the marketplace. This includes e-CRM, electronic bill presentment, E-HR, and call management.

Section III, "Wireless and Mobile Business Solutions" examines what could be the next major hot area in the IT industry. This section looks at the business opportunities that are available in the wireless world and at the underlying technology infrastructure and standards. This section also discusses programming, design, and hardware considerations in this space.

Section IV, "Project Approaches and Life Cycle" identifies some of the primary methodology and management considerations for building E-enabled Business Solutions. This includes a review of prototyping methods, component based development, and how to reuse legacy investments.

Section V, "E-Enabled Architecture and Design" explains how to architect and design business solutions for the Internet. This includes a review of usability concepts, Web design, and how components can be selected. Middleware, such as CORBA, COM, and SOAP, are also discussed in this section.

Section VI, "The E-Enabled Toolkit" examines some of the dominant tools for building business solutions. This includes a review of XML, JAVA, C++, and Linux.

Section VII, "Solution Testing" reviews testing considerations in the Internet space. This includes automated testing, Web-based testing, firewall testing, object-based testing, and penetration testing.

Section VIII, "Solution Performance and Security" focuses on two concepts that are critical to the ongoing success of business solutions over the Internet. This includes defining a framework for security planning and identifying opportunities to improve application performance.

Section IX, "Advanced Topics" examines a set of advanced concepts, such as development standards and knowledge classification technology.

SANJIV PURBA
October 2001

Section I
Strategy and Planning

In the last few years, it was really the Business-to-Consumer (B-to-C) side of the Internet that caught the attention of the public and quickly created a near-panic in individuals and businesses to claim a portion of the Internet sweepstakes. Was the Internet just a fad? If we define fads as temporary events or situations that attract an enthusiastic following, which is temporary, fleeting, and easily forgotton, the answer is a clear "no." Despite all the ups and downs, the Internet is still here. Business transactions are conducted over it. In fact, the Internet is a part of most business solutions being deployed today.

While the fervor and unrealistic expectations may have disappeared, there is a lot of Internet strategy and planning being done in organizations around the world. This is the Business-to-Business (B-to-B) side of Internet architecture, which is quietly being implemented and expanded on a global basis. While the B-to-C side of the architecture did not live up to the unrealistic hype of the last few years, the fact that B-to-B solutions are being implemented implies that Internet architecture has entered the mainstream of the Information Technology industry and is expected to be around for a long time.

This section considers several strategic initiatives that focus on business exchanges, making money on the Internet, application service providers, Web services, and legal issues. All of these areas should be considered in building an Internet strategy and plan for your organization. The chapters in this section cover the following material:

"Business-to-Business Integration Using E-Commerce" (Chapter 1) explains how to create an integrated and seamless supply chain by integrating business-to-business processes. This is accomplished by leveraging various E-commerce applications and the Internet to integrate supplier and manufacturer's back office systems.

"Business-to-Business Exchanges" (Chapter 2) discusses the types of business exchanges that have evolved since 1999 along with the costs and benefits. The chapter focuses on providing managers regarding the selection and effective use of exchanges.

"The ASP Model: Evolution and Future Challenges" (Chapter 3) shows how the application server provider model is evolving with the Internet, in addition to some of the tools and techniques (e.g., Microsoft.Net) that are available to leverage it in an effective manner.

"Enclaves: The Enterprise as an Extranet" (Chapter 4) examines strategies that organizations can leverage to create highly secure suborganizations that can benefit from Internet-based technology without suffering from the security problems and occasional unavailability experienced by full Internet business applications.

"The Legal and Regulatory Environment of the Internet" (Chapter 5) explores the critical areas of the law and legal liabilities for organizations doing business over the Internet. The chapter also presents some relevant business cases.

"Writing and Implementing Internet Acceptable Use Policies" (Chapter 6) discusses policies for effective and acceptable Internet use in order to protect organizations from illegal usage and acts. This chapter explains how such policies should be designed, to whom they should be applied, and how they can be managed.

"Designing Equitable Chargeback Systems" (Chapter 7) examines how Internet and other IT services can be charged out to their users. This is a fundamental building block for the future success of the Internet as the advertising model is being challenged.

Chapter 1
Business-to-Business Integration Using E-Commerce

Ido Gileadi

Now that many of the Fortune 1000 manufacturing companies have implemented ERP systems to streamline their planning and resource allocation as well as integrate their business processes across the enterprise, there is still a need to be integrated with the supply chain.

To reduce inventory levels and lead-times, companies must optimize the process of procurement of raw materials and finished goods. Optimization of business processes across multiple organizations includes redefining the way business is conducted, as well as putting in place the systems that will support communication between multiple organizations each having its own separate systems infrastructure and requirements.

This type of business-to-business electronic integration has been around for some time, in the form of EDI (electronic document interchange). EDI allows organizations to exchange documents (e.g., purchase orders, sales orders, etc.) using standards such as X.12 or EDIFACT and VANs (value-added networks) for communication. The standards are used to achieve universal agreement on the content and format of documents/messages being exchanged. EDI standards allow software vendors to include functionality in their software that will support EDI and communicate with other applications. The VAN is used as a medium for transferring messages from one organization to the other. It is a global proprietary network that is designed to carry and monitor EDI messages.

The EDI solution has caught on in several market segments but has never presented a complete solution for the following reasons:

0-8493-1161-6/02/$0.00+$1.50

- High cost for setup and transactions: smaller organizations cannot afford the cost associated with setup and maintenance of an EDI solution using a VAN.
- EDI messages are a subset of all the types of data that organizations may want to exchange.
- EDI does not facilitate online access to information, which may be required for applications such as self-service.

With the advance of the Internet both in reliability and security and the proliferation of Internet-based E-commerce applications, E-commerce has become an obvious place to look for solutions to a better and more flexible way of integrating business-to-business processes.

The remainder of this chapter discusses a real-life example of how internet and E-commerce technologies have been implemented to address the business-to-business integration challenge.

BUSINESS REQUIREMENTS

The business requirements presented to the E-commerce development team can be divided into three general functional area categories:

1. General requirements
2. Communicating demand to the supply chain
3. Providing self-service application to suppliers

General requirements include:

- 100 percent participation by suppliers: the current EDI system was adapted by only 10 percent of suppliers
- Minimize cost of operation to suppliers and self
- Maintain high level of security both for enterprise systems and for data communicated to external organizations
- Utilize industry standards and off-the-shelf applications wherever possible; minimize custom development
- Supplier access to all systems through a browser interface

Demand requirements include:

- Send EDI standard messages to suppliers
  — 830: Purchase Schedule
  — 850: Purchase Order
  — 860: Purchase Order Change
- Provide advance notice of exceptions to demand through exception reports

Exhibit 1 describes the flow of demand messages (830, 850, 860, exceptions) between the manufacturer and supplier organization. The demand is generated from the manufacturer ERP system (Baan, SAP, etc.). It is then

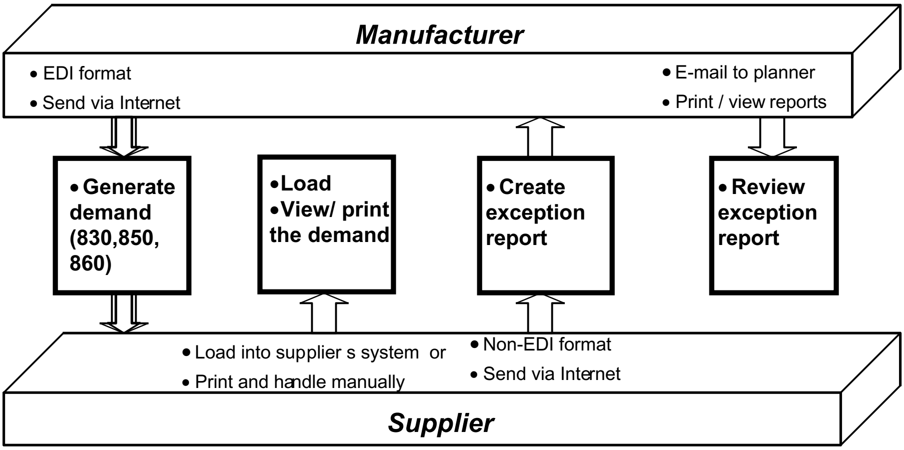


Exhibit 1. Demand Flow

delivered to the supplier through one of several methods (discussed later). The supplier can load the demand directly into its system or use the supplied software to view and print the demand on a PC. The supplier can then produce an exception report, indicating any exception to the excepted delivery of goods. The exception report is sent back to the manufacturer and routed to the appropriate planner. The planner can view the report and make the necessary adjustments.

Self-service application requirements include:

- Ability for suppliers to update product pricing electronically, thereby ensuring price consistency between manufacturer and supplier
- Provide online access with drill-down capabilities for suppliers to view the following information:
  — Payment details
  — Registered invoices
  — Receipt of goods details
  — Product quality information

TECHNICAL REQUIREMENTS

The technical solution had to address the following:

- Transport EDI messages to suppliers of various levels of computerization
- Provide complete solution for suppliers that have no scheduling application
- Support small and large supplier organizations seamlessly

- Provide batch message processing and online access to data
- Provide security for enterprise systems as well as data transmission
- Utilize industry standards and off-the-shelf products

Once again, the technical requirements are divided into three categories:

1. General requirements
   a. Low cost
   b. Low maintenance
   c. High level of security
   d. Industry standards
2. Batch message management
3. Online access to enterprise information

In reviewing the three main categories of technical requirements it is apparent that one needs a product to support message management (EDI and non-EDI), and the same or another product to provide online access. The selected products will have to possess all the characteristics listed under general requirements.

E-COMMERCE PRODUCT SELECTION

Selection of E-commerce products to construct a complete solution should take the following into consideration:

- What type of functionality does the product cover (online, batch, etc.)?
- Is the product based on industry standards or is it proprietary?
- Does the product provide a stable and extensible platform to develop future applications?
- How does the product integrate with other product selections?
- What security is available as part of the product?
- What are the skills required to develop using the product, and are these skills readily available?
- Product cost (server, user licenses, maintenance)?
- Product innovation and further development?
- Product base of installation?
- Product architecture?

The E-commerce team selected the following products.

WebSuite and Gentran Server from Sterling Commerce. This product was selected for handling EDI messages and communication EDI and non-EDI messages through various communication mediums. This product provides the following features:

- Secure and encrypted file transfer mechanism
- Support for EDI through VANs, Internet, and FTP

- Browser operation platform using ActiveX technology
- Simple integration and extendibility through ActiveX forms integration
- Simple and open architecture
- Easy integration with other products
- EDI translation engine

Baan Data Navigator Plus (BDNP) from TopTier. This product was selected for online access to the ERP and other enterprise applications. The product has the following main features:

- Direct online access to the Baan ERP database through the application layer
- Direct online access to other enterprise applications
- Integration of data from various applications into one integrated view
- Hyper Relational data technology, allowing the user to drag and relate each item data onto a component thereby creating a new more detailed query providing drill-down capabilities
- Access to application through a browser interface
- Easy-to-use development environment

Both products had just been released when the project started using them (summer 1998). This is typically not a desirable situation because it can extend the project due to unexpected bugs and gaps in functionality. The products were chosen for their features, the reputation of the companies developing the products, and the level of integration the products provided with the ERP system already in place.

E-COMMERCE SOLUTION

Taking into account the business and technical requirements, a systems architecture that provided a business and technical solution was put together. On the left side of the diagram are the client PCs located in the supplier's environment. These are standard Win NT/95/98 running a browser capable of running ActiveX components. Both the applications (WebSuite and TopTier) are accessed through a browser using HTML and ActiveX technologies. As can be seen in Exhibit 2, some suppliers (typically the larger organizations) have integrated the messages sent by the application into their scheduling system. Their systems load the data and present it within their integrated environments. Other suppliers (typically smaller organizations) are using the browser-based interface to view and print the data as well as manipulate and create exception reports to be sent back to the server.

Communication is achieved using the following protocols on the Internet:

- HTTP, HTTPS: for delivery of online data
- Sockets (SL), Secure Sockets (SSL): for message transfer

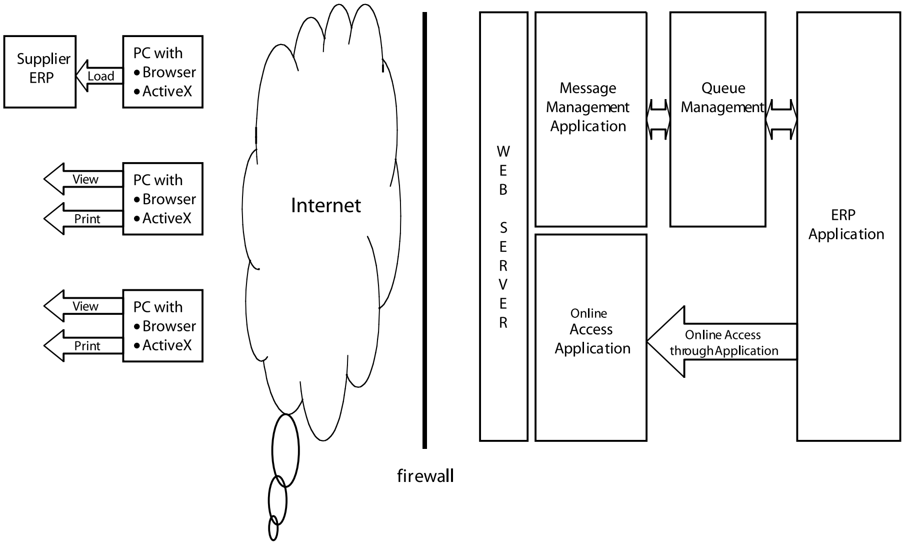


Exhibit 2. Firewall

All traffic enters the enterprise systems through a firewall for security. Security is discussed in the following section.

On the enterprise side, the client applications first access a Web server. The Web Server handles the HTTP/HTTPS communication and invokes the server-side controls through an ASP page.

The online application (TopTier) intercepts the HTTP/HTTPS communication address to it and interprets the query. It then provides a result set and integrates the result set with an HTML template to be sent back to the client PC as an HTML page. The online access application communicates with the ERP application through the application API or through ODBC.

The message management application (WebSuite) communicates to the message queue using server-side ActiveX controls and FTP to send and receive files between systems. The message management application communicates with the ERP and other enterprise applications using a set of processes that can read and write messages to a shared mounted disk area.

The above system architecture supports a mechanism for transferring messages in a secure and reliable fashion as well as providing online access to data residing in the enterprise systems — all through a browser interface with minimal requirements from the supplier and minimal support requirements.

SECURITY

The are two categories of security that must be handled:

1. Enterprise systems security from outside intrusion
2. Data security for data communicated over the Web

Security for the enterprise is intended to prevent unauthorized users from accessing data and potentially damaging enterprise systems and data. This is handled by various methods that are far too numerous to discuss meaningfully in this chapter. One can review the steps taken to secure the system on this project; these are by no means the only or the complete set of measures to be taken. In addition, each organization may have different security requirements. For this project the following steps were taken:

- Use a firewall that provided the following:
  — Limitation on IP and PORT addresses
  — Limitation on protocols allowed (HTTP, HTTPS, IP)
  — User Authentication at the firewall level
  — Abstraction of Server IP address
- Authentication:
  — Front-office application layer
  — Back-office application layer
  — Operating system layer
  — Firewall layer
- Domain settings:
  — The Web server machine is not part of the enterprise domain
  — The Web server machine has IP access to other servers

Data security is required to protect the information that is transferred between supplier and manufacturer over the public domain of the Internet. The intent is to secure the data from unauthorized eavesdropping. There are many methods to protect the data; these methods can be grouped into two main categories:

- Transferring data through a secure communication channel (SSL, HTTPS). This method utilizes:
  — Authentication
  — Certificates
  — Encryption
- Encryption of data. This method is typically used in conjunction with the previous method, but can be used on its own. There are various encryption algorithms available. The encryption strength (cipher strength), which can be defined as how difficult it would be to decrypt encrypted data without the keys, can vary and is designated in terms of number of bits (40 bit, 128 bit, etc.). This project employed

Microsoft Crypto API, supported both by the Web server (IIS 4) and by the client browser (IE 4). The cipher strength selected was 40 bits to allow non-United States and Canada access to the application; 128-bit cipher strength is not available for browsers used outside of the United States and Canada.

CONCLUSION

Manufacturing organizations striving to reduce inventory levels and lead-times must integrate business processes and systems with their supply chain organization. E-commerce applications utilizing the Internet can be used to achieve integration across the supply chain with minimal cost and standard interfaces.

When implementing E-commerce applications, it is recommended to select application that can be used as an infrastructure to develop future business solutions to address new requirements. Selecting applications that provide technology solutions with a development platform, rather than applications that provide an integrated business solution, will provide a platform for development of future business applications as the use of E-commerce proliferates through the organization.

ABOUT THE AUTHOR

Ido Gileadi is a senior manager with Deloitte Consulting – ICS, in Toronto, Ontario, Canada. He is a technical leader in the Baan practice.

Chapter 2
Business-to-Business Exchanges

Marie Tumolo

Business-to-business (B2B) exchanges are central, electronic marketplaces in which multiple buyers and multiple suppliers come together to exchange goods and services. Exchanges are a significant component of the business-to-business electronic commerce market, estimated to reach $600 billion to $3 trillion in U.S. revenues by 2003.

Exhibit 1 outlines the three major aspects of exchanges. Exchanges are used to match buyers and suppliers and facilitate transactions between the two. They also maintain a technical, institutional, and compliance infrastructure that supports their offerings. Exchanges are typically run by independent, third-party intermediaries rather than by individual buyers (e.g., General Motors Corp. or General Electric Co.) or suppliers. In the single-firm case, a buyer opens an electronic market on its own server and invites suppliers to bid on specified parts or services needed, or a supplier sells its products or services only to approved customers. Third-party exchanges, on the other hand, do not take title or physical possession of goods but facilitate the matching of buyers and suppliers. The terms "exchange" and "marketplace" are used interchangeably in discussions of electronic commerce. Exchanges serve a variety of industries including aerospace, agriculture, automobiles, banking, chemicals, education, employment, energy, food, hospitality, insurance, paper, and steel.

Exchanges are a form of outsourcing, enabling a company to shift much of the work performed by the purchasing function to a third party. The exchange searches for suppliers matching the buyer's request, compares prices and product features, and provides recommendations. All of these services replace ones now usually performed by employees in the purchasing department. For suppliers, exchanges provide another channel of distribution, one that does not require support by sales personnel. Exhibits 2 and 3 illustrate the differences between using an exchange for transactions and using it for conventional means.

0-8493-1161-6/02/$0.00+$1.50

Exhibit 1. How Exchanges Work

Matching Buyers and Suppliers

- Establishing product offerings
- Aggregating and posting different products for sale
- Providing price and product information, including recommendations
- Organizing bids and bartering
- Matching supplier offerings with buyer preferences
- Enabling price and product comparisons
- Supporting negotiation and agreement between buyers and suppliers

Facilitating Transactions

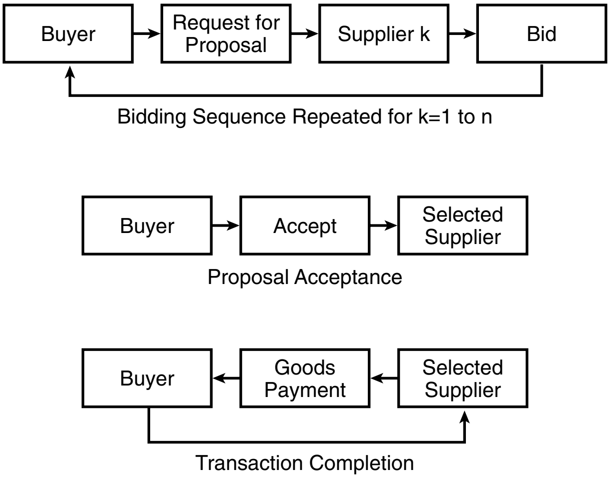
- Logistics: delivery of information, goods or services to buyers, identification of company administrator to:
  - — Provide billing and payment information including addresses
  - — Define terms and other transaction values
  - — Input searchable information
  - — Grant exchange access to users and identify company users eligible to use exchange
- Settlement of transaction payments to suppliers, collecting transaction fees
- Establishing credibility: registering and qualifying buyers and suppliers, communicating exchange transaction and other fees, maintaining appropriate security over information and transactions

Maintaining Institutional Infrastructure

- Ascertaining compliance with commercial code, contract law, export and import laws, intellectual property law, rules and regulations of appropriate agencies
- Maintaining technological infrastructure to support volume and complexity of transactions
- Providing interface capability to standard systems of buyers and suppliers
- Obtaining appropriate site advertisers and collecting advertising and other fees

Exchanges operate throughout the supply chain, facilitating everything from the acquisition of raw materials to the sale of finished goods. When exchanges are integrated with automated procurement processes and customer requirements management systems, the supply chain is streamlined within and across organizations and industries. A typical supply chain includes the components shown in Exhibit 4.

Estimates of the number of exchanges in existence vary from 600 to 1000. Announcements of the formation of new exchanges continue to appear in the press, despite the announcements of exchanges closing down or scaling back. Many of the exchanges are unlikely to last beyond the initial press release or survive beyond the first year. AMR Research predicts that only 50 to 100 of the current exchanges will survive through 2001.

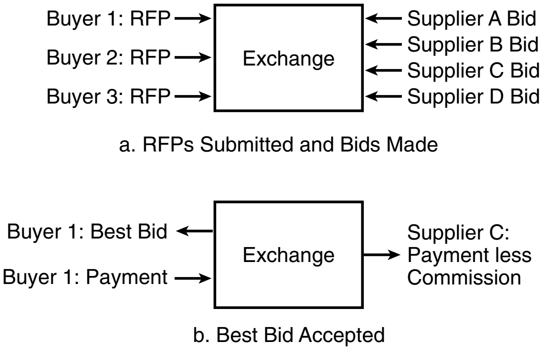


Bidding Sequence Repeated for k=1 to n

Proposal Acceptance

Transaction Completion

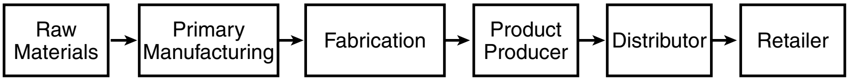
Exhibit 2. Conventional Process



a. RFPs Submitted and Bids Made

b. Best Bid Accepted

Exhibit 3. Exchange Process



Source: J. R. Galbraith, "Strategy and Organization Planning," in *The Strategy Process: Concepts, Contexts, Cases, 2nd ed.,* edited by H. Mintzberg and J. B. Quinn (Englewood Cliffs, NJ: Prentice Hall, 1991), p. 316.

Exhibit 4. Typical Supply Chain

How Exchanges Evolved

Since the 1980s, IT experts have predicted that information technology would reduce many of the coordination costs incurred by businesses, such as the costs of gathering information, negotiating contracts, or protecting against opportunistic behavior. Coordination costs, along with other factors such as the degree of complexity of a particular product and the level of specificity of assets used in a particular business, influence the decision whether to use markets for economic activity or to directly control more aspects of business through ownership and hierarchical management. The reduction of coordination costs along with lower asset specificity leads to higher use of market mechanisms. Exchanges are market mechanisms that use information technology and the Internet to reduce the cost of gathering information and negotiating contracts for products that are fairly standard either across or within industries.

B2B E-commerce has its roots in electronic data interchange (EDI) networks established between large buyers and suppliers within a specific industry. The automobile, aerospace, and chemical industries used EDI extensively to reduce costs and improve operational efficiency. EDI consists of private networks between companies that facilitate communication of orders, status, invoicing, and payment. Expensive to build and maintain, EDI networks were often limited to the largest companies within that industry. Many of the buyer- and supplier-oriented marketplaces evolved from EDI, facilitated by the online automation of the procurement process made possible by the Internet.

Exchanges came about when several companies (predominantly Ariba, Inc., Commerce One, Inc, and W. W. Grainger, Inc.) took their E-procurement software and used it to establish open markets on their own servers. Many of the initial exchanges were online catalogs. However, continued technological development is increasing the scope of services provided by the exchanges. A number of exchanges are beginning to provide additional functions. For example, Messmer[11] points to such functions as:

- Sharing of synchronized, real-time updates on prices and shipment information
- Pushing and pulling data directly from corporate back-end enterprise resource planning and database systems
- Flagging errors before problems multiply on the production and shipping end

In addition, Trombly (15) describes the following functionality:

- Generating Extensible Markup Language (XML) forms, which can be viewed with a Web browser

- Evaluating product availability, by linking into a supplier's inventory application
- Pacing orders from a buyer's procurement application, then following through with a supplier's order-fulfillment package
- Supporting different contract terms or purchasing agreements for different buyers
- Fulfilling orders electronically, which requires real-time order validation and downloads for applications such as software or content sales
- Electronic invoicing and payments including automated clearinghouse payments
- Credit checks and financing
- Security and authentication

Types of Exchanges

Most exchanges can be categorized as either horizontal or vertical. *Horizontal exchanges* provide many commodity products that can be used across most industries. Typical horizontal exchanges involve the purchase and sale of

- Office supplies
- Uniforms
- Furniture and equipment
- Maintenance services
- Electronic components
- Repair and operating supplies

The primary benefits of horizontal exchanges are the variety of products offered and lower prices.

Vertical exchanges focus on a specific industry, providing participants with

- Specialized products
- In-depth industry knowledge
- Greater opportunities for collaboration

There are also meta-exchanges that combine aspects of both vertical and horizontal markets and support a full range of market-connecting mechanisms, including bid/ask exchanges, auctions, and reverse auctions.

Horizontal and vertical exchanges can take various forms. Specifically, the major forms of exchanges are

- Aggregate catalogs (Chemdex)
- Trading (TradeOut)
- Online exchange of goods (FreeMarkets)
- Labor exchange (Guru.com)
- Online auctions and reverse auctions (PlasticsNet)
- Fully automated with order matching such as stock exchanges (Altra Energy Technologies).

Exhibit 5. Major Forms of Exchanges

Exchange Name	Type	Form	Functions
Chemdex	Vertical	Catalog	Academics, scientists, pharmaceutical firms buy and sell science research products
OrderZone	Horizontal	Catalog	Merged with Works.com. Sells uniforms, office supplies, maintenance, repair, laboratory and safety equipment, and electronic components
TradeOut	Horizontal	Trading	Virtual storefront posting surplus assets representing over 100 product categories
FreeMarkets	Horizontal	Online	Purchase and sale of industrial parts, raw materials, and commodities and services
Guru.com	Horizontal	Labor	Posting of projects requiring contract workers
PlasticsNet	Vertical	Online Auction	Auctions and reverse auctions for used and excess products in the plastics industry; materials and news for buyers and suppliers
Altra Energy Technologies	Vertical	Fully Automated	Trades natural gas and other energy; traded $4 billion in 1999
e-STEEL	Vertical	Automated	Entire steel industry; supports prime and non-prime steel products
PaperExchange	Vertical	Automated	Buys and sells all grades of paper
Covisint	Vertical	Catalog, auction, and automated	Formed by auto industry manufacturers

Examples of these forms are shown in Exhibit 5.

Exchange Technology

Exchanges are basically Web sites that use a standard language, XML, to facilitate application-to-application data exchange, similar to EDI. XML allows information regarding orders, purchases, payments, and products to be easily understood by other computers. XML, in effect, makes the benefits of EDI accessible to organizations of all sizes.

Three software companies, Ariba, Commerce One, and Oracle, dominate the exchange software market by providing packages, installation help, and consulting. These software packages evolved from buy-side software developed to help purchasing departments.

Although XML is a standard language, different versions exist, hampering the ability for exchanges to communicate with one another. In response, a number of software companies such as i2 Technologies, Extricity, Mercator, and IBM are developing software to help buyers and suppliers use multiple exchanges to transcend the barriers erected by using different versions of XML.

Full automation of the supply chain is one of the primary benefits often claimed for B2B E-commerce. That is, the software covers all functions from sales force and materials buying to billing. However, only 10 of 600 exchanges tracked by AMR Research as of April 2000 actually provide integration from the exchange to a supplier's or buyer's back end systems.

Sell-side electronic commerce systems enable the exchange to tap into the suppliers' systems to determine available quantities and price for quotations. Software companies dominating the supply side include Calico, Ironside, and SAP. On the buy side, E-procurement systems such as those developed by Oracle, Ariba, and Commerce One allow buyers to streamline the processes of requisitioning parts and services, retrieving necessary documentation, obtaining and evaluating bids, and receiving the items.

BENEFITS OF EXCHANGES

Exchanges promise significant benefits including

- Cost savings
- Increased operational efficiency
- Improved information.

Most of these benefits have yet to be realized, although some participants predict annual cost savings of anywhere from 7 percent to 30 percent. Benefits vary among buyers and suppliers, but both parties achieve the benefit of better information. Exchanges are information tools, providing buyers and suppliers with the ability to screen and compare products, prices, sources, terms, availability and potential substitute products. In addition, some exchanges provide product recommendations, often one of the most valued features for participants. In addition, both buyers and suppliers have the added benefit of reduced negotiation costs because the exchange enables both parties to a transaction to meet electronically and come to agreement rapidly.

Buyer Benefits

Because exchanges bring together multiple buyers and suppliers, buyers can expect to pay lower prices when purchasing through an exchange. Smaller orders can be aggregated by the exchange so that each individual buyer receives the high-volume discount. Buyers have more suppliers to choose among and gain greater price transparency. It is much easier in an exchange to see how prices for the same product may vary based on geographic region, size of the order, or customer relationship. In addition, customers may compare the price they typically pay with the current exchange price and ask why any difference exists. Using an exchange also enables buyers to obtain information about product availability and potential substitute products more quickly than before and at a lower search cost.

Even more significant are the cost savings that can be achieved by automation of the procurement process and integration of a company's systems with those of the exchange. Lower administrative costs result from reductions in the number of employees required to support the purchasing function, streamlining the approval process for purchases and enabling managers to make purchasing decisions directly by using the exchanges. Integration of a company's back-office systems with those of the exchanges also facilitates improved inventory management.

Supplier Benefits

Suppliers that participate in exchanges are able to expand their markets, acquire new customers at a very low cost, aggregate smaller orders into larger bundles, and service customers at a lower cost. Suppliers reduce their dependence on their sales forces and eliminate the expensive costs of continually producing expensive catalogs. In addition, suppliers can often eliminate traditional market intermediaries because the exchange acts as an automated intermediary. It may be possible for suppliers to appropriate and retain a portion of the discount previously given to the middleman. Integration of exchange information with customer relationship management information already under development by the major software companies will allow suppliers to obtain customer information such as purchasing history and to automate sales and services.

COSTS

Fees vary by exchange: some exchanges charge a fee per transaction; others charge a percentage of the revenue on the transaction or a percent of the cost savings achieved. Some exchanges also charge membership fees, but competition for participants is reducing the number of exchanges charging such fees. Recent trends indicate that exchanges are increasingly favoring open (non-fee) membership to qualified participants. Downward pressure continues to be exerted on all types of fees. Additional information on the example exchanges and their fees is shown in Exhibit 6.

Suppliers often are required to pay a fee to post items for sale or for sales transactions that occur. Suppliers, however, are critical to the success of exchanges. For an exchange to be successful, it must provide

Exhibit 6. Additional Information on Example Exchanges, Including Fees

Exchange	Date Opened	Ownership	Membership	Available Fee Data	Supply Chain Integration
Horizontal:					
TradeOut	1998	Independent	Open	Supplier: 5% of -$10 listing fee -$1000 annual membership fee	No
Works.com (OrderZone)	1999	Consortium	Open	Service fee: $1 per order	Pending
FreeMarkets	1995	Independent	Open	Seller: 5% of winning bid	No
Guru.com	1999	Independent	Open	$200 per project or $1000/qtr	No
Vertical:					
Altra Energy	1996	Independent	Qualified		Yes
eSTEEL	1999	Independent	Qualified	Supplier: 0.875% of transaction value	Pending
PaperExchange	1998	Commercial investors	Qualified	Supplier: 3% of total purchase price	No
PlasticsNet	1997	Independent	Qualified		Yes
Covisint	Pending regulatory approval	Consortium	Qualified	To be determined	Future enhancement
Chemdex		Independent	Qualified		Yes

buyers with a large number of suppliers from which to choose products and services. To attract suppliers, exchanges must keep fees charged to suppliers reasonable. With so many exchanges being developed and as yet few clear dominant players, most exchanges are more concerned with attracting suppliers than making money from high fees.

Buyers' costs can include membership fees as well as transaction fees, depending on the exchange. The actual cost of the goods to the buyer includes the amount paid to the supplier, the commission paid to the exchange (if any), and the freight costs. In particular, if a buyer selects a supplier located further away, the savings from lower costs of goods may be eaten up by the additional freight charges.

DISADVANTAGES OF EXCHANGES

To achieve the benefits of the exchange in terms of a more efficient process, buyers must make a large number of transactions over this channel. In effect, the exchange becomes the buyer's single (or major) source for supplies or vital inputs. Although cost savings are associated with using the exchange, the buyer assumes the risk of exchange failure or deterioration. Buyers also still run the risk that changing suppliers when buying through the exchange may result in poor product performance, particularly when buying critical parts and components. Exchange recommendations and comparisons may provide insufficient information regarding input specifications unique to an individual buying company.

From a supplier's standpoint, the primary disadvantage is that exchanges may dominate over all other selling channels, leaving those companies that do not join the right exchange out of the bulk of the business. Mediocre suppliers (in terms of quality and price) face particular disadvantages because the widespread use of exchanges will most likely force out of business those suppliers that cannot meet the exchange's standards.

CRITICAL SUCCESS AND FAILURE FACTORS FOR EXCHANGES AND PARTICIPANTS

Critical Success Factors

- Mass. Exchanges require sufficient mass, that is, enough buyers and suppliers to make participation worth it for both. Mass also increases liquidity of the exchange and enables the exchange to improve services offered. Because exchanges are so new, it is difficult to determine what sufficient mass is for each market. Dollar volume of transactions and the stability of transaction fees are considered to be important criteria of exchange liquidity.

- Seamless integration. Increasingly, exchange members want to integrate their company's back-office systems seamlessly with the exchange as well as clearing the financial transactions with each party's bank. As a result of this integration, exchanges can be expected to evolve from marketplaces to full supply chain automation. Exchange members also want to use information generated by the exchange to forecast demand for their products better as well as to compare prices, product characteristics, availability, terms, and sources.
- Income. The problem for exchanges is how to balance the need for revenues to keep going against the need for participants to realize the cost-saving benefits of participation. Having a sufficient number of suppliers is critical to the survival of an exchange because it increases the willingness of buyers to participate. Yet many suppliers are increasingly concerned about participating in exchanges that are too focused on price as the only determinant of the purchasing decision. Suppliers have little incentive to encourage deterioration of their own profits.

Critical Failure Factors

Many of the exchanges that closed recently — for example, IndustrialVortex, a marketplace for industrial automation products; M-xchange, a horizontal exchange for minority-owned suppliers; and Fleetscape.com, a marketplace for commercial truck aftermarket parts and service — were unable to obtain additional funding. Venture capitalists are becoming more reluctant to invest in exchanges with heavy reliance on transaction fees. So many exchanges have been formed that one analyst at Keenan Vision estimates that some 4,200 exchanges will exist by 2003. The numbers are driving down transaction fee revenues in many markets. Originally, exchanges anticipated charging 1 percent to 3 percent of each transaction amount. Downward pressure led many exchanges to reduce their fees to 1/4 percent to retain sufficient membership and liquidity. For this reason, exchanges that are backed by significant players in an industry are considered most likely to survive.

IMPLICATIONS OF B2B EXCHANGES

Business Issues

Exchanges need to ensure that suppliers listed on the exchange are able to supply the quality and quantity of goods demanded by buyers and have the integrity to be participants to a contract. Buyers will quickly stop using an exchange if the products and services offered by the supplier do not meet expected standards. Exchanges that will survive are those that exert an effort to prequalify both buyers and suppliers in terms of conventional business factors such as reputation, creditworthiness, size, and experience.

Distribution logistics are also an important consideration when using an exchange. Although exchanges bring together multiple buyers and suppliers, enabling each to expand its scope of operations, basic logistics require the ability to actually deliver products, limiting the transactions to those companies best able to deliver where and when the buyer needs it.

As with any intermediary function, exchanges are subject to concerns by participants that the exchange is reputable, complies with all relevant laws and regulations governing transactions, and has adequate procedures in place to qualify participants, secure private information, and safeguard financial assets.

Antitrust Considerations

Both the FTC and the European Commission are examining exchanges and their potential impact on the competitiveness of various industries. The exchanges attracting the most regulatory attention are those that strive for full supply chain automation within a specific industry and that evolved from industry consortiums. For this reason, the FTC continues to look closely at Covisint, the automakers' exchange, after granting it guarded approval on September 11, 2000. The European Commission is looking at MyAircraft.com, a joint venture for the sale of aircraft spare parts and engines set up by Honeywell International, Inc., United Technologies Corp., and i2 Technologies, Inc. One potential violation of competition rules concerning the EU is the ability of the exchange to compare price and other sensitive information — in effect, forming a cartel. Both regulatory agencies are concerned with the ability of exchange members to share information about prices and the potential for that to lead to price fixing. They are also concerned that exchanges may so dominate an industry that nonmember firms are forced out of business and that price sharing may lead to downward price pressure, squeezing out smaller players and creating oligopolistic situations.

Exchanges are carefully constructing firewalls and other security measures designed to alleviate potential antitrust concerns. Many exchanges, such as eSTEEL, do not allow individuals from other companies to see final agreed-on prices between buyers and suppliers. Automated exchanges are concentrating on commodities and using existing commodity exchange markets as guides.

Relationship Management

One of the potentially most interesting effects of exchanges is their impact on supplier relations, customer loyalty, and customer retention. During the past ten years, academics and the popular business press advocated the formation of deep relationships between buyers and suppliers, partially facilitated by EDI and customer relationship software. Strengthening relationships with key suppliers enabled companies to reduce costs and defects of parts and raw materials. Customer intimacy enabled companies to respond to the evolving needs of their targeted customers with tailored products and services. Flexible customer response often depended on speed to market, increasing the importance of strong supplier relationships.

The exchanges have the potential to fundamentally change the nature of those relationships as buyers become aware of new suppliers and increase their ability to compare prices and service across a broader range of businesses. Yet at the same time, customer/supplier intimacy is increasingly critical to a company's ability to differentiate itself from the competition.

The depth and criticality of customer/supplier relationships will drive the type of exchange used by a company. For noncritical supplies and equipment, a third-party horizontal exchange is appropriate. Companies would use vertical exchanges to purchase industry-specific and commodity items and to monitor changes and evolutions in the industry that may affect future customer/supplier relationships.

LESSONS FOR MANAGERS

- *Determine what role exchanges should play in your business.* Although exchanges are considered most effective for commodity products and services, the development of industry-specific exchanges with full supply chain integration may lead to the use of exchanges for acquisitions of strategic materials and services as well. Many exchanges are working diligently to improve security around requisitions, bids, negotiations, and transactions. Strategic functions, products, and services need to be carefully identified, and the process of developing and acquiring strategic components needs to be carefully examined to determine how best to take advantage of developing opportunities. Fragmented industries are particularly able to benefit from the use of exchanges.
- *Identify all potential exchanges.* Look at exchanges being used by other companies in the same industry. Determine whether companies of similar size with similar business models are using certain exchanges more than others. Closely examine geographic dispersion of suppliers and buyers in a particular exchange in light of the unique logistic issues specific to the company's product or service.
- *Evaluate the exchanges based on how the company will use them.* Look for focus and participants that are appropriate for the company's buying or selling needs. Review content carefully to determine whether products offered are of acceptable quality and quantity, include name brands, and are priced competitively. Examine exchange investors and partners. Most of the exchanges that have closed are independent

third-party exchanges with no equity participation by any significant players in the industry. Look at the history of the exchange and fee structure to determine how sustainable the exchange is.

- *Select the right exchanges to join.* Realistically, companies will probably need to join more than one exchange. The most successful exchanges to date focus on a particular industry or type of product but have sufficient breadth to attract many buyers and suppliers. Liquidity is key as well as integration and the ability of the exchange to facilitate linkage to other exchanges, enabling a buyer or supplier to participate seamlessly in the most appropriate exchanges. Although many experts define liquidity as sufficient buyers and suppliers, it also is important to look at the nature of transactions — whether buyers and suppliers will find sufficient products and services to keep using the exchange over the long term.
- *Extract value from the exchanges.* Closely monitor cost savings achieved compared with those anticipated. Use the exchanges' information-gathering and reporting capabilities to improve planning and forecasting of product demand. Monitor enhancements offered by the exchanges and carefully consider how they may help you improve your business.
- *Evaluate results.* Calculate your return on investment, cycle time improvements, and impact on the business. Look at cost savings over time. Periodically review trends over time in prices. Keep a close eye out for process savings. Determine what information gained from participating in the exchange means in terms of improving business performance or avoiding costly mistakes.

CONCLUSIONS

Peter F. Drucker tells us that the best way to determine what will happen in the future is to look at what has already happened. For example, a number of independent exchanges closed down or merged into other exchanges because they had liquidity problems. Consortium-led exchanges, such as Covisint and MyAircraft.com, are experiencing regulatory scrutiny as well as delays due to the need for consensus among industry members. These instances do not mean that exchanges will disappear, rather that only the best will survive, resulting in easier decisions for managers trying to determine which to join. The number of enhancements being developed to improve exchange functionality and service is another indication of which exchanges may ultimately deliver the greatest value to participants.

References

1. eMarketer (2000). *Changing B2B Exchanges* [Business 2.0 Web site]. August 14.
2. Backes, A., & Butler, S. (2000). "New eCommerce: B2B Report Examines Wide Range of Projections for B2B Growth." *eMarketer,* August 8.
3. Bakos, Y. (1998). "The Emerging Role of Electronic Marketplaces on the Internet." *Communications of the ACM* (August), 35.
4. Copeland, L. (2000). "Trade Exchange Closes Virtual Doors." *Computerworld* (July 24).
5. Drucker, P. F. (1997). "The Future That Has Already Happened." *Harvard Business Review,* 75 (5, Sept.–Oct.), 20–24.
6. Greenemeier, L. (2000). "Buying Power." *Information Week,* 780 (Apr. 3), 67–68.
7. Gubman, Edward I. (1995). "Aligning People Strategies with Customer Value." *Compensation and Review* (Jan.–Feb.), 15–22
8. Kaplan, S., and Sawhney, M. (2000). "E-Hubs: The New B2B Marketplace." *Harvard Business Review,* 78 (3, May–June), 97–103.
9. Lundegaard, K. (2000). "FTC Clears Covisint, Big Three's Auto-Parts Site." *Wall Street Journal.*
10. Malone, T. W., Yates, J., and Benjamin, R. I. (1987). "Electronic Markets and Electronic Hierarchies." *Communications of the ACM,* 30(6), 484–497.
11. Messmer, E. (2000). "Online Supply Chains Creating Buzz, Concerns." *Network World,* 17 (Apr. 24), 12.
12. Nash, K. S. (2000). "Reality Check for E-Markets." *Computerworld* (June 5), 58–59.
13. Segal, R. L. (2000). "Online Marketplaces: A New Strategic Option." *Journal of Business Strategy,* 21 (2, Mar./Apr.), 26–29.
14. Sweat, J. (2000). "E-market Connections." *Information Week,* 780 (Apr. 3), 22–24.
15. Trombly, M. (2000). "Top U.S. Bank to Open B-to-B Marketplace." *Computerworld,* 34 (15, Apr. 10), 6.
16. Vizard, M. (2000). "Business-to-Business-to-Consumer Signals the Next Generation of e-Business." *InfoWorld,* 22 (14, Apr. 3), 111.
17. Walker, L. (2000). "B2B: Almost as Old as the Internet." *The Washington Post,* April 5, p. G03.
18. Wilson, T., and Mullen, T. (2000). "E-Business Exchanges Fight for Survival." *InternetWeek* (August).

ABOUT THE AUTHOR

Marie Tumolo is Assistant Professor at California State University, Fullerton.

Chapter 3
The ASP Model: Evolution and Future Challenges

John Davies

The increasingly competitive global marketplace requires organizations to be extremely *agile*, i.e., to be able to quickly and efficiently respond to changes, such as a merger, competitive threat, necessity to deliver new products with rapid development cycles, or driving costs down.

The role of information technology (IT) has become important in achieving this agility. Business via the Internet has grown exponentially.[1] Technology is enabling companies to become more efficient, to achieve a broader and more effective market reach. It is becoming the lifeblood of many organizations from both a revenue generation and cost-control perspective. No longer behind the scenes where system failures were considered an inconvenience, IT has achieved very high profile. Unavailability, for any reason, can severely impact an organization, and have direct impact on revenue and in some cases stock prices.[2] The pressure on IT departments to deliver ever-higher levels of performance and reliability has increased significantly.

The application service provider (ASP) model[3] has emerged as a way to deliver software applications to organizations over the Internet efficiently, reliably, and at low cost to the end user. These applications range from office suites to enterprise resource management (ERP) and eprocurement systems. This provides a promising role for the ASP model to allow corporate IT departments to successfully outsource components of their IT application and technology infrastructure and to enhance organizational agility and cost-effectiveness. This chapter briefly describes the ASP model and its evolution and future, and suggests that there remain some challenges to overcome before it can claim to be a successfully mature and mainstream paradigm.

0-8493-1161-6/02/$0.00+$1.50

WHAT IS THE ASP MODEL? WHY HAS IT EMERGED?

Outsourcing IT data processing is an old concept, dating back to the mainframe "service bureaus." The ASP model clearly has its roots here, although it has extended the concept. As the ubiquity of PC with browser and Internet connectivity has increased, and internal IT resources (often scarce) are getting more expensive, companies are reconsidering outsourcing IT tasks that fall outside the business' core competencies. The Internet, high-speed networks and "thin-client" computing that allow software applications to be provided over the Internet have been the enabling technologies behind the growth in ASPs. But more than being just a technology-driven shift, current management thinking about the extended enterprise, reduction in capital investments, and focus on core competencies has also been an important factor in the reinvigorated interest in this form of outsourcing.

ASPs develop and deliver a service shared by multiple customers, provide these services for a usage-based fee, and supply these services from a central location, over the public Internet or a virtual private network (VPN). These providers often emerge as spin-offs from IT services organizations, enterprise software providers, and Internet hosting companies.

Businesses that use the services of ASPs can realize a number of benefits. A monthly fee based on usage increases predictability of cash outflows and reduces the risk of high up-front investments in application licenses or the technical environment (and its high rate of obsolescence) to run it. It also reduces the internal costs and difficulties in recruiting and retaining short-supply labor skills. Organizations can focus their labor pool, their capital investments, and their management attention to the heart of their business — what they do best. Because the ASP has already deployed the technology, implementation timeframes are typically much shorter than from-scratch internal projects and hence benefits can be realized sooner.

ASP Solution Providers — Playing Various Roles

There are a number of roles played by companies delivering ASP solutions:

- *Application Service Managers (aka Aggregators)* — sell, manage, customize, and provide support for packaged application solutions (those they develop as a solution developer, if any, plus those from other application service developers; a company that does not develop, but just aggregates third party solutions, is often referred to as a "Pure Play").
- *Application Service Developers* — focus on the development of thin-client application software, or application components that can be accessed as services over the Internet.

- *Application Aggregators Platform Enablers* — own and operate commercial data centers; offer managed hosting platforms for their applications or that third-party ASPs can use.
- *Agents* — provide consulting and systems integration.

Telcos also play a key role in delivering ASP solutions. Some research organizations have included this group in the ASP model.

As will be discussed later, while ASPs often play multiple roles, it is increasingly likely that the end-to-end application service will be provided by more than one player, each being an ASP in its own right. From the client's perspective, however, single point of contact and responsibility for the application service is preferred. This is, or should be, the role of the application service manager.

THE ASP MARKET

In 1999 the worldwide ASP market[4] was forecast to grow from $1 billion in 1999 to $3.6 billion in 2000, to $14.7 billion[5] by 2003, and to more than $25.3 billion by 2004. The U.S. market had revenue[6] of $1.4 billion in 2000, with no vendor able to claim market dominance.

Small and mid-sized companies[7] currently represent the majority of ASP users, since they are less likely to have a substantial IT staff, but by 2004, larger companies with more than 2500 employees are expected to account for 56 percent of the ASP market.

Larger businesses are recognizing the value of selectively adopting the ASP model and applying it to their existing legacy applications. 19 percent of large enterprises (500 to 100,000 employees) currently use ASPs for internal applications, while 7 percent use ASPs for E-commerce applications. By 2004, 65 percent of large corporations are expected to use ASPs for internal applications, and 72 percent for E-commerce.

North America has led the worldwide ASP market (65 percent of revenue in 1999), but the market will become much more global. In 2004, the North America region is forecast to represent 45 percent of ASP revenue. Europe accounted for 20 percent of the ASP market in 1999 and is projected to have 32 percent in 2004.

INDUSTRY SUPPORT OF THE ASP MODEL

Many major software and hardware companies are building the infrastructure and applications to support the evolving ASP model. Sun, IBM, Oracle, Novell, Intel, SAP, Hewlett-Packard, Microsoft, and a host of others have embraced the ASP model. Microsoft, with its strategy to evolve its DNA architecture to an XML-based .NET architecture, described below, is certainly a major proponent of the applications as services approach what is the cornerstone of the ASP model. Microsoft announced its Complete Commerce program (1999) as the result of an ASP pilot project. Microsoft is seeding ASP hosting of Microsoft Exchange Server messaging, Microsoft Office 2000 collaboration tools, corporate purchasing, media streaming, and LOB application services on Windows 2000 Server. They have recently[8] launched the Web Developers Community to lead the independent software vendors (ISVs) through transition to software as a service in .NET Environment. Microsoft ASP Certification will provide market recognition for industry partners who demonstrate a consistent, high quality delivery of specific hosted or outsourcing services built on Microsoft technology. Sun is also offering a similar certification program. In each case, the purpose is to foster confidence in the ASP model.

Microsoft's ASP Vision: Web Services and the .NET Architecture

Microsoft's recently announced .Net Strategy is based on the vision that applications should be provided as Web services, available anywhere, anytime, and on any device (PC, cell phone, or PDA). Based on the Web services model, the .NET platform will enable corporate applications to be managed locally, while the services to support them (user authentication, file storage, user preference management, calendaring, mail, etc.) can be seamlessly accessed. The ability to seamlessly combine internal and external services will enable simplified creation of applications that bring together corporate data with associated data from vendors and partners, resulting in an unprecedented level of functionality and a much improved user experience. IT professionals will be able to focus more on delivering value to their businesses without being burdened with nonvalue-added tasks.

The .NET Architecture. The Microsoft .NET architecture provides the key components needed to create and leverage Web services. All the software components are built from the ground up for interoperability, supporting XML and SOAP as the glue enabling applications to communicate.

The main elements in the .NET architecture are:

- *The .NET Framework and Visual Studio.NET* — will facilitate component development and integration of these Web services into applications by calling APIs.
- *.NET Servers* — are infrastructure services that most applications will need: Windows 2000, SQL Server 2000, BizTalk Server 2000, and others.
- *Building Block Services* — will provide a range of basic service components that will be available to facilitate the Web experience. For example, Microsoft Passport.NET will enable users to access Web services by providing a common broker that authenticates the user and decides what information they can access.

- *.NET Device Software* — client devices from PCs to cellular phones and pagers will have the functionality of full use of a device's capabilities.

Microsoft has been investing in making its own products more reliable, scalable, and easy to integrate. At the same time, Microsoft is working to enable partners in the ASP industry to connect with one another, and to work together in reaching out to customers.

Many new applications are now created as true Web services. Microsoft has stated it is "betting the business" on the strategy of interconnected and interchangeable Web services, and if this model is to be successful, solution providers will have to start building their point solutions based on this technology.

Microsoft's strategy and .NET server and development products will possibly take a dominant role in ASP solutions development marketplace. Microsoft's .NET application development platform will turn "the world's largest software vendor into the Internet's biggest application service provider."

EXAMPLES OF ASP SERVICES APPLICATIONS

There are many examples of applications that are being offered as outsourced ASP services, available over the Internet. Many vendors of client/server "fat client" software that is traditionally licensed and installed at user premises are beginning to offer their software as services via the ASP subscription or pay-per-use model. This includes everything from ERP (financial, manufacturing) software from SAP, Oracle, PeopleSoft, and others to CRM systems from Siebel and Pivotal, to Supply Chain Management software from such companies as Descartes and QLogitek, to office suites and other desktop applications from Microsoft and others. Example ASP solutions are described below.

E-Procurement/Supply Chain Management

Supply Chain Management is perhaps the best example of a B2B service offered over the Web. The ASP hosts a trading exchange through a portal that is developed by an application service developer. This essentially brings major buyers (hubs) and their suppliers (spokes) to a common portal. Browser-based functionality offered to the trading partner varies, but at its heart consists of establishing electronic trading profiles, sourcing of goods, issuing purchase orders, delivery fulfillment (e.g., advanced ship notice, logistics) and payment through invoices, evaluated receipt settlement, or EFT. EDI or non-EDI documents and data are securely exchanged over the public Internet (SSL, certificates) or Virtual Private Network (VPN-IPsec). Trading partners typically pay a subscription or transaction fee for use of the portal.

The ASP builds and hosts the trading exchange using such products as Microsoft Windows 2000/Active Directory, Commerce Server 2000 (profiling and catalog management), SQLServer 2000 (DBMS, data mining), and BizTalk Server 2000. BizTalk Server is a key product in Microsoft's .NET arsenal, advancing enterprise application to facilitate the seamless connectivity of both buyer and suppliers' back-end systems to the ASP's exchange portal.

Bringing Host Applications to Web Services

Not all ASP solutions are new stand-alone applications. Existing investments in mainframe and other legacy systems can be maintained by an enterprise while using an externally hosted ASP to bring a back-office legacy application to a portal as a value-added B2B Web service. This is accomplished by having an Application Service Developer design a Web interface with specific new functional workflow and database that then resides with the ASP. The browser application accesses the enterprise host data, maintaining transactional consistency across the platforms, without changes or disruption to the existing host application or its environment.

Microsoft's Host Integration Server (HIS), with its built-in XML support, is one product that provides the infrastructure for host communication and data-level access to common host databases (e.g., DB2). Other Microsoft .NET server products would be deployed by the ASP as well (such as Windows 2000, SQL Server 2000, Exchange 2000) depending upon the application, but HIS would be the key technology facilitating host-to-Web integration. The result is an extended Web service application hosted by the ASP, which provides a B2B service at a fraction of the cost of rewriting the entire application or having the enterprise invest in the new technology platforms.

Web Services Components: Hailstorm

Developing Web services which can be used by others as part of a larger application involves creating programmable components that are accessible and can be activated over the Web with a simple XML/SOAP-based message. An HTTP request is made of the component and it responds via XML. SOAP is the dialect used to transact the service. WSDL (Web Services Description Language) is used to describe the service in terms of formats and ordering of messages. UDDI (Universal Description, Design, and Integration) is the emerging industry standard directory for finding Web services. Collectively, this is the infrastructure for true Web services.

Hailstorm (code name) is Microsoft's core set of user-centric XML Web services, which will become building block services for other applications. Microsoft will provide this infrastructure and a standard set of "identity-based" services, such as address, profile, calendar, inbox, contacts, location, favorites, and will work with other business and industry groups to create new services to add to this .NET infrastructure. An ASP (developer) can create an application in less time by leveraging the hailstorm identity-based services, when permitted by the individual (for example, federating security services into an application by using Passport authentication services). In this way, various Web services effectively collaborate to produce an efficient Web application that provides a richer user experience.

THE FUTURE: ASP CHALLENGES

The ASP model is evolving and maturing. Some rationalization will take place. This is natural and from a customer perspective, the only issue that should be of concern is whether the ASP you select will stick (while there have been a few high profile ASP failures, these have been relatively few and, in most cases, the customer has managed to switch to another ASP without more than modest financial inconvenience). To a lesser extent, the stickiness of the overall ASP industry is no slam-dunk, but if the ASP community contains its irrational exuberance and addresses the challenges logically it is more a question of the sharpness of the "hockey stick" growth curve rather than whether the growth will be there at all.

So what are these challenges? They differ somewhat between the United States and Canada, but here is a partial list that generally applies and should be illustrative.

- Many companies remain unaware of the ASP model. Others are generally skeptical that the ASP model makes sense and is viable. Education and evangelism will be required for some time. This requires patience. This requires cash, particularly for the high quality of service infrastructure providers. The dot.com fallout has made cash more scarce. Have a well-defined market-segmented business plan (am I going to be an ASP developer, aggregator, infrastructure provider, or all the above) and be prepared to commit to flawless execution. Be prepared to be flexible and reinvent yourself (things are evolving, after all). The softening economy will likely have a negative impact on the ASP adoption rate forecasts cited earlier (references typically pre-January 2001); however, the ASP model is intended to reduce risky upfront capital investments and generally reduce costs. It is an approach that has appeal where a focus on costs is increasing in importance.

  Note: There has been some heavy investing into the construction of hosting facilities and infrastructure. The big players have targeted the high-end market where quality of service is critical and pockets are relatively deep. The mission-critical LOB applications will rely on these big players. Since many firms cannot justify or afford the high costs of 5–9's service and need a more economical solution, a niche for smaller ASPs providing a more basic level of hosting service definitely exists. Big and small players will try to establish their respective beachheads, then likely battle for the mid-market.

- Many companies do not trust running their business applications and storing their data outside their premises. If your solution is not secure, you will not be successful. But more than that, you have to convince those who are paid to fret about this sort of thing that all that security you have invested in is as good as (normally much better than) what they can provide themselves. In addition, aside from the obvious requirement to have a secure and fully redundant architecture, achieving extremely high levels of availability requires state-of-art processes, such as change and configuration management. ASPs would do well to invest in process management as well as infrastructure, and consider a certification program to reassure customers that they have focused on both. Sun, Microsoft, and others have recently initiated ASP certification programs to help establish credibility for those ASPs that have achieved best practices in managing all facets of their services.
- Many IT departments have no interest whatsoever in a model that is perceived to move the good stuff out of their domain. If it smells like outsourcing, and of course a goodly part of the ASP model is outsourcing, then many IT departments will resist. The economics, security, reliability, and other factors need to be very compelling, given this natural resistance. The role of IT departments is slowly evolving to focusing on being good IT strategists, good IT services shoppers, and good IT program managers, as opposed to trying to be completely self-sufficient. But this evolution is slow.
- To date, the ASP model has not completely adopted a single point-of-responsibility approach; it remains fragmented. Customers must often manage multiple relationships for delivery of services. Managing multiple relationships sounds like a lot of work. It is. Customers are going to want (or should want) someone to assume complete responsibility for the service. If you are an ASP providing a service that you want customers to use, figure out how to ensure that someone (you or someone else) will assume complete responsibility for the solution on behalf of the customer. Someone needs to be the Application Service Manager and the buck stops there. This is a lesson that smart IT shops have learned regarding their internal customers. ASPs that learn from this lesson will satisfy their customers. Service Level Agreements (SLAs) are important and ASPs must develop solid metrics around availability, reliability, security, performance, and customer service response.

CONCLUSION

On balance, the ASP model will be an exciting evolution in the advancement of the overall delivery of IT solutions, based on a maturing model of outsourcing which recognizes the value of sticking to one's core competencies and shopping for expertise that economies of scale will allow to be provided better outside your organization. At the same time ASP outsourcing will reduce dependence on capital expenditures and getting solutions implemented faster and more reliably. The phenomenal advancements in Internet and other enabling technologies such as XML and SOAP and the relatively unanimous support of the ASP model provided by the major industry players to promote software as Web services are all factors that favor this model becoming mainstream. ASPs have some challenges if they collectively want this model to advance faster rather than slower and of course it will be fiercely competitive so they also have challenges to sustain themselves as profitable entities. All in all, success is predicted and early adopters will reap benefits sooner than their competition.

References

1. Application Service Providers: Current Status and Future Trends, Report by International Engineering Consortium. http://www.iec.org/pubs/asp_status.html
2. Application Service Providers: Evolution and Resources. http://www.microsoft.com/ISN/downloads/ASP_Evolution_Resources.doc
3. A lot of to-date information on the ASP model can be found at: http://www.aspnews.com/, http://www.aspindustry.org/, and http://www.aspscope.com
4. ASP Market to Reach $25.3 Billion by 2004. http://cyberatlas.internet.com/big_picture/hardware/article/0,1323,5921_434091,00.html
5. Great ASP Aspirations. http://msdn.microsoft.com/library/periodic/period00/asp.htm
6. http://biz.yahoo.com/bw/010516/2037.html
7. ASPs Must Reach Out to Larger Companies. http://cyberatlas.internet.com/big_picture/hardware/article/0,1323,5921_475581,00.html
8. Microsoft Launches Web Developers Community to Lead ISVs Through Transition to Software as a Service. http://www.microsoft.com/PressPass/press/2001/Apr01/04-26DevCommunityPR.asp

ABOUT THE AUTHOR

John Davies is Vice President, Systems Development, with QLogitek.

Chapter 4

Enclaves: The Enterprise as an Extranet

Bryan T. Koch

Even in the most secure organizations, information security threats and vulnerabilities are increasing over time. Vulnerabilities are increasing with the complexity of internal infrastructures; complex structures have more single points of failure, and this in turn increases the risk of multiple simultaneous failures. Organizations are adopting new, untried, and partially tested products at ever-increasing rates. Vendors and internal developers alike are relearning the security lessons of the past — one at a time, painful lesson by painful lesson.

Given the rapid rate of change in organizations, minor or incremental improvements in security can be offset or undermined by "organizational entropy." The introduction of local area networks (LANs) and personal computers (PCs) years ago changed the security landscape, but many security organizations continued to function using centralized control models that have little relationship to the current organizational or technical infrastructures. The Internet has brought new threats to the traditional set of organizational security controls. The success of the Internet model has created a push for electronic commerce (E-commerce) and electronic business (E-business) initiatives involving both the Internet itself and the more widespread use of Internet Protocol (IP)-based extranets (private business-to-business networks).

Sophisticated, effective, and easy-to-use attack tools are widely available on the Internet. The Internet has implicitly linked competing organizations with one another, and linked these organizations to communities that are opposed to security controls of any kind. There is no reason to assume that attack tools developed in the Internet cannot or will not be used within an organization.

0-8493-1161-6/02/$0.00+$1.50

External threats are more easily perceived than internal threats, while surveys and studies continue to show that the majority of security problems are internal. With all of this as context, the need for a new security paradigm is clear.

The time has come to apply the lessons learned in Internet and extranet environments to one's own organization. This chapter proposes to apply Internet/extranet security architectural concepts to internal networks by creating protected *enclaves* within organizations. Access between enclaves and the enterprise is managed by *network guardians*. Within enclaves, the security objective is to apply traditional controls consistently and well. Outside of enclaves, current practice (i.e., security controls at variance with formal security policies) is tolerated (one has no choice). This restructuring can reduce some types of network security threats by orders of magnitude. Other threats remain and these must be addressed through traditional security analysis and controls, or accepted as part of normal risk/reward trade-offs.

SECURITY CONTEXT

Security policies, procedures, and technologies are supposed to combine to yield acceptable risk levels for enterprise systems. However, the nature of security threats, and the probability that they can be successfully deployed against enterprise systems, have changed. This is partly a result of the diffusion of computer technology and computer networking into enterprises, and partly a result of the Internet.

For larger and older organizations, security policies were developed to address security vulnerabilities and threats in legacy mainframe environments. Legacy policies have been supplemented to address newer threats such as computer viruses, remote access, and e-mail. In this author's experience, it is rare for current policy frameworks to effectively address network-based threats. LANs and PCs were the first steps in what has become a marathon of increasing complexity and inter-relatedness; intranet (internal networks and applications based on IP), extranet, and Internet initiatives are the most common examples of this.

The Internet has brought network technology to millions. It is an enabling infrastructure for emerging E-business and E-commerce environments. It has a darker side, however, because it also:

- Serves as a "proving ground" for tools and procedures that test for and exploit security vulnerabilities in systems
- Serves as a distribution medium for these tools and procedures
- Links potential users of these tools with anonymously available repositories

Partly because it began as an "open" network, and partly due to the explosion of commercial use, the Internet has also been the proving ground for security architectures, tools, and procedures to protect information in the Internet's high-threat environment. Examples of the tools that have emerged from this environment include firewalls, virtual private networks, and layered physical architectures. These tools have been extended from the Internet into extranets.

In many sectors — most recently telecommunications, finance, and health care — organizations are growing primarily through mergers and acquisitions. Integration of many new organizations per year is challenging enough on its own. It is made more complicated by external network connectivity (dial-in for customers and employees, outbound Internet services, electronic commerce applications, and the like) within acquired organizations. It is further complicated by the need to integrate dissimilar infrastructure components (e-mail, calendaring, and scheduling; enterprise resource planning (ERP); and human resources (HR) tools). The easiest solution — to wait for the dust to settle and perform long-term planning — is simply not possible in today's "at the speed of business" climate.

An alternative solution, the one discussed here, is to accept the realities of the business and technical contexts, and to create a "network security master plan" based on the new realities of the internal threat environment. One must begin to treat enterprise networks as if they were an extranet or the Internet and secure them accordingly.

THE ONE BIG NETWORK PARADIGM

Network architects today are being tasked with the creation of an integrated network environment. One network architect described this as a mandate to "connect everything to everything else, with complete transparency." The author refers to this as the One Big Network paradigm. In this author's experience, some network architects aim to keep security at arm's length — "we build it, you secure it, and we don't have to talk to each other." This is untenable in the current security context of rapid growth from mergers and acquisitions.

One Big Network is a seductive vision to network designers, network users, and business executives alike. One Big Network will — in theory — allow new and better business interactions with suppliers, with business customers, and with end-consumers. Everyone connected to One Big Network can — in theory — reap great benefits at minimal infrastructure cost. Electronic business-to-business and electronic-commerce will be — in theory — ubiquitous.

However, one critical element has been left out of this brave new world: security. Despite more than a decade of networking and personal computers, many organizational security policies continue to target the legacy environment, not the network as a whole. These policies assume that it is possible to secure stand-alone "systems" or "applications" as if they have an existence independent of the rest of the enterprise. They assume that attackers will target applications rather than the network infrastructure that links the various parts of the distributed application together. Today's automated attack tools target the network as a whole to identify and attack weak applications and systems, and then use these systems for further attacks.

One Big Network changes another aspect of the enterprise risk/reward equation: it globalizes risks that had previously been local. In the past, a business unit could elect to enter into an outsource agreement for its applications, secure in the knowledge that the risks related to the agreement affected it alone. With One Big Network, the risk paradigm changes. It is difficult, indeed inappropriate, for business unit management to make decisions about risk/reward trade-offs when the risks are global while the benefits are local.

Finally, One Big Network assumes consistent controls and the loyalty of employees and others who are given access. Study after study, and survey after survey, confirm that neither assumption is viable.

NETWORK SECURITY AND THE ONE BIG NETWORK PARADIGM

It is possible that there was a time when One Big Network could be adequately secured. If it ever existed, that day is long past. Today's networks are dramatically bigger, much more diverse, run many more applications, connect more divergent organizations, in a more hostile environment where the "bad guys" have better tools than ever before. The author believes that it is not possible to secure, to any reasonable level of confidence, any enterprise network for any large organization where the network is managed as a single "flat" network with "any-to-any" connectivity.

In an environment with no effective internal network security controls, each network node creates a threat against every other node. (In mathematical terms, where there are n network nodes, the number of threats is approximately n^2.) Where the organization is also on the Internet without a firewall, the effective number of threats becomes essentially infinite (see Exhibit 1).

Effective enterprise security architecture must augment its traditional, applications-based toolkit with *network-based tools* aimed at addressing network-based threats.

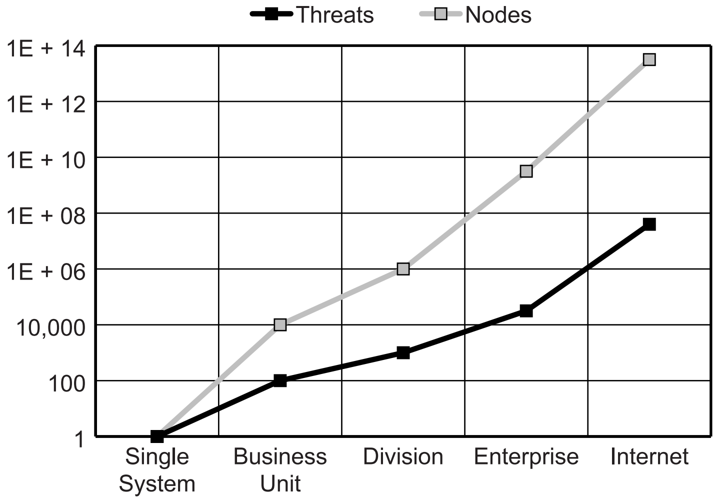


Exhibit 1. Network Threats (log scale)

INTERNET SECURITY ARCHITECTURE ELEMENTS

How does one design differently for Internet and extranet than one did for enterprises? What are Internet/extranet security engineering principles?

- *Simplicity.* Complexity is the enemy of security. Complex systems have more components, more single points of failure, more points at which failures can cascade upon one another, and are more difficult to certify as "known good" (even when built from known good components, which is rare in and of itself).
- *Prioritization and valuation.* Internet security systems know what they aim to protect. The sensitivity and vulnerability of each element is understood, both on its own and in combination with other elements of the design.
- *Deny by default, allow by policy.* Internet security architectures begin with the premise that all traffic is to be denied. Only traffic that is explicitly required to perform the mission is enabled, and this through defined, documented, and analyzed pathways and mechanisms.
- *Defense in depth, layered protection.* Mistakes happen. New flaws are discovered. Flaws previously believed to be insignificant become important when exploits are published. The Internet security architecture must, to a reasonable degree of confidence, fail in ways that result in continued security of the overall system; the failure (or misconfiguration) of a single component should not result in security exposures for the entire site.

- *End-to-end, path-by-path analysis.* Internet security engineering looks at all components, both on the enterprise side and on the remote side of every transaction. Failure or compromise of any component can undermine the security of the entire system. Potential weak points must be understood and, if possible, managed. Residual risks must be understood, both by the enterprise and by its business partners and customers.
- *Encryption.* In all Internet models, and most extranet models, the security of the underlying network is not assumed. As a result, some mechanism — encryption — is needed to preserve the confidentiality of data sent between the remote users and enterprise servers.
- *Conscious choice, not organic growth.* Internet security architectures are formally created through software and security engineering activities; they do not "just happen."

THE ENCLAVE APPROACH

This chapter proposes to treat the enterprise as an extranet. The extranet model invokes an architecture that has security as its first objective. It means identifying what an enterprise genuinely cares about: what it lives or dies by. It identifies critical and securable components and isolates them into protected *enclaves*. Access between enclaves and the enterprise is managed by *network guardians*. Within enclaves, the security objective is to apply traditional controls consistently and well. Outside of enclaves, current practice (i.e., security controls at variance with formal security policies), while not encouraged, is acknowledged as reality. This restructuring can reduce some types of network security threats by orders of magnitude. Taken to the extreme, all business-unit-to-business-unit interactions pass through enclaves (see Exhibit 2).

ENCLAVES

The enclaves proposed here are designed to contain high-value securable elements. Securable elements are systems for which security controls consistent with organizational security objectives can be successfully designed, deployed, operated, and maintained at any desired level of confidence. By contrast, nonsecurable elements might be semi-autonomous business units, new acquisitions, test labs, and desktops (as used by telecommuters, developers, and business partners) — elements for which the cost, time, or effort required to secure them exceeds their value to the enterprise.

Within a secure enclave, every system and network component will have security arrangements that comply with the enterprise security policy and industry standards of due care. At enclave boundaries, security assurance will be provided by network guardians whose rule sets and

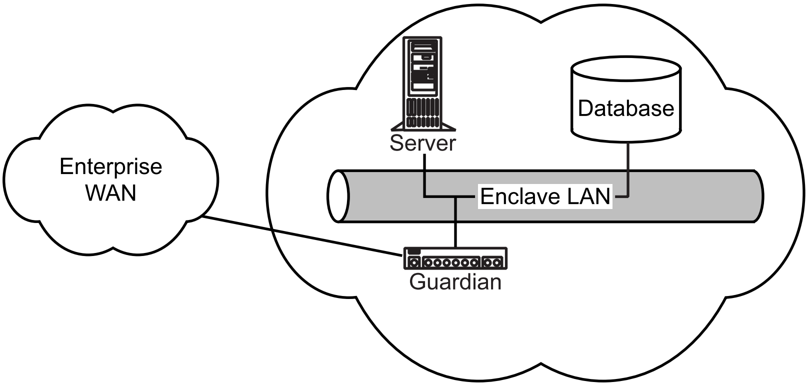


Exhibit 2. Relationship of an Enclave to the Enterprise

operational characteristics can be enforced and audited. In other words, there is some level of assurance that comes from being part of an enclave. This greatly simplifies the security requirements that are imposed on client/server architectures and their supporting applications programming interfaces (APIs). Between enclaves, security assurance will be provided by the application of cryptographic technology and protocols.

Enclave membership is earned, not inherited. Enclave networks may need to be created from the ground up, with existing systems shifted onto enclave networks when their security arrangements have been adequately examined.

Enclaves could potentially contain the elements listed below:

1. Mainframes
2. Application servers
3. Database servers
4. Network gateways
5. PKI certificate authority and registration authorities
6. Network infrastructure components (domain name and time servers)
7. Directories
8. Windows "domain controllers"
9. Approved intranet web servers
10. Managed network components
11. Internet proxy servers

All these are shared and securable to a high degree of confidence.

NETWORK GUARDIANS

Network guardians mediate and control traffic flow into and out of enclaves. Network guardians can be implemented initially using network routers. The routers will isolate enclave local area network traffic from LANs used for other purposes (development systems, for example, and user desktops) within the same physical space. This restricts the ability of user desktops and other low-assurance systems to monitor traffic between remote enclave users and the enclave. (Users will still have the ability to intercept traffic on their own LAN segment, although the use of switching network hubs can reduce the opportunity for this exposure as well.)

The next step in the deployment of network guardians is the addition of access control lists (ACLs) to guardian routers. The purpose of the ACLs is similar to the functionality of "border routers" in Internet firewalls — screening incoming traffic for validity (anti-spoofing), screening the destination addresses of traffic within the enclave, and to the extent possible, restricting enclave services visible to the remainder of the enterprise to the set of intended services.

Decisions to implement higher levels of assurance for specific enclaves or specific enclave-to-enclave or enclave-to-user communications can be made based on later risk assessments. Today and for the near future, simple subnet isolation will suffice.

ENCLAVE BENEFITS

Adopting an enclave approach reduces network-based security risks by orders of magnitude. The basic reason is that in the modern enterprise, the number of nodes (n) is very large, growing, and highly volatile. The number of enclaves (e) will be a small, stable number. With enclaves, overall risk is on the order of $n \times e$, compared with $n \times n$ without enclaves. For large n, $n \times e$ is much smaller than $n \times n$.

Business units can operate with greater degrees of autonomy than they might otherwise be allowed, because the only data they will be placing at risk is their own data on their own networks. Enclaves allow the realignment of risk with reward. This gives business units greater internal design freedom.

Because they require documentation and formalization of network data flows, the presence of enclaves can lead to improved network efficiency and scalability. Enclaves enforce an organization's existing security policies, at a network level, so by their nature they tend to reduce questionable, dubious, and erroneous network traffic and provide better accounting for allowed traffic flows. This aids capacity planning and disaster planning functions.

By formalizing relationships between protected systems and the remainder of the enterprise, enclaves can allow faster connections to business partners. (One of the significant sources of delay this author has seen in setting up extranets to potential business partners is collecting information about the exact nature of network traffic, required to configure network routers and firewalls. The same delay is often seen in setting up connectivity to newly acquired business units.)

Finally, enclaves allow for easier allocation of scarce security resources where they can do the most good. It is far easier to improve the security of enclave-based systems by, say, 50 percent, than it is to improve the overall security of all desktop systems in the enterprise by a similar amount, given a fixed resource allocation.

LIMITATIONS OF ENCLAVES

Enclaves protect only the systems in them; and by definition, they exclude the vast majority of the systems on the enterprise network and all external systems. Some other mechanism is needed to protect data in transit between low-assurance (desktops, external business partner) systems and the high-assurance systems within the enclaves. The solution is a set of confidentiality and authentication services provided by encryption. Providing an overall umbrella for encryption and authentication services is one role of public key infrastructures (PKIs).

From a practical perspective, management is difficult enough for externally focused network guardians (those protecting Internet and extranet connectivity). Products allowing support of an enterprisewide set of firewalls are just beginning to emerge. Recent publicity regarding Internet security events has increased executive awareness of security issues, without increasing the pool of trained network security professionals, so staffing for an enclave migration may be difficult.

Risks remain, and there are limitations. Many new applications are not "firewall friendly" (e.g., Java, CORBA, video, network management). Enclaves may not be compatible with legacy systems. Application security is just as important — perhaps more important than previously — because people connect to the application. Applications, therefore, should be designed securely. Misuse by authorized individuals is still possible in this paradigm, but the enclave system controls the path they use. Enclave architecture is aimed at network-based attacks, and it can be strengthened by integrating virtual private networks (VPNs) and switching network hubs.

IMPLEMENTATION OF ENCLAVES

Enclaves represent a fundamental shift in enterprise network architecture. Stated differently, they re-apply the lessons of the Internet to the enterprise. Re-architecting cannot happen overnight. It cannot be done on a cookie-cutter, by-the-book basis. The author's often-stated belief is that "security architecture" is a verb; it describes a *process*, rather than a destination. How can an organization apply the enclave approach to its network security problems? In a word, planning. In a few more words, information gathering, planning, prototyping, deployment, and refinement. These stages are described more fully below.

Information Gathering

Information is the core of any enclave implementation project. The outcome of the information-gathering phase is essentially an inventory of critical systems with a reasonably good idea of the sensitivity and criticality of these systems. Some readers will be fortunate enough to work for organizations that already have information systems inventories from the business continuity planning process, or from Year 2000 activities. A few will actually have accurate and complete information. The rest will have to continue on with their research activities.

The enterprise must identify candidate systems for enclave membership and the security objectives for candidates. A starting rule-of-thumb would be that no desktop systems, and no external systems, are candidates for enclave membership; all other systems are initially candidates. Systems containing business-critical, business-sensitive, legally protected, or highly visible information are candidates for enclave membership. Systems managed by demonstrably competent administration groups, to defined security standards, are candidates.

External connections and relationships, via dial-up, dedicated, or Internet paths, must be discovered, documented, and inventoried.

The existing enterprise network infrastructure is often poorly understood and even less well documented. Part of the information-gathering process is to improve this situation and provide a firm foundation for realistic enclave planning.

Planning

The planning process begins with the selection of an enclave planning group. Suggested membership includes senior staff from the following organizations: information security (with an emphasis on network security and business continuity specialists), network engineering, firewall management, mainframe network operations, distributed systems or client/server operations, E-commerce planning, and any outsource partners from these organizations. Supplementing this group would be technically well-informed representative from enterprise business units.

The planning group's next objective is to determine the scope of its activity, answering a set of questions including at least:

- Is one enclave sufficient, or is more than one a better fit with the organization?
- Where will the enclaves be located?
- Who will manage them?
- What level of protection is needed within each enclave?
- What is the simplest representative sample of an enclave that could be created within the current organization?

The purpose of these questions is to apply standard engineering practices to the challenge of carving out a secure enclave from the broader enterprise, and to use the outcome of these practices to make a case to enterprise management for the deployment of enclaves.

Depending on organizational readiness, the planning phase can last as little as a month or as long as a year, involving anywhere from days to years of effort.

Prototyping

Enclaves are not new; they have been a feature of classified government environments since the beginning of computer technology (although typically within a single classification level or compartment). They are the basis of essentially all secure Internet E-commerce work. However, the application of enclave architectures to network security needs of large organizations is, if not new, at least not widely discussed in the professional literature. Further, as seen in Internet and extranet environments generally, significant misunderstandings can often delay deployment efforts, and efforts to avoid these delays lead either to downward functionality adjustments, or acceptance of additional security risks, or both.

As a result, prudence dictates that any attempt to deploy enclaves within an enterprise be done in a stepwise fashion, compatible with the organization's current configuration and change control processes. The author recommends that organizations considering the deployment of the enclave architecture first evaluate this architecture in a prototype or laboratory environment. One option for doing this is an organizational test environment. Another option is the selection of a single business unit, district, or regional office.

Along with the selection of a locale and systems under evaluation, the enterprise must develop evaluation criteria: what does the organization expect to learn from the prototype environment, and how can the organization capture and capitalize on learning experiences?

Deployment

After the successful completion of a prototype comes general deployment. The actual deployment architecture and schedule depends on factors too numerous to mention in any detail here. The list includes:

- *The number of enclaves.* (The author has worked in environments with as few as one and as many as a hundred potential enclaves.)
- *Organizational readiness.* Some parts of the enterprise will be more accepting of the enclave architecture than others. Early adopters exist in every enterprise, as do more conservative elements. The deployment plan should make use of early adopters and apply the lessons learned in these early deployments to sway or encourage more change-resistant organizations.
- *Targets of opportunity.* The acquisition of new business units through mergers and acquisitions may well present targets of opportunity for early deployment of the enclave architecture.

Refinement

The enclave architecture is a concept and a process. Both will change over time: partly through organizational experience and partly through the changing technical and organizational infrastructure within which they are deployed.

One major opportunity for refinement is the composition and nature of the network guardians. Initially, this author expects network guardians to consist simply of already-existing network routers, supplemented with network monitoring or intrusion detection systems. The router will initially be configured with a minimal set of controls, perhaps just anti-spoofing filtering and as much source and destination filtering as can be reasonably considered. The network monitoring system will allow the implementers to quickly learn about "typical" traffic patterns, which can then be configured into the router. The intrusion detection system looks for known attack patterns and alerts network administrators when they are found (see Exhibit 3).

In a later refinement, the router may well be supplemented with a firewall, with configuration rules derived from the network monitoring results, constrained by emerging organizational policies regarding authorized traffic (see Exhibit 4).

Still later, where the organization has more than one enclave, encrypted tunnels might be established between enclaves, with selective encryption of traffic from other sources (desktops, for example, or selected business partners) into enclaves. This is illustrated in Exhibit 5.

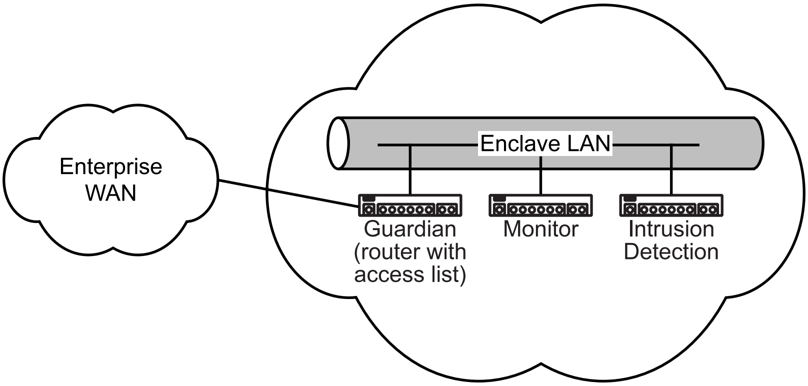


Exhibit 3. Initial Enclave Guardian Configuration

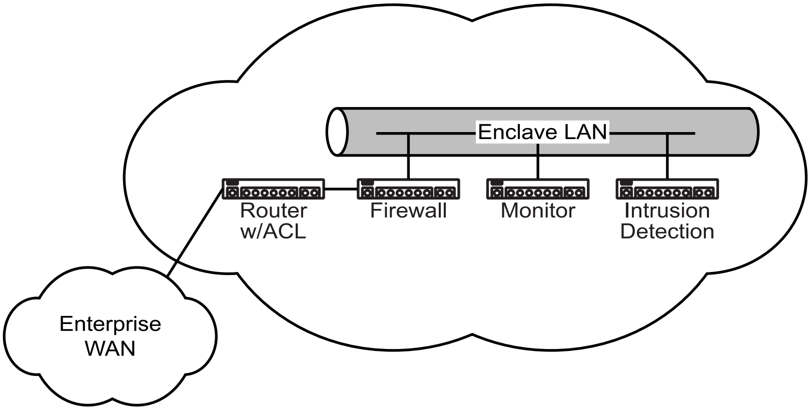


Exhibit 4. Enclave with Firewall Guardian

CONCLUSION

The enterprise-as-extranet methodology gives business units greater internal design freedom without a negative security impact on the rest of the corporation. It can allow greater network efficiency and better network disaster planning because it identifies critical elements and the pathways to them. It establishes security triage. The net results are global threat reduction by orders of magnitude and improved, effective real-world security.

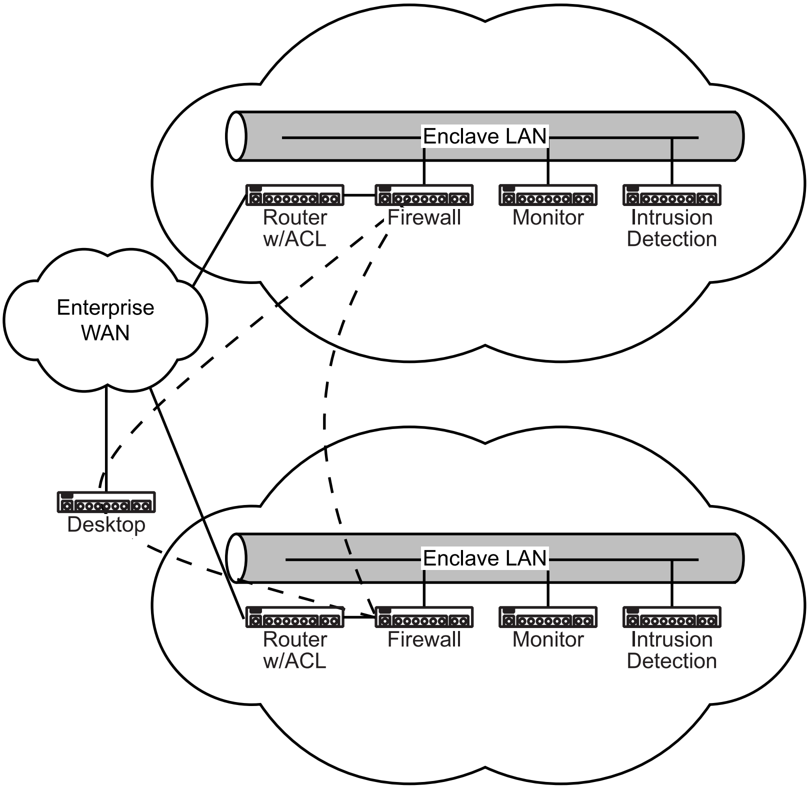


Exhibit 5. Enclaves with Encrypted Paths (dashed lines)

ABOUT THE AUTHOR

Bryan T. Koch, CISSP, is principal security architect for Secure Computing Corporation.

Chapter 5

The Legal and Regulatory Environment of the Internet

Lawrence D. Dietz

The legal environment of the Internet has often been compared with the Wild West in the days of the American frontier. This analogy is used to convey the wide open and freewheeling atmosphere that pervades this area of the law. The Internet, like many other technological phenomena, is developing along several parallel directions. The main line is the technology direction — those facets of networking, computing, software, and databases, which, when combined, add up to the ability to access the array of interconnected computers known as the Internet.

The second area of development is the nature of business on the Internet. How can information be exchanged? How can goods or services be bought or sold? What aspects of today's business rules can be employed effectively as rules for tomorrow's net-based business?

It is only after business transactions are in process or, more properly, when business transactions do not turn out according to the expectations of the participants, that the law enters into the picture. Astute managers do not resort to the law to correct a problem; rather, the law is supposed to be used as a guideline to avoid problems or to minimize the consequences if things go wrong.

Therefore, it follows that the law of the Internet is an emerging and evolving beast. Large end-user organizations are turning to technology first and then are immersed in the process of figuring out how to use that technology. Once the technology is employed, the true nature of its influence on business relationships can be determined. The deeper and more

0-8493-1161-6/02/$0.00+$1.50

mission-critical the use of technology is, the more severe the effects are if something goes awry. When it is clear that a simple business solution will not work to resolve a problem, lawyers are called.

DEFINING THE SITUATION

To understand the sheer magnitude of the dilemma of law on the Internet, it is useful to look at the Internet through an analogy. John Anderson, the former Presidential candidate, speaking at the annual RSA Conference in Redwood City, California in January 1995, compared the Internet with interstate highways. He pointed out how, back in 1955, no one could have foreseen the economic fallout of the interstate highway network. Originally set up during the Eisenhower administration as a key part of its Civil Defense strategy, the interstate highway network not only spawned a multibillion dollar auto and truck industry, but also had a profound effect on shipping and population concentration. No one could have predicted the way highways would either spur or destroy commerce and communities.

At present, the Internet and its subsequent progeny are similarly unknown and unpredictable. From a legal perspective, this situation becomes particularly perplexing. When stripped to its core, the purpose of a legal system is to form a bulwark upon which a set of governing behaviors can be determined. The split of law into civil and criminal areas historically has been used to divide the legal world into two segments: a part that deals with interaction among parties (i.e., civil law) and a part that governs an individual's (or organization's) behavior with respect to society (i.e., criminal law).

From a U.S. perspective, civil law may be divided further into multiple areas: areas of legal specialty and jurisdictions. As law relates to the Internet, areas of consequence are:

- Contract law
- Intellectual property
- Torts and negligence
- Criminal law

Before addressing each one of these areas in turn, it is important to point out that laws are not enforced in a vacuum. A critical aspect of the law is jurisdiction. Jurisdiction has two dimensions: the party and the law. The first jurisdictional issue is: does the court have the power to control an individual or entity? In the United States, a court's jurisdiction may be a city, county, state, several states, or the entire country.

The second question is one of subject matter or rules. Plaintiffs (i.e., the parties bringing a suit) determine where they will bring the action; that is, the legal forum. If the plaintiffs and defendants (i.e., those charged by the action) are from different locations and if the matter at hand occurred in yet another location, various procedures are set in motion by both sides to decide which set of rules (i.e., body of law) will be applied by the court. This is especially true in a federal court that, although located in one state, will often be compelled to follow the law of another. This jurisdictional issue is called diversity of citizenship, whereby the law of one state may be applied to adjudicate a dispute in a court of another state.

There is no greater challenge to jurisdiction than that of an indefinable web of computers and the various media that link them. The Internet is conceptually stateless and countryless so jurisdictional issues are wide open. Some of the key aspects of litigation, such as forum shopping (i.e., picking the best place to bring the action), is discussed in a later section. Moreover, courts need two kinds of jurisdiction to try a case: personal jurisdiction over the parties and subject matter jurisdiction over the matter. One is not enough; both are necessary.

The next sections highlight significant areas of the law and how they relate to the Internet and its security and integrity.

Contract Law

The Internet may become not only a transportation medium for business transactions, but the subject matter as well. Existing rules of law and terms and conditions that govern business transactions, such as the Uniform Commercial Code (UCC), must be modified to bend to the Internet's way of doing business.

A number of reforms have been under way for several years on different fronts. A key area of contract law that is evolving is electronic data interchange (EDI). Organizations can employ EDI to replace paper transactions. If so, clear terms and conditions must be extended to encompass new concepts of contractual relationships. Key terms, such as acceptance, rejection, and remedies for breach of contract, must be couched in terms appropriate for the Internet world. Potential failures or compromises of Internet-based transactions, failure to perform by Internet service providers, as well as action or inaction by suppliers and customers must be considered in developing contracts between organizations using EDI.

Areas of particular interest to Internet security practitioners include the use and acceptance of digital signatures in lieu of written signatures. A digital signature is the use of an algorithm as a substitute for an individual's authorized, holographic signature. The purpose of the signature is to commit the signer. It is an authentication of the signer's intent and proof of his or her acceptance or authoring of the document at hand.

The main reason behind the push for digital signatures is the UCC. Used by 49 states (the exception being Louisiana), the UCC requires both parties to sign a writing for transactions in excess of $500. As a side note, there are other branches of the law, such as real estate, where a signed writing is also required for the transaction to be valid. Digital signatures could be employed in a number of other areas in which the legitimization of documents is important.

Intellectual Property

Intellectual property is made up of several key components: patents, trademarks, copyrights, and trade secrets. For the most part, intellectual property is a part of state craft; that is, federal law controls its validity and use. Only trade secrets are governed by state law. From the perspective of the Internet security practitioner, a fair number of rules are already in place. Violations or, more commonly, infringement of intellectual property rights can occur throughout large end user organizations. The availability and convenience of e-mail and the Internet as a transportation medium increase the reach of a potential infringer. The literally unlimited horizons of the Internet raise the stakes for intellectual property problems.

The astute practitioner will bolster him or herself through aggressive policies and extensive education. Employees and others with access to an organization's intellectual property should be placed under contractual control not to use that property improperly. As with other aspects of employee-related legal issues, notice and consent are critical. Organizations must be obligors on notice as to what information is a trade secret, and they must place conspicuous notice on copyright or trademarked items.

Organizations must remember that the Internet is another way in which employees can transport protected property to unauthorized parties. They must guard against the possibility of this occurring as they would with more traditional vulnerabilities.

Torts and Negligence

In cases involving the Internet and in others involving negligence, courts will apply (and will instruct jurors to apply) classic test factors. The common law test applied in this situations has the following elements:

- Gravity of the harm: How extensive was the damage?
- Likelihood to occur: Given the surrounding circumstances, how likely was the event to happen?
- Cost to prevent: Given the size of the potential harm and its likelihood, what would have been the cost to prevent the harm, and how reasonable would it have been to expend those funds?

- Duty of care: What responsibility did the defendant have to the plaintiff? For example, because banks hold their depositors' money and are considered fiduciaries, they are held to a higher standard of care than a simple vendor of stationary goods would be.
- Standard of care: What do other similar persons or organizations do under the same circumstances? Do 95% or more of similar victims of a crime perpetuated over the Internet employ firewalls? How sophisticated is the victim as an Internet user or provider?

These factors will continue to be the yardsticks by which negligent actions will be measured.

Product liability is an area within tort law in which products used in Internet applications are included. By way of analogy, the New Jersey Supreme Court in *Roberts* v. *Rich Foods, Inc.,* 139 N.J. 365 (1995) found that a computer used in a motor vehicle was defective. This computer was used by truck drivers to record mileage and fuel data. The court judged it as defective because the device could be operated while the vehicle was in motion. It was reasoned that operating the computer would divert the driver's attention from operating the vehicle so that if there was an accident, the design of the computer would be a factor in that accident, and liability of the computer manufacturer for improper design had to be considered.

Criminal Law

Criminal law is a creature of the government. The plaintiff is the government or the people. To be guilty of a crime, one must have broken the law or violated a particular statute. Typical criminal law statutes require a voluntary or involuntary action (i.e., *actus reus*) in legal jargon and an intent (i.e., *mens rea*). Usually, Internet and other computer crime laws require voluntary acts (as opposed to involuntary or unconscious acts) and purposeful intent. Therefore, government prosecutors must be able to prove both. This proof must be to the higher standard known as beyond a reasonable doubt, which contrasts with civil law, where the standard is preponderance (i.e., majority) of the evidence.

Often, as with other laws, computer crime laws are shaped out of well-known past rules. For example, criminal harassment activity, stalking, and similar behavior have been a part of the legal landscape for some time. In June 1995, the state of Connecticut joined the ranks of computer crime pioneers by amending its existing harassment law to include a computer network as a means by which a defendant could employ with the intent to harass, annoy, alarm, or terrorize. Details can be found in the Connecticut General Statutes, sections 53A to 182b, and 183.

Another important aspect of Internet criminal law that is currently being addressed is the issue of sentencing guidelines. Sentencing guidelines are issued by various jurisdictions and are used by judges in dealing with the post-trial punishment of defendants who have been found guilty. Among the aspects of sentencing guidelines is sexual abuse or exploitation. (For reference, look at the *United States Sentencing Guidelines*, section 2G2.2[b][4].) The First Circuit Court, based in Boston, MA, felt that the transmission of child pornography over the Internet (in this case, America Online, or AOL) did not constitute sexual abuse or exploitation under the guidelines. The case in question was *United States* v. *Chapman,* 60 F.3d 894 (1st Cir. 1995).

In this case, according to the court, there was considerable evidence that the defendant used AOL to transmit child pornography on a number of occasions. The court concluded that these transmissions were not abuse or exploitation under the guidelines; therefore, these transmissions should not be considered a factor in deciding an appropriate sentence.

The Computer Fraud and Abuse Act of 1986 serves to protect computer systems, particularly federal computers. United States Code Section 1030 (a)(5)(A) states that its penalty provisions apply to "anyone who intentionally accesses a Federal interest computer without authorization, and by means of one or more instances of such conduct, alters, damages, or destroys information in any such Federal interest computer or prevents authorized use of any such computer or information …" and thereby causes loss of $1,000 or more.

It is important to note that the term Federal interest computer broadens the scope of the law to more than just federal government computers. It logically would include contractors to the federal government and perhaps computers privately owned by U.S. federal government employees that are being used for the benefit of the federal government. It is also interesting to note that loss of use receives protection under the statute as well as damage or alteration.

The most well-known conviction under this statute, upheld on appeal, was the case of the Cornell graduate student, Robert Morris (son of the NSA cryptographer), who was convicted for releasing the worm, a computer virus that replicated itself over the Internet, causing multiple crashes. Among those computers affected were a significant number of Federal interest computers. The appeals court's opinion may be read at *United States* v. *Morris,* 928 F.2d 504 (2d Cir.), *certiorari* denied by the Supreme Court in 502 US 817 (1991).

Export Control and International Traffic in Arms Regulations (22 CFR Parts, 120 through 128 & 130)

These regulations are used to control export of anything that could harm the security of the United States, including weapons, weapons systems, and cryptography. Vendors seeking to export must secure an export license. The approval process weaves a circuitous route among the Departments of Commerce, Defense, and State. Although the process has been a thorn in the side of U.S. software exporters, it has spawned a specialized consulting niche. This niche has been addressed by a number of independents, most recently by RSA Data Security, in Redwood City, CA. RSA recently announced a new division in the company, which will be headed by a former employee of the National Security Agency (NSA), to assist companies in obtaining export licenses. There are also a number of independent consultants, such as Cecil Shure, president of CSI Associates, in Washington, D.C., who specialize in exporting.

In fairness, the Clinton administration sent a number of signals that it was willing to relax the draconian regulations under certain circumstances. Among these was the vendor's willingness to give the government access to key-breaking information when the government asks for it or as a part of the approval cycle. This was both good news and bad news for vendors. On the positive side, the approval process appeared at last to be getting more export friendly. On the negative side, non-U.S. customers will perhaps be unwilling to employ a product knowing that the U.S. government is able to read their mail.

LIABILITY ISSUES

Anyone can be named in a lawsuit or charged with a crime. The defendant can be an individual or an organization. Ancillary potential plaintiffs and defendants in Internet matters can include suppliers, customers, government agencies, and trade associations, to name a few possible candidates.

An International Perspective

In an unusually frank spirit of cooperation, the forum for suit was broadened in Europe to allow defamation plaintiffs domiciled in a Brussels Convention country to pursue remedies either where the publication originated or where the harm occurred. The choice of litigation therefore could be based on a greater likelihood of success under that country's laws or the reputation for plaintiff sympathy. (Plaintiffs choose where to bring actions; defendants merely respond.) Reference for the European Court of Justice is C-68.93, and the United Kingdom (UK) reference is *Shevill* v. *Presse Alliance CA* (1992) a AllER. The defendant was a French publisher, and the action was brought in the United Kingdom because the plaintiff felt that it was a more sympathetic jurisdiction. The court noted that circulation was greater in France than in the United Kingdom, but that was not material to the selection of forums.

A Role in the Events

In general, a party cannot be found liable unless it had some part in the problematic acts. For purposes of the law, the party just as easily can be an organization (government or private) as well as an individual. Of particular interest to the Internet community is the issue of publisher liability. If there is a liability issue, an entity who creates or edits the news is far more likely to be found culpable than one who merely distributes or transmits the news.

A New York case, *Stratton Oakmont Inc.* v. *Prodigy Services Co.,* No. 31063/94, 1995 WL 323710, 23 Media Law report 1794 (N.Y. Supreme Court 1995), was decided against the online service. The facts involved comments posted by an unidentified bulletin board user in October 1994. These comments on Money Talk contained allegedly libelous statements about Statton, an investment banking firm. Stratton sued both the poster and Prodigy.

The rationale behind this decision covered a number of relevant points. Prodigy employed moderators for the panels. These board leaders had a number of responsibilities over the bulletin board. They were charged with enforcing the content guidelines set up by Prodigy (the guidelines themselves were considered another reason why Prodigy had control over content) and could use a special delete function to remove offending material. The court also noted that Prodigy employed software to screen postings for offensive language. Another critical aspect of this case was that the board leader of Money Talk was found to be an agent of Prodigy and that agent liability attached.

The opposite ruling (that is, finding that the service provider was not a publisher) was the 1991 case in the Southern District of New York, *Cubby Inc.* v. *CompuServe Inc.*, 776 F. Supp. 135 (S.D.N.Y. 1991). In this case, the court felt that CompuServe did not post any guidelines, take any role in controlling content, or promote itself as a family-oriented service, as Prodigy had.

Organizations can be found liable for the actions of their employees or agents under the legal doctrine of *respondeat superior.* Simply stated, employers can be liable for the acts of their employees acting within the scope of their employment. Therefore, software developers who accidentally unleash a virus or worm, as Morris did, may bring liability upon their employers. In addition, plaintiffs will continue to search for defendants with money. Often, employers have more financial wherewithal than their employees and become the targets of legal action.

Some areas of the law look to what management actually knew or should have known given due diligence of the reasonable person under similar circumstances. Intentional acts by employees that can or should have been prevented by more direct action by management may also result in liability applying to the organization, even for intentional acts.

Another rule of law, that intentional criminal acts are a bar to liability, may also be applied in Internet security cases; however, there are no guarantees. Juries often have gone against facts that appear to be overwhelming and appellate jurisdictions have often labored to reach a decision based on abstract theories of society goodness. The absence of historical precedent makes legal actions by and about the Internet perfectly positioned for inconsistent decisions. Security practitioners who go down this uncharted road do so at their peril.

Product Liability

Anyone in the stream of commerce can be included as a party in a product liability matter. Included in the stream of commerce are designers, developers, manufacturers, distributors, representatives, and retailers. An aggressive plaintiff and competent counsel will seek to embroil any potential defendant in litigation. This is especially true if the defendant has significant financial resources or a track record of trying to settle rather than litigate matters. This undoubtedly will be an important aspect of future Internet legal activity.

LIABILITIES AND AVAILABLE REMEDIES

The ultimate purpose of remedies is to put the aggrieved party back into the position that he or she would have been in if the wrongdoer had not acted in the way that he or she did. Remedies also can be used to deter future negative behavior and compensate the plaintiff for wrongs against society committed by the defendant.

Money Damages

A court can award substantial sums of money to the aggrieved parties. The court's rationale can be real or imagined, and amounts can be rational or irrational. Experts are often used to prove up damages. The role of the expert witness is to clarify facts for the court. As shown in a recent celebrated criminal trial in Los Angeles, CA, scientific, expert testimony does not necessarily ensure victory for the presenter. In addition to damages as a result of the defendant's act or failure to act, damages can be awarded based on a bad intent on the part of the offender. These punitive damages often can be twice or three times the amount of actual damages.

Injunctions

An injunction is simply a court order prohibiting a party (or parties) from doing a specific action. To get an injunction during the pretrial phase, plaintiffs have to demonstrate (among other things) that they will suffer irreparable harm if the injunction is not invoked, that the plaintiff is likely to win on the merits of the case (which will result in a permanent court ruling), and that the court will be able to enforce the injunction.

Criminal Liability

In criminal cases, remedies are spelled out by the statute that was violated and the sentencing guidelines that jurisdictions often issue to accompany the laws. The most common punishments include fines, community service, and incarceration. Incarceration can take many forms: county, state, or federal prisons, and a growing number of other more innovative programs such as confinement to one's home.

Courts sometimes have gone to great lengths in computer-related crimes to remove a convicted defendant's access to the tools of the computer trade. The incorrigible nature of some defendants and the magnitude of the harm they caused, combined with their lack of remorse, have often induced judges to impose heavier and more creative sentences than in comparable cases of noncomputer-related crimes.

Lawyer Liability

Lawyer liability is a phrase that the author uses to describe other harm. Time spent with attorneys and money spent on attorney's fees are not trivial. In the days of downsizing and rightsizing, employee productivity is guarded zealously. Time spent that does not either increase revenue or decrease costs is wasted time. The effort and resources needed to pursue and win a legal action should be considered before the action is undertaken. Fees for attorneys, as well as other expenses, such as court costs and expert fees, are substantial. Often plaintiffs have to spend an inordinate amount of time educating their counsel about the nature of their businesses and the nature of the action. Combine this time investment with the uncertain nature of law as related to the Internet and the general lack of computer literacy in the legal profession and there are the makings of a true disaster in terms of the expenditure of resources versus the likelihood of benefit or gain.

This approach could be applied to computers as well. Given the seemingly pro-conservative bent of the American electorate and the desire to win the family vote of leading politicians, a strong push to repeal such infringements is not likely to come from elected officials. Rather, it will be up to pioneering plaintiffs, perhaps aided by the Electronic Freedom Foundation (EFF) or Computer Professionals for Social Responsibility (CPSR) or other similar rights advocates to step up to employ legal action to block enforcement.

Such activity is not unprecedented. The California proposition 187, limiting educational and other entitlements of immigrants, which, although passed by the electorate, was blocked due to potential constitutionality problems, could be a model for such protestations. However, computer and information freedom does not have a readily identifiable homogenous group of affected persons who will take direct, immediate, and costly action, at least not at this time. Furthermore, championing of pornography is not a popular view that will capture the hearts and minds of the electorate or the media.

AVOIDING PROBLEMS

System administrators must be mindful of the need for notice and disclosure. They must ensure that users or subscribers are fully aware of who has access to the system. They must indicate clearly how monitoring and control may be or is exercised on the system. Employee handbooks should spell out exactly what employees are expected to do in terms of use of the company's information resources. All employees should sign an acknowledgment that they have read the rules, understand them, and agree to be bound by them.

Prosecution of Hackers

It is important to remember that a criminal prosecution is run by the office of the local District Attorney (DA), not by the victim. The goal of the DA is to get a conviction, not to ensure that the victim is compensated nor to prevent similar occurrences in the future. A decision to proceed with prosecution is also a decision to cooperate fully with law enforcement authorities. Cooperation may require a significant amount of time, money, and resources from the company. This commitment may not fit with the company's goals of minimizing bad publicity, fixing the leak, and controlling the course of legal events.

Should the decision be made to proceed, it is important to be mindful of the rules of evidence and the critical need to keep a pure chain of custody. One person's opinion that a piece of evidence is damning does not mean that it is. More importantly, it does not mean that it will be admissible and, if it is, that it will be understood by the trier of fact, whether judge or jury.

It is important to recognize that experts may be needed and that they may come from the ranks of the company or the company's suppliers. Victims may not be in a position to recommend, to supply, or to compensate needed experts, and the DA's budget may not permit hiring the right kind of talent.

History has shown that a defendant with financial muscle is not to be taken lightly. Should the defendant be well funded, he or she might not get convicted and might turn around and sue the plaintiff for defamation or malicious prosecution.

Before opting for a criminal prosecution, a lot of pertinent information can be found in a Government Printing Office document: the *Criminal Justice Resource Manual*, prepared by the Department of Justice. It contains excellent advice concerning the types of computer crimes, evidence, likely perpetrators, and other related material.

Companies contemplating this type of prosecution also should be sensitive to the track record of the local DA with respect to this type of white-collar crime case. Obviously, some jurisdictions (such as Austin, TX; Boston, MA; and Santa Clara, CA) are better venues for technology-related cases due to the high population of computer literate potential jurors and high-tech companies.

It is critical to remember that when the lawyer is called, whether a civil counsel, corporate counsel, or the local DA, someone loses time, resources, and money.

CONCLUSION

An organization should determine its goals early in the process and balance the practical results that it wants to achieve against the legal hurdles that will have to be navigated to get them. Often, compromise is a faster, cheaper, and better alternative than pursuing legal remedies. Simple themes are always better than complex ones.

ABOUT THE AUTHOR

Lawrence D. Dietz, Esq., is vice president of Knowledge Centers at the Giga Information Group in Santa Clara, CA.

Chapter 6
Writing and Implementing Internet Acceptable Use Policies

James E. Gaskin

Because more companies are now using the Internet as a means of communication and research, IS executives are writing and implementing an acceptable usage policy for corporate use of the Internet. Such a document may be called an Internet use policy, the networking portion of the computer use policy, or the Internet addition to the personnel manual. The goal of this policy is to list the rules and standards for employees using computers, networks, and, particularly, the Internet. Although the acceptable use policy can be incorporated into other existing documents, it generally provides more company protection if it is a separate document.

Why is the acceptable use policy so important today? Legal liability for Internet actions can quickly shift from the employee to the employer. If management allows access to inappropriate Internet sites without either warning users or blocking access, management can become liable, along with the employee performing illegal actions.

WRITING AN ACCEPTABLE USE POLICY

IS managers, or a department employee, must write the acceptable use policy. It is better to have the fewest number of people — preferably one person — writing the acceptable use policy. Although this suggestion may conflict with many corporate cultures, the acceptable use policy is different from a product manual or marketing white paper. The acceptable use policy is a legal document that binds the behavior of employees within certain boundaries.

0-8493-1161-6/02/$0.00+$1.50

With fewer authors the number of viewpoints within the acceptable use policy is limited. Your employees must be clear about the purpose of the acceptable use policy, their Internet and computer use responsibilities, and the penalties for misuse of company resources, including time. More authors, or up-the-line editorial changes, will muddy the acceptable use policy. Internal contradictions within the acceptable use policy will leave loopholes for employee lawyers to exploit.

After the policy is written, a committee that oversees employee compliance with the terms of the agreement should meet and approve the acceptable use policy before distributing the document. This is the time for any comments, suggestions, additions, or deletions. While the committee is welcome to offer changes, only the author should implement them. Again, the consistency of viewpoint is important.

Legal review comes after the committee has approved the acceptable use policy. Here a philosophical decision must be made. Often, lawyers want long, complicated documents that spell out every possible infraction and associated punishment, while business managers want short documents that can be interpreted in the company's favor. The length and level of detail should reflect the corporate culture and views of upper management.

The document should be a part of the employee handbook. In some states, these handbooks are regarded as a legal contract. Corporate counsel will be able to anser that question for the states where the company operates.

Be aware that the longer the policy, the fewer the number of employees who will read it to the end. In most states, employees are bound by the conditions of the policy. However, holding employees liable for a document they have not read will be seen as a cold, heartless corporate maneuver. Employees who feel betrayed contact lawyers far more often than those who feel they were treated fairly. Although it is legal in some states for companies to ignore the promises they make in employee handbooks, the antagonism employees may feel as a result guarantees lawsuits.

POLICY SCOPE AND OVERVIEW

Does your company already have computer-use policies? How about company telephone, fax, and mail use? Is there a security policy in place?

Some companies, remiss in providing policies in the past, try to incorporate all these into the acceptable use policy. Although this is legal, it is confusing to employees. The acceptable use policy will be more valuable if targeted strictly to Internet and other computer networking concerns.

E-Mail

Because e-mail is the most popular Internet application, control over its use is important. The good part of e-mail is that an appropriate analogy can be made to traditional mail.

One company includes this statement: "Remember that e-mail sent from the company travels on the company's electronic stationary. Your e-mail appears to the recipient as if it were sent on company letterhead."

Your security policy, if separate, should cover information about e-mail accounts, such as forging identities. Instructions concerning appropriate e-mail use can also be included in the acceptable use policy.

Other e-mail guidelines that some schools and companies prohibit:

- Sending harassing, obscene or other threatening e-mail
- Sending junk mail, for-profit messages, or chain letters
- Sending or receiving sexually oriented messages or images
- Transmittal of confidential company information
- Divulging employee medical, personal, or financial information
- Personal messages

Also often included is a request that reasonable precautionary means be taken against importation of computer viruses.

Employees may also be reminded of the importance of e-mail to communications within a company. Whether an employee must be told when the company monitors communications is advisable according to some lawyers, but not others. Either way, if every employee signs the acceptable use policy accepting e-mail monitoring on a random basis, they may pay more attention to the following the rules.

Employees should have no expectation that e-mail messages are private and protected by a privacy law. Make sure each user understands that some messages will be read by management, even if messages are only spot-checked.

Do not keep e-mail message for longer than 90 days. Lawyers are now routinely demanding e-mail archives during lawsuit discovery. If your company is sued for any reason, the opposing lawyers will try to read all internal and external e-mail messages for the time in question. No e-mail archives means no embarrassing quotes and off-the-cuff remarks that will cost you in court. Some large companies refuse to back up e-mail files for this reason.

World Wide Web Resources and Newsgroups

The Web is often criticized as a giant productivity sink hole. Corporate managers rank employee time wasted on the Internet as their number two concern right behind security. Management often wonders how many employees are frittering away hours on company time perusing the Web on company equipment.

While newsgroups full of equivalent professionals in other companies provide great benefit to your company employees, the nontechnical press focuses on the “alt.sex.*” hierarchy of newsgroups. Someone in your management will be determined to limit access to all newsgroups, just to keep the alt.sex. groups out of the company.

Newsgroups are where the majority of defamation happens; “flame wars” are when people become angry and make unprofessional statements, which result in legal action against the employee and the company represented by the employee. Often, other readers of the newsgroup will send copies of messages to the postmasters of the flame war participants. Management should counsel employees accused of involvement and if this does not work, unplug them from the newsgroup access list. There is no sense risking a lawsuit when there may be a good chance of statements being made that have no positive value to the company.

Be upfront with management about the existence of inappropriate Web servers and newsgroups. But also note that some Web servers and newsgroups are valuable. Also be sure to mention that each user can be monitored and the name, date, time online, and amount of material downloaded from any inappropriate network source can be obtained.

This will allow the actions of each and every corporate user during each and every network communication to be logged. If the proper firewall or proxy server to monitor users is not in place, make this a priority.

Realize that some time will be wasted on the Web, just as time is wasted reading through trade magazines looking for articles that apply to your company. Every profession has trade magazines that offer articles and information in exchange for presenting advertising to the reader. The Web, to some people, is becoming nothing more than a huge trade magazine, offering helpful information interspersed with advertising. Some employees research information more than others and will use their Web client more. Know which employees should be using the Web.

Web guidelines may be mentioned in the acceptable use policy. Sample restrictions may include:

- Viewing, downloading, displaying, or distributing obscene images
- Limiting Web browsing during work hours to business-related searching

Remind your employees regularly that obscenity in the workplace will not be allowed. Modify the second bullet point to match the company's comfort level regarding employee use of the Web.

Other restrictions may include:

- Downloading or uploading of nonbusiness images or files
- E-mailing of harassing, obscene and/or other threatening messages
- E-mailing of junk posts or "for-profit" messages
- Posting of articles to groups unrelated to the article's subject matter
- Posting of company advertisements in any newsgroup
- Posting of messages without an employee's real name
- Copying of newsgroup information to any other forum

Several acceptable use policies address defamation obliquely. Some examples of the language included in those policies include statements restricting "comments based on race, national origin, sex, sexual orientation, age, disability, religion, or political beliefs" or "send[ing]/receiv[ing] messages that are racist, inflammatory, sexist, or contain obscenities."

Whether these are politically correct or good business sense depends on the individual company. These same courtesy restrictions apply to e-mail, but e-mail lacks that extra edge brought when thousands of readers see your company name attached to the ranting of one overwrought employee.

IRC (Internet Relay Chat) and MUDs (Multi-User Domain) have not been mentioned because they have no redeeming professional use. Employee use should not be tolerated.

In case employees are confused about the company's rights to monitor employee computer use, include a line such as: "All computer communications are logged and randomly reviewed to verify appropriate use."

The term "appropriate use" is carefully chosen. If the acceptable use policy says the words "dirty pictures" or "indecent," employees (and their lawyers) can argue that "dirty" and "indecent" is in the eye of the beholder. "Inappropriate" covers more activities than any other term. Another option is "obscene," which is a legal term that applies just as well to computers as to magazines, books, and videos.

Penalty for misuse should range up to and include termination. If an employee must be terminated, do so for work-related causes, rather than Internet causes. Free speech advocates can get involved when an employee is fired for inappropriate use of the Internet, but not when an employee is terminated for wasting too much time on the job and disobeying orders.

Netiquette Addendum

Some companies spell out appropriate e-mail, newsgroup, and Web communication guidelines within their acceptable use policy. This is a noble endeavor, but slightly misguided. Company guidelines on Internet communications are likely to change more often than your restrictions on inappropriate Internet use and discipline for infractions.

Because an acceptable use policy should be signed by each employee, any changes to netiquette embedded in the acceptable use policy will require a new signature. The logistics of this process can quickly become overwhelming.

Put the rules of Internet behavior in a separate addendum. Changes to e-mail rules, for instance, will not negate the acceptable use policy, nor will a new signature be necessary.

ACTIVATING THE POLICY

Getting employees to sign an acceptable use policy can be tricky. Small- to medium-sized companies can handle the logistics of gathering signed copies of the acceptable use policy, although there will still be considerable time expended on that effort. Large companies may find it impossible to ship paper policies all over the world for signatures and get them signed, no matter how much time and effort they devote.

The best case is to get a signed acceptable use policy from each employee before that person is connected to the Internet. Training classes offer an excellent chance to gather signatures. If software must be installed on client computers, the policy should be presented, explained, and signed during software loading.

Unfortunately, many companies already have granted Internet access before developing an acceptable use policy. This is not the wisest course, but is common. Other companies do not offer training or cannot gather signed copies.

It is important to send copies of the policy to each employee with Internet access. Copies should also be posted in public places, such as break rooms and department bulletin boards. The policy should also be added to the existing personnel manual or employee handbook. An e-mail should also be sent to users every quarter reminding them of the acceptable use policy and where they can read a copy. These efforts should stop any employee contentions concerning Internet restrictions.

THE ACCEPTABLE USE POLICY COMMITTEE

An Acceptable Use Policy Committee should be formed from employees from each department. Each member should be notified in advance of the first meeting and have adequate and timely background information on the task of the committee.

The following list contains the requisite committee positions and their expected contributions:

- Computer systems manager. Provides technical details of Internet access and monitoring.
- Company lawyer or human resources official. Provides legal aspects of workplace rules.
- Executive management representative. Guarantees your committee will not be ignored.
- Union representative. Has knowledge of laws for union employees.
- Employee representative. Represents employee concerns and interests.

This committee will discuss all Internet concerns, and should probably meet every two weeks. Once the Internet connection is old news, once a month may be enough. The interval is dictated by the number of security incidents and employee discipline actions to be resolved.

In extreme cases, such as an employee action that could result in company liability or criminal prosecution for someone, the committee must meet immediately. The grievance policy in cases of Internet abuse should be clear and well known to all employees. It also is important that employees know who is on the committee. Secret committees are repressive, but open committees can encourage good will within the company. Strongly consider setting up an internal e-mail address for your committee, and use it for questions and as an electronic suggestion box.

The most effective deterrent to misdeed is not the severity of discipline but the inevitability of discovery. Remember, the goal is to make the Internet serve the company, not to find excuses to discipline or fire employees. At the first committee meeting, the following questions should be answered:

- Will employees be fired for Internet misuse?
- What is the penalty for the first offense? The third? The fifth?
- Will the police be called for stolen software or obviously obscene images?
- Should other employee policies be modified to support the Internet connection?
- Are any insurance policies in place to protect against hackers or employee misdeed? Should some be added?
- How often will employees be reminded of company Internet guidelines? How will this be done?

Discipline is particularly tough. After all, if an employee is wasting hours per day on the Internet, the department manager also should be disciplined for improper management. Waste of time on the Internet is not a technology issue, but a management issue.

Outsiders with an executive mandate to punish miscreants are never popular and often are sabotaged by the very employees they oversee. Keep department managers in the loop as long as possible. Exceptions include security violations and illegal acts: department managers must be informed in these instances, but company security or local police will handle the situation.

RECOMMENDED COURSE OF ACTION

The job of the acceptable use policy is to outline acceptable Internet and/or computer use and behavior. The committee dedicated to enforcing the provisions of the policy must publicize the policy and monitor employee compliance. Infractions must be handled quickly, or employees will assume the Acceptable Use Policy is not important, and compliance levels will shrink. Proactive Internet management will drastically lower the chances of Internet-related lawsuits, arguments, and misunderstandings.

ABOUT THE AUTHOR

James E. Gaskin is an author and consultant specializing in technical subjects and technical policy issues, such as corporate politics and the Internet. He can be reached at james@gaskin.com.

Chapter 7
Designing Equitable Chargeback Systems

Christine B. Tayntor

The concept of charging for computing services is not a new one. It has been in existence for as long as computers have played a key role in the business world. Because the initial use of large-scale computers in most companies was to automate accounting functions, and because the IS department often reported to the chief financial officer, it is easy to understand why cost-accounting techniques were used in allocating the expenses associated with computing.

In the early days of computing, when computers were expensive and the support staff (i.e., programmers and operators) relatively inexpensive, most chargeback systems focused on allocating the costs of the mainframe. Later, some companies incorporated operational costs and added charges for programming. As the costs of mainframes fell in relation to the rising costs of programmers, people-related costs became part of the chargeback equation, although the focus was still on the mainframe. With the current emphasis on downsizing and the movement to even less expensive computing resources, it is important to consider all computing platforms when designing a chargeback system.

PROS AND CONS OF CHARGEBACKS

For companies that do not currently have chargeback systems, the first step is to recognize the arguments for and against instituting one. Although the CFO may believe that full recovery of costs is essential for a fiscally responsible organization, the IS department's customers may have a different view.

When to Use a Chargeback System

There are three primary reasons why companies institute a chargeback system.

0-8493-1161-6/02/$0.00+$1.50

Chargeback Systems Increase the Customer's Accountability. When the IS department does not charge for its services, or when the costs are allocated to departments according to factors other than use, there is no incentive for a customer to perform a cost/benefit analysis before authorizing a systems project. Nonessential projects may be undertaken and less expensive solutions may not be investigated. When the customer department's bottom line is affected, requests for additional services are more likely to be evaluated like any other goods or service purchased, based on their value to the department.

Chargeback Systems Encourage Conservation of Expensive Resources. When online systems were first developed, many IS departments faced a dilemma. Their computers were overloaded during prime shift, when customers used the new online systems, but they were underutilized during second and third shifts. By instituting variable rates for central processing unit (CPU) cycles and charging a premium for first shift, IS managers could persuade customers to move batch reporting and less time-critical processes from prime shift, thereby eliminating or at least deferring the purchase of a larger mainframe. A similar approach has helped many companies wean their customers from dependency on tapes, which requires operator intervention, to disk storage. In both cases, without a differential charge for the resource there would have been no incentive for the customer to conserve the resource.

Chargeback Systems Increase the Customer's Perception of the IS Function's Value. "You get what you pay for" may be a trite saying, but many people believe that it is also a true one. If there is no charge for IS staff and computing resources, many customers perceive those services to be of little value. Forcing customers to pay for IS services frequently has the effect of making the customer departments regard more highly the IS services they actually use and rely on.

When Not to Charge Back

Although there are reasons why a company would implement a chargeback system, there are also three primary reasons why it might not.

Chargebacks Can Lead to a Short-Term Focus. Although a chargeback system encourages accountability by affecting the customer department's bottom line, it can also result in tactical rather than strategic decisions. Because most companies evaluate financial performance quarterly or annually rather than over a period of years, a system with long-term benefits to the department may not be approved because of the high one-time cost of developing and implementing it. This is particularly true when programming costs are charged back, because they cannot be amortized over multiple years as hardware expenses can.

Chargebacks Require Overhead to Administer. In addition to the costs involved in developing a chargeback system, there are ongoing operational costs. These include the computer resources required to run the system as well as the administrative time required to oversee it and explain charges to customers. It can be argued that this increased cost has a negative impact on the company's overall bottom line.

Chargebacks Can Create Adversarial Relationships between IS and Customers. During the initial stages of implementing a chargeback system, customers may resent paying for what were previously free services. They may seek alternative sources for these services, such as using outside contractors for programming, and they may try to replace mainframe processing with spreadsheets and other programs on their microcomputers. In addition, they may not consult IS on key new initiatives because of the internal chargeback. For an IS department that is seeking to establish a partnership with its customers, this may be a serious deterrent to instituting a chargeback system.

CHARGEBACK METHODS AND RELATED ISSUES

The decision to implement a chargeback system is one that should be made only after considering all factors and consulting major customers. After a company has decided to charge for its services, the next issue is identifying which services should be included and how they should be charged.

Many companies have implemented complex chargeback schemes, using a different method for each of the services the IS department provides. The most commonly used methods and the issues that each raises are discussed throughout the rest of this chapter.

MAINFRAME OPERATIONS

Because the mainframe was the first computing component to be charged, it is also the one with the most clearly defined chargeback methods and associated systems to automate the charging. There are two primary methods used to bill for mainframe services: one is based on the specific resources consumed and the other is a flat charge per CPU hour (with premiums for prime shift). Each approach has its advantages.

Specific Resources. This is the most precise method of charging because it prices each service individually. CPU cycles are charged at a different rate than tape mounts; the cost of disk storage differs from a line of print. Although it is also the most complex method of charging, there is a variety of packaged software available to calculate usage of each component. The true complexity lies in determining the correct rate for each of the resources to be charged.

CPU Hour. The primary appeal of this approach is its simplicity, because it requires measuring only one usage component: CPU cycles. It is, however, less accurate than the specific-resources charge because it makes no distinction between people-intensive tasks, such as tending printers, and fully automated ones, such as disk access. It also undercharges customers who use extensive disk storage but process data infrequently.

LAN HARDWARE AND NETWORK OPERATING SYSTEM

With the almost ubiquitous use of local area networks (LANs), companies must charge for both the hardware and the operating system software. Unfortunately, there is little automated software available to assist in the chargeback process.

Determining What to Charge

Before a chargeback scheme can be implemented, IS should ask a series of questions designed to determine which components to include in the costing algorithm.

- What comprises the LAN?
- Who paid for the LAN components?
- What services will be included in the charge?

Determining What Comprises the LAN. Although most companies would agree that the servers, network operating software, and wiring are the primary LAN components, others would include PCs and workstations. Still other companies would include peripheral devices such as async servers and fax gateways and standard applications software, including word processing, e-mail, presentation graphics, and spreadsheets.

Determining Who Paid for the LAN Components. When initiating chargebacks, it is important to know whether IS or the individual customer departments purchased the equipment. If the LAN was acquired by the customer departments, IS will probably want to transfer the assets to its own books and give prorated charges to the departments until their initial investment is recovered.

Determining Which Services to Include in the Charge. Before a chargeback algorithm can be developed, it is necessary to determine the components of the costs to be recovered. There are three categories of costs to consider: hardware, software, and services.

Hardware. The cost of the LAN hardware (identified by answering question one) is the most typical component charged back to customers. The current year's depreciation is used for hardware whose costs are amortized over

Exhibit 1. Components of Personnel-Related Costs

- Primary costs (costs most companies include)
- Salaries
- Benefits
- Rent or occupancy charge
- Training
- Travel and entertainment
- General expenses (e.g., stationery and other supplies)
- Secondary costs (costs that should be included if total chargeback is desired)
- Software purchases
- Hardware purchases and depreciation

Note: As a rule, the salaries, benefits, and other expenses of the employees who actually perform the service should be part of the costing equation; some companies also include prorated costs of supervisors and managers.

several years. Purchases that are considered expense items are fully charged in the year in which they are made.

Software. In addition to the initial purchase price of network operating software, the cost of annual maintenance or periodic upgrades should be factored into the costing equation. The same expense components should be considered for all other types of software that were determined to be part of the LAN.

Services. A LAN does not run itself, and any chargeback scheme that seeks to bill out the total cost of running the LAN must include the people-related operating expenses. Before these can be calculated, however, IS must determine which services should be included in the total LAN charge. These may include administration (e.g., adding new user IDs or changing passwords), regular data backup and offsite storage, software upgrade installations, network monitoring, and scanning for viruses. Initial installation of LAN workstations may be included in the total cost or may be billed as a separate one-time charge.

When charging for services, all related costs must be considered before developing a charge, particularly if IS wishes to achieve full chargeback. If, for example, LAN services require two full-time employees, all the costs associated with those employees should be included in the chargeback equation. Exhibit 1 lists typical components of the personnel-related costs.

Billing for LANs

Once IS has determined which costs will be included in its chargebacks, it must determine how to implement the charges. There are three primary approaches to billing for LANs: a per-user charge, one that is based on use, and a hybrid approach.

Per-User Charges. The primary advantage of this approach is simplicity. To determine the charge, IS calculates its total annual LAN costs, then divides them by the number of users. If the cost calculations were accurate, full chargeback is guaranteed.

This approach is not only simple, but it also can be easily explained to the customers. It does, however, have several drawbacks, the most important of which is that it penalizes occasional users of the LAN by charging them the same amount as the power users. (For companies that seek to encourage LAN use, this may be considered an advantage rather than a disadvantage, because it does not discourage extensive use.)

The second potential problem using the per-user charge is that during a period of rapid growth, when many new users are being added to the network, IS may actually over-recover its costs if the pricing scheme is based on the number of users at the beginning of the year. To avoid the perception that IS is becoming a profit center, costing can be done by projecting the number of new users and the dates when they will be added to the network, then including those usage months in the per-user charge.

Usage-Based Charges. Similar to the mainframe billing scheme that charges for specific resources, this is the most precise method of charging for LAN services. It is predicated, however, on identifying the costs of each component of the LAN as well as each customer's use of that component. Because of the scarcity of automated tools to measure LAN use at this level of detail, few companies employ the approach.

Hybrid Systems. Some companies have adopted a hybrid chargeback that consists of a base charge (for connection to the LAN and use of the Network Operating System) and additional flat charges for specific LAN components (applications software or FAX servers). The charge-by-component approach is discussed in the following section on LAN software.

LAN Software. Some companies prefer to group LAN software with the LAN hardware when charging customers. Many other companies consider these as two separate components and charge for them separately. Although LAN software can consist of either purchased packages or in-house-developed systems, for this discussion only packages purchased from a third party will be considered. (Charges for custom in-house development, whether for a LAN, a client/server platform, or a mainframe, are reviewed in a subsequent section on programming.)

Generic LAN software, such as word processing, spreadsheets, e-mail, and presentation graphics, may be charged separately from a package purchased for a single department. In both cases, IS must consider several of the questions that were applied to LAN hardware, specifically: who purchased the software initially and what services should be included in the charge? Software-related services may extend to purchasing and applying upgrades as well as responding to customer questions.

Two of the most common ways of charging for LAN software are a flat charge per application and a charge that is based on use. Once again, the lack of precise monitoring tools keeps many companies from instituting a usage charge. Although software exists that can track the length of time a user has access to a specific program, it is rudimentary in its monitoring capabilities and requires additional administrative effort by IS if it is to be used in a chargeback program.

In determining a flat charge for an application, IS should consider the following factors:

- *Purchase price and payback period.* Assuming that IS has purchased the software, it must determine the period over which it wishes to recover costs. Many companies adopt a payback period of two to three years and, in effect, amortize the cost of the software to customers over that period. (This approach, however, will not result in complete chargeback during the year that software was acquired.)
- *Maintenance or upgrade costs.* Usually this is an annual charge that can be fully recovered during the year in which it is paid.
- *Related services.* Like LAN hardware, software requires support including installation, monitoring, and problem resolution. The cost of providing these services should be included in the total cost of LAN software.

When the costs to be recovered within a year have been determined, IS can calculate the flat charge by dividing that cost by the number of customers. This approach is effective for both generic and department-specific applications.

CLIENT/SERVER APPLICATIONS

Although client/server applications differ from LAN-based ones, they involve chargeback issues similar to those for LANs. As in the case of LANs, few tools are available to assist in formulating usage-specific chargebacks for client/server computing.

Here again, companies must determine which components to include in the charge. These typically include hardware, operating systems, database software, and applications software. The issues for client/server hardware and applications software are similar to those used for the equivalent LAN components. Client/server operating systems and database software involve issues more analogous to those involving mainframes, and they raise two questions that need to be answered before the costing method is determined:

1. *What are the corporate standards?* Although many companies have a list of approved operating systems and database managers, they may allow for acquisition of others.
2. *Should a premium be charged for nonstandard packages?* Because the more powerful operating systems and database managers associated with many client/server applications require active monitoring and support, some companies have established a two-tiered charging scheme. Like the mainframe surcharge for peak shift processing, the two-tiered approach is designed to encourage compliance with the corporate standards.

Here again, once IS has determined the cost components of the chargebacks, it must decide how to recover them. There are three approaches to billing for client/server applications: a per-user charge, a charge that is based on percentages, and a hybrid approach.

Per-User Charges. Although this approach is simple, it can result in charging less for an application with a few active users, high disk storage, and high CPU usage than for one with many occasional users and little resource consumption.

Percentage-Based Charges. Under this approach, each application is charged a percentage of the overall cost. Unless the application's owners are involved in the establishment of the percentages, IS is likely to be perceived as arbitrary for implementing this type of chargeback.

The Hybrid Approach. A combination of per-user and per-MB of storage charges can help to mitigate the disadvantage of pure per-user costing without requiring excessive administrative overhead.

THE INTERNET

The growth of the Internet as a business tool has led companies to begin developing chargeback methods for Internet use. Three methods are currently in use:

1. Flat charge per month
2. Per-hour charge
3. Hybrid approach

Flat Charge per Month. This method is the simplest. Like all per-user charges, however, the advantage of being easy to implement is accompanied by the drawback of penalizing low usage.

Per-Hour Charge. A charge based on per-hour usage is highly equitable but requires more effort to administer. It may also result in the IS function under- or over-recovering its costs because — at least initially — there is a limited baseline of hours used to factor into the cost equation.

The Hybrid Approach. The third chargeback algorithm is a combination of the first two. A flat monthly charge is used for up to a specified number of hours, then a per-hour charge is used. Although this approach gives IS a constant base of revenue and reduces the penalty for low usage, it is the most difficult of the three to administer.

PROGRAMMING: NEW SYSTEMS

Many companies that have implemented full chargeback schemes for their mainframe computers do not charge for programming services. In most cases these decisions reflect concerns over the potentially negative effects of chargebacks (as outlined earlier in this chapter). Recognizing the need for establishing system priorities when there is no charge for services, many of these companies have developed executive-level steering committees to review major projects and determine which should be funded. For those IS departments that choose to charge individual customers directly for the programming services provided, there are three primary chargeback methods: an hourly rate, fixed price, and an hourly rate with a not-to-exceed clause.

Hourly Rates

This approach is the safest for IS because it ensures that the department is paid for all work it performs. One disadvantage is that it makes budgeting difficult for customer departments unless IS is able to provide an accurate estimate of the amount of work it will provide to each department. For companies whose customer departments' performance is measured by the bottom line, including internal charges, this approach may result in some work being curtailed or deferred to improve the customer department's profits.

When implementing an hourly-rate charging scheme, IS must determine how many rates to use. Companies employ four types of hourly rates: a single rate for the entire department, one that varies according to the individual and is based on salary, one which is fixed by job grade, and one based on the type of work being performed.

Single Rate. This is the simplest approach because it involves only one calculation. Implementing this approach is also relatively simple because once the rate has been determined, the department need only record time and generate bills. Although most IS departments use automated tools for time recording and billing, this approach can be manual in small- to medium-sized departments.

The major disadvantage to having a single rate for the entire department is that it makes no distinction between the value of work provided by entry-level programmers and the most senior staff. Customers may balk at paying what appears to be an inflated rate for a junior person or may request only the highly experienced staff for their projects.

Individual Rate. From the IS view, this is the most complex rate scheme to develop and administer because it requires calculating a separate rate for each staff member. It has the added disadvantage of making salary variances, which are usually confidential, easy to determine. It does, however, allow IS to distinguish between junior and senior staff and to establish a direct correlation between actual cost and the charge.

Rate by Job Grade. A variation of the individual rate method, this approach boasts the advantages of individual rates without the major disadvantages, because it reduces the number of calculations required and removes the ability to determine which employees are more highly compensated.

Rate by Type of Work. Some companies have established variable rates that are based not on the person performing the work, but on the type of work being performed. Using this philosophy, senior employees who are temporarily performing an entry-level function (e.g., coding from detailed specifications) will bill out at a lower rate than when they are doing higher-level work, such as designing the system or writing specifications. This approach recognizes that some types of work are inherently more valuable than others; from an IS view, however, this charging scheme is the most complex to administer. Not only does it require all work to be type coded, but IS must be able to estimate how much of each function will be performed during the year in order to properly cost them.

Fixed Charges

The second approach companies use to charge for the development of new systems is fixed price. In essence, IS estimates the number of hours a project will require and multiplies that by the hourly rate to calculate a fixed price. A fudge factor may also be added.

The primary advantage to this method of chargeback is that it reduces friction between IS and the customer department; there are no surprises, no cost overruns, and no need for monthly variance explanations. This approach also aids customer budgeting. There are, of course, drawbacks. Unless IS is accurate in its estimates, full chargeout may not occur. To reduce this risk, most IS shops with fixed-price billing bid on only one phase of a project at a time, rather than provide a single price for the entire project. Accurate records of the time spent are important in estimating future projects.

Hourly Not-to-Exceed Pricing

The hourly not-to-exceed pricing scheme is a combination of hourly and fixed-price rates. IS charges an hourly rate for the work it performs but places a cap on the charge. This approach is often used as a selling tool with customers because it removes both the customers' fear of giving IS a blank check and their concern that a fixed price includes padding. The disadvantages are to the IS department. Not only does this method require slightly more overhead in billing, but it introduces the possibility of under-recovering expenses if the estimate was too low.

PROGRAMMING: MAINTENANCE AND SUPPORT

When charging for maintenance and support of existing systems, most IS departments use either an hourly rate or a maintenance contract. As expected, each has its advantages and drawbacks.

Hourly Rate. This is the classic method of charging for maintenance, and many of the issues and concerns that apply to charging hourly rates for new systems development are equally applicable here. The primary disadvantage to IS is that customers may decide not to have some maintenance performed, leaving IS with idle resources and expenses that are not fully recovered.

Maintenance Contract. Like its fixed-price equivalent in new systems development, the maintenance contract provides a guaranteed revenue stream to IS and permits customers to accurately budget their IS expenditures. The IS department's primary concerns in establishing maintenance contracts should be twofold:

1. *Ensuring that customers understand what is included.* A contract should include fixed-price support of the system (e.g., fixing bugs and responding to customers' questions about the system's operation). It may also include mandatory system changes (e.g., upgrades from a vendor or regulatory changes) but usually does not include enhancements. The services to be provided (and those not provided) should be clearly outlined and agreed to by both IS and the customer.
2. *Accurately estimating the amount of work required.*Without reliable records of how much time has been spent on system maintenance in the past, IS will be unable to develop a fair price for a contract and may under- or over-recover its expenses.

GAINING CUSTOMER ACCEPTANCE

After a company has decided to implement a chargeback program and determined the methods it will use to charge for each of its IS services, the next step is to introduce the concept to customers and obtain their buy-in. A three-step plan is helpful in gaining customers' understanding.

Meeting with the Customers

If chargebacks are being implemented for the first time, it is essential to explain to customers why they are being asked to pay for services that were previously free. Customers need to understand how costs will be calculated and should receive an estimate of their department's charges. IS should meet with each customer department individually and review the answers to three key questions: why, when, and how much?

Comparing Internal Rates to Contractor Charges. IS should also be prepared for questions comparing its internal rates to those of outside contractors. Customers may complain that outside contractors are less expensive. To determine if this is true, IS should obtain rates for comparable work from contracting firms. Because most contract programmers work on their customers' premises and use customer computers, the rates must be adjusted to include that overhead. The specific costs to be included are:

- Rent or occupancy charge
- General expenses (stationery and other supplies)
- Hardware depreciation
- Software purchases (if applicable)
- Management

These costs are a subset of the costs included in the calculation of an in-house hourly rate. The contract rate substitutes for salaries, benefits, training, and travel-and-entertainment expenses in the in-house calculation.

When contract programming costs are fully loaded, they are frequently higher than in-house programming because they are designed to generate a profit for the contracting firm. Most IS departments seek only to recover their costs.

Establishing Formal Contracts

Although contracts are not mandatory, they help to reduce ambiguity and the interdepartmental conflicts that can result from misunderstandings about services and when they are to be provided and at what cost. Even if IS proposes a single hourly rate for all services, it should give its customers written confirmation.

Instituting a Memo-Only Period

Although this step is not possible for all companies, a memo-only period — in which charges are calculated and reported to the departments but not charged against their budgets — is useful in gaining customer buy-in. It gives customers a chance to see the size of charges they are incurring before they affect their budgets and also familiarizes customers with the billing process. Benefits accrue to IS as well. The memo-only period allows IS to fine-tune and work out any bugs in its billing system without affecting customers.

SUCCESSFUL CHARGEBACK SYSTEM DESIGN

A successful chargeback system is marked by two primary characteristics. The first, and most important, is simplicity. An effective chargeback system is easy to understand and to explain to customers. Not only are arcane algorithms and complex formulas difficult to explain, but they make customer departments wary of the IS department's motives. A chargeback system should also be easy to administer. If it requires substantial overhead to record resource consumption and to bill customers, no one benefits, least of all the customers, because their charges will have to be increased to pay for the billing process.

The second desirable characteristic is a correlation between the charge and the use of a resource. If a charge appears to be arbitrary, such as a flat allocation of costs to a department based solely on the department's census and not on its actual use of IS services, the basic goals of a chargeback system cannot be accomplished. Specifically, there will be no incentive to conserve resources, and customers will not be held accountable for their use of resources. An effective chargeback system charges customers fairly for their use of all computing resources and rewards them for reduced consumption.

RECOMMENDED COURSE OF ACTION

Although it is unlikely that customers will fully welcome chargebacks, a program that is simple, equitable, and clearly communicated has a high chance of success. A company planning to implement a chargeback system should take these five actions.

Determine Goals. Identifying the forces that are driving the need for a chargeback system is the first step to determine the methods to be used. For example, if IS needs to recover all of its costs, it must charge for all services. If it only seeks to reduce use of prime-time CPU cycles, it may implement a charge for only computer-related costs. Goals should be clearly understood and outlined in writing before the program is defined.

Define the Simplest Way to Meet Those Goals. Not only should the billing algorithm be easy to understand, it should also be easy to administer. For most companies, this means buying or developing an automated time recording and billing system.

Keep Channels of Communication Open. Both the IS staff and customer departments will have concerns about the new system. These concerns can be diffused by having clear, open communications with both groups starting as soon as the chargeback plan has been developed.

Establish a Trial Period. By starting the program in midyear and implementing chargebacks on a memo-only basis for six months, IS can gain customer acceptance of the system at the same time that it streamlines its own procedures.

Be Prepared to Change. It is possible that the initial approach to chargebacks may not work. It may be too complex; it may not meet customers' needs; it may not result in full recovery of the IS department's costs. IS should carefully monitor the program and be prepared to modify its approach.

ABOUT THE AUTHOR

Christine B. Tayntor is manager of corporate staff applications at Allied-Signal, Inc. in Morristown, NJ.

Section II

E-Enabled Business Solutions

Aside from the technical challenges of running businesses over the Internet, the main challenge faced by Internet suppliers (e.g., anyone involved in building, deploying, and maintaining an application on the Internet) and users (e.g., anyone using an Internet application) is to determine how to make the Internet profitable — or at least self-sustaining. Perhaps for this reason, more than any other, the IT industry is focusing on defining and constructing business solutions over the Internet. In some cases, this means migrating an existing application to the Internet — E-enabling it, in fact. In other instances, it means constructing a business solution from the ground up with the Internet as a core component of the application architecture. On the whole, it is easier to justify a price on a business solution that provides a quantifiable business benefit to the end user — the person generally paying the bills — than individual system components that provide no easily quantified end value in themselves.

This chapter examines some of the more popular E-enabled business solutions in the industry through the following chapters:

“Building an E-Business Solution” (Chapter 8) discusses some tool suites for building competitive Web sites that are user-friendly, efficient, and effective. These solutions employ Internet standards that are popular in the IT industry.

“Customer Relationship Management: New Technology, Same Rules” (Chapter 9) examines technology that allows businesses to build solutions to improve their customer relationships and continue to bring customers back into their store, whether the operation is a total E-commerce entity or primarily a bricks-and-mortar shop.

“e-CRM Is Not eASY” (Chapter 10) details approaches for utilizing Internet-based technologies for building effective e-CRM business solutions. At the heart of effective e-CRM are the old-fashioned virtues of knowing your customers and making them feel they are the most important people to your company.

“Electronic Bill Presentment and Payment” (Chapter 11) discusses the emergence of this business solution and how it enables the goods/service provider to present the invoice details electronically to the consumer, who can access the information from any geographic location and authorize payment. Using EBPP, this entire process can be completed without generating even a single piece of paper.

“Are Human Resource Departments Ready for E-HR?” (Chapter 12) explores the human resource function as one of the latest functions undergoing Web enablement and changing the way HR professionals do their jobs and database managers hire their staffs.

"Call Management and the Internet" (Chapter 13) demonstrates how corporations can merge call centers and call distributors with the Internet to reduce equipment and personnel costs. This is examined in the context of telephony solutions.

Chapter 8
Building an E-Business Solution

Michael Simonyi

E-business, the newbuzz word for the next millennium, is forcing the corporate world to rethink its computing strategies. How can we reach our customer? How can we provide our customers with what they need when they need it? Perhaps a more important question is who are your customers, and what categories of customers are there? Most E-commerce solutions have typically catered to the retail consumer. In the corporate world that consumer could be brokers, subdivisions, subsidiaries, departments, suppliers, or perhaps even the board of directors.

E-business is not just about selling a product over the Internet anymore — it is about providing secured information to one's customer(s) so that they can make informed decisions that will ultimately impact the bottom line. Whether a company is selling widgets to the public, feeding new product information into a brokerage channel, or providing senior management with production statistics, E-business is about dissemination of information to the customer in a manner that allows them to use it effectively.

THE ROUTE TO E-BUSINESS

Delivering an E-business solution can be as simple as buying a package and plugging it in, using a service organization to build it, or using individual tools to handcraft the solution. The determination of needs and requirements are paramount before embarking on E-business development. Jumping into the game without knowing what one's needs are and what one's customers needs are can have disastrous effects on perception and unwanted publicity.

Determining the desired E-business requirements is the key to delivering an effective long-term solution. Figuring out what, when, who, where, and why will be the building blocks that form the foundation of any

0-8493-1161-6/02/$0.00+$1.50

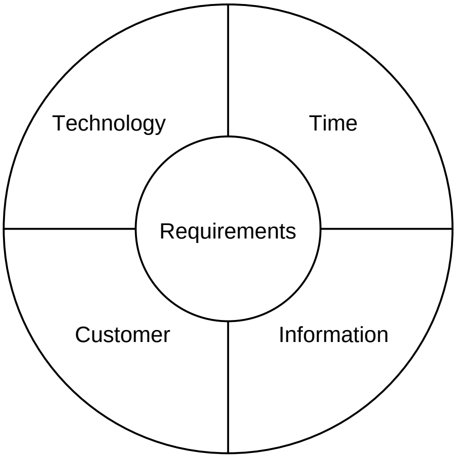


Exhibit 1. Requirements for E-business

E-business strategy. Exhibit 1 depicts the typical interdependencies that requirements can pose during needs and requirements definition stages. Exhibit 1 allows one to begin answering some fundamental questions about E-business strategy. For example:

1. What is the implementation timeframe?
2. What information is one going to disseminate?
3. What does one need to disseminate?
4. Who does one need to disseminate this information to and when?
5. What technology is available to make it happen?
6. Will it work within the existing environment?
7. What is startup cost going to be?

Conversely, one can also begin looking at questions from the customer's point of view. For example:

1. What information does the customer need?
2. What information does the customer want?
3. How much time will it take for does the customer to get the needed information?
4. How difficult will it be to get the information?
5. What technology does the customer require to get at the needed information?

The basic rule to follow here is to remember that it is not just what one thinks needs to be available to the customer. It is about what the customer is expecting to be available.

THE PRESENTATION

Perhaps one of the most important aspects of delivering an E-business site will be its presentation. The presentation layer is the one area that fully exposes the corporate presence to the customer. The design of the site motif will be critical to its acceptance by the customer base. However, this does not mean there should be an excessive use of "eye candy" to sell the site. An overabundance of graphics, video, audio, or text will make the site unusable and prone to failure in the long term.

The presentation layer is composed of two separate functional components: (1) the client layer, which is typically the browser, and (2) the server layer, which is the Web server. Together, these two layers comprise the full range of client functionality exhibited on the browser.

With proper use of the presentation tools, it is possible to craft efficient page download characteristics, design uniformity, and pattern future maintenance cycles. Because the lowest common denominator for all Web presentation layers is HTML, the most practical design approach should be solely based upon its use. As the introduction of forthcoming standards such as XML and XSL become available on the horizon, emphasis should be placed on where these technologies can be inserted for future maintenance release cycles. During the design phase of the project, areas of improvement should be identified where these new technologies can be utilized.

The general approach used today to architect Web presentation layers is modeled after the thin-client approach. HTML lends its self quite well to this type of implementation. The use of Java, ActiveX, or other downloadable components or support controls should be avoided. These types of technologies will impact initial page access times, affect future maintenance releases, increase testing cycles, and limit browser compatibility. The nature of these components allows for distributed processing. This type of process unloading at the client will, over time, become a maintenance burden and support nightmare. The most unwanted feature of this type of technology, however, is browser compatibility. Limiting the site's access to a specific browser will limit one's customer base.

In addition to HTML, JavaScript can be used to complement the design of the presentation layer. More robust and functional tools such as Java, CGI, Perl, or VB Script can be used on the server-side presentation layer to augment complexities necessary for the application that would not be suitable for processing on the client browser. These tool sets can be designed to provide transactional components for the middle layer. Transaction servers can use Java Beans or other high-level languages such as C, C++, or Visual Basic to form the foundation of the business tier. The types of tools that can be used will, of course, be limited by the Web server platform of choice. For example, Apache is best-suited for Java development; Lotus Domino can support both Java, and Lotus Script, and can interface with Microsoft's IIS platform to extend the capabilities of Domino with COM (Component Object Model) technology. For the purposes of this article, the Microsoft Internet deployment strategy is referenced.

Considerations for Web Interface Design

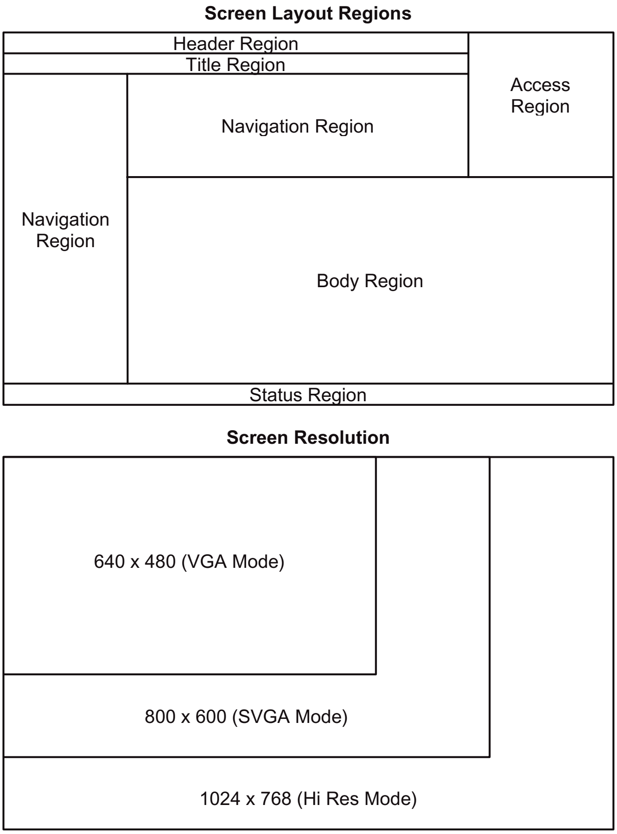
Planning for the interface design is one of the most important aspects of the overall site design process. The interface design process can be divided into four primary sections: screen layout, page content, technology requirements, and corporate requirements.

Screen Layout. The site layout or screen layout (see Exhibit 2) should be based on display regions. Regions are used to define areas of the screen that pertain to specific content and functionality. These regions are typically broken down into six regions, Header region, Title region, Access region, Navigation region, Body region, and Status region. Exhibit 2 depicts a typical region layout scenario; multiple combinations can be arranged, and finding the desired design layout will take time and should ideally be designed by a graphics artist. Knowing the presentation real estate will also contribute to the overall layout. The greater the layout area, the more creatively the content can be assembled and presented.

Page Content. The page content covers many factors — from the actual information content that is to be presented to items such as terminology of proprietary data, date, and currency formatting. It also deals with methods of content linking, linking to support material or external systems. Page content should be designed to stand on its own. Content should also take into consideration the dating of the material. Information content should be as accurate as possible as of a specified date.

Technology Requirements. The minimum technology requirements are the defined absolute minimum requirements necessary to view, access, and interact with the site. This area deals with technical issues such as screen resolution size requirements, (640×480 VGA, 800×600 SVGA, or 1024×768), color depth requirements for each of the resolution sizes, minimum connection speed such as 28.8 bps, 64 Kbps ISDN, ADSL, cable or T1 connections. Browser support for Microsoft Internet Explorer or Netscape Navigator and base-supported versions, that is, 4.x and up. It also defines printing requirements for portrait, landscape, or A4 sizes. It defines page download timeframes. For example:

- Static content should be available in two to five seconds at peak loading.
- Graphical content should be available in five to fifteen seconds at peak loading.
- Dynamic content should be available in five to twenty seconds at peak loading.



Region	Description
Header	Corporate identification and application name
Title	Page or form identification
Access	User role and access level information
Navigation	Site navigation, hyperlinks etc.
Body	Site interaction area
Status	Site status information for user requests

Exhibit 2. Screen Layout

Working with defined download timeframes should provide for optimal page loading even as site load characteristics increase above regulated traffic patterns.

Corporate Requirements. Corporate requirements revolve around known entities and standards. These standards are based on corporate logo requirements, color schemes, fonts, font sizes, and type display requirements. It should be determined up front if these will be extended to the

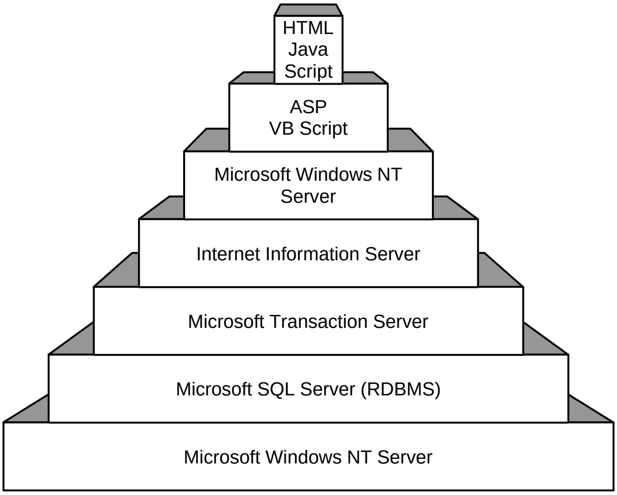

HTML
Java
Script
ASP
VB Script
Microsoft Windows NT
Server
Internet Information Server
Microsoft Transaction Server
Microsoft SQL Server (RDBMS)
Microsoft Windows NT Server

Exhibit 3. Technology for E-business

Web or if only a portion will be extended to the Web and how they will be extended. These are important requirements and should not be distorted just for the site.

MICROSOFT'S CORE TECHNOLOGY FOUNDATION

The core foundation technology of Microsoft's Internet strategy is based on the following components. These have been layered from lowest to highest in order to display the technology dependency pattern (see Exhibit 3). These components are defined in Exhibit 4.

The individual layers define the delivery mechanisms for the technology. Each of these layers provides its own opportunities for development and is feature-rich to allow for system scalability. The three major components that define the Web solutions strategy are the Internet information server, transaction server, and SQL server. These components are the foundation for the construction of any Internet-based E-commerce solution. Pooled together properly, an effective, robust, and scalable solution can be handcrafted to construct extremely complex and business-sensitive sites.

Internet Information Server

Internet Information Server (IIS) is a Web server computing platform ideally suited for developing, deploying, and managing high-performance, scalable, and robust enterprise Internet server applications. Its ability to tap into a vast array of development tools and business support products

Exhibit 4. Microsoft's Core Technology Foundation Components

Layer	Description
HTML, JavaScript	This layer is dedicated to the browser's client presentation layer. HTML and JavaScript are used for speed, flexibility, and effective uniform page design. Both technologies provide for effective use between the major browser vendors.
ASP, VB Script	This layer provides highly specific functional application development for pre-processing of data and presentation characteristics for that data. This layer makes up the other half of the client presentation layer.
Site Server Commerce Server	This is a layer that has been shown for functional purposes only. This layer is a generator layer. These products are used to create E-commerce site swiftly, based on the above presentation layers. In addition, these products use additional controls and add-ins to add functionality to the site.
Internet Information Server	This layer is the Web server platform. This constitutes the bulk of the presentation services layer. All the above layers operate within the confines of this component. It also provides the services for HTTP, SMTP, FTP, and other Web-related services.
Microsoft Transaction Server	This layer is the business services layer. The transaction server provides business-level services to the presentation layer to perform business-related functions. The transaction server layer is typically broken down into multiple layers in order to define specific business operations and data access operation in the context of a business function.
Microsoft SQL Server	This layer formulates the data service layer. This is the relational database management system (RDBMS). The data is constructed in accordance with the business-level requirements in order to facilitate a strong bond between the business layer and the data layer.
Microsoft Windows NT Server	This is the operating system platform for the environment. Each of the major services defined above runs within the context of the operating system platform. Only the browser client can function outside of the area.

makes it a prime candidate for E-business applications development. Other Web server computing environments must rely on third-party application development components to attempt to even come close to Microsoft's rich product offerings. Although other products such as Apache and Lotus Domino provide avenues for construction of high-volume, scalable, and robust E-business sites, their dependency on third-party products can impact development timeframes.

The IIS runtime infrastructure makes application development, deployment, and management simple by providing a comprehensive and easy-to-use set of subsystem services for application development and system administration. Utilities and administrative support offerings make the IIS platform easy to use, deploy, and troubleshoot.

The integration of the Microsoft Management console allows for easy configuration and support of the platform via an explorer-type interface. Performance analysis tools such as Web Cat, Homer, Orville, and iNetMonitor aid in site development and performance optimization.

Transaction Server

Microsoft Transaction Server (MTS) is a transaction processor. It provides a middle-tier deployment infrastructure that is designed to work with a variety of technologies. The development technologies currently supported by MTS are Visual C++, Visual Basic, and Visual J++. Each of these development languages can be used to develop and deploy robust transactional components for a large-scale, mission-critical application.

Although MTS only runs on top of the Microsoft Windows NT operating system, the environment does support access to other relational databases such as Oracle and Sybase. This support provides an even further reach to access data that may only be available via legacy systems.

As with IIS, MTS is also supported by the Microsoft Management console. This console interface allows for easy maintenance and support of the environment, and also allows for the setup of security at both the role and component levels.

The ability for MTS to pool both database connections and object instances make it ideal for high volume transaction processing systems. The integration of the development tools suite allows developers to take advantage of MTS's offerings without having to code complex transactional or concurrency routines. These features alone provide latitude over competing products by allowing developers to concentrate on business-related issues as opposed to technical-related issues.

SQL Server

The data repository of choice is Microsoft SQL server. The power, stability and manageability of MS SQL server round off the entire development and deployment platform very nicely. The strong foundation of an enterprise class RDBMS is paramount to beginning any E-business development and MS SQL Server fits that role nicely.

In addition, Microsoft SQL Server also provides administration and maintenance via the Microsoft Management Console.

CONSTRUCTION REQUIREMENTS OF THE APPLICATION TIERS

Presentation Tier

This tier refers to the browser. Any browser should be able to form the presentation tier as long as the browser supports the minimum technology requirements (i.e., HTML version support, etc.).

Construction of this tier should primarily focus on HTML code and JavaScript code to ensure low deployment overhead and maintenance, and excellent response time. Although the browser cannot be administered, checks can be put in place to make sure required options are set by the browser prior to application entry. Any deviations in browser technology should be handled via HTML or JavaScript as best as possible. However, any major deviations should be highlighted or explained via a simple site note such as, "This site is best viewed through Netscape 4.x."

Client Tier

The client tier within the context of a multi-tiered application in this case is Microsoft Internet Information Server (IIS). IIS is considered the client tier in this context because it performs the physical interaction between the business-level tier and the Active Server Pages (ASPs) to construct a display request.

This tier acts as a buffer zone to preprocess any information prior to its formatting in HTML to be sent to the presentation tier. The IIS server also provides an additional layer of security shielding. The added layer of security is accomplished by only allowing the viewing of HTML markup content to the browser. This masks the ASP markup code and any server-side scripting that would otherwise expose back-end systems to the Internet.

Construction of the client tier should take into account the centralized nature of the application, and its low deployment overhead and maintenance characteristics. The centralized nature of this tier can be safe-guarded through the use of load balancing and clustering. The key benefit to this approach is the centralized distribution of the pages that make up the site.

Business Tier

Microsoft Transaction Server (MTS) forms the basis of the business tier. The role of the business tier is to encapsulate all business-level logic. Business rules, processing, or other business-specific items operate within this domain. This logic resides within the confines of the MTS.

The business layer is typically subdivided into two or more physical layers. Each higher-level business layer pertains to actual processing requirements or business-level logic. The lowest business-level layer pertains to data access. In most scenarios, there is a one-to-one relationship between a business-layer process and a business layer data access object. Each business-layer object interacts with at least one data access object. These objects are dynamically bound at runtime only. The isolation of the layers allows for future migration to alternate data models or even databases, if so desired. For example, if the database currently being referenced is an

Oracle database that will be migrated to an SQL server database, only the business data access layer needs to be rewritten.

The ability to break down the business components provides the capability to load-balance objects across multiple servers to achieve higher levels of scalability. In addition, the object pooling[1] characteristics of MTS allows for a single object instance to be used by multiple clients, which provides lower memory utilization that in turn reduces server loading during peak access intervals.

Data Tier

The relational database itself forms the data tier. The physical manifestation of the database can be accomplished by way of a variety of different databases, including but not limited to Microsoft SQL server, Sybase SQL server, or even Oracle.

The data tier forms the physical data store of the application. Each of the major database vendors noted above supports the critical features necessary. These features, such as stored procedures, triggers, and other techniques, are utilized for increasing performance throughput. Stored procedures are primarily used to enhance data retrieval times for the business layer's data access functions. It is critical to maintain a database structure that closely follows the business data access objects to prevent the use of dynamically generated SQL statements.

If the database model is well designed, the need for dynamic SQL statements will be negligible. However, should dynamically generated SQL statements become the desired method of accessing the database, the data model should be redressed immediately. Use of dynamic SQL leads to extremely complex SQL scripts that eventually become unmaintainable. It also points to an inherent weakness in the data model, which will eventually become a serious problem.

Great care should be given to the design of the data model. Without a well-conceived data model, the foundation of the application will be weak and lead to a maintenance nightmare. The development of a strong data model ultimately leads to enhanced performance, scalability, and manageability.

Conceptual Application Model

The conceptual application model (see Exhibit 5) depicts the graphical representation of an entire infrastructure and separation between layers.

DEPLOYMENT TECHNIQUES

There are a number of deployment techniques available for this platform. Primary consideration should be given to security — for obvious reasons.

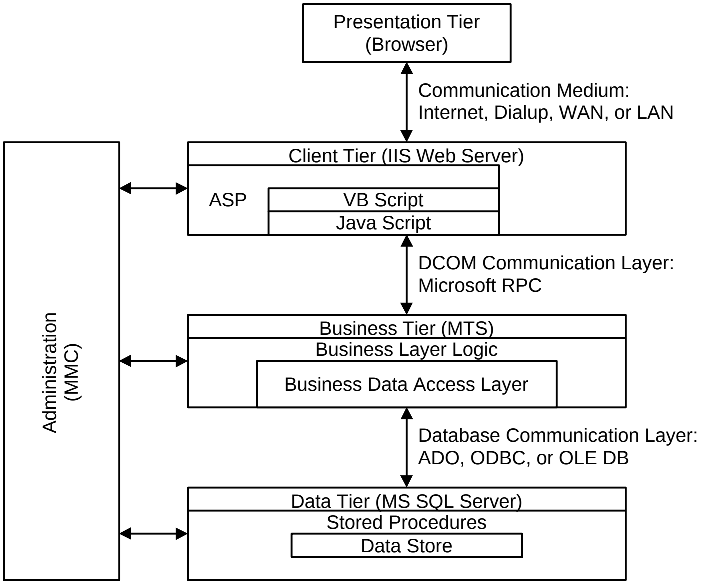


Exhibit 5. Conceptual Application Model

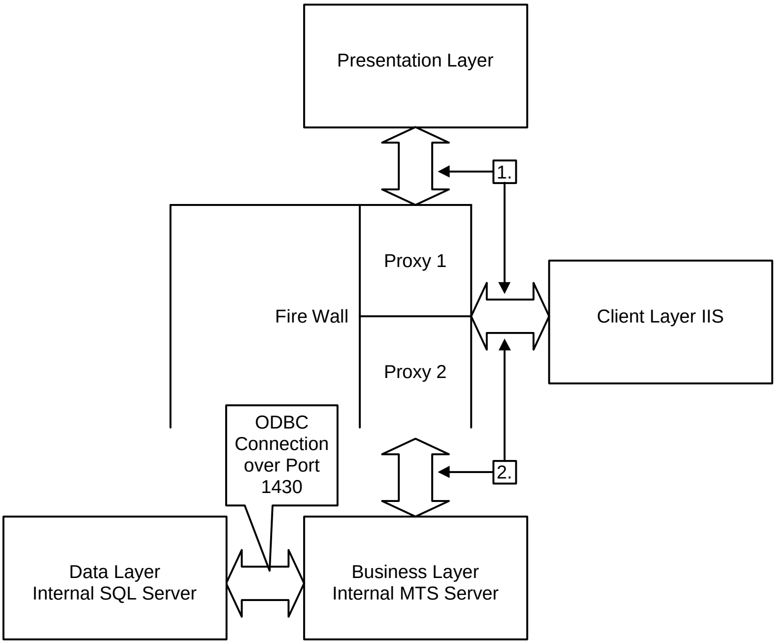
The security mechanisms for the presentation layer and client layer data transport requires encryption for all confidential transactions.

Confidential transactions are carried out over Secured Sockets Layer (SLL — port 443). To support this level of encryption, a server-side certificate is necessary. These certificates can be acquired from Versign, Entrust, or other security vendors. Additional requirements will be necessary for consumer-related sites that require credit card transactions. The integration of credit card payment into an E-commerce site is actually not that difficult when utilizing Microsoft's Pipeline technology. There are actually component object model (COM) components that can be integrated into an E-business solution quite swiftly.

Additional security concerns should be addressed with the inclusion of a firewall. The firewall will actually round off the overall security for the site. If the site will service departmental offices within a corporation, a proxy server can be used in place of a firewall. Exhibit 6 depicts a typical firewall implementation for a site.

FUTURE DIRECTIONS IN E-COMMERCE

With the introduction of Windows 2000, Microsoft introduces a new technology platform for developing and deploying highly available E-commerce systems. New generations of tool sets will allow developers to design and construct even more powerful business solutions. Windows 2000 promises


Presentation Layer
1.
Proxy 1
Fire Wall
Client Layer IIS
Proxy 2
ODBC Connection over Port 1430
2.
Data Layer Internal SQL Server
Business Layer Internal MTS Server

1. Internet Connectivity comes in on Port 80 (Standard HTTP traffic) and Port 443 (Secured HTTP traffic), IP Address aaa.bbb.ccc.ddd. Traffic is re-directed to IP Address eee.fff.ggg.hhh over Ports 80 or 443. No other Ports are required to be open form the Internet side for this application.
2. The Web Server takes requests from the IIS Server and directs them to the MTS over Port 135, the Microsoft RPC NT administration Port. It is not possible to use Network Address Translation to map a logical IP address to a physical IP address due to COM/DCOM limitations. This will be addressed with COM+ in Windows 2000.

Exhibit 6. Typical Firewall Implementation

to deliver higher levels of reliability and performance, but also maintain its ease of use.

Windows DNA (Distributed interNet Architecture) will fulfill the requirements to design and build flexible, reliable, and highly scalable E-business applications. The tool sets that will deliver the foundation of Windows DNA will allow one to build on top of the previous generation of tools. Most notable among these tools will be the ability to manage server farms by way of windows load-balancing technology and cluster services.

The Windows DNA will consist of the components listed in Exhibit 7. These tools build upon an already strong foundation of E-business development tools. As the tool sets mature, developers will be able to focus more on the business requirements of the commerce site model than with the technical issues that surround the technology.

Exhibit 7. Windows-Distributed interNet Architecture Components

Tool	Application
Visual Studio	Application development platform
AppCenter Server	High Availability/Server farm management
Commerce Server	Business-to-consumer E-commerce platform
BizTalk Server	Business-to-business E-commerce platform
Host Integration Server	Legacy system connectivity
SQL Server	"Shiloh" next-generation RDBMS
Windows 2000	Application server platform

ABOUT THE AUTHOR

Michael Simonyi is an IT professional working for private sector enterprise organizations. He has more than twelve years of practical and theoretical experience, from mainframe systems to PC client/server networks. His areas of expertise center on practical systems management, networking, databases, and application architecture, with emphasis on quality.

Chapter 9
Customer Relationship Management: New Technology, Same Rules

Curtis Cook

Sam Walton, the legendary founder of Wal-Mart, was quoted as saying, "There is only one boss — the customer. And he can fire everybody in the company from the chairman on down, simply by spending his money somewhere else." Despite Mr. Walton's exclusionary language regarding women consumers, this is one marketing message that continues to ring true, even in the new economy. The customer rules. The increased use of technology in both brick-and-mortar operations, and "E-commerce-only" firms has not changed the balance of power in the least. If anything, technology has simply made it more apparent to companies that they must find new and innovative ways to address customer needs through multiple channels.

Ten years ago, customers came through the front door, called on the phone, or mailed an order — maybe they even used a fax machine. Today, the Internet, electronic mail, and the many faces of Web commerce have dramatically altered the ways in which customers and potential customers communicate with their vendors of choice. And businesses that understand that the customer rules are exploring all avenues to ensure that they can retain existing customers and attract new clients, whether by phone, fax, e-mail, or in person. In most cases, these companies are turning to one of the fastest growing and lucrative software areas in recent years — that of customer relationship management.

0-8493-1161-6/02/$0.00+$1.50

Customer relationship management (CRM) software is more than a fancy name for e-mail and a personal digital assistant (PDA). It is an automated adjunct to existing client service initiatives, designed to support all facets of business–customer interaction that traditionally were handled (or mishandled) by sales managers, salespeople, customer support personnel, and all of their respective software tools. CRM integrates the e-mail and PDA mentioned above with the day planner, the electronic scheduler, the client database, and a number of other business management tools, creating a single point from which to manage relationships with customers. Simply put, if it has something to do with maintaining existing clients and customers, a useful CRM system will have it. A more useful CRM solution may offer tools to attract new clients as well. In either case, the technology will put the customer at the center of the client relationship — a task at which many businesses fail.

CRM system vendors are attempting to convince businesses to clean up this critical element of operations. By all accounts, they have been successful messengers. The META Group Inc. forecasts that global CRM sales will reach $46 million by 2003, making the business of creating profitable customer relationships very profitable indeed. And, while the use of technology to manage customer accounts and related data is not new, it has evolved at a blistering pace.

CRM TECHNOLOGY: IN THE BEGINNING

First-generation CRM software focused on technology that helped the business owner determine who his or her most profitable customers were by tracking numerous aspects of the customers' interaction with the business. A database of current valuable customer information, combined with functions to manage sales leads, forecast potential future sales or new clients, and contact management capabilities, often served as the foundation for the "employee" component of the CRM solution. As for the customer, basic features to follow up on orders and lodge complaints were also integrated into the technology. Employees willing to use the technology found that, at its best, CRM software was arranged around the contact record, the customer. Every tool required to see the customer's historical relationship with the business, access the customer's contact information, and contact the client by various electronic means were seamlessly integrated with the customer name (for example) as the focal point. This ease of use took the focus off the application itself and placed the emphasis on effectively meeting client needs.

Early CRM efforts were aimed at perfecting the type of application described above. And while the software has evolved rapidly, it has only been in the last couple of years that is has caught up with the scorching pace of change in E-business. Two years ago, CRM insiders were predicting more seamless integration for remote use and that Web-based functionality would lead the way in CRM development, and they were half-right. The latest evolution has resulted in a migration of CRM solutions to the Internet. It has also changed the face of CRM completely.

GETTING TO KNOW THE CUSTOMER

If a customer spends enough time shopping at the same store, it is quite likely that the staff will get to know that customer. If a customer buys enough cars at the local dealership, the salesperson will start addressing that customer by first name and calling that customer when the new arrivals hit the lot. These individuals begin to get an idea about a costomer's preferences, budget, and other valuable nuggets of information that they can use to their advantage to make a sale. Early CRM was no different; the database of existing customers was managed effectively and sales information was analyzed to ensure that the most effective contacts were made with customers at the most opportune times.

Enter the Internet. It has been some time since the initial recognition that the Web would be a great medium for selling products to people. It took a little bit longer for all of these E-business pioneers to realize the severe limitations of customer relationships through computers. Clearly, most businesses underestimated the complexity of the Internet as a distribution channel, and those who recognized its complexity assumed that technology could provide the answer. They were half-right.

Today, most E-businesses realize that they cannot simply display a graphic of a consumer good with a competitive price and have their Web site overrun with online shoppers. They are learning that they must know more about their customers to compete because service matters — even in cyberspace. Yet, when one cannot see, hear, or even use one's own intuition with a customer, technology becomes the tool of choice.

THE MORE THINGS CHANGE, THE MORE THEY STAY THE SAME

It is not uncommon to hear from CRM vendors that the Internet and new, supporting technologies are critical factors influencing customer relationships. This can be interpreted in more than one way and, depending on one's preference, can be a dangerous approach to customer relationships. First, if one believes that widespread Internet use has changed the face of consumer/business relationships by providing more opportunity to access goods and services, not too many people will argue. This is a primary reason to jump on the E-commerce bandwagon and implement a Web-based customer strategy.

However, many CRM proponents believe that E-commerce is synonymous with a customer revolution — that somehow all this technology has created a major shift in the market forces that puts the power in the hands of the purchaser rather than the buyer. Consequently, businesses must now fight for customer loyalty and need the latest tools and applications to do it. This philosophy is dangerous for a couple of reasons. First, it presupposes that there is a power struggle between customers and businesses. That is not a healthy attitude upon which to build a business/customer relationship. And, second, it suggests that in the past, the vendor was in the driver's seat when it came to relationships with customers. No doubt many businesses may have believed this and may have instilled this thinking in their corporate approach to clients. Quite likely, these businesses have struggled with customer relationship issues.

Continued belief in this philosophy while adopting CRM solutions will only compound an existing problem. Technology or not, the customer, the customer's needs, and the best ways to keep or acquire customers are the driving forces on which businesses must focus. They always have been, and they always will be. The successful business will learn how technology can be integrated into the customer service strategy to enhance its focus on these driving forces.

REMINDERS FOR SUCCESSFUL CRM

There is no single approach for implementing CRM applications into a business. Every business is different — products and services are diverse, target markets respond differently, strategies are often fluid, and budgets vary. Yet, there are guidelines that will help any CRM effort stay on track while all of these operational issues are blended into the solution. Do not bother with the time and expense required to set up CRM applications for a business if "No" is the response to any of the following questions.

- *Are any of the "bosses" actively involved in the planning, design, and implementation of the customer management relationship solution?* At the very least, the CIO and the marketing and sales executives should be driving the CRM initiative. Additionally, involvement from key individuals in the trenches — the salesforce, marketing, communications, and others are all integral to developing the relationship the company is trying to create with its customers and potential customers. They are the people who realize, for example, that the customer relationship is going to be different in an E-commerce-only environment due to limited exposure to the customer. As such, it may comprise the collection of all the information that is obtained during an online transaction, as well as during a visit to the Web site when the experience does not necessarily include a purchase. This information is then used to develop improvements for future customer interactions, increasing the likelihood of a purchase or return visit. When CRM initiatives are thrust upon individuals who lack credibility, decision-making capacity, and resources, failure is a likely outcome regardless of the individuals' motivation.

- *Is CRM a business process issue for the company?* If CRM is regarded as a technology issue — approved by the CEO or the president and passed off to the IT department — it is going to fail. It does not matter how well intentioned and brilliant the technology folks are, the company is negating its possibility for success. Most CRM implementations either fail outright or are delivering unsatisfactory results. Like many "technology flavors of the month," companies are jumping on the CRM bandwagon with no idea about how it can work for them, let alone how to implement it; and they are paying the price. This leads to the next question.
- *Is there a CRM strategy?* While it seems obvious, so did enterprise resource planning (ERP) applications that consumed huge sums of corporate money with varying degrees of success. There are many routes that a strategy can take, as mentioned above, but there is always one great point of departure: keeping customers happy so that they return to do more business. It costs much more to acquire a customer than it does to keep a customer. That is why a sale to a customer is *good*, but a relationship with a customer is *great*. Web-based businesses are quickly learning that establishing such a relationship involves much more than bargain prices and gimmicky promotions.
- *Does the customer relationship initiative extend beyond the CRM application?* CRM is best used as a tool, not a company strategy to reach and retain customers. It plays more of a support role to strengthen existing efforts. As such, it will fail if it is used as a replacement or a cure-all for a company's weak approach to customers. At the end of the day, CRM software is simply software. To make it work, there must be commitment from the organization on two fronts: (1) commitment to learn how to maximize the capabilities of the CRM technology, and more importantly, (2) commitment to the ideals of great customer service.
- Pondering the former issue at the expense of the latter often pulls the focus of CRM into the realm of technological capability. This can be dangerous if one considers the objective: customer relationships. Consequently, it is best to avoid a CRM solutions provider that focuses too much, or completely, on the capabilities of the technology, while ignoring the type of customer experience envisioned by the organization. Finding the balance between the technology, the process, and the organization is a good objective.
- While CRM gravitates toward the Net, there is a potential downside to an organization. Granted, there may be increased efficiency in many types of dealings with customers and potential customers; however,

there is also the risk that the reduced flexibility that can only exist in human-to-human interaction will have a negative impact. After all, it remains impossible to build a "relationship" between a human, a server, and a Web site.

- Do you know what your customers want? It may take some effort to figure this out, but it is worth it. Once you understand your customers' preferences and your company's objectives with regard to client service and relationships, you have a solid foundation from which to build your CRM system. And, once the CRM system is in place, it will be collecting the data you require to spot future trends and keep abreast of your customers' ever-changing likes and dislikes.
- Are customer relationships paramount in the organization? Sam Walton did not need wall-to-wall technology to keep customers happy; he just needed to know that his livelihood — and that of his employees — ultimately depended on retaining customers and attracting new ones. Today, Wal-Mart's global chain of retail stores uses advanced technology to manage virtually all facets of its operations, including customer management relationships. Still, they have not said goodbye to the smiling face that greets each and every customer upon arrival. As hokey as that may seem in this impersonal world, it can still make the customer feel like a somebody.

ABOUT THE AUTHOR

Curtis Cook, CITP, is an international business strategist with Global Trade Solutions. He is the author of *Competitive Intelligence: Create an Intelligent Organization and Compete to Win* and numerous articles on business and technology. He can be reached at ccook@go-global.net.

Chapter 10
e-CRM Is Not eASY

Alex Lee

Customer relationship management (CRM) is the process of developing and reinforcing the relationships between customers and vendors. Based on the concept that in the modern business environment customers would rather purchase relationships than products, CRM promotes the cultivation and leverage of these relationships in order for companies to maximize their business potential. While its predecessor, sales force automation (SFA), focused more on the acquisition of customers and sales, CRM affects all aspects of an organization including management, research and development, finance, marketing, and support. It has been embraced by practically every major industry as the methodology by which organizations will conduct business in the 21st century.

Too often though, CRM is associated directly with technology. Many mistakenly believe that CRM hinges on its use of technology and that technology problems lead to its failure. While technology does significantly increase the effectiveness of CRM initiatives, CRM is really about designing business processes and methods to manage and analyze your business relationships regardless of technology. The focus is in better understanding your client's needs and requirements. It also provides the necessary framework to effectively solve problems with your customers when they arise.

eCRM is the application of e-technology (or Internet-based technology) to achieve CRM objectives. eCRM is to CRM what e-mail is to interpersonal communication. The recent popularity of eCRM has been motivated by the exploding popularity of Internet access through various devices (e.g., desktops, laptops, handhelds, mobile PCS, and television sets). Companies are beginning to recognize the potential in acquiring and maintaining customers through online means. eCRM applications can also offer organizations other benefits, like simplified remote access and management.

The challenges for CRM initiatives are well documented. The Gartner Group Inc. recently estimated that 60 percent of all CRM software implementations fail. This can largely be attributed to three main causes. First, most companies fail to make the necessary (and difficult) changes

0-8493-1161-6/02/$0.00+$1.50

required to implement CRM successfully. Hence, acceptance of new CRM methods is very low. Second, organizations fall to the temptation of continually customizing their CRM applications before they have been up and running for a period of time. As such, systems remain in a constant state of development, without any opportunity to mature. Third, many vendors tend to exaggerate the capabilities of CRM software, and thus, inflate the expectations of organizations. The shortfall results in a lack of commitment by users to the solution because they feel misled.

In order to have any success with CRM, a company must first be dedicated to the CRM concept. CRM can require changes in an organization and can force shifts in responsibilities and personnel. A willingness to accept change can be the key success factor in a CRM implementation. Organizations should also designate someone to champion or take ownership of the CRM project. Having a prominent and dedicated advocate, with authority over CRM initiatives, can greatly increase the chance of acceptance by the core staff.

The technical issues with CRM are not unlike those of any software development project. First, objectives and specifications must be well defined and documented. Tools and technology must be selected based on relevant criteria (e.g., features and costs). Implementation milestones are then established according to business timelines and availability of resources. A significant testing phase is recommended to ensure functionality so problems can be corrected on schedule. Final delivery of the application should also be accompanied by a maintenance plan for regular housekeeping issues (e.g., backups, synchronization with remote locations, and database maintenance).

Most organizations fail to leverage the value of their CRM data into their management and marketing decisions. Companies, such as Hyperion that develops CRM analysis software, emphasize the power of analyzing your CRM data to better understand the behavior and needs of your clients. They also promote using metrics from your CRM data to formulate vital signs for your company.

Hyperion uses the term *touch point* to describe each potential customer contact. Traditional touch points consist of things like phone calls, meetings, seminars, and correspondence (see Exhibit 1). The use of Internet-based technologies introduces many new additional touch points such as e-mail, Web site visits, newsgroups, chat groups, and Webcasts. Hence, the excitement over eCRM is driven by how many new touch points are potentially created between an organization and its customers. The other major advantage of eCRM is that many Internet touch points can be automated and their usage can be recorded. This makes it even easier to produce statistics to contribute to the CRM effort.

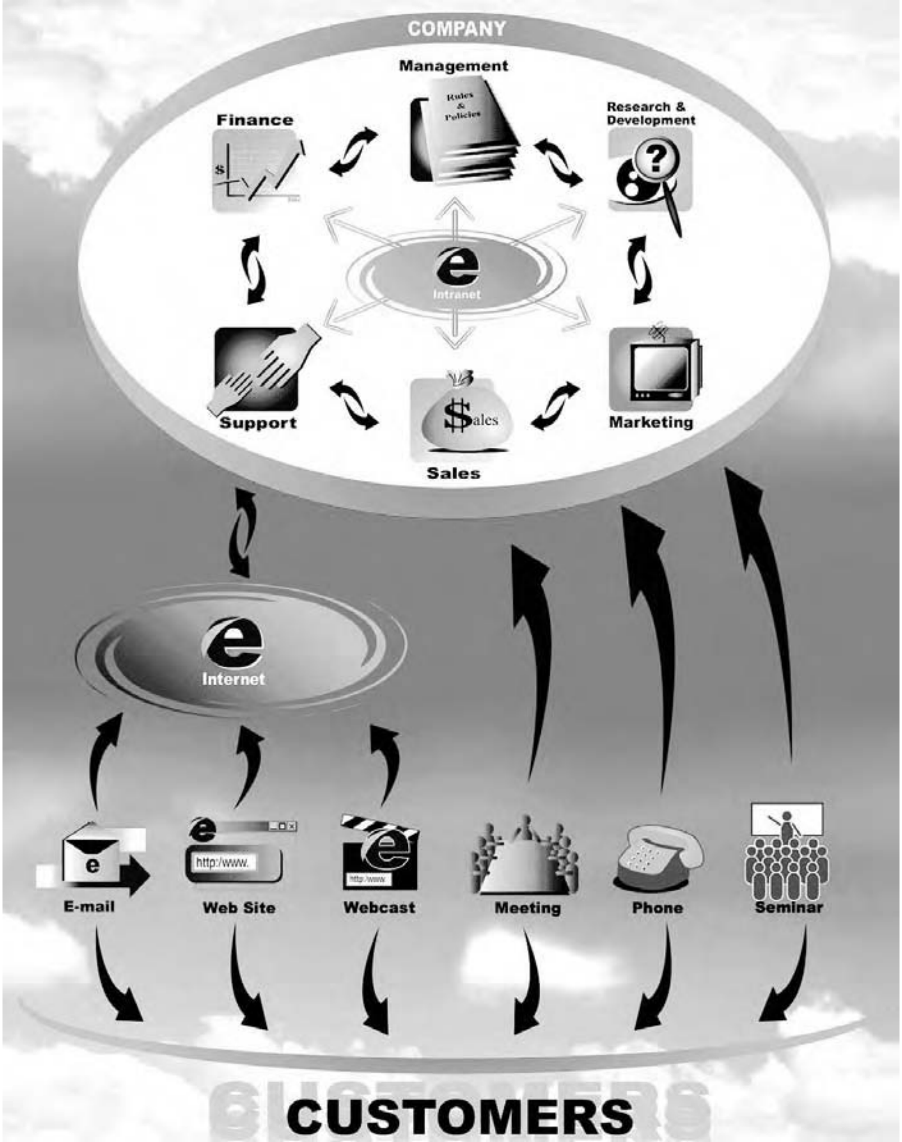


Exhibit 1. eCRM Model

Consider a scenario where a company is promoting a new product line by inviting existing customers to a seminar. At the seminar, the presenter lets everyone know that a promotional package will be sent to everyone who submits a business card to the reception desk. Feedback forms are passed out to all attendees with the promise of a company-logo key chain for all those who complete the form. After the seminar, data entry staff will enter all the business card data and feedback forms into a database, which may or may not be used to actually send out those promotional packages to every single card contributor. More likely, an array of sales people will start calling each attendee, hoping to generate a face-to-face meeting.

In the eCRM world, the presenter would let everyone know that the promotional package can be downloaded from the corporate Web site in a PDF (Adobe's Portable Document Format) format. When customers access the Web site, the site will ask them to enter their name, address, and e-mail information. If the customer appears on the list of seminar attendees, then the site will pop up the feedback form and ask the user to complete the form prior to downloading the promotional package. Sales people can then e-mail those people who downloaded the package for follow-up.

The foregoing scenarios illustrate the fundamentals of eCRM and, more importantly, the subtle advantages of the eCRM situation. For example, the eCRM solution avoids much of the data entry overhead. It also avoids many costs, in particular, not having to mail out the promotional packages. But the real potential comes from being able to answer some interesting questions. For instance, how many attendees went to download the promotional package? How many people completed the feedback form? How many people requested the package but backed out once the feedback form was presented? How many days after the seminar did people access the Web site? How often did a customer revisit the site after the seminar? Such statistics cannot be easily derived through traditional means.

Now let us reconsider the scenario above, but with a few additional eCRM enhancements. When the customer accesses the Web site, he (or she) will now be prompted to enter his name and company. Based on the attendance list from the seminar, the customer is asked to confirm his identity and then enter his e-mail address if it is not already in the system. The promotional package will come in the form of a Flash movie with background audio and commentary. After viewing the movie, the site offers to connect the customer to a live sales associate via a Webcam. As soon as the customer has left the site, the system also notifies the appropriate sales representative who can follow up with the customer at a later date.

This scenario is undoubtedly more "slick" and it illustrates two design elements of effective eCRM applications. First, it is sensitive to the fact that customers are tired of constantly providing information that a company should already have. Having a customer provide his name and address at every touch point is akin to forcing him to introduce himself every time he meets with the same sales person. It is inefficient and, more importantly, it reflects to the customer the company's inability to manage its own information.

The second design element is not as obvious. In order to achieve the "slick" eCRM scenario, the company has to evolve from the prior eCRM scenario. It is extremely difficult and hazardous to go from no eCRM methodology to a sophisticated eCRM scenario without taking some intermediary steps. The business processes and workflows have to be well designed for CRM to work. Personnel at each stage also need to be sufficiently trained as to how to handle each possible situation and contingency. These processes and procedures do not happen overnight and are the kinds of challenges that organizations must overcome in order to make CRM a success.

One of the common misconceptions is that CRM starts with purchasing a CRM software package. Many small businesses (without realizing it) practice CRM using nothing more sophisticated than a simple database. Technology is only the tool that is used to implement CRM. The quality of an organization's CRM project is no more reflected in the choice of CRM software than is the quality of a house reflected by the choice of hammer used to build it. With that said, many of the CRM packages on the market are very good in assisting an organization with CRM initiatives. Also, it would be very difficult to implement any type of large-scale CRM objectives without using a significant amount of technology (just like it is tough to build a house without a hammer).

Another idea that has been promoted, even by some CRM advocates, is that an organization should avoid CRM packages that cannot conform to its existing business processes. This is a little misleading. What is really being suggested is organizations should avoid CRM packages that appear overly rigid or inflexible as far as customization is concerned. If your current business processes are working (and you have the statistics to prove it), then it makes little sense to reorganize your company to accommodate a particular piece of software. However, in most cases, a company is implementing CRM (or eCRM) because it wants to overhaul its business processes. In such cases, implementing CRM software might be the impetus that a company needs to get started.

Designing an eCRM solution begins with identifying the aspects of your business that can benefit from CRM. Typically, most organizations first look to eCRM for client acquisition (which usually means a Web site). However, as previously suggested, one's Web site can be useful in understanding your potential and current customers and their behavior. It also helps to reveal the portions of your site that work and those that do not. Organizations are also only beginning to understand the power of e-mail in attracting customers. While not promoting spam e-mail, a well-designed, well-timed e-mail message can be very influential in attracting customers.

Again, eCRM methods allow you to easily measure the response rates for such campaigns.

As the name suggests, eCRM should be about managing relationships with your customers. This is an aspect where eCRM is superior to traditional CRM because of the additional touch points introduced by the Internet and the increasing ease with which people access the Internet. It is also important to note that eCRM can also leverage *internal* touch points and help an organization manage the relationships among departments.

E-commerce is another area where eCRM has huge potential. Besides the apparent benefits of electronic client transactions, perhaps the most compelling illustration of the potential of eCRM in e-commerce is seen in sites like Amazon.com, which not only show you books of a similar nature to the one you are selecting, but will also remember your past purchases in order to make suggestions the next time you visit. The power of this feature is due to the site's ability to personalize itself to each user automatically. In essence, you are creating a unique sales person who knows every client very well and offers tailored advice.

Information technology (IT) vendors understand better than most the power of eCRM in providing after-sales support for customers. In providing support, eCRM recognizes that 90 percent of customer issues are likely repeat issues that will be shared by many customers. In that regard, eCRM encourages automating responses to clients using facilities such as knowledge bases and FAQs (frequently asked questions). Discussion forums also provide an excellent electronic means for customers to gather and resolve issues. In many cases, another customer will likely answer a customer's concern. These same mechanisms can easily be applied to other businesses, with similar effectiveness.

eCRM need not just benefit businesses, but any organization. For example, some hospitals now allow doctors to monitor patients over the Internet, even from mobile devices. Charity organizations can use eCRM methods to garner sponsors and bring awareness to their causes. Schools could also use eCRM to build better relationships between parents and teachers. Even government, which is constantly trying to balance policy with public opinion, can benefit from the concepts that eCRM advocates.

In terms of specific solutions, it is difficult to declare an outright leader among the CRM vendors. Certainly names like Pivotal and Siebel are well recognized by most organizations. However, a significant number of lesser-known vendors offer extremely powerful software that may be well suited to your particular needs. Most vendors have also recognized the potential of eCRM and have made great efforts in providing better integration with the Internet. In fact, the Internet-based solutions are becoming so popular that some organizations forgo the traditional CRM client software in favor of an entirely Web-based interface. An excellent resource at the time for writing for comparing various CRM solutions is the popular CRM Guru Web site (www.crmguru.com), which offers a CRM Solutions Guide.

eCRM is not about a specific software or methodology. It is an innovative new approach in dealing with clients and customers, utilizing Internet-based technologies. At the heart of effective eCRM are old-fashioned virtues of knowing your customers and making them feel they are the most important people to your company.

ABOUT THE AUTHOR

Alex Lee is a co-founder and president of QUEUE Systems Inc. (www.QUEUE-Systems.net). Alex graduated from the University of Waterloo with a degree in systems design engineering. Established in 1989, QUEUE Systems is a multidisciplinary consulting firm that provides complementary resources in its IT Consulting, Placement Agency, and New Media divisions.

Chapter 11
Electronic Bill Presentment and Payment

Rahul Kumar

The emergence of the Internet has deeply affected the way in which business is conducted. It has simplified transactions between entities by connecting them "virtually." It has enabled people to transact business online (i.e., order goods and services) as well as pay for them. However, until some time back, although the consumers used their credit cards to pay online, they could not see their credit card invoice online and had to wait for a paper copy of the credit card invoice. This situation was not restricted to credit cards only. Many utility companies like telephone, cable, and water used to send the monthly invoices in the form of a paper copy. This meant that the consumers could know their balance only at the end of the month, had to wait for the paper bill to arrive in the mail, and only then send a check (another paper transaction) to pay for the service. The entire process had a lot of delay designed into it and the information being sent and received by the parties involved was never dynamic.

This scenario has changed with the emergence of electronic bill presentment and payment (EBPP). EBPP enables the goods/service provider to present the invoice details electronically to the consumer who can access the information from any geographic location (having access to the Internet) and authorize the payment. This process can be completed without transacting even one piece of paper. The impact is enormous in light of the fact that in the United States alone, 1.5 billion invoices are sent to the consumers every month. This translates to 18 billion invoices a year or $6 billion in processing fees (according to the research firm International Data Corp.). The market opportunity for EBPP, and especially the firms who provide the systems to enable it, is huge.

0-8493-1161-6/02/$0.00+$1.50

EMERGENCE OF EBPP

Remote banking and payment of invoices via telephone have been around for two decades. However, they have not met with huge success because of the cumbersome nature of the transactions and the primitive infrastructure (though they continue to be used by the people who are "laggards" in the technology adoption curve).

The first electronic interactions between the bank and the consumers took place in the 1980s with the emergence of personal computers (PCs). Personal finance management (PFM) systems (like Meca's Managing Your Money, Intuit's Quicken, and Microsoft's Money) made it possible for users to download their bank statements electronically and authorize payments. This was done using a private network. The drawback in this process was that the data resided on one's own PC and could not be accessed from any other location.

With the emergence of the Internet in the 1990s, it became possible to connect all users to their banks, electronically, without the use of a proprietary link. The World Wide Web allowed the users to access the central secure location where the banks could post the account details using a bill payment software and enable the users to make payments by transferring funds from their accounts to the biller's. This marked the beginning of the popularity of EBPP.

WHAT IS EBPP?

As has been stated earlier, EBPP is the process in which the goods/services providers present invoices to its customers (either on their own or through a third-party provider) in an electronic format, which the customers can access using the Internet, view the details, and authorize their payments through their banks. The entire process is electronic with minimal manual intervention.

Here is how EBPP can work in real life: A telephone company, for example, extracts the billing data from its databases and converts it into an invoice in a format that can be hosted on the Web. This invoice is posted on either the company's own Web site or on the customer's financial institution's (bank) Web site. The customer accesses her account on the bank's Web site using her unique identification number and clicks on the icon representing her bills from various providers. She brings up the details of her telephone bill by clicking on the appropriate icon and verifies them. Upon satisfaction, she clicks on the "Pay Bill" icon that results in an electronic payment from her bank account to the telephone company.

The following sections describe the complete process of EBPP, the players, technology, and its drivers in details.

PARTICIPANTS IN THE PROCESS OF EBPP

Different entities participate in the EBPP process at different stages. These entities are:

- *Customers (or Payers)* — This is the entity that is responsible for making the final payment for the goods and services rendered to them. They could be individuals (e.g., buyers, consumers), other businesses, and government organizations. The payers are the ones that receive the invoice for their purchases and it is the replacement of this paper invoice by the electronic format that is the main objective of Electronic Bill Presentment. Electronic Bill Payment takes the process a bit further by enabling the payers to make the payment electronically, thereby eliminating the need for paper used for such things as checks and money orders.
- *Billers* — These are the entities that provide the goods and services that the payers desire. Billers can be individuals, businesses (e.g., telecommunications service providers), or government agencies (e.g., Tax Authorities). Billers are responsible for producing the invoice to be sent to the payers. Traditionally, this invoice has been a paper document, but not anymore with EBPP.
- *Banks (or Aggregators)* — Most of the non-cash payment methods require banks to play a major role. The same is the case in EBPP where banks, as representatives of the payers and billers (there may be two different banks representing the payer and the biller), complete the payment and posting functions on their behalf. They make sure that the billers get the payment owed to them by the payers once it has been authorized. The processing takes place behind the scenes through entities like clearinghouses and lockboxes (where payments are physically processed in financial institutions).
- *Processors (or Consolidators)* — Processors, as the name suggests, are the entities that are responsible for invoice preparation, delivery, and tracking of the invoice-related items. They are also known as consolidators or aggregators as they are a central point for the above-mentioned activities. For the purposes of processing the data for the billers, the processors charge them a fee. They are usually third-party providers (different from banks and billers). Another set of players that have joined the processors is the portals. Portals are like aggregators that perform the function of aggregating invoices for a set of customers that frequently access a network where all their billers would like their invoices to be displayed. E.g., the Web sites of Internet service providers (providing Internet access to their customers) could serve as the home site for invoices of all its customers.

Although the roles of each of the entities mentioned above are different, it does not exclude them from performing multiple roles. For example, a

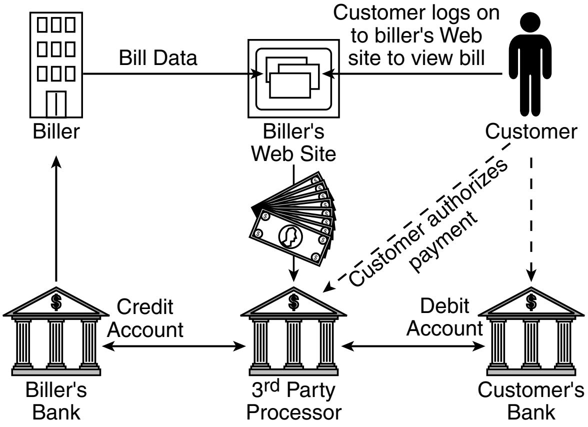


Exhibit 1. Direct Biller Model

bank can act as a biller (for its loans) as well as a processor for all its services and loans to its customers.

EBPP MODELS

Almost all of the variations of EBPP process can be structured under two models: *direct biller* and *consolidator models*.

Direct Biller Model

In the direct biller model, the biller presents the bill directly to the customer. The biller has complete control of all the data. Customers access their bills by logging onto the biller's Web site.

Billers create the electronic invoice by aggregating data from their own billing systems. Since they host the details of the bills, they have full access to monitor the customers' activities when they log onto their Web sites to view their bills. This provides the billers with a great opportunity to market and sell more products and services to them. At the same time, they are responsible for customer enrollment and authentication and thus also have complete information on the customer profiles, which enables them to perform targeted marketing as well. Exhibit 1 shows the transactions that occur in a direct biller model.

Since individual billers present their own bills, a typical customer would have to log onto multiple Web sites to access all their bills. This acts as a roadblock to the success of this model. The direct biller model is useful for billers who provide services for which the customers need to see a detailed bill. For example, customers making substantial long distance calls need to verify the details before paying for them. Many of the telephone companies host the long distance service bills for its customers and make them accessible online thereby successfully implementing the direct biller model.

The other area where this model is successful is for corporate customers (biller-to-business relationship) for whom one bill represents a large percentage of their total spent, so that they require the capability to route the bill through multiple departments for reconciliation purposes.

One of the other drawbacks of this model is that customers need to make multiple payments (one each at all the biller Web sites) to clear their dues. The model would work well for billers who use the *direct debit* mode of payment extensively. E.g., Sympatico, the largest ISP in Canada, provides detailed electronic bills to its customers who pay via direct debit of their checking account or credit card. The direct biller model also represents huge investments in technology by the biller. Billers need to set up systems that can handle the conversion of raw billing data into an Internet-compatible presentation format as well as build capabilities to accept payment online. The huge costs associated with setting up the infrastructure do not always make a strong business case for the direct biller model execution.

Consolidator Model

For customers who have multiple service providers and need the convenience of accessing all their bills from a single source, the direct biller model is cumbersome. The consolidator model addresses these needs by aggregating multiple bills and presenting them at a single location. In this model, the customers need to log onto just one site to access summaries and details of all their bills.

This model transfers control of data from the billers to the consolidators who consolidate multiple customer bills and present them through an aggregator (e.g., banks, financial institutions). The aggregators use online applications to provide customers an interface to electronic bills. With the consolidators acting as the central link in this model, it is their responsibility to invest in infrastructure and open systems that communicate with numerous billers as well as financial institutions.

Consolidators perform the function of enrollment of customers thereby eliminating the need for customers to enroll with multiple billers. By doing so, the consolidators get access to key customer data that they can use to drive traffic to their Web site. They also manage the payment-processing functionality by providing debit information to consumer banks and sending credit information to the biller banks.

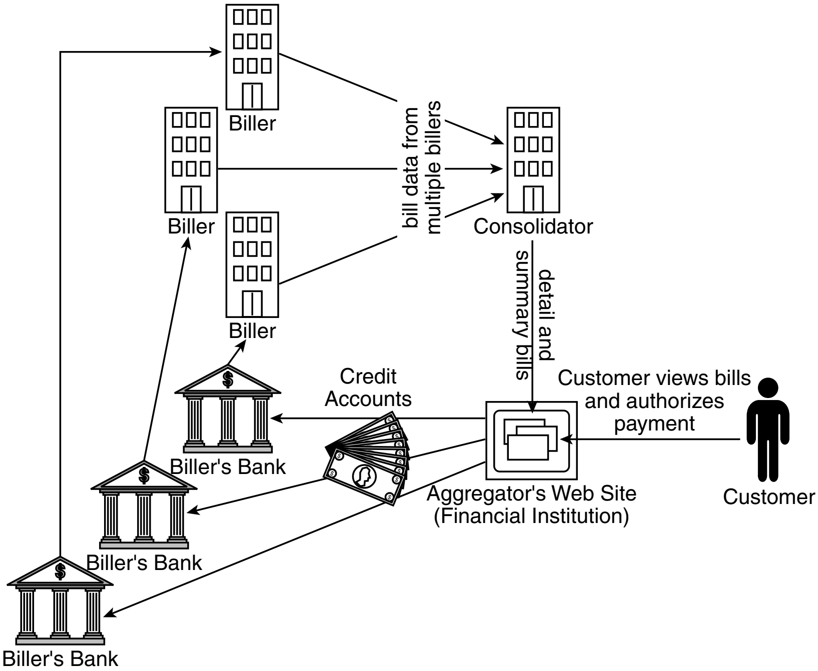

Biller
Biller
Biller
bill data from multiple billers
Consolidator
detail and summary bills
Credit Accounts
Customer views bills and authorizes payment
Customer
Aggregator's Web Site (Financial Institution)
Biller's Bank
Biller's Bank
Biller's Bank

Exhibit 2. Thick Consolidator Model

There are two basic variations to the consolidator model, namely, *thick consolidator* and *thin consolidator.*

Thick Consolidator. In the thick consolidator arrangement, the biller has the least control over the bill and customer ownership. They send the billing data to the thick consolidator who is responsible for both presentment and payment. The thick consolidator hosts the bill summary and details and presents the bills to the customer through an aggregator branded Web site. The aggregator, in most cases, is the customer's financial institution. Thus, the consolidator has access to all customer data that it can use to its own advantage. CheckFree and eroute Inc. are two primary vendors pushing the thick consolidator model. Exhibit 2 shows the transactions that occur in a thick consolidator model.

Thin Consolidator. The thin consolidator model allows the biller to retain some control over the bill. In this arrangement, the billers produce the bill summary and details from raw billing data and send only the summary information to the consolidator or aggregator who aggregates multiple bills and presents them on its Web site. This summary bill is enveloped in the URL of the biller's Web site so that the customer can easily access

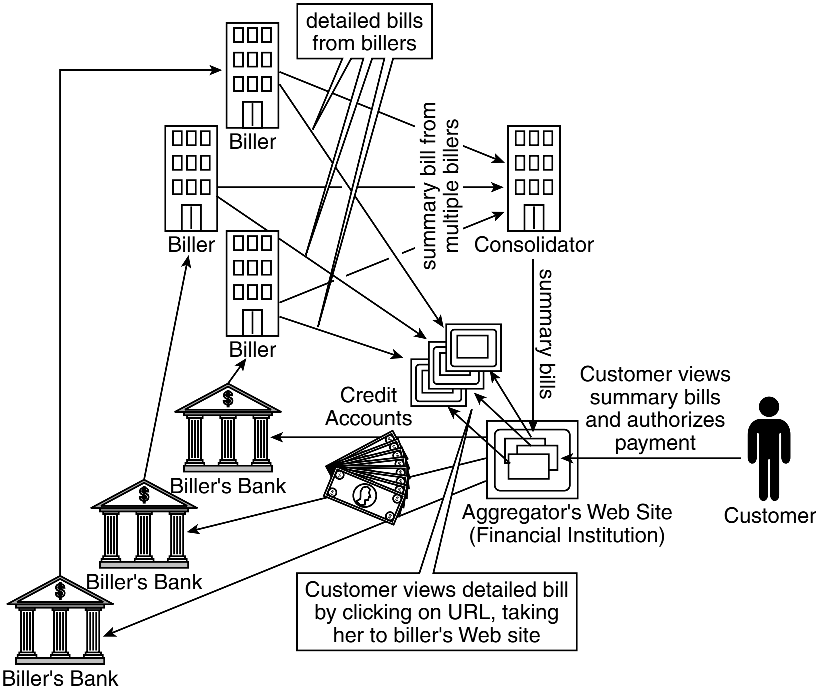

detailed bills from billers
Biller
Biller
Biller
summary bill from multiple billers
Consolidator
summary bills
Credit Accounts
Customer views summary bills and authorizes payment
Customer
Biller's Bank
Biller's Bank
Biller's Bank
Aggregator's Web Site (Financial Institution)
Customer views detailed bill by clicking on URL, taking her to biller's Web site

Exhibit 3. Thin Consolidator Model

the details by clicking on the hotlink. The aggregators in this model include personal financial managers and financial sites of portals such as Yahoo or Netscape. The thin consolidator model allows only summary bills to be hosted by the consolidator and requires the biller to communicate with multiple aggregators, as its customers may use different institutions as the aggregators. Exhibit 3 shows the transactions that occur in a thin consolidator model.

STAGES THAT A BILL GOES THROUGH

Before a bill reaches a customer and is paid via the customer's bank, there are numerous stages that the data, which make up this bill, go through in the complex process of EBPP. Each stage has its own importance and implications. The different stages are:

1. *Extraction of Data* — Data required for billing the customers needs to be extracted from the biller's systems before it can be sent to the customers or consolidators and be presented and paid. This may not always be easy because of the different formats in which it is stored in the legacy systems. Multiple systems that house this data

also complicate the extraction and affect the integrity of the data. At the same time, from the perspective of customer care, this data (and bills) also need to be stored in-house in order to assist the customer care representative in answering queries from customers. They need to look at the data in the same manner as it is presented to the customers.

2. *Profiling* — One of the most difficult areas to tackle, "profiling," refers to any addition, change, or deletion of customer information to or from the customer database. To perform activities that come under the profiling banner, information in multiple files and databases needs to be managed. One of the important considerations in profiling is the decision on who performs the additions, deletions, or changes to the file. It could be either the biller or the consolidator/processor depending on the type of EBPP model being used. In the case of the direct biller model, the biller is responsible for profiling, whereas in the case of the consolidator model, the processor has this responsibility. However, in all cases, it is the processor that needs to authenticate the existence of the customer with the biller.
3. *Content Creation and Formatting* — This is the stage where the content of what the customers see is decided. The content can be developed in-house or the raw data can be sent directly to the consolidator who then works on the composition and format of the content. Content development is not merely the formatting of the raw data. It involves much more as the raw data itself may be residing in multiple systems in multiple formats. The task of formatting involves converting the different formats into Internet-friendly HTML and XML formats. Electronic bills also provide the opportunity of posting marketing messages that help in cross-selling and upselling the biller's products and services. These events are also managed through the process of content creation and formatting.
4. *Audit and Tracking* — This is a key part of the EBPP process and refers to the task of keeping an audit trail of all the activities that the data goes through until it reaches the customer in the form of an electronic bill. This becomes important in cases when any errors in the process need to be tracked. Customer habits and interactions can also be tracked as they access the invoice on the Web (through the use of cookies). However, who gets this data depends on the type of EBPP model being executed. In the direct biller model, the biller can make use of this information to enhance its customer care and marketing approach, whereas in the consolidator model, the consolidator/aggregator benefits from capturing this information. Audit of the events also enables a better customer care process as the billers/consolidators can better answer customer queries by knowing exactly what has occurred in the process until then.

5. *Notification* — Notification about the invoice can be sent to the customer in a variety of ways. The most common way has been e-mails. However, this method is very much limited in the graphic and dynamic content capabilities. A modified version of this method of notification is sending the customer the URL of the Web site that hosts the invoice. Customers can also access their bills by directly visiting the Web sites of the biller, consolidator, portal, or the aggregator depending on the model being used (e.g., banks like the Toronto Dominion Bank, that provide multiple services like loans and credit cards to their customers and host invoices for these services). Nowadays, with the boom in wireless services, customers can also choose to get notified about their bills via their mobile phones and wireless capable PDAs — the possibilities are immense.
6. *Bill Presentment* — This is the stage where the biller or the aggregator/consolidator presents a copy of the bill to the customer in an electronic format. This is the "moment of truth" for the customers as this activity can make or break their experience. The billers may choose to present the customers with a summary or complete details of the invoice. This activity also provides an opportunity to the billers and consolidators to interact with the customers, guide them to their areas of interest, capture their preferences and other important information (for future use), cross-sell and up-sell, and build loyalty by providing them with an enhanced or compelling experience. The service providers (billers) include dynamic features alongside the invoice like providing the customer any pending order status, details of past payments, and cumulative usage of services since the last bill (e.g., telecommunication service providers — ISPs, Mobile Service Providers, ILECs, etc. can post details of the usage till date).
7. *Customer Care* — It has been observed that more than 70 percent of the customer queries received by service providers concerns billing. This makes it imperative for the billers and consolidators to design the resolution of the most frequently posed inquiries into the presentment system itself to resolve the issues in an efficient and effective way. Some of the ways of achieving this are providing self-care features like querying present and past data, capability to chat live with a care representative, ability to log trouble tickets online to dispute charges, and posting quick and easy answers to FAQs that help the customer self-diagnose and trouble shoot the problems. Another important consideration to keep in mind is the fact that both the biller and the consolidator (if used) should be able to handle questions from the customers. This means that both the parties

should have access to all the shared information required to completely answer customer queries on any issue.

8. *Payment and Posting* — The last stage in the process of EBPP is payment processing and posting. Billers and consolidators need to provide the customer with multiple payment options that include direct debit, one-time payment, cheque payment, etc. Customers have a choice of making multiple payments to individual providers or one consolidated payment, for multiple services, to the consolidator/aggregator (depending on the model being used). Posting refers to the process of documenting information on when, how, and how much the customer paid for the services he used, and updating the accounts receivables systems. For this, the billers have to build the capability to interact well with the banks, remittance processors (lockboxes), and credit card payment processors in order to process payment and remittance information and facilitate reconciliation.

DRIVERS FOR EBPP

As is evident from the discussion of the EBPP process above, there are a number of drivers for implementing EBPP — cost reduction, customer relationship management, improved customer service, customer demand, and others. A survey of some of the leading billers by one of the Big 5 consulting firms found the following as the most important drivers for them to implement an EBPP solution:

- *Improved Customer Service* — By presenting the bills in an electronic format and providing access to varied levels of details about the bill, the billers make it easier for their customers to manage their own accounts and trouble shoot any problems. At the same time, billers can provide them with the ability to interact with the customer service representatives online and resolve their queries. And, of course, the customers have 24 × 7 access to their account details, 365 days of the year.
- *Greater Customer Loyalty and Retention* — Once a customer signs up for EBPP services, it makes it harder for them to leave, as it would mean changing ten to fifteen billing accounts. At the same time, with the control of the bills in their hands, the billers can track the customer movement across their Web site and use this information to learn about the customer preferences and demographics. Armed with these data, the billers can further enhance the customer experience by providing them services and offers of their choice and cross-sell/up-sell in the process. This directly measures up in the loyalty that customers show toward the service providers, as they are all looking for targeted offers rather than the “junk” ones that have become the norm over the years. Given the fact that a bill may be the only medium through which

some billers come in contact with the customers, an enhanced customer experience in viewing the bill would go a long way in retaining them as loyal customers.

- *Competitive Necessity* — "If you don't do it, somebody else will." This is the dictum that guides the billers toward implementing an EBPP solution. With so many players vying for this nascent market, the early mover advantage would make the difference between the winners and the losers in this race to keep the customers. Providing an enriching customer experience in this "moment of truth" is the competitive advantage that will make the billers gain market share.
- *Cost Reduction* — Implementation of EBPP can save money for both the billers and the customers. The areas in which these savings can be realized are billing costs, processing costs, paper costs, envelope costs, postage costs, exception handling costs, lockbox costs, and customer service costs (owing to reduced volume of call traffic into the call centers). With EBPP, the cost of producing a single bill can be brought down to as much as $0.38 as compared to the regular costs of anywhere between $0.65 and $1.50. McKinsey estimates the potential savings for the billers to be as high as $2 billion (annually) by the year 2002. That is an amount lucrative enough for many players to intensify the competition.
- *Customer Demand* — And last but definitely not the least, the customer demand itself is a huge driver for embracing EBPP. The convenience factor of being able to access the bills anywhere, at anytime, ability to access the minute details, and the comfort of paying the dues by the mere click of a button is incentive enough for the customers to shun the traditional method of paper transactions and embrace the new way of viewing bills. With technology advancing at an ever-increasing pace, customers have already started demanding access to these capabilities through their cellular phones, PDAs, and other mobile devices.

Though the above-mentioned factors make a strong business case for embracing EBPP, its ultimate success would be determined by the ease and perfection of its execution. Technology lies at the heart of execution.

TECHNOLOGY — THE BACKBONE OF EBPP

Given the fact that the raw billing data needs to go through so many stages and different entities (based on the EBPP model being executed), data integrity and communication among the different players in the process pose a major issue. Seldom do the legacy systems used by the billers or financial institutions have an open architecture. In order to make these disparate systems (based on different platforms) "talk" to each other, some kind of a standard/protocol needs to be followed. Open Financial Exchange (OFX) and IFX are two bodies that have defined these standards.

Open Financial Exchange (OFX)

OFX has defined two standards for EBPP: OFX-PRES for bill presentment and OFX-PAY for bill payment. These define the industry format and protocol for exchanging financial information (like bill presentment) and managing financial transactions (like bill payment) over the Internet. OFX allows communication between consolidators and billers in a standard way so that biller systems can "talk" to the consolidator systems and pass information between themselves.

OFX-PRES defines the standard for bill presentment by providing a communications channel for transfer of (1) sign-up information from the consolidator to the biller and (2) bill summary or details from the biller to the consolidator. This, however, does not imply that OFX also dictates the look of the bill. It just provides the common channel for exchange of information.

IFX

IFX is a forum of independent organizations that has developed business-level technical requirements to build an interoperable online bill presentment and payment solution — IFX 1.0. The participants in this development includes financial institutions, billers, and technology providers. The Data Interchange Standards Association (DISA) provides administrative support for the IFX forum. IFX builds on the industry experience of OFX and GOLD standards and covers the following areas:

- Funds transfers
- Consumer payments
- Business payments
- Bill presentment and payment
- Communication between banks, brokerages, insurance companies, merchants, payment and bill processors, financial advisors, and government agencies

Although standards exist, it does not imply that all solutions make use of them. In fact, some of the popular solutions like Transpoint's (Transpoint was acquired by CheckFree) Biller Integration System solution, which supports the consolidator model, do not make use of OFX but utilize the other popular technologies to enable the EBPP process.

Transpoint's Biller Integration System (BIS). BIS is a system that is implemented at the biller's site and is based on Microsoft's NT technology. The BIS environment sends batches of statement data and associated templates to the transpoint data centre (that serves as the consolidator). This environment uses MS Visual InterDev as a foundation for designing the templates. Also used is MS Visual Studio that uses COM-based interface as a language to write translators to the legacy system in order to integrate with third party products like customer support, reporting, etc.

Bill templates are designed using the Visual Basic Scripting edition that supports active server pages. Visual Source Safe is used to manage code and content documentation. The active server technology pulls data from the billing databases and converts it into HTML format so that customers can access the information over the Internet.

Transpoint also uses a Web-based integrated third party tool that enables the customers to initiate requests in electronic form. The tool, based on the request context, routes it to the appropriate party — biller in case of billing query, bank if a payment inquiry. At all times, the communication is protected by 128-bit SSL.

CONCLUSION

In conclusion, EBPP, with its numerous benefits, has already built a strong business case for its adoption. It promises benefits to all the participants in the process — from the customers to the billers. The market for these services is currently in the growth phase with consolidations and alliances already starting to take place (CheckFree, the biggest EBPP player in the United States, recently acquired a major rival transpoint to provide customers a more complete and integrated offering). The race for capturing the huge potential market is on. The winner would be the one who can build a solution based on an open platform or the industry standards, to integrate the disparate legacy systems of all the participants in the process and deliver an enhanced customer experience.

ABOUT THE AUTHOR

Rahul Kumar is a consultant with Deloitte Consulting. He specializes in business and technology solutions.

Chapter 12
Are Human Resource Departments Ready for E-HR?

Marie Karakanian

Watching the sweeping changes that the E-wave is throwing at the shores of all business disciplines one cannot help but wonder if HR is ready for E-business? The question for HR is whether to embrace and formally describe the meaning of E-HR, or remain a spectator for competitor actions and miss the boat dashing across the E-waves. Of course, these questions raise a slew of others, like whether HR has finally managed the time to reach the comfort zone in their HRMS systems or if HR is ready to declare the addition of its client/server applications to the ranks of legacy systems.

Is this technology wave truly important for HR, considering that despite all the hoopla surrounding the digital revolution shaping the new economy, HR continues the struggle to align to the ever-dynamic business strategies, to hunt rare talents, and to develop compensation strategies for high-tech and high-touch people? How will this E-wave impact the HR agenda? Will it facilitate HR's attainment of its objectives, or will it eradicate HR? Should HR jump on the E-procurement and supply chain bandwagon, clarify the enterprise integration points, and wave the business case flag once again and convince everyone of how the HR activity chain can add value and strengthen the E-business promise?

All valid questions that have no definite, clear-cut answers. However, as the promise of global network technology shapes the lives of all sizes of businesses today and questions the very existence of so many of those businesses, HR has no option but to go back to the drawing board and pull together the blueprint of its E-existence.

0-8493-1161-6/02/$0.00+$1.50

WHAT DOES E-BUSINESS MEAN?

E-business is the overall business strategy that redefines the old business models and uses digital media and network technology to optimize customer value delivery. It relies on Internet-based computing, which is the platform that supports the open flow of information between systems. It capitalizes on an existing technology backbone consisting of front-end and back-end enterprise business systems, and it makes effective use of component technology and interacts with customers via business portals established over the Internet. Technology is used in this case both as the actual cause and also driver of business strategy. It is used not only to develop the product or the service, but also to provide better choices to customers and enhanced delivery options.

E-business requires a complete replacement of the old business designs — new outsourcing and partnership alliances that not only reduce costs and speed solutions but also improve customer options. In one word, it is the re-invention of the old ways of doing business and aligning business strategies, partnerships, processes, applications, and people — truly fast and right.

HOW TO DESCRIBE E-HR

When mapping the above description of E-business to the world of human resources, one can say that E-HR is the overall HR strategy that lifts HR, shifts it from the HR department and isolated HR activities, and redistributes it to the organization and its trusted business partners old and new. E-HR ties and integrates HR activities to other corporate processes such as finance, supply chain, and customer service. Its premise is that HR is the owner of the strategy and, when required, it is the service broker as opposed to the provider.

What this philosophy demands is dedicated HR homework; executive participation; excellent appreciation of technology and utilization of technology, including a well-developed and integrated HRMS system; and wise use of network technologies and various communication channels such as Web, wireless, and perhaps kiosks. The HRMS system acts as the HR data and business rules backbone; it interfaces to the enterprise intranet and it connects to HR service suppliers and business partners via an extranet and links to the Internet via HR portals.

Thus, a potential critical dimension of the HR role becomes that of a services broker as opposed to a deliverer. Considering the proliferation of the variety of HR-related services within the marketplace — including business process, application services, candidate search, survey data, and a variety of function-specific expert service providers — the concept of HR service broker will probably soon turn into a larger reality than it is today.

WHY WOULD INTERNET-ENABLED HR TECHNOLOGY BE IMPORTANT FOR HR?

Here are some of the reasons why Internet-based technology is important for HR business:

- Provides cost-effective universal access to HR data to all authorized parties, including employees, managers, executives, HR service providers, relevant communities, corporate customers, and also the public-at-large
- Offers more cost-effective options for HR information systems management, especially for small- and mid-sized organizations via Application service providers (ASPs)
- Allows the capture of significant amounts of data truly at source, thus almost eliminating data collection turnaround time and enhancing data accuracy
- Reduces the uncomfortable distance between the HR department and its internal customers by optimum integration between the corporate processes, potentially covering purchasing to payables, new employee request to candidate identification, and compensation surveys to performance increases
- Enables the globalization of corporate HR information and its accessibility at significantly reduced costs

The next section focuses on some business scenarios to illustrate the potentials currently available for E-HR.

Scenario A

A typical HR business scenario can explain the use of all the above links. Say the Canadian headquarters of a multinational pharmaceutical company called B-Tec is looking for Biotechnologists and has outsourced this process to a specialized global service provider, BPO.Com. BPO.Com posts the vacancies on the Internet. Applicants e-mail their résumés to this service provider, which shortlists three candidates for interviews and e-mails their résumés to B-Tec. B-Tec stores these résumés on its HRMS system, and selects and authorizes two candidates to create their profiles using the HR portal on the company's Web site. BPO.Com arranges electronic tickets to the candidates from two different countries using its own travel business partner. At the completion of the interviews, B-Tec authorizes the selected candidate to access and accept a job offer waiting for her on its Web site. The candidate accepts the job and advises B-Tec of the candidate's preferred arrival date subject to an employment visa. BPO.Com processes the employment visa for the selected candidate and assists with accommodation arrangements.

B-Tec creates a temporary Employee ID upon candidate's acceptance of the offer and grants her access to the self-service portion of its Web site. The potential employee listens to a welcome Webcast by the president of the company, chats with her new manager, and familiarizes herself with the company's organizational chart. The candidate also selects her benefit preferences and registers for a company orientation session based on her arrival date. A B-Tec-assigned coach and the candidate chat on the Internet regardering her questions. E-mail from BPO.Com notifies her of her employment visa at the Canadian consulate.

Upon arrival to her country of employment, she presents herself to an HR consultant, who transfers her temporary employee record from the company's Web site to its employee database to complete the hiring process. The built-in business rules and edits ensure that the record has complete data.

A computer is allocated to the new employee right away. B-Tec's e-mail system advises the new employee of her temporary password to the Web site that has a to-do checklist waiting for her. One of the to-do items is her expense claim from BPO.Com. BPO.Com deposits owed moneys to the employee's bank account.

Sound like a fairy tale? Examine the next scenario.

Scenario B

A small U.S. branch (1000 employees) of a Japanese giant automaker H-Flyer finally decides to replace its existing Y2K-patched HRMS system. One of the requirements is delivery to Japan, on a biweekly basis, of a variety of HR information, including employee core data, head count, training information and budgets, compensation changes, and employee turnover and reasons. After a long analysis including multi-layer security administration requirements, the U.S. branch decides to use the services of an ASP as this turns out to be more cost-effective, less disruptive, and more quickly implemented. This approach allows all employees of the branch to update their personal and benefit information via self-serve; it allows the HR department to access/maintain HR data in traditional ways; and it also allows managers access to a variety of authorized employee data. All data maintenance and access is enabled over the Internet, considering that H-Flyer's HRMS database resides on the ASP's servers. The ASP provides the application implementation, data conversion, and maintenance and post-implementation support services, some of which it subcontracts to expert partners such as Speedo Consulting Services and the E-Infrastructure Gurus.

In addition to a variety of information outputs from the database, the ASP provides three files in the format required by the Japanese parent and posts it on a dedicated Web site accessible by the Japanese headquarters and the Canadian HR branch. These files are in a predefined format required by the Japanese headquarters, who in turn download them via its translation engine into Japanese and update its global HR data repository, which is maintained in Japanese.

The organizations in both scenarios A and B can be described as E-HR-enabled, wherein all employees and managers can interact directly with the supporting technologies, HR professionals can communicate directly with HR service suppliers and vendors, and the public-at-large can access company information to the extent the companies allow them. HR assumes accountability for the coordination and delivery of the right service at the right cost to the right party at the right time.

THE TRENDS SHAPING E-HR

Similar to E-business, there are certain market and business trends that are shaping the world of E-Human Resources. These trends are driven by users/clients, processes/services, organizational entities, and, of course, technology. Exhibit 1 describes these trends and provides contextual examples. However, risk and security management is perhaps more crucial to HR-related information than any other because it involves private and highly sensitive individual data. The disclosure and cross-border movement of HR data is a critical issue that needs to be managed very carefully based on country, organization-specific, as well as individual authorizations. Thus, data and multi-platform security aspects are perhaps the most serious factors that need to be taken into consideration during the formulation of an organization's E-HR strategy.

The value of the HR chain of events is derived from the unique combination of business strategy, people and knowledge assets, technology resources, and business processes within a given environment. This value should be recognizable not only by the employees of an organization, but also by its customers in the form of affordability, quality accessibility, and usability of its products and services.

CONCLUSION

One can safely say that Web technology is here to stay and its muscle is strengthening day-by-day and impacting business strategy in a way not experienced heretofore. Therefore, as HR becomes more of a refined business discipline, with processes more sophisticated than those of the traditional back office, it should optimize the use of all available technology to support and help accomplish the business goals. The HR department, however, needs to recognize some of the current limitations related to Web technology and its integration into the HRMS backbone. These challenges

Exhibit 1. Trends and Realities Impacting HR Business

Driver	Trend	Impacts	Examples/Application Related to HR
Users/clients	Service delivery speed	Employees Managers Executives Job candidates	Reimbursements Recruitment processes New system implementations
	Self-service and self-sufficiency	Employees Managers Executives Business partners	Change of personal information Performance management and feedback Career planning and training Pay information
	Integrated solutions	Employees Managers Executives	Performance-based increases and bonuses, including Web-enabled employee communication Web-enabled CBTs with updated competency profiles Operational budgeting and monitoring Strategic information management, trend analysis, performance score cards
Process	Integration Partnership/alliances Adaptation Consistency Convenience Enterprise extension	Employees Managers Public-at-large Business partners Service providers Competitors	E-recruiting, hiring, and orientating new employees using multimedia Sharing costs and benefits of same sources, such as job candidate databases and technology infrastructures Dedicated portals over company intranet targeted to various employee groups such as sales staff, executives, high-tech staff, etc. Use of repeatable, consistent, well-organized multi-channel communication based on specific questions related to benefits, pensions, policies, and procedures Accessing of multi-channel information for 24 hours from anywhere Company property management and tracking

New business models	Outsourcing: BPO, ASP Innovation	Employees (jobs) Business partners Technology vendors	Reviewing shortlist of candidates from an executive search partner's online system and scheduling interviews Accessing and maintaining own HR information over the Internet
Technology	HR systems bridging Enterprise application integration Multi-channel integration Internet business portals	Employees Managers Customers Public-at-large Technology vendors	Feeding overtime information from work management systems to the payroll system Linking HRMS to customer relationship management systems to improve customer service Linking HRMS to company supply chain system to integrate internal and external processes, such as supply of training material, receivables, and payables Accessing expert systems provided by external suppliers, such as compensation surveys and benchmark jobs
Globalization	Profitability Growth Sharing knowledge assets	Shareholders Governments Employees	Using the Internet to link to shared HR data across all operational countries where a company operates Using a combination of the intranet, network, and other technologies to extend operations across different countries Using existing technologies to develop consistent company image, culture, and processes across geographies where possible Accessibility to internal knowledge resources across the globe through networking and groupware technologies

include multi-platform security, the inability to perform extensive transaction processing, and concurrent Web site and database updates. There is no question about the complexity of a technology environment that operates on the principle of ubiquitous availability, yet needs multi-layered technologies to help slice and dice the unrestricted cyberspace. Similar to most E-business ventures, the security of private HR information is a top priority. Organizations looking seriously into Internet enabling of their HR business should evaluate the authentication, security, access rules, and audit trails related to service providers' networks, servers, and applications.

Web technology is currently not capable of capturing employee time, checking pay rates, and running transactions such as payroll processing all at the same time. However, it allows the capture of employee time data via self-serve that can be transported to a backbone HRMS system, and get validated and calculated into employee pay dollars, thanks to payroll engines — some of which still use good old COBOL programs. Data captured on a corporate intranet — such as benefit changes, overtime data, or organizational change — that is using Web technology needs to be fed somehow to the backbone HRMS database or the payroll system. HRMS vendors have started exploring the potential for concurrent updates of the Web sites as well as the HRMS database, regardless of which end is accessed for updates.

ABOUT THE AUTHOR

Marie Karakanian is a senior manager with Deloitte Consulting and specializes in human resources issues.

Chapter 13
Call Management and the Internet

John Fiske

Web sites can be thought of as rich-media interactive voice response (IVR) systems. Most people are familiar with IVRs — those devices that play a recorded menu of choices to telephone callers and prompt the callers to press keypad numbers to access desired information. If the caller does not get an option that helps to solve issues properly, the caller can "zero out" to reach a live person. However, the same button-pressing offers a much greater array of functions to the Internet user. Rather than accessing only audio information with the push of a button, the Internet user can access information-rich content with a click of the mouse. Although the basic Web site may have a significant amount of information, it cannot provide the degree of personalization that can be delivered by a live person. An Internet-enabled commerce center allows the online customer to receive a considerable degree of personalization by linking the customer to a live person.

How does this work? If E-shoppers want to speak to a live customer service or support agent, they simply click on a "Talk to Us" button displayed in their browser. With a standard multimedia PC, the online customer can connect with an agent to conduct an audio or video call. The technology that enables such interaction resides at the Internet commerce center, and more specifically, in the corporate call center. Outfitted with an Internet telephone switch and its accompanying automatic call distributor (ACD), the commerce center can handle incoming calls sent as packets of information transmitted via the Internet.

Combining Internet media with customer interaction provides customers with the benefit of optimal service in a convenient setting. In the old days, if a customer had a problem with something that was purchased, or wanted to find out about new products that the vendor was offering, he would have to go back to the store where the product was purchased in order to receive service. This method allowed for quality, personalized

0-8493-1161-6/02/$0.00+$1.50

service, but it took too much of the customer's time. Enter telephone-based customer support, which saved time but took away much of the personalization that comes from face-to-face contact with service representatives. Next came the Internet, which saved even more time for the customer but took away even more of the personalization as the service representative's voice was taken away. With the advent of live multimedia customer interaction over the Internet, the convenience of home shopping remains, while personalized service returns in the form of not only the service agent's voice but also the agent's face.

The benefits to the company are also enormous. With International Data Corporation predicting more than $250 billion in E-commerce transactions by the year 2002, companies have a tremendous opportunity to grow their businesses on the Internet. At the same time, companies face a great deal of risk as they expand their electronic businesses. Poor customer service or confusing Web sites drive both potential and current customers away and onto the Web sites of competitors. The risk becomes greater when one takes into account the ease with which customers can browse from store to store simply by clicking a mouse button. Multimedia customer interaction helps companies hold onto their customers and draw in new customers. Using live customer interaction, a company can build customer loyalty by putting a face on service agents and by offering service that is simply not possible with old-school technology. The end result is that the company can take advantage of the opportunities available from the E-commerce explosion.

The benefits do not stop with the bottom line, however. Contact center representatives can handle an Internet interaction as easily as they can handle a traditional telephone call because the company can retain the model of the old call center while increasing the effectiveness in the new Internet contact center. They can place a caller on hold, consult with a supervisor, transfer a call to another department, or conference in a third party. But those are just the standard call features. Multimedia customer interaction allows the agents to go further than the traditional call center does. While the customer is on hold, the agent can run multimedia streaming video to further explain the product or service of interest to the customer. The agent can also direct the customer around the site and even help complete any necessary forms through Web browser sharing, file transfers, and data collaboration. Best of all, this is accomplished with something most online customers already have sitting on their desks' standard multimedia PC. The result is that the agent can explain things easily, and the agent is better able to serve customers who would benefit greatly from a visual presentation.

As the boundaries and definitions of E-commerce continue to evolve, live Internet customer interaction will be instrumental in creating new ways for corporations to leverage their Web sites. For example, think of the impact multimedia customer interaction will have on transactions where face-to-face contact is crucial, such as online banking. Customers nationwide are connecting to their banks online, making account transactions, and even applying for loans and financing via the Web. However, to complete the high-value transactions — to actually get a loan once a customer is approved — most customers have to take a trip to the bank. With an Internet-enabled commerce center, loan officers and loan applicants can conduct face-to-face meetings from their homes and offices, thus allowing them to complete transactions in a timely and convenient manner.

This technology also overlaps to benefit customer support and help desk efforts in tandem with E-commerce strategy. By taking advantage of data collaboration available with the various Internet media types, an online customer support agent can help the caller fill out forms and view related information, remotely download software applications and upgrades directly to the caller's machine, and assist with installs and usability.

The end result is a more successful Web transaction for customers and businesses alike. The technology allows customers to feel more comfortable with online transactions as they are able to glean the information they need in realtime — the benefits of a face-to-face meeting without leaving the home or office. Best of all, the enabling technology that makes this possible boosts online sales, builds the customer loyalty that is imperative in an online environment and, in the end, helps companies to take advantage of a multi-billion-dollar industry.

ENTERPRISE DEPLOYMENT OF THE IP-ACD

Voice Is Data

Before long, voice will be considered another data type on the enterprise network. Whether on a PBX or a call center, IP voice is coming. Look for voice to occupy an increasingly large space on the data network.

The Internet is swiftly creating new ways to service customers, providing unprecedented access to rich, predefined media content. Advancing technologies are enabling live interaction between the customer and the Internet-based call center.

The Automatic Call Distributor

Call centers use an automatic call distributor (ACD) to maintain high-productivity environments for the agents. The ACD in any call center matches customer needs to agent capability (service, sales, etc.).

Today's call center evolved from an audio-only facility receiving and routing voice calls, into one able to handle a variety of media types, such as e-mail and fax. The call center must respect the demands of the caller, including the timing and media type used for the connection.

Interactive Voice Response

In order to increase the efficiency of the audio-only call center, interactive voice response (IVR), computer telephony integration (CTI), and other technologies have emerged. IVR systems free call center agents from such repetitive tasks as entering account information or providing responses to frequently asked questions.

Voice on the Internet

Telephony communication over the Internet is defined by H.323, the International Telecommunications Union (ITU) specification for audio, video, and data telephony over IP. It is a packet-oriented protocol. An integral component of H.323 is T.120, the ITU standard for audiographics (otherwise known as data collaboration). The H.323 standard was approved and implemented several years ago. Endpoints that are H.323-enabled have been widely distributed. For example, the NetMeeting product from Microsoft has been shipped with every Microsoft Windows 95 software license since early 1997. The H.323 standards are important because they facilitate peer-to-peer communications and peer-server-peer communications in a nonproprietary fashion, enabling audio (and optionally video and data collaboration) communications, using the Internet.

The Web-Enabled Call Center

The Web gives people far greater access to information than any previous medium or resource. Information is presented in both an audio and graphic fashion, and customers make accurate choices and can complete transactions. Users of well-designed IVR phone systems know they can almost always "press zero to reach the operator" when they need additional information, or if their transaction does not follow the preconceived and preprogrammed call flow. E-business customers are beginning to demand this same ability, either by clicking a Web page button or by calling on Internet phones. This is a major shift.

Connection Methods

There are three general techniques to connect caller and agent together for audio, audio/data, and audio/video/data calls using the Internet. The techniques are *call-back*, *call-through*, and *switched connection*. Call-back places an audio call back to the customer. Call-through places a point-to-point H.323

call, calling around any switch. An H.323 call in the switched connection configuration calls through the H.323 switch.

Call-Back. By clicking the CALL ME BACK button, the browser sends a HyperText Transfer Protocol (HTTP) message to the Web server, which typically reacts by directing the browser to point to a URL containing a form requesting the customer's name and telephone number, and asking, "What's the best time to call?" Next, the Web server creates an e-mail to send to an agent, or invokes an outbound dialer, generating an outbound call from the call center's ACD. From this point, the call-back follows one of two scenarios.

If the customer has two phone lines (one for an ISP connection plus a line for regular audio calls that can be reached by direct dial), the call center may call on the second line, while the customer is still on the Web site. Some companies have developed technologies that enable the customer and agent to view the same Web page, even permitting the agent to change the Web page the caller is viewing. This is called browser sharing.

To share browsers, an applet must be downloaded onto the customer's PC. Browser-sharing technologies typically perform a bit image copy. Periodically (once a second), the Web server application that provides the browser sharing function "reads" the image on the agent's screen (or the customer's screen) and then paints the image on the screen of the other party.

If the customer only has one connection, he or she disconnects from the ISP and stays off the phone, waiting for the call center to make an audio call-back to that line. There is no browser sharing in this model.

Drawbacks. The call-back solution is less than perfect. In the better call-back model, the customer would have two phone lines. The main problem is that people have become conditioned to expect immediate connections to agents, with solutions delivered during a single call. Hanging up and waiting for a specific call does not provide the level of service that customers expect. The customer may view Web-initiated telephone calls as new technology and, for a short while, be willing to tolerate the scheduled or immediate call-back.

The privacy issue is also a concern. Customers who are business employees sitting behind a PBX may not want to give out their phone numbers. There is a perceived anonymity about the Web that customers enjoy. From the call center's perspective, call-back is a quick fix for Internet-enabling the call center with little incremental investment, beyond that of a blended (inbound and outbound) call center. For the inbound-only call center, call-back requires the development of costly outbound calling practices, equipment, and management skills.

Browser-sharing solutions are fine for solving certain problems. For example, if the customer is not able to find the correct Web page for the information he or she required, browser sharing works well. The agent can "move" the customer's browser to a different page.

Some implementations allow the agent to fill in some browser-based forms and allow the customer to see those forms. However, browser sharing is typically accomplished through proprietary bit-copying techniques. Because browser sharing is not realtime, interactive collaboration, simultaneous changes to the browsers of the customer and the agent result in one person losing their changes — only one browser is "in charge," and the other follows. Because this technology relies on bit copying, if the page being shared extends beyond the size of the browser window, only part of the browser is viewed by the customer (e.g., the caller and the agent may not be seeing the same information), causing confusion. If the agent scrolls the browser to see the bottom of the window, the customer does not see the scrolling effect. To avoid this problem, the Web master must restrict the amount of information to be viewed on any page. However, browser sharing does not support the capabilities of H.323/T.120 standards-based data collaboration. (Data collaboration includes: application sharing, remote application control, file transfer, text chat, and whiteboarding — all in a realtime, interactive fashion. An example of data collaboration is an agent sharing a chart set or a spreadsheet with the customer in order to make a presentation.)

Call-Through. From a call flow perspective, "call-around" would be a more accurate description of this type of connection because the customer directly calls an appropriate agent. However, the call does not go through a switch — it avoids the switch, choosing to go around it.

The customer, interacting with the Web server, clicks a "connect me" button and is connected to an agent. They then communicate using products such as NetMeeting that enable audio, video, and data collaboration.

How It Works. When a customer is browsing a Web site, the browser exchanges HTTP messages with the Web server. When the customer clicks the CONNECT ME button, the Web server executes a script, which sends a CGI message to an ACD application. The ACD application monitors agent availability and, at some point in time, detects an available agent. When the agent becomes available, the ACD application provides the agent's IP address to the Web server, which in turn provides the agent's IP address to a preinstalled applet on the customer's PC. The applet starts the H.323 phone, instructing the H.323 phone to call the agent's IP address.

The connection type is a function of the customer and the agent capabilities and is determined at call setup. H.323 call setup includes "capabilities exchange," a dialogue between the endpoints used to establish the call.

Capabilities information includes protocol identifiers (G.723.1 audio encoding), video capabilities (none, one-way, two-way), and data session establishment. Once this point-to-point call is established, customers and agents can speak to each other and potentially see each other. Most importantly, data collaboration can begin. Data collaboration in this context includes whiteboarding, file transfer, text chat, and remote application control.

Remote application control enables the agent to perform actions on the customer's PC. An example might be an agent performing remote diagnostics on the customer's PC in order to solve a problem with a software driver. Using file transfer, a software patch can be downloaded and installed. Of course, the customer must agree to these actions and can easily be provided with the ability to click on a "panic button" to stop these activities. Also, when the call is established, a standard browser-sharing application can be used to push and share Web pages.

In order to make a connection from a caller to an agent with the call-around architecture, the IP address of the agent must be publicly accessible. If the agent's IP address is not publicly visible, the point-to-point call cannot be placed, since one function of the firewall is to re-map real IP addresses into virtual IP addresses. The function provided by the ACD is to broker the availability of the agents and then tell the customer's application when to make the call. Because the IP addresses of the agents are public, anyone — including hackers — can attempt to connect or perform any TCP/IP-based application, at any time.

The customer call experience in the call-around model matches the audio experience that they understand. The customer asks for a connection and a connection is made.

Drawbacks. From the call center's perspective, there is a rather large problem: security. Agents are directly accessible from the Internet; otherwise, the point-to-point H.323/T.120 call could not be connected. This means the call center agent is not protected by a firewall.

Call center technology managers need to ask questions about call-around. What risk is there by exposing agents to the Internet? If my agents are CTI-enabled, how do they reach the CTI server? Do my agents have to sit on multiple LAN segments? How is that accomplished? Initially, Internet-based, multimedia applications will be rolled out to informal workgroups, where security may not be as much of an issue. However, they will eventually integrate with the existing call center operations, requiring airtight security.

Switched Connection. The switched-connection model is essentially that of an audio ACD. In the audio model, the pilot directory number (DN) of the group is published, not the DNs of the individual agents. Through various directory services methods (directory assistance, telephone book, advertising), the call center's pilot DN becomes known. In the switched-connection model, the IP address of the ACD group — not the address of the agents — is published and publicly accessible.

How It Works. The customer may start by browsing the Web site or going through an Internet directory service to make a direct Internet telephone call. When the customer initiates the call, the ACD with integrated H.323/T.120 firewall proxy is called. H.323 call set-up occurs between the customer's H.323 phone and the ACD. When an agent is available, the call is connected between the customer and the agent.

All of the H.323 (including T.120) packets go through the ACD. The agent is provided call context information upon call arrival. Call context information might include the customer's name, currently viewed URL, subscriber service level, etc. If a browser collaboration session is valuable for this call, the agent can use the call context information to synchronize with the customer, through the Web server.

As each call is connected through the ACD, packets pass through an integrated H.323/T.120 firewall proxy, ensuring the security of the connection. During transmission through the firewall, each packet address is remapped to route it to the appropriate agent. This process allows agents to be hidden from direct public access, easing LAN topology management issues and resolving security concerns.

Since all of the H.323/T.120 protocol packets pass through it, the switched-connection model can provide standard audio call center agent and supervisor features. It is mandatory to process these protocols to implement basic call center features such as transfer and conference.

From the customer's perspective, the customer places a call. Respecting the media choices of the customer, the call center accepts the call and connects to an agent. From the call center's perspective, inbound agents are still inbound agents. The switched connection performs as a standalone Internet ACD, providing basic call center features and functions, including call routing, agent and supervisor features, and management information.

Questions to Consider

In building out a call center to handle IP traffic, one may want to examine the following points.

- Is there risk by exposing the agent to the Internet?
- If the agent is CTI enabled, how does he or she reach the CTI server?
- Does the agent have to sit on multiple LAN segments? How is that accomplished?

The findings will help shape the call center to a particular organization's needs.

SUMMARY

Because call centers are mission critical, changes to them will be gradual and iterative. Hybrid call centers may become the standard. Traditional calls coming through PSTN will continue to be handled by agents, just as they are today. Calls coming in through the Internet will come to a Web server through the router firewall, using its own software smarts in routing the call based on information received from the Web-URL stack. This information will help the call center drive the call to an agent equipped for multimedia customer support.

These two call centers can coexist as call traffic gradually shifts from black phone to Internet. The agent pool will eventually shift from a traditional to Internet-enabled group.

The IP ACD promises to do a much better job at call routing — getting the customer to a call center agent and a better match, resulting in customer satisfaction in less time.

Note

This article was derived from white papers published by PakNetX Corp., Salem, NH. www.paknetx.com.

ABOUT THE AUTHOR

John Fiske is an independent writer specializing in enterprise networking and management technology. He lives in Prides Crossing, MA, and may be reached at fiske@tiac.net.

Section III

Wireless and Mobile Business Solutions

With Internet technology becoming part of the Information Technology (IT) mainstream, the next "hot" areas are expected to be wireless and mobile solutions. These two do not always go together. For example, a desktop computer can be wirelessly connected to an organization's network. This makes the user wireless, but not very mobile — unless he wants to carry a large monitor and CPU box with him. By and far, the most exciting wireless solutions are mobile. These include wireless laptops and personal device assistants (PDAs). The objective of all these devices is to allow users to access their technology from any location at any time.

This section discusses wireless solutions from several perspectives, including defining and constructing wireless solutions using a variety of popular standards. The following topics are examined in this section:

"Living in a Wireless World: Wireless Technology 101" (Chapter 14) examines wireless technology and related issues, with a focus on compatibility issues pertaining to the use of various types of devices, costs of using wireless technology being prohibitive sometimes, lack of suitable content for wireless computing, and security challenges.

"Wireless: A Business and Technology Perspective" (Chapter 15) explores the fundamentals of the wireless Internet and the drivers propelling its current and future growth. The following topics are reviewed: the key attributes of mobility and how they create business value, the impact of wireless technologies on business processes, key industry sectors and personal productivity, enabling technologies in wireless, and the future of the mobile Internet.

"Building a Wireless Web Solution: Tools and Justification for Building Wireless Web Solutions" (Chapter 16) focuses on wireless Web development. This includes a discussion of the user interface (UI) restrictions due to low bandwidth and the small screen formats that are available. HGML, ActiveX Data Objects, Active Server Pages, XML/XSL, WAP, and WML are also reviewed in this chapter.

"Putting Data in the Palm of Your Hand" (Chapter 17) examines how organizations can gain an edge over the competition with wireless technology. This is a technical chapter that tries to answer questions such as: What devices to support? How fast? What kind of data source? How much data? How many users? Perhaps more so than with other IT choices, the success of implementing wireless data access is dependent on thoughtfully planned objectives and well-defined expectations. This is, in part, due to the inherent risks and high costs required for any sort of large-scale deployment of wireless data and devices.

"Programming Wireless Applications" (Chapter 18) examines methods of integrating the capabilities of the Internet and cellular telecommunications. The Wireless Application Protocol (WAP) and the Wireless Markup Language (WML) are reviewed in this chapter. Some basic approaches for development programs with WML are also included.

Two chapters, "Wireless Communications for the Data Center: Part I" (Chapter 19) and "Wireless Communications for the Data Center: Part II" (Chapter 20), examine the state of wireless communications in data centers. The chapters discuss fixed, handheld, and microprocessor-controlled radio frequency transmitter/receiver units. Hardware configuration, including LAN configuration, are also discussed.

"Wireless Internet Security" (Chapter 21) explains the significant security issues facing the wireless Internet security industry. This chapter could have been placed under the security section; however, it also focuses on portable Internet devices and standards that can be leveraged in a more traditional development or architectural context.

Chapter 14
Living in a Wireless World: Wireless Technology 101

Michelle Cook

Like it or not, we are living in a wireless world of cellular phones, personal digital assistants (PDAs), pagers and messaging services, remote e-mail, computers, and Blackberries. This wire-free revolution happened so quickly that few people even noticed it. Yet, it is upon us. For anyone who works or plays in the wired world, the benefits of being free of wires are obvious. Wires are not the most attractive by-product of the digital age nor are they user-friendly or time-conscious. These unsightly creatures require lengthy effort to move (any mobile professional using a notebook computer with a half dozen wires connected to it surely understands) and there is always the risk of being tripped up.

Using wireless technology, on the other hand, has its obvious benefits: checking e-mail from airports, firing off reports from a vehicle while traveling, checking banking and investment news in some spare time that might otherwise be wasted, and chatting with your colleague while being stuck in traffic. These rewards of wireless technology are a testament to its growth in popularity. Most business people own at least one wireless device, while many own several. And its popularity is steadily increasing. According to a 2000 survey by Forrester Research of 9000 North Americans, the growth rate for adoption of wireless technology is likely to be as fast as that experienced for Web access via personal computers (PCs). Other estimates suggest that wireless Internet access will surpass PC access to the Net within a few short years.

With the ever-increasing popularity of wireless technology, a clear understanding of the technology and the issues that accompany it is critical. Plus, as the technology "matures" and becomes more widespread, the

0-8493-1161-6/02/$0.00+$1.50

challenges will change. There are many issues, but the main ones can be classified into:

- Compatibility issues pertaining to the use of various types of devices;
- Costs of using wireless technology are sometimes prohibitive;
- Lack of suitable content for wireless computing;
- Security challenges are prevalent; and
- Speed is relatively slow.

An understanding of these issues and their role in the business world may prevent a few headaches and, quite possibly, a panic attack or two.

COMPATIBILITY ISSUES PERTAINING TO THE USE OF VARIOUS TYPES OF DEVICES

With the advent of personal digital assistants, Blackberries, pagers, cell phones, and mobile notebook computers, the issue of compatibility has been on the rise. These devices use different technologies and networks so there are common compatibility issues to consider when implementing different devices into your work or personal life. While many representatives of the companies selling the devices claim that they will be free of any possible compatibility problems, in reality the devices rarely have seamless communication.

There are two main types of networks: circuit-switched and packet-switched, or circuit and packet networks as they are typically called. Cellular phones can "talk" over circuit-switched networks (as opposed to packet-switched networks) like CDMA, GSM, or TDMA very effectively. Although PDAs can use circuit-switched networks effectively, they rarely do. Packet-switched networks enable the PDAs and other portable devices of the world to communicate rather effectively as well. Circuit networks tend to be voice-centric while packet networks are usually data-centric.

The companies whose devices use either type of network have a vested interest in claiming that the devices are compatible, but that has yet to be proven. Some experts predict that over the next couple of years circuit-switching networks will migrate to packet-switched networks to better enable voice and data to transmit over the same network at the same time.

In the interim, the compatibility issue looms large, and while there have been attempts to overcome the problems and some technologies are faring rather well, other technologies have had limited success.

Consider, for example, Edge, which is a type of packet-switching technology. It allows the TDMA networks (used largely by cellular phones) to provide true packet-switching data capabilities.

On the other side of the picture, there is a form of layering technology that many people criticize for compatibility problems. In these types of cases, a form of packet-switching technology is merely layered over a circuit network and does not produce true packet-switching network capabilities. One example of this technology is known as GPRS.

Wireless Application Protocol (WAP) is a platform that standardizes the delivery of content between wireless devices and is currently in widespread use. However, experts have discounted it as a possible future standard because it has difficulty handling the content- and graphics-laden Internet.

COSTS OF USING WIRELESS TECHNOLOGY ARE SOMETIMES PROHIBITIVE

In North America, the cost of using wireless technology on a widespread level throughout organizations has been prohibitive. There are significant up-front costs to purchase new technology or replace existing technology. As well, there are the ongoing usage fees that can add up quite quickly for cash-strapped small businesses or employee-laden large corporations. In other parts of the world, cost is not necessarily an issue. For example, in parts of Asia and Europe, wireless technology is often more cost-effective than its wired counterparts. This is particularly true of cellular phones and their usage fees. They are less expensive than using traditional phones and land lines. Wireless technology has typically been around longer in these parts of the world and therefore the market for it is more mature. It may take a while to reach that point in North America, but as more players constantly seem to be jumping on the wireless bandwagon, competitiveness should force market prices down.

Determining the costs involved in using wireless technology may not be as straightforward as determining the costs for other technology. For example, most network managers know exactly how many servers and terminals will be needed to upgrade computer systems within an organization. However, using wireless technology is less clear-cut. It is critical to determine exactly the processes with which the wireless technology is to assist. If its main purpose is to help a business stay in touch with its clients remotely, then cellular phones will do the job nicely. If the main purpose is paging, a pager is best suited for this purpose. While these points may sound obvious, with all the multifunction devices popping up, most of which have overlapping functions, the picture gets muddled. For example, many cell phones now provide the capacity for sending and receiving e-mail remotely. Yet, if you have ever tried using a telephone to type e-mail messages, you will clearly understand that this device may not meet an organization's needs in this capacity. So, comparing prices for the technology also starts to get murky. It is difficult to compare pricing for PDAs and remote e-mailing services to a cell phone that provides remote e-mail.

LACK OF SUITABLE CONTENT FOR WIRELESS COMPUTING

The content issue with wireless technology really has more to do with the providers of the content itself: the providers of Web content or the individuals sending e-mail. Because the Web is so graphics-intensive, it tends to preclude many wireless devices from accessing it with any amount of sophistication. Even using wireless devices for e-mail may pose some difficulties when trying to view a large message on a single-lined cell phone or pager. Your eye doctor will agree.

Wireless Internet content is still not easy to find. Again, it is more readily available in parts of Europe and Asia where greater use of wireless technology has demanded it. When considering adding wireless devices to an organization, consider whether or not the content you will need to access is truly available at this stage of the wireless game. In other words, the issue may transcend wireless devices themselves to become a lower-tech concern: Have you selected the right content providers? For example, if you require remote banking and investment updates, then the content issue may be whether you selected the right bank or investment house.

Some makers of wireless PDAs are establishing their own wireless networks for their devices and have contracted major Web sites to provide a Web “clipping” service. This truncated form of browsing will be far more suitable to the current limitations of remote computing.

This may be a trend that starts to take place with other makers of wireless devices because the Web is becoming increasingly graphics- and animation-heavy. So there will be a greater demand for truncated Web services (at least until the wireless devices mature a fair bit). This is an area where wireless computers (like notebooks) have a bit of an edge simply due to the larger screen size and memory capacity, although there is a trade-off, because wireless notebooks, unlike PDAs, cannot handle longer distance remote computing. They still need to remain a relatively short distance away from their hub (a wireless computer’s equivalent of a server).

SECURITY CHALLENGES ARE PREVALENT

While people in Finland may be quite comfortable making purchases from vending machines using their cellular phones, most people still question the level of security while using various types of wireless devices, particularly while transmitting confidential data or accessing financial information over the Web.

According to a recent study by the Irish company, Covado Ltd., the average mobile data transaction is vulnerable at two points. Covado suggests that most WAP phones do not really support the wireless transmission layer security (WTLS) standard incorporated into the WAP standard. In other words, data in transmission from cellular phones to the telecommunications company can be intercepted.

The other problem, according to Covado, is that even where the WTLS standard is implemented, sensitive data is at risk in the mobile company's own system until it arrives at the server.

Because there is a lack of an "end-to-end" security solution in the marketplace, the security of transferring data using PDAs and wireless devices exists.

In the short term it is essential to carefully select robust hardware and a network that offers the maximum security possible at this stage of the technology. While this cannot guarantee security of all your transmissions, you will at least reduce the chances of data interception.

SPEED IS RELATIVELY SLOW

The current speed of most wireless devices is approximately one sixth the speed of a 56K modem. Thus, remote computing is not currently known for its speed and, of course, this presents some challenges in the business world. When getting the right information as quickly as possible is critical to success in today's business environment, wireless computing may not be keeping pace, unless the options are not getting the information or getting it slowly.

In all fairness, it is not always the remote devices causing the slowdown. In fact, when it comes to accessing data over the Internet, usually the Internet itself presents the bottleneck. So, in some cases, even if the devices get faster, access to data over the Internet will still present problems.

There is still a lack of stability in the technology because wireless technology is similar to radio technology: it is affected by weather conditions, humidity, and electromagnetic interruptions. In the same way that the radio will occasionally have static and the signal may even be lost during storms, wireless technology can suffer from the same type of interruptions.

While there is no doubt that the wireless industry is actively working to overcome compatibility issues between the devices — sometimes prohibitive pricing, suitable content, security challenges, and slow speeds — these are not likely to be overnight changes, particularly with such a complex industry. And these issues will probably change over time as the whole industry evolves and matures.

THE WIRELESS LEXICON

Are you having trouble keeping up with the jargon of the wireless world? This glossary provides a brief overview of some of the most commonly used terms.

1G, 2G, 2.5G, or 3G — The "G" stands for generation and refers to maturity of the technology according to the industry. 1G represents the first generation of cellular phones, for example, while 3G is the future of wireless technology.

802.11b — The LAN technology that enables wireless computers and devices to be "networked" and communicate with one another over short distances.

Bluetooth — The short-range, wireless, high-speed data connection technology allowing different types of wireless devices to "talk" to one another using a common language. Currently, it is typically limited to a 10-meter range.

Circuit-switched networks (or circuit networks) — The network that enables wireless devices to be used. It is mainly a voice network and therefore largely supports cell phones but can handle PDAs as well. Its counterpart is packet-switched networks.

Edge — A type of packet-switching technology that enables true packet-switch data capabilities.

Packet-switched networks (or packet networks as they are typically called) — Another type of network that enables the use of wireless devices. Currently, packet-switched networks support PDAs and allow the "always on" messaging and e-mail services offered by some paging and PDA companies.

SMS — "Short messaging service," which means the capacity to send AND receive e-mail messages.

T-9 — The latest technology that enables cell phones to send messages or e-mails with less effort. Instead of having to continuously hit the same key to get the last letter listed on it, T-9 uses probability logarithms to anticipate what you are trying to type, thereby saving you keystrokes.

WAP (wireless application protocol) — A platform that standardizes the delivery of content over wireless devices.

Wireless notebooks — The most recent contenders to join the world of wireless computing. They are notebook computers without all the wires that typically use Bluetooth technology.

ABOUT THE AUTHOR

Michelle Cook is a founding partner of Global Trade Solutions, a management consulting firm specializing in competitive intelligence and market research, particularly in technology industries. She is the author of *Competitive Intelligence*: *Create an Intelligent Organization and Compete to Win,* currently distributed worldwide. Ms. Cook is also a recipient of the Forty Under Forty award as one of the top business people under 40 years old in Canada's Capital Region. She is a regular contributor to *Database Management.*

Chapter 15
Wireless: A Business and Technology Perspective

Mark Barnhart
Ritu Joshi
Dean Peasley
Jeffrey Staw

The intent of this chapter is to educate the reader on the fundamentals of the wireless Internet and the drivers propelling its current and future growth. The chapter is written with the assumption that the reader has a base knowledge of current technologies.

For the purpose of this discussion, any Internet device that has real-time or asynchronous wireless access to the Internet is considered to be a part of the "Wireless Internet."

One of the key benefits of wireless technology is mobility. Hence, the terms wireless and mobile are used interchangeably. This study is more focused on the benefits of the mobile Internet, therefore certain static wireless applications like wireless networking have not been included in this discussion.

The chapter addresses the following topics:

1. The key attributes of mobility and how they create business value
2. The impact of wireless technologies on business processes, key industry sectors, and personal productivity
3. Enabling technologies in wireless
4. The future of mobile Internet

CREATING VALUE THROUGH THE MOBILE INTERNET

Mobility in the wireless context implies more than just the convenience of moving devices without the constraints of wires. It goes beyond connectivity

0-8493-1161-6/02/$0.00+$1.50

Exhibit 1. Creating Value through the Mobile Internet

Capabilities	Implications
Real-time information	Instant access to current information Real-time reach to a person/device
Accessibility	Unrestricted access To different data types To different applications Through a multitude of devices and interfaces Across geographical areas
Location specificity and tracking	Location information and context Remote monitoring and controlling

to actually make data available in *real-time*, make applications accessible through the wireless devices, and provide location-specific information that allows dynamic management of content and applications based on the user's location and immediate environment. The combinations of these factors, as shown in Exhibit 1, create opportunities and challenges that drive many of the new developments in wireless technologies and business applications.

BUSINESS IMPACT

Technology has powered the economic expansion during the last decade. Computers and the Internet have galvanized businesses and created new standards in efficiency and expectations. Technology has heightened the competitive environment and compressed time-lines for change. Considering the enormous worldwide investment in wireless infrastructure, it is anticipated that wireless technologies will produce an economic impact comparable to the previous computer and Internet revolutions. Wireless technologies provide new opportunities to boost productivity at a time when businesses have had a taste for both the promise and the pitfalls of new technologies. It is also a time when most leading companies have already adopted technology as an integral part of business strategy. In the current environment, emerging wireless technologies will be judged more objectively and adopted more on the basis of their impact on business rather than for their investor appeal. The major areas of opportunity and impact of wireless technologies are based on the impact of mobility in improving personal productivity, improving business efficiency, creating new businesses through wireless services in different verticals and horizontals, and in providing new opportunities in wireless networking and service providers.

Each of these areas represents a major opportunity and presents significant revenue opportunities and creates competitive advantages.

Improving Personal Productivity

Wireless promises to improve personal productivity through providing communication and collaboration access services like SMS (short messaging service), voice mail, e-mail, video calls, unified messaging, and groupware messaging. At the same time information access services such as corporate databases, personal files, and external information services (e.g., news, industry information, market data, and virtual assistance services like your calendar, mobile ID, or virtual secretary) will also become available through wireless. These individual productivity gains will have greatest impact to businesses in the knowledge and service economy where real-time accessibility to information and transaction processing creates a more efficient work force and a distinctive competitive advantage.

Improving Business Efficiencies

Wireless is an enabling technology that will create value for businesses through productivity gains translating into cost savings. It can also generate revenues by achieving competitive edge through enhanced customer relationships.

Businesses have become increasingly dependent on communication technology, computerized applications, information storage, corporate Intranets, and the Internet. However, workers are cut off from these resources the minute they leave their desks or offices. Imagine a salesperson who could check inventory while in front of the customer, or a mobile stock trader who could reallocate a portfolio while waiting for a flight at the airport. Creating wireless information access and transaction processing will be vital in many industries where a well-informed work force can supply a distinctive competitive advantage.

Revolutionary new services will be driven by unique benefits of wireless, e.g., location information. Location information is unique to the present-day wireless networks that can pinpoint the physical location of the mobile subscriber and communicate it to services on the network. This means that your mobile phone now knows its geographical location and can serve content and applications most relevant to that location. For instance, your phone can now provide you information about restaurants in your immediate vicinity, flight information for the nearest airport, and also connect you with services in the area. Right information at the right location is the next logical productivity gain, which is unique to wireless.

Evolutionary services are based on extending existing wired services through integration of wireless to multi-access solutions, e.g., customer relationship management.

Exhibit 2. Impact on Business Processes

	Delivered Wireless Benefit by Business Function						
Function	**Location-Specific Information**	**Accessibility**	**Real-Time Information**	**Remote Monitoring**	**Remote Adjustment**	**Potential for Wireless Services**	**Killer Application/Services**
Purchasing	P	F	G	F	P	F	Inventory management
R&D	P	F	P	P	P	P	Access to expert systems
Operations	F	P	F	G	G	F	Wireless process monitoring
Logistics	E	E	VG	G	P	VG	Fleet management, location tracking
Sales and marketing	G	VG	G	P	P	G	Field sales automation, order-entry
After-sales service	E	E	VG	VG	G	VG	Field service staff dispatch, parts and supplies ordering

Note: E = excellent, VG = very good, G = good, F = fair, P = poor.

Impact of Wireless on Business Processes and Industry Sectors

While wireless will produce overall productivity gains across business functions, our study of the relative importance of the various attributes of mobility (location information, real-time information, mobile reach and accessibility, remote monitoring, and remote adjustment) in different business functions shows that different attributes have varying degrees of importance. For instance, location information is most importance in logistics, whereas it is irrelevant in purchase or research and development. The relative importance of these five general benefits of wireless across common business functions is mapped in Exhibit 2 to reveal important focus areas in wireless applications that aim to optimize business processes. A similar analysis across industry sectors also reveals varying importance of these attributes of wireless in different industry segments.

Creating New Businesses in Different Verticals and Horizontals

Since wireless presents opportunities to create value in multiple vertical and horizontal segments, early adopters have an opportunity to establish their competencies in these emerging markets and leverage them to create

Exhibit 3. Impact on Key Industry Sectors

	Delivered Wireless Benefit						
Industry	**Location-Specific Information**	**Accessibility**	**Real-Time Information**	**Remote Monitoring**	**Remote Adjustment**	**Potential for Wireless Services**	**Killer Application/Services**
Utilities	F	F	G	E	VG	VG	Meter readings and problem alerts
Security	G	E	G	E	P	VG	Wireless surveillance, access control
Transportation	E	E	F	F	P	G	Fleet management systems
Healthcare	F	G	G	VG	P	G	Patient monitoring, remote consultation
Manufacturing	F	G	F	G	F	F	Wireless production monitoring
Financial services	P	F	E	P	P	F	Wireless trading and financial database access information push
Retail	F	G	E	G	P	F	Wireless POS systems, inventory management

Note: E = excellent, VG = very good, G = good, F = fair, P = poor.

new businesses based on wireless services. The key is to identify opportunities complementing the core competencies of the company and partnering with other companies to rapidly capture these emerging opportunities. Some of the product and service opportunities (killer applications) in different industry sectors are presented in Exhibit 3.

Opportunities for Wireless Network and Service Providers

Wireless environments create new opportunities in providing software, hardware, and services. However, wireless infrastructure requires heavy investment and large up-front risk. Many service providers have already made heavy investments in 3G wireless infrastructures. Most of Europe has already auctioned off the 3G spectrum at huge cost to the carriers, which plan to roll out wireless services based on higher network speeds as high as 2 Mbps. But it is not certain if consumers will pay for these higher speeds. If the Japanese i-mode experience is any guide, consumers may be happy enough with simple services and unwilling to pay much more for the promise of higher data rates. Such consumer content is a frightening prospect for the carriers who have just paid so much in the hopes of serving greater demand. Further, since a large part of the value created by wireless is through its integration with the Internet and since the wired Internet is to a large extent a minimal cost, it is unlikely that additional usage-based wireless charges will be acceptable to the buyers of wireless services.

This indicates further consolidation in the wireless service providers. Many wireless providers carry enormous debts — and the interest payments are crippling for all but the largest players. Smaller companies will have trouble raising the cash to remain competitive, leaving firms like Vodafone, Orange, and AT&T wireless to dominate the market.

As a result of this pressure network providers have also been moving downstream and attempting to derive a cut of the transaction and service and content revenues. This has created an uneasy feeling among application providers who potentially face new competition from large wireless service providers. Wireless application providers will have to give importance to managing both customer and carrier relationships to be successful in this market.

TECHNOLOGIES

General Wireless Architecture Model

Wireless architecture enables communication between disparate devices and protocols. From a logical perspective the architecture is relatively simple in form. Transactions are driven from a source to a destination through a set of protocols and translation engines that allow communication between devices that do not have the same native communication standards.

Wireless Architecture Using WAP Communications

The wireless architecture model consists of a few basic elements that, when interconnected, enable inter-protocol communication. The basic elements are wireless devices, communication infrastructure, gateway devices, and application engines. See the Appendix for more information on WAP architecture.

Communication Protocol. The architecture is essentially split into two analogous parts that can be distinguished by the data transfer protocol (see Exhibit 4). The wireless side of the architecture is defined by wireless communication protocols (WSP), while the wired side is defined by conventional wired communications protocols (HTTP). The selection of protocols to use on either side of the architecture is an application- or device-dependent decision. In the case of the wireless side, the choice depends on

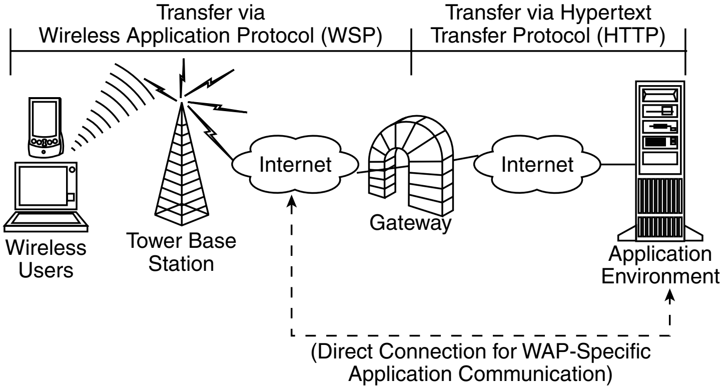


Exhibit 4. Communication Protocol

the capabilities of the device and the speed at which data may be transferred to and from the device. Today the most common protocol is WAP, but any generic protocol may be substituted in its place as long as the remaining elements in the infrastructure can support its use. For example, in an environment where bandwidth is great, the protocol of choice may be HTTP due to the additional robust features that it provides. The wired side of the architecture usually includes a land-based application engine that traditionally communicates via HTTP protocol. Like the wireless side of the architecture, the application engine does not necessarily need to use the common HTTP protocol; it may choose any established protocol that the rest of the architecture can support (e.g., ftp, gopher, SMTP, SNMP, etc.).

Communication Infrastructure. Facilitation of communications requires that some special telecommunications infrastructure exists in the geographical region surrounding the desired area of wireless access. For example, to communicate with a wireless device a user must be within 2 to 20 mi (based on protocol, frequency, atmospheric conditions, and other factors) of a tower base station. Quality of transmission will depend to a great degree on the quality of the connection that is established with the application environment. The path to the application environment is dictated by the wireless transmission from a handheld device to the closest base station tower, the connection between the tower and the gateway device, and the connection between the gateway device and the application environment. In almost all cases the bottleneck in information transfer exists between the tower and the wireless device. The data rates that are currently available for tower to device communication are generally on the order of 19.2 Kbps (sometimes higher depending on infrastructure), significantly

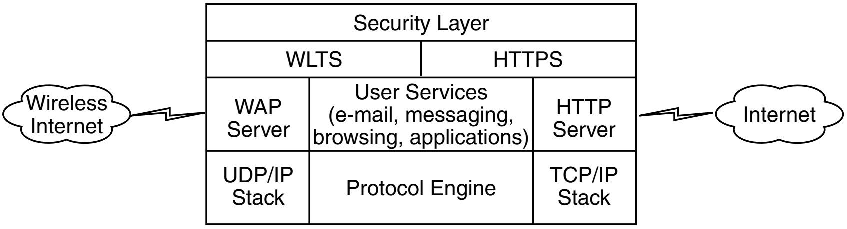


Exhibit 5. Gateway

lower than LAN or WAN communications that comprise the bulk of the information transfer in the rest of the architecture.

Gateway. The device that bridges the gap between the wireless world and the wired world is the Gateway device (depicted in Exhibit 5). The gateway can be either purchased as a component from an outside vendor such as Ericsson or can be developed as an in-house application. The Gateway device is very similar in logical function to the traditional LAN router that allows communication between disparate networks. Physically, the Gateway resides between the application environment and the wireless transmission point of origin. The Gateway, like a router, can be implemented as software or hardware solutions. In general, the more sophisticated solutions require hardware Gateways.

The gateway performs *protocol translation*, provides *security*, and acts as an *application engine*.

Protocol translation is the primary purpose of the Gateway device. The Gateway parses requests and transfers between HTTP and WSP. The Gateway works in both directions and can translate multiple protocols to WSP. For example, the Gateway device supports the translation of secure HTTP or HTTPS.

Security services provided by the Gateway are consistent with the security model that is pervasive on the Internet and World Wide Web. The Gateway employs a horizontal security layer that acts using a similar methodology to the protocol translation engine. Security implementation differs from wired devices to wireless devices. The security layer is split between the wired and wireless interfaces of the Gateway, each using their own version of the traditional Secure Socket Layer (SSL).

The Gateway *application engine* provides API level interfaces that allow application designers to build applications that reside on the Gateway rather than behind the Gateway. The primary advantage of applications residing on the Gateway device is related to application specification and

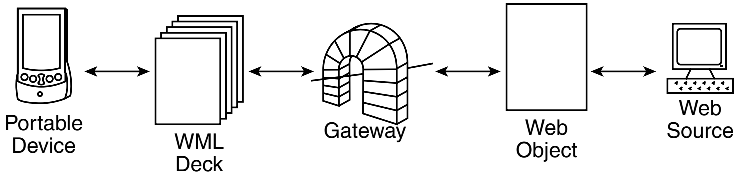


Exhibit 6. Logical Request Model

the level of customization that can be applied to an application if it is designed to run on a specific Gateway platform. Additionally, resident applications need only make HTTP transactions for information that is not resident on the Gateway device. Nonresident information includes database access and nonconventional data requests such as rich media that are not readily formatted for use on a wireless device.

Logical Request Model. The logical request is analogous in concept to a simple client/server transaction. The transaction model is as follows (see also Exhibit 6):

1. User makes request on cellular phone from preexisting list of available Web sites or services.
2. Request is sent from cellular phone as a WML object back to the Gateway device.
3. The Gateway device performs necessary translations and manages security between networks.
4. The Gateway passes a Web object from the Gateway to the application server or Web server to process the request.
5. The requested Web object is passed back to the Gateway where the translation engines convert the information into a WML deck.
6. The deck is passed back to the phone where it will be viewed by the user.

Web Clipping Wireless Architecture

Web Clipping is a proprietary system developed by Palm Inc. for delivering data over a wireless link. To minimize the amount of wireless communications while providing a high quality user experience, Web clipping applications are divided into static and dynamic sections. Static information such as a company logo is stored in the device's local memory, while dynamic content like a stock quote is stored on a land-based server. Dynamic content is then accessed via the wireless link. Currently over 400 Web sites support the Web-clipping technology.

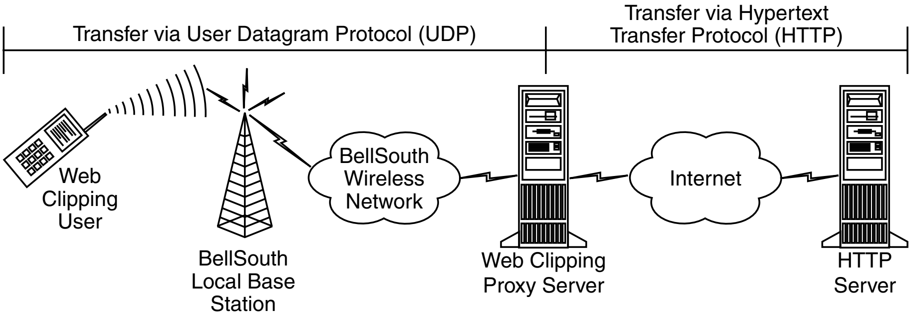

Transfer via User Datagram Protocol (UDP)
Transfer via Hypertext Transfer Protocol (HTTP)
Web Clipping User
BellSouth Local Base Station
BellSouth Wireless Network
Web Clipping Proxy Server
Internet
HTTP Server

Exhibit 7. Web Clipping Communications Protocol

Palm offers four handheld devices capable of running Web clipping applications: the Palm III, Palm V, Palm VII, and the new M100 model. Of these products, only the Palm VII has an integrated wireless capability. Third-party hardware is necessary to provide the wireless capability for the other three products.

Communication Protocol. The Web clipping architecture is completely independent of the underlying telecommunications protocol (see Exhibit 7). However, the Palm VII's integrated wireless communication hardware is specifically designed for the Mobitex standard and service is provided exclusively through Cingular's (formerly BellSouth Wireless) Interactive Intelligent Wireless Network.[1,2] Mobitex is a digital packet-switched technology capable of delivering 8 kbps over a 12.5-kHz radio channel. Dependence of the Palm VII on only one communication standard has complicated the expansion into international markets where GSM networks are more common.[3]

The Palm V, Palm III, and M100 handheld devices require additional hardware to provide the wireless communications. A "clip-on" wireless modem using the Cellular Digital Packet Data (CDPD) standard is available from both Omnisky and Novatel. By overlaying TCP/IP on existing AMPS cellular networks, CDPD can transmit packets of information at data rates of up to 19.2 kbps.[4]

In addition to wireless modems, Palm devices may also be coupled with a digital cellular phone, via cable or infrared port, to provide wireless communications. However, since this method requires that a dedicated wireless connection be maintained even when no data is being transferred it is less efficient than using a CDPD modem.

Logical Request Model. A Web clipping transaction model for a Palm VII device typically has the following flow (see Exhibit 8):[5]

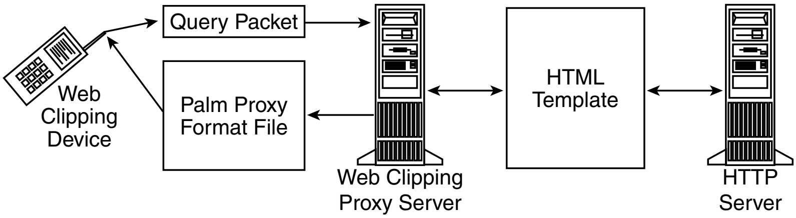


Exhibit 8. Web Clipping Logical Request Model

Exhibit 9. WAP versus Web Clipping

	Web Clipping		
	Palm VII	**Palm III, V, and M100**	**WAP Devices**
Technology standard	Proprietary	Proprietary	Open
Communication provider	BellSouth	OmniSky, Novatel	All 2.5G and 3G wireless providers
Data rates	8 kbps	19.2 kbps	150 kbps to 2 Mbps
Access both WAP and Web clipping	No	Yes; requires WAP browser installation	No
Markup language	HTML 3.2 subset	HTML 3.2 subset	WML (application of XML)
Web site access	Preplanned only	Preplanned only	Surf any site

1. User requests information by tapping a link on the screen.
2. If the information requested is a static part of the Web clipping application stored then it is immediately displayed, otherwise a query packet (~50 bytes) is passed to the Web clipping proxy server.
3. The Web clipping proxy server converts the request to a HTTP/HTML and then forwards the request to the HTTP destination server.
4. The HTTP destination server processes the request and returns the appropriate HTML content to the Web clipping proxy server.
5. Back at the Web clipping proxy server the HTML content is compressed and converted to a UDP packet format (~500 bytes).
6. The content is then sent back to the device where it is rendered by the Web clipping viewer.

WAP versus Web Clipping

Exhibit 9 highlights several key differences between the WAP and Web clipping architectures as related to technological and usability.[4,6]

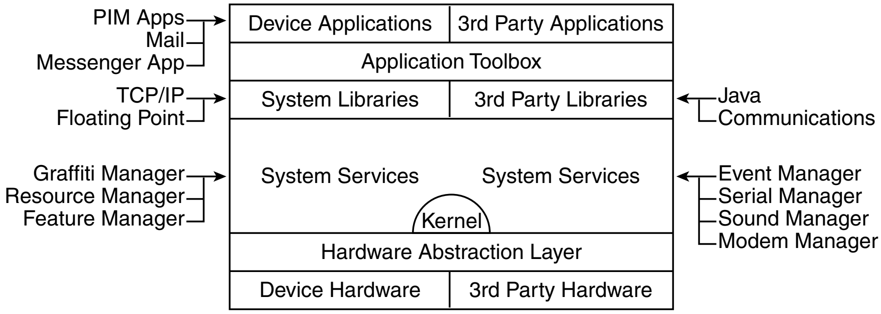


Exhibit 10. Palm OS Schematic

Operating Systems: Palm OS versus Windows CE

Perhaps the greatest factor in choosing a PDA is the operating system because it manages the components of the device and runs the applications. Palm has dominated the market from its inception, presently holding approximately 80 percent market share, as opposed to only 15 percent by Microsoft.[7] Palm has established this position by gaining the initial exposure and acceptance by the individual consumer for its non-wireless enabled devices. The Palm OS is a clean, simple, and familiar solution to users. Microsoft has released numerous versions of the more robust, yet cumbersome, Windows CE operating system (including mobile versions of PowerPoint, Excel, and Word) in the last few years, in an attempt to establish itself as the all-purpose solution in mobile devices. Nevertheless, the dominance Microsoft has enjoyed in the PC market has not yet been duplicated in the PDA market. However, the April 2000 release of the newest version of Windows CE, now termed the PocketPC, and the devices that support it (HP Jornada, Compaq IPAQ, Casio Cassiopeia) has greatly improved the usability and thus has brought this competition to a critical point.

Palm's dominance is based on "Simplicity, Wearability and Mobility."[8] To expand on this philosophy, the operating system has been developed considering the needs of the target user. A Palm device is not intended to be a mini-PC, but rather a unique tool designed for information management, providing easy and seamless access to personal, corporate, and Internet-based information. Palm does not strive to supply the user with all the applications that would be available at a desktop, but rather the critical data to make one's job and/or daily life more productive and efficient.

PALM Operating System. The PALM philosophy on PDAs is reflected in the design of its operating system. Exhibit 10 provides a schematic of the Palm OS.[9]

The kernel, which is the heart of the operating system, is built on top of the hardware layers. Aligned with the Palm philosophy, the kernel is precisely designed to produce optimal performance for an information management machine. Furthermore, its design is highly specific for only the hardware platform designed by Palm Computing. This is considerably different than the PocketPC, which is designed to support numerous hardware platforms from different vendors. The kernels must support not only different CPUs, but different screens, modems, and other peripheral devices. In other words, there is only one version of the Palm OS kernel as opposed to at least four for the PocketPC.[10] This consistency creates simplicity and stability for the Palm OS, but also results in limited functionality. Above the kernel are the system services; some of these include I/O through the graffiti manager, communications through the modem manager, and memory through the resource manager. Systems libraries are accessible to developers to interface custom applications with the Palm. The simplistic architecture has facilitated over 50,000 developers globally who create new products (most commonly in C++) tailored to the information they need.[11] Above the systems services is the application layer, where only one application can run at a time. The OS is event-driven, which is based on user interface from the stylus and from system calls. The lack of multitasking is designed to streamline the usability of the device to efficiently process the desired request from the user.

The openness of the Palm OS architecture has led to alliances with leading software firms such as Computer Associates, Lotus, Oracle, PeopleSoft, Remedy, SAP, Sun, Sybase, and Vantive to develop a vast library of management and communication applications. Also, Palm has partnered with companies such as Handspring, QUALCOMM, Symbol, TRG, Nokia, and Sony to develop other devices based on the Palm OS, such as smart phones.[12] The Palm OS has allowed Palm to gain considerable loyalty from both consumers and business partners. The challenge for Palm will be to continually innovate new features while still remaining true to its objectives of simplicity, wearability, and mobility, and at the same time be able to manage the rising competition from the PocketPC.

WindowsCE. The WindowsCE operating system in the handheld device industry has not been able to replicate Microsoft's success in other markets. Windows CE is a pared down version of a PC operating system (Windows), whereas Palm OS was designed specifically for a PDA. Upon viewing the product specifications, it appears that Windows CE should be the clear functional choice. There is an endless list of features that far exceed the Palm OS. For example, it allows multitasking of applications such as Word, Excel, Money, and full Web browsing. Windows CE devices have significantly more memory than the Palm, about double the resolution (and of course in color), 206 MHz processing as opposed to 20 MHz for Palm, automatic synchronization with

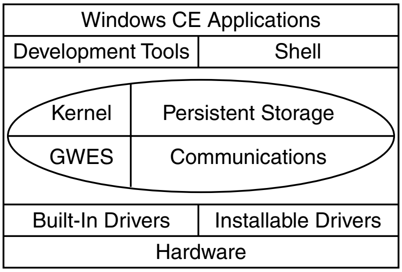


Exhibit 11. Windows CE Schematic

desktop software, natural handwriting recognition, voice recognition, packaged mapping software, an MP3 player, and animation.[13] And as mentioned earlier, at least four different versions of the kernel have been developed to support vendors such as Casio Computing, HP, Compaq, and Symbol. Opposite the Palm OS philosophy of simple, directed architecture, Windows CE is robust and modular. Microsoft has struggled to harness all these capabilities into a simple, easy-to-use product that the average consumer can understand and operate. The PocketPC is the latest attempt to manage all of this mobile functionality. Exhibit 11 presents a schematic of Windows CE.[14]

The four sections of the oval represent the major modules in the architecture. The kernel supports the basic system, such as CPU and memory processing. It is a preemptive, priority-based thread system. The filing system is controlled by persistent storage. GWES is the graphic windows and events subsystem that controls the windows interface. The communications module interacts with the exchange of information with other devices. Similar to the Palm OS, the kernel and system services are built on top of the hardware and support the development tools. The shell is the command interpreter, which is a windows-based system similar to Windows98. Finally, the CE applications reside on the uppermost layer.[15]

As mentioned, Windows CE is more indicative of a complete operating system, allowing it to support vast functionality and capabilities. While being able to support numerous interfaces and complex applications, CE has some performance, security, and ease-of-use issues, not to mention it requires a physically larger device for the user to carry, in order to support its complex multimedia capabilities. The target customer is one who is looking for a wealth of functionality, but is also concerned with practicality. These are typically individuals utilizing the device for personal management or specific professional functionality. So far, it is evident from sales that consumers have felt that past Windows CE technology could not support all its features in a way to offset the simplicity and convenience of the Palm.

As more businesses familiarize themselves with the capabilities of mobile units and as technology advances, handheld devices will most likely become a pertinent and standardized part of the corporate environment. Because of this trend, Microsoft has the potential to gain significant market share because new hardware technology will provide Microsoft the ability to continually pare down the physical size of the device, which is one of its leading drawbacks. Windows-based networks are already established throughout many industries, and with the continued development of PocketPC's interface with client/server and wireless networks, it would be the more preferred option for conformity and support. Also, as users learn to rely more and more on mobile features, the PocketPC has clear advantages with its more comprehensive suite of tools.[16]

FUTURE OF WIRELESS

The future of wireless will be driven by advances in wireless technology and by adoption of new and existing wireless technologies. New technologies in wireless networking, integration platforms, and mobile access will spawn new applications for the technology and also lower the cost of service, hence spreading wireless technology to mass markets. The future of wireless is closely tied to the development and adoption of technologies in three key areas — 3G networks, new wireless integration platforms, and the adoption of wireless access technologies.

The Impact of 3G Wireless Networks

The next wave in wireless networking technology is high bandwidth "3G" Wireless networks. These networks will bring the wireless Internet and computing experience closer to the wired counterpart. 3G wireless networks are expected to have significant availability by 2004. They will allow high-speed data applications up to 144 kbps/s while on the move and 2 Mbps/s when stationary. They will also enable the creation of a "virtual home" environment allowing users to carry their profile with them and hence providing consistent Internet experience through wired and wireless networks.

According to an Ovum group report on "3G Mobile Market," the market for 3G services first emerged in Japan, moved to Europe, and will gain mainstream acceptance in the United States by 2003. By 2003, 59 percent of the European wireless market, 22 percent of Asia Pacific, and 19 percent of the Japanese market will be using 3G wireless networks.[17]

High bandwidth wireless networks will provide convenience, personalized access, and drive cost efficiencies in certain market segments.

Although this will initially create efficiencies in many of the "high opportunity" markets (see the section on Business Impact), the overall economic impact of these applications will be significant and would create the next wave of Internet productivity tools.

Today, most wireless networks are circuit-switched and offer data rates much less than 56 kbps and require that users dial into the network to retrieve information. Each time a user connects to the network can take 15 to 30 s and then starts the transmission of data. Although this may not seem like a significant amount of time, users that frequently interact with their wireless device will need to continually dial into the network, creating unacceptable service levels. 3G networks are packet-switched, which means that the device is constantly connected to the network. When an e-mail is received at the Gateway server it can be directly sent to the device without the user initiating a connection. 3G networks will eventually have data rates of 2 Mbps, however, within the next few years are expected to provide 384 kbps.[18]

To capitalize on the additional bandwidth provided by 3G technologies computational capacities also require improvement. Both Intel and Transmeta are developing advanced CPUs for mobile devices. Additional performance also comes at the cost of power consumption. The lower the power consumption the longer the battery life. Transmeta's Crusoe chip is the leader in this respect, requiring only 1 to 2 W compared to the Intel chip at 10 to 15 W.[19]

The Impact of New Integration Platforms

The development of wireless networks will also drive the need for platforms that integrate wireless applications to the wired. This includes new developments in operating systems, protocols, database applications, E-commerce, and content applications. Much of the early development will be directed toward achieving better integration with existing Internet applications and establishing new industry standards. The business value created will be through combining the availability and portability benefits of wireless technology with the benefits of the Internet and computing in making the business applications and services available through wireless devices. Services like E-commerce (m-commerce), content, e-mail, scheduling, and business-specific applications will be available through wireless devices. There are many fantastic business scenarios narrated in the media; however, after discounting the hype, according to the Durlache Mobile commerce report, it is applications like e-mail, unified messaging, instant messaging, and E-commerce that will form the main applications of wireless Internet.[20] New platforms that support mobile commerce, wireless security, PKI, and content management will shape the development of the wireless market.

The Impact of Access Technologies

Access technologies bring the wireless network and the Internet to the user. These include end-user connectivity technologies like Bluetooth and mobile commerce terminals, operating systems for mobile devices, physical terminals, access software like micro-browsers, access hardware like smart cards, and security protocols like PKI and authentication devices. All these devices/software operate at the user level on the PDA, cell phone, or other wireless device. Competition in these products will accelerate the development of devices that are more secure, more integrated, and also easier to use. This will also drive the use of wireless products.

User Interface

The entire concept of the user interface must be redesigned for the future of wireless technologies. The idea that a typical Web site can look the same way on a wireless device that it does on a PC is not realistic. The fact of the matter is that users will be increasingly pushed or pulled toward the small form factor exhibited by the cell phone and the generic PDA. There is a significant question as to how the precise form factor will turn out, but it will be in some ways similar to what we can relate to today with a bit of James Bond mixed in.

The killer device of the future will likely be a fusion device — a wireless device that acts as a combination of all wireless devices that are currently taking positions within the wireless market. The next-generation device must support a user interface that allows an intuitive look and feel that is generally accepted as a usable device.

The killer application on the killer device will be those applications that are designed for the specific interfaces that the fusion devices support. Applications that support one-click transactions and voice/video interfaces will likely gain approval with a wide variety of increasingly demanding users. To achieve the one-click design much work will need to be done in the area of human factors engineering employing scientific methods of engineering and the human understanding of cognitive psychology. Significant research is required to develop new interface paradigms that will support advanced application and infrastructure technologies coupled with fairly limited display capabilities.

As third-generation devices evolve into fourth-generation devices the user interface requirements of the fusion wireless device may tend to be closer to what is currently understood as the wired Web browser. A fourth-generation device will have the architectural and network capability to stream high bit rate video and support innovative collaborative activities that involve sharing multiple data types with many concurrent users.

Mobile Commerce

Mobile commerce is currently in its infancy with text-based news and e-mail leading today's killer applications, but the future holds a wide array of possibilities including mobile entrainment, such as video streaming and interactive multiplayer games, and an electronic wallet application, which can hold payment solutions and identification. While these applications are not necessarily new to Internet users, mobile devices can significantly enhance the value of almost any E-commerce application or Web site. A typical person has his mobile phone within arm's reach almost 24 hours a day and 7 days a week, whereas a person may only be in front of a wired computer for a few hours per day. In addition, the next generation of wireless telecommunications will provide information about the physical location of the user, and also will provide immediate delivery of information because devices are always connected to the Internet. Thus, mobile commerce does not create a "new" distribution channel, but rather a "ubiquitous, instant, and location-specific" distribution channel that can increase the value of a business' products and services.

Although the recent meltdown of investor confidence in E-commerce companies has certainly reduced the excitement surrounding m-commerce services, the main issue delaying the widespread market introduction is technological. Problems include mobile device form factors currently having small low-resolution displays, the wireless communication networks are still mainly circuit switched requiring dial-up connection, and the location technology has not yet been introduced. Once these technological constraints have been resolved, all E-businesses may need to develop mobile commerce sites just to retain their position in the market. For example, if Yahoo! were to develop a mobile site for its online auctions this would provide a significant advantage over its competitors in this product space, since frequently much of the biding on items is done in the last few hours before the item is sold, requiring participants to continually access the site during this period. Even if the situation is non-competitive, business could benefit from location-specific advertising provided by mobile technologies.

In the future, wireless will work in conjunction with traditional technologies and will be readily adopted by business functions and industries where real-time information, accessibility, and location specificity impact growth and create new opportunities.

APPENDIX 1. WAP ARCHITECTURE[21]

The Wireless Application Protocol (WAP) is a combination of a communication protocol and an application environment that facilitates accessing the Internet from a wireless device. WAP is functionally similar to protocols used on the Internet; however, the high overheads and latency of HTTP and

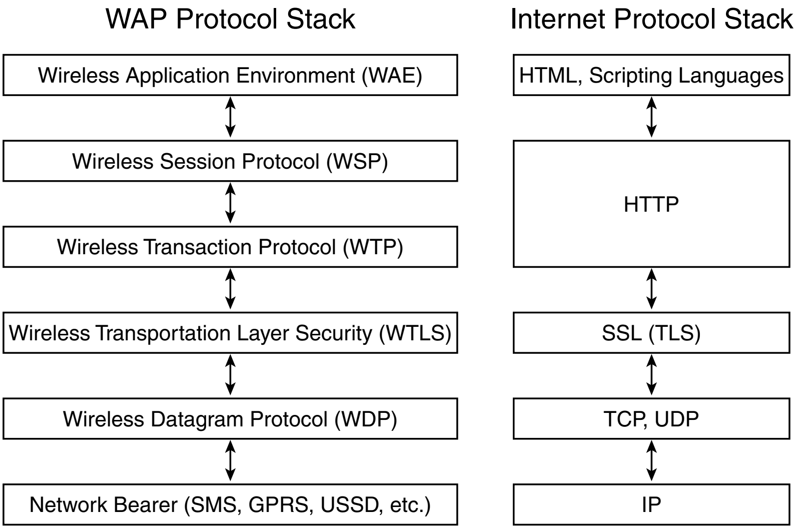

WAP Protocol Stack
Internet Protocol Stack
Wireless Application Environment (WAE)
Wireless Session Protocol (WSP)
Wireless Transaction Protocol (WTP)
Wireless Transportation Layer Security (WTLS)
Wireless Datagram Protocol (WDP)
Network Bearer (SMS, GPRS, USSD, etc.)
HTML, Scripting Languages
HTTP
SSL (TLS)
TCP, UDP
IP

Exhibit 12. WAP Protocol Stack

TCP/IP make these protocols not well suited for mobile devices. In addition, Internet markup languages, such as HTML, make the assumption that all devices have similar display sizes, memory capacities, and software capabilities. This assumption does not hold for wireless devices. The WAP architecture has been optimized for handheld wireless devices possessing little memory and low process speeds, and that communicate over low bandwidth. The WAP architecture is divided into five different layers of functionality (depicted in Exhibit 12), which compose the WAP protocol stack.

Wireless Application Environment (WAE)

The WAE layer (see Exhibit 13) provides application development and execution environments. WAE is logically divided into a user agent layer, and services and formats layer.

A user agent is a software program that interprets content on behalf of a user. The most fundamental type of user agent in WAE is a WML user agent, also known as a WML browser. A WML browser can transform the content of WML and WML Script documents to a form easily understood by the end user. The exact transformation made by the WML browser is not specified in WAE, which allows the display to be optimized for the features and capabilities of a particular device.

Two other common user agents found on wireless devices is a message editor and a Wireless Telephony Application (WTA). The message editor

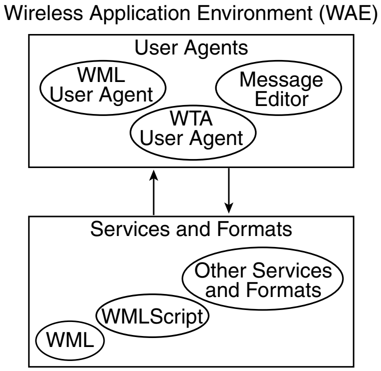


Exhibit 13. Wireless Application Environment Layer

user agent can be use to view and write e-mail. WTA is an application that interacts with telephone-related functions provided by the phone network provider, such as phone book and calendar applications.

The WAE specification includes services and format layers that focus on ensuring interoperability among various user agent implementations and optimizing documents for low data rate transport. The main features of the specification are *WML*, *WMLScript*, and *methods for encoding WML and WMLScript* to minimize the size of the files.

The *Wireless Markup Language* (WML) is an XML-based markup language that provides a way to organize data without indicating how the data should be rendered or displayed. This is important for wireless devices since the size of the display can vary greatly. A WML document is organized around a "deck of cards" system (see Exhibit 14). A deck contains one or more cards, with each card containing both content and navigational controls. When a WAP device requests an application from a server, the server will send back a deck; the cards will then be displayed one at a time. The deck system provides an effective means of controlling the amount of information displayed and reducing latency since the next card has already been transferred.

WMLScript is a procedural scripting language based on JavaScript. It adds intelligence to the application without needing to contact the server. Before the WMLScript is transported to the wireless device it is complied into space-efficient bytecode at the gateway.

To increase transmission efficiency of WML and WMLScript and to minimize the computational needs of the device, WML tokenizes and WMLScript uses a bytecode format. In the case of WML the tag name is replaced

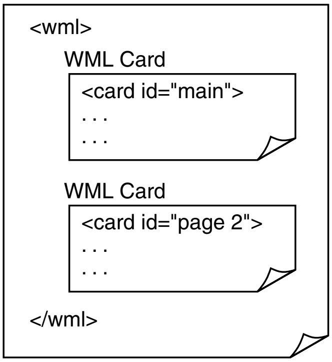


Exhibit 14. The WML "Deck of Cards" System

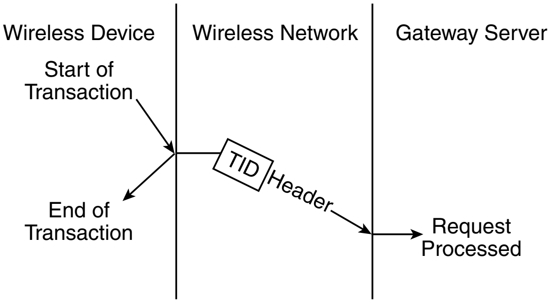


Exhibit 15. WSP Session Protocol

by an eight-bit token. For example, an "anchor" tag is replaced by the hexadecimal token number of "22."

Wireless Session Protocol (WSP). WSP provides an organized method of data transfer for the application layer (see Exhibit 15). WSP is functionally the same as HTTP/1.1, but specifies the use of a compact binary encoding for the methods, status codes, and other parameters included in HTTP headers. This reduces protocol overhead and increases the transmission efficiency over the wireless network.

WSP has two different types of session services for connecting to the server gateway (see Exhibit 16). The "connectionless" session service is the simpler of the two — it bypasses the WTP layer and operates directly over the WDP layer. However, the simplicity also makes it an unreliable method of transport because the server does not confirm the reception of a communication. A connectionless session can be used to send a method

Exhibit 16. Two Types of Session Services for Connecting to the Server Gateway

Connection Mode	
Session management	• Enables device to connect with server • Facilitates transfer of protocol options • Exchange of attributes, use for the duration of the session
Method invocation	• Allows device to request an operation from the server • HTTP methods or user-defined extension operations
Exception reporting	• Allows the server to notify the user of events that do not change the state of the session and are not related to a particular transaction
Push facility	• Enables the server to send unsolicited information to the device • This service may be either confirmed or unconfirmed
Session resume	• Provides means to suspend a session until the device wants to resume
Connectionless	
Method invocation	• Allows device to request an operation from the server • HTTP methods or user-defined extension operations
Push facility	• Enables the server to send unsolicited information to the device • Unconfirmed service only

to a server (i.e., GET) or for "push" services. To track a session a Transaction Identifier (TID) must be appended to each header. If the server replies to a request the TID is needed to find the requesting user.

Connection-mode is the second type of session service. It operates over the WTP layer, which provides reliable data transmission between device and server. To begin a new session, a header containing information that will remain constant over the life of the session is exchanged between the user agent in the device and the gateway server. This header might include information on content type, character set encoding, or languages.

Wireless Transaction Protocol (WTP). The WTP protocol uses the WSP layer to provide both reliable and unreliable transactions over the inherently unreliable underlying datagram service. A datagram service has advantages for wireless applications because no explicit connection setup or teardown phases are required. Reliability is achieved through the use of unique transaction identifiers, acknowledgments, and re-transmissions.

Each transaction is uniquely identified by a socket pair (source address, source port, destination address, and destination port) and a transaction identifier (TID). The TID is included with the WTP protocol data unit, which is located in the data portion of the datagram. When the gateway server receives a message from WTP, an acknowledge message can be sent back to inform the device that the message was received. If an acknowledge

Exhibit 17. Three Classes of Transaction Service

WTP Transaction Class	Description	WSP Facility Usage
Class 0	Unreliable message (no result or acknowledge message)	Session management Session resume Unconfirmed push
Class 1	Reliable message (acknowledge message, no result message)	Confirmed push
Class 2	Reliable message (acknowledge and result messages)	Session management Method invocation Confirmed push

message is not received within a defined amount of time, the message is then re-transmitted.

To provide flexibility in the degree of reliability, WTP specifies three classes of transaction service (see Exhibit 17). Class 0 transaction is an unreliable service that is used to send less important messages to a particular socket. WSP uses Class 1 transaction exclusively for reliable "confirmed push" messages. When a device receives the "push" message, it will reply with an acknowledgment message. The most commonly used transaction class is Class 2. Here a device can contact a server with a request message; the server then compiles the result message and sends it back to the device. The device closes the loop by sending the server an acknowledge message.

Wireless Transport Layer Security (WTLS). WTLS (see Exhibit 18) is an optional layer for security solution provided in the WAP architecture that is based on Internet protocol SSL version 3.0. It provides the same level of 1024-bit encryption security and services that ensures privacy, server authentication, client authentication, and data integrity. However, there is reason for additional security concern over that of the Internet. Information is sent from the originating server over SSL to the WAP gateway; then the gateway must unencrypt the information and translate it to WTLS before sending the information on to the wireless device. This means that the information is in plain text for a brief period at the gateway. Because of the security concern this creates, for applications where security is paramount the business may wish to implement its own WAP gateway server.

Wireless Datagram Protocol (WDP). The services offered by WDP (see Exhibit 19) must allow applications to operate transparently regardless of the capabilities of the underlying bearer service. These services include application addressing by port numbers, segmentation and reassembly,

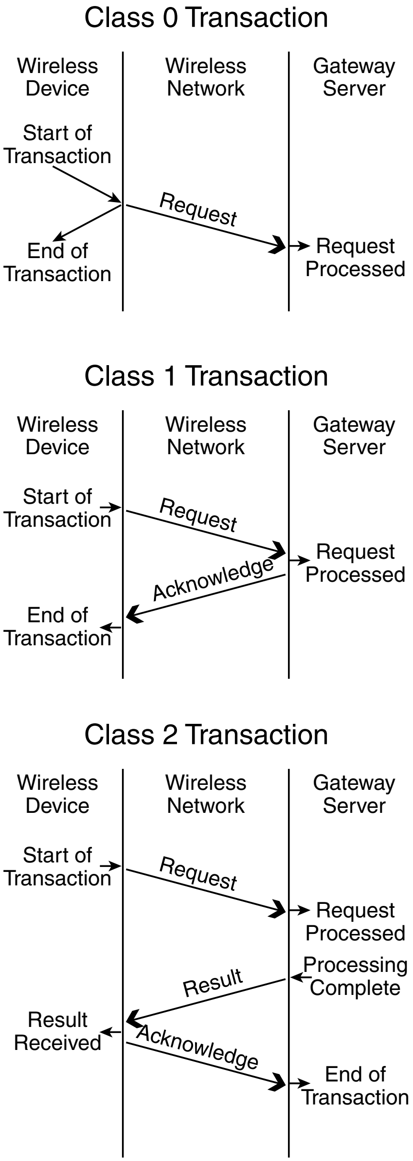


Exhibit 18. Wireless Transport Layer Security (WTLS)

and error detection. Some of these services may be provided by the underlying bearer service, in which case that element is eliminated to increase transmission speed.

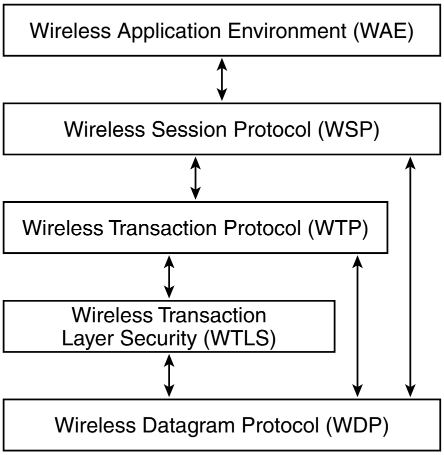

Wireless Application Environment (WAE)
Wireless Session Protocol (WSP)
Wireless Transaction Protocol (WTP)
Wireless Transaction Layer Security (WTLS)
Wireless Datagram Protocol (WDP)

Exhibit 19. Wireless Datagram Protocol

Notes

1. Palm VII Connected Organizer, http://www.palm.com/pr/palmvii/7whitepaper.pdf January 12, 2001.
2. http://www.bellsouthwd.com/index.php February 3, 2001.
3. Web Clipping — Not Web Browsing, http://www.ericsson.se/wireless/products/mobsys/mobitex/subpages/mmark/msegm/mmob/webcbrow.shtml February 10, 2001.
4. What is CDPD? http://www.nais.com/business/cdpd.asp February 9, 2001.
5. Web Clipping Developer's Guide, James Brook, 2000, Palm Computing, Inc.
6. Wireless Application Protocol Version 1.3, 1999, Wireless Application Forum, Ltd. http://www.wapforum.org
7. PocketPC Targets the Palm, John G. Spooner. http://www.zdnet.com/zdnn/stories/news/0,4586,2472028,00.html July 23, 2000. *ZDNET.com.* March 22, 2000.
8. The Philosophy Behind the Palm OS. http://www.palmos.com/platform/philosophy.html July 23, 2000. *Palm.*
9. A Flexible Architecture for Innovative Solutions. http://www.palmos.com/platform/architecture.html July 23, 2000. *Palm.*
10. Technical Analysis: Comparing Windows CE with Palm OS, Jason Perlow, March 1999. http://www.palmpower.com/issuesprint/issue199903/ninotwo.html July 23, 2000. *PalmPower Magazine.*
11. http://www.palmos.com/platform/architecture.html July 23, 2000.
12. http://www.palmos.com/platform/philosophy.html
13. http://www.microsoft.com/mobile/pocketpc/compare.asp
14. http://www.microsoft.com/TechNet/WCE/technote/chapt1.asp
15. http://www.microsoft.com/TechNet/WCE/technote/chapt1.asp July 23, 2000.
16. http://www.semperaptus.com/reviews/r6100a.html July 25, 2000.
17. The Ovum Group Report on 3G Mobile Market, 2000.
18. Professional WAP, 2000 Wrox Press.
19. Professional WAP, 2000 Wrox Press.
20. Durlache Mobile Commerce Report, 2000.
21. WAP Forum Specifications. http://www.wapforum.org/what/technical.htm, November 5, 2001.

ABOUT THE AUTHORS

Mark Barnhart holds a B.S. in Physics from the University of Missouri — Rolla and an M.B.A. and M.S. from Boston University. Mr. Barnhart has professional experience with Eaton Corp. in semiconductor product improvement.

Ritu Joshi is cofounder of BrightOne Inc., a product management consultancy. Mr. Joshi holds a Bachelor's degree in Economics and Statistics and a Master's degree in Information Systems from Boston University.

Dean Peasley has an M.B.A. with Honors and an M.S. in Information Systems with Honors from Boston University. Mr. Peasley is currently a senior business analyst for Liberty Mutual.

Jeffrey Staw is the director and owner of J. Staw Consulting, a firm specializing in E-business models and Internet architectures. Mr. Staw holds a B.S. in Human Factors Engineering and Management from Tufts University and an M.B.A. and M.S. in Information Systems from Boston University.

Chapter 16
Building a Wireless Web Solution: Tools and Justification for Building Wireless Web Solutions

Sylvain Duford

This chapter is an introduction to wireless Web development. We will discuss the concepts, requirements, and restrictions you will encounter when developing a wireless Web solution. This includes a discussion of the User Interface (UI) design guidelines imposed by the low bandwidth/high latency nature of the wireless Web, as well as the small screen format of most wireless devices.

Next, we will talk about the proper usage of HTML, ActiveX Data Objects (ADO), and Active Server Pages (ASP) to display dynamic data coming from a database. We will then discuss how you can use XML/XSL technology to target multiple types of wireless devices from the same code base. We will also introduce you to WAP, WML, and HDML, and how they enable Web access via mobile phones with very small screens.

Finally, we will discuss an exciting new technology coming out of Microsoft's Adaptive UI group called Mobile Controls for ASP.NET. This technology promises to take away most of the pain one currently encounters when trying to support multiple wireless Web devices.

This chapter assumes that you already have a basic understanding of HTML, XML, and JScript, as well as Web servers and databases.

0-8493-1161-6/02/$0.00+$1.50

WIRELESS WEB USER INTERFACE REQUIREMENTS

Designing a Web user interface so it will work well on wireless devices is very different than designing regular Web applications. There are several constraints and guidelines you must follow for your wireless Web application to be successful. First we will discuss what kinds of applications make sense in the wireless world.

When Does It Make Sense to Implement a Wireless Solution?

First, let it be clear that not all applications belong on the wireless Web. Accessing a Web application through a low bandwidth and costly wireless connection, and with a device that has a small screen and limited input capabilities, is often a real chore. This means that users will not use it unless it brings them value. Users will not use the site if they can just wait and do the same thing more easily on their desktop machines. This brings up what is, in my view, the single most important success factor for wireless Web applications: there should be a definite time-related advantage. This can either be in the form of time-sensitive data or the type of information you need to get while on the go. For example, stock trading is a highly time-sensitive activity; selling or buying securities at just the right time can save you many dollars, as will the ability to stay abreast of what is happening in the markets anywhere, anytime. This makes wireless stock trading and banking a compelling application. Another example would be flight and weather information or driving directions — information that you need while away from your desk.

Wireless Access versus Synchronization

So far, we have only discussed accessing the Web in real-time over a wireless connection. Another way to take the Web on the road is in a disconnected fashion, through periodic synchronization. In this scenario, your browser will synchronize the content stored on your device while connected through a conventional Internet or LAN connection. This is usually accomplished through a *channel*. You then disconnect and take this static Web content with you on your handheld device. If you want more information on channels and Web browser synchronization, you should look into the AvantGo technology or Microsoft's offering for WinCE devices.

What Are the Restrictions?

Applications that require lots of data entry, large screen displays, or high bandwidth are not currently viable for wireless access. There are countless examples of applications out there that have failed or go unused because they just do not make sense on a small format wireless device. So before going forward with a wireless implementation, you should ask yourself: Does it make sense? Would I use it?

Some Wireless User Interface Guidelines

There are four main UI design factors to keep in mind when putting together your wireless application:

1. A small screen format with limited display and navigation capabilities
2. Limited data-entry capability
3. Low bandwidth and/or high cost of transmitting data
4. Network latency and other delays

The first two points mean that you really need to pay attention to how you design your screens, both for data display and data entry. For example, you should not use graphics (except for small icons on some devices), you should keep your tables small and mostly vertical, and you should not use complex control elements like cascading menus or calendar controls. Some of these are possible to do on some devices, but most will not translate well to smaller devices with lesser capabilities.

You need to pay particular attention to the navigation — you must keep it as simple and linear as possible. It is very easy to get lost in a maze of screens and menus. Also, try to avoid going more than three levels deep as it quickly becomes frustrating for your users to access the information on the lower levels.

An even more important consideration is data entry. It can be very arduous to enter large amounts of data on a Pocket PC; it is nearly impossible on a phone. So you should make extensive use of pick lists and menus, and you should store the user's information and preferences so she does not have to enter the same data more than once.

The last two design factors have to do with response time, latency, perceived performance, and usage cost. Some wireless networks charge by the amount of data transmitted, so you should strive to keep the transmitted data to the bare minimum necessary for your application. In addition, most wireless networks have a lot more built-in latency than traditional networks do. This includes things like the carrier's WAP server delay, encryption and authentication overhead, and the regular Internet and transmission delays. Many wireless devices, especially cellular phones, use only a 9.6-kbps transmission rate. To add insult to injury, some wireless browsers will not display a page until most or all of it has been downloaded to the device. These factors all add up to a perceived low performance and high latency. While most users expect a somewhat higher latency on wireless devices, you should not try not to go beyond 10 s; that is about the limit at which most users will give up on your application. Additionally, most wireless browsers will time out after 30 s, leaving the user with the impression that the server failed. Once again, the best way to limit the impact of these problems is to keep your pages simple and the amount of data small.

While these restrictions vary greatly according to the specific type of device you target, you pretty much have to go with the lowest common denominator. If you have the luxury of targeting a higher-end mobile device like a sub-notebook or Pocket PC, then you will have a lot more design leeway. However, the reality is that most wireless Web applications have to target most types of devices. So you have two choices here: you can target the lowest common denominator and not use the capabilities of the larger devices, or you can have two or more sets of Web pages, targeted at the different classes of devices. In the latter part of this chapter, we will discuss a new technology coming from Microsoft that promises to greatly reduce the effort required for supporting multiple wireless devices (see the section *Using Microsoft's Mobile SDK for ASP.NET*).

USING HTML IN A WIRELESS WORLD

Most developers think that you need to use wireless specific technology for serving content to wireless devices. In fact, good old HTML is perfectly capable of doing the job, as long as you keep in mind the restrictions we just discussed. As a bonus, you can make use of your old trusted HTML editor and you can leverage XML and XSL to target multiple devices.

If you are targeting Pocket PCs, sub-notebook computers, or any other high-end wireless devices, than using HTML is probably your best bet. You will have lots of flexibility and you will be able to use well-known development tools. However, you probably will not be able to use the latest version of HTML. Most wireless devices only support a subset of HTML 3.2, while some only support HTML 2.0. Therefore, you need to look at which devices you are targeting and select the subset of HTML tags that is common to all of them, and then limit your page designs to those tags.

The Pros and Cons of Using HTML

There are many reasons you might want to use HTML. HTML is well known, stable, and easy to use. HTML is powerful, flexible, and there are many good HTML editing tools on the market to help you. However, HTTP and HTML were not designed from the start to be used over wireless networks and they present a few drawbacks.

HTTP/HTML uses a lot more bandwidth than a pure wireless protocol like WAP/WML. In addition, HTML was not designed for the small and limited displays of small devices like Web-enabled phones. This means that you have to exercise extreme care in how you design your pages; you need to minimize the amount of data on your pages, and you need to restrict yourself to the subset of HTML tags that is supported by your target

Exhibit 1. Typical HTML Document for Wireless Browsers

```
<HTML>
<HEAD>
   <META NAME="HandheldFriendly" CONTENT="true"/>
   <META NAME="PalmComputingPlatform" CONTENT="true"/>
   <BASE HREF=http://www.cactus.ca/WirelessDemo/StockMenu.html/>
   <TITLE>Stock Trading</TITLE>
</HEAD>
   <BODY>
      <H1 ALIGN=center>Stock Trading Menu</H1>
      <P>Choose an action:</P>
      <MENU>
         <LI><A HREF="sell.html"</A>Sell Shares</LI>
         <LI><A HREF="buy.html"</A>Buy Shares</LI>
         <LI><A HREF="quote.html"</A>Get Price Quote</LI>
         <LI><A HREF="positions.html"</A>Get Positions Summary</LI>
      </MENU>
   </BODY>
</HTML>
```

devices. There are many tags that should be avoided because they are either not widely supported or simply do not work well with all wireless browsers. In particular, font support varies greatly from one wireless browser to the next, so you should avoid using specific font or typography tags. Among others, you should avoid the following HTML tags:

`<ISINDEX>, <LINK>, <SCRIPT>, <STYLE>, <EMBED>.`

All the raw font control tags like:

`<B>, <I>, <U>, <FONT>.`

And, most advanced tags like:

`<FRAME>, <IFRAME>, <DIV>.`

You should consult the documentation for each of the wireless browsers you are targeting for the list of supported tags. You should also avoid the use of client-side cookies, because, once again, support for cookies varies greatly from one wireless browser to the next.

A Sample HTML Document for Wireless Browsers

The sample shown in Exhibit 1 demonstrates and describes a typical HTML document designed for wireless browsers. We will not describe the basic HTML features but will concentrate on the ones that are relevant for wireless applications.

The first section declares some useful meta tags that provide information to wireless browsers. These tags are ignored by browsers that do not understand them, but they provide useful cues to those that do. The HandheldFriendly tag tells an AvantGo browser that this document has been properly formatted and that it should not try to reformat it for the small screen. PalmComputingPlatform does something similar for Palm Web Computing applications. You need to be careful not to add any unnecessary meta tags as all that extra data has to be downloaded by the browser. Many HTML editing tools in particular will add a whole series of meta tags for their own purposes. These should be stripped out before putting the pages on a production server.

The "BASE HREF" line sets the base URL so that further HREFs do not need to specify the entire URL. For example, without the base URL, the menu item lines would have to be much longer and would look something like this:

```
<LI><A HREF= "http://www.cactus.ca/WirelessDemo/sell.
html"</A>Sell Shares</LI>
```

That base URL directive can greatly reduce the amount of data being transmitted for each page. The rest of the page is straight HTML except that it is purposely kept simple and the lines are kept short.

WIRELESS DATA ACCESS

The simple HTML page we looked at in the previous section is completely static. In most cases, you will need to get data out of some storage mechanism and dynamically build your Web pages based on those data. This can easily be accomplished using Microsoft's ASP technology and ADO. The techniques described here are nothing new and we assume that you already have an understanding of ASPs and you can understand VBScript code. For the purpose of this chapter, we will make use of these technologies with a wireless frame of mind.

Using ASP and ADO

Using ASPs, we can easily gather data on the server side and combine it with the HTML before sending the page out to the browser. This is accomplished by inserting VBScript or JScript code in the ASP page. This code is executed on the Web server, and it can very easily access any database server. However, some of the traditional ASP techniques that you may want to avoid are returning large record sets, doing data binding with complex HTML controls, and displaying large tables. All of these will go against the design principles we discussed at the beginning of this chapter.

To give you a taste of what wireless data access looks like, we will go through a simple ASP application (see Exhibit 2) that returns a list of stock positions from a fictitious portfolio. The data will be displayed in a simple table format.

At the top of the page, the METADATA line sets a reference to the ADO 2.5 type library. The code block that follows the display header accesses the

Exhibit 2. Simple ASP Application

```
<%@ LANGUAGE = JScript %>
<!--METADATA TYPE="typelib" uuid="00000205-0000-0010-8000-00AA006D2EA4" -->
<HTML>
<HEAD>
   <TITLE>Stock Portfolio, Positions</TITLE>
</HEAD>
<BODY BGCOLOR="White" topmargin="10" leftmargin="10">

<!-- Display Header -->
<H2>My Stock Portfolio</H2>

<%
   var oConn;
   var oRs;
   var curDir;
   var Index;

   // Map physical path to authors database
   curDir = Server.MapPath("stocks.mdb");

   // Create ADO Connection Component to connect
   // with sample database
   oConn = Server.CreateObject("ADODB.Connection");
   oConn.Open("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +curDir);

   // Create ADO Recordset Component
   oRs = Server.CreateObject("ADODB.Recordset");
   oRs.ActiveConnection = oConn;

   // Set Recordset PageSize so that it only holds 10 rows
   oRs.PageSize = 5;

   // Get recordset
   oRs.Source = "SELECT Symbol, Shares, Price, CCur([shares]*[price]) AS
[Value]FROM Positions"
   oRs.CursorType = adOpenStatic;

   // Open Recordset
   oRs.Open();
   %>

   <TABLE border = 1>
      <!-- Table Header -->
      <CAPTION>Current Positions:</CAPTION>
      <TR>
         <TH VAlign=top>Ticker</TH>
         <TH VAlign=top>Shares</TH>
         <TH VAlign=top>Price</TH>
         <TH VAlign=top>Value</TH>
      </TR>

   <%
   // Create table rows from recordset
   var RecordCount;
   RecordCount = 0;
   while ((!oRs.eof) && (RecordCount < oRs.PageSize)) { %>
      <tr>
         <% for(Index=0; Index < oRs.fields.count; Index++) { %>
            <TD VAlign=top><% = oRs(Index)%></TD>
         <% } %>
      </tr>
      <%
         RecordCount = RecordCount + 1;
         oRs.MoveNext();
   }
   %>
   </TABLE>
   <%
      oRs.Close();
      oConn.Close();
   %>
   </BODY>
</HTML>
```

database and retrieves a RecordSet for display. After declaring a few variables, we create an ADO Connection object and open a connection to the Stocks.mdb database file. After that, we prepare a RecordSet object and open it, which executes the query and loads the data from the database.

After the script block, we are back to HTML mode and we create a table header and table to display the data from the RecordSet. This is followed by another script block that loops through the records and inserts each one into a table row. Following this loop, we complete the HTML table and close the ADO objects. It is that simple! We have now loaded data from a database and displayed it in a small HTML table suitable for Palm Pilots and Pocket PCs.

USING WAP AND WML

WAP stands for Wireless Application Protocol and it is the equivalent of TCP/IP in the Internet world. Just like TCP/IP, WAP defines a whole suite of low-level protocols dedicated to transporting data to and from wireless devices.

WAP was created by a group of industry heavyweights including Microsoft, Nokia, Ericsson, Motorola, Sony, Palm, and Phone.com. Because it was designed from the start as a wireless protocol for cellular phones, WAP is much more efficient in its usage of bandwidth and screen real estate than HTML is.

What Is WML?

WML stands for Wireless Markup Language and it is a standard part of the WAP specification. It is an XML-based language that looks a lot like HTML, so it is easy to learn for any experienced Web developer. One big difference with HTML is that WML uses a card and deck metaphor. A card is a single page (usually one screen) that contains standard UI elements like lists, tables, action buttons, and navigation buttons. A deck is a single physical file that contains one or more related cards.

The card metaphor and the way WML handles action buttons and navigation forces you to design your pages with the restrictions of small mobile devices in mind. This means your designs will naturally be simple, small, and efficient.

When Should You Use It?

Simply put, WAP/WML is your best bet if you need to support cellular phones and other small format devices. It is the most efficient protocol for those devices, and it forces you to design your pages so they will be appropriate for this type of device.

However, when targeting advanced devices like Palm Pilots, Pocket PCs, and sub-notebooks, you are probably better off using HTML. It is simply more powerful and more flexible. This means that you should probably use both WML and HTML if you want to support the widest possible range of wireless devices.

A WML Example

The example in Exhibit 3 is a very simple WML deck that displays a list of cities to pick from and displays the respective weather information. When you select a city, the browser navigates to the corresponding card. This is obviously oversimplified: a real application would retrieve the weather information from a database or Web service. If you want to try this sample, you should obtain the free WML SDK and a WAP phone emulator from http://developer.phone.com.

As you can see this WML file, called a deck, is a well-formed XML document with a DOCTYPE set to the WML 1.1 DTD. The root tag for this document is simply <wml>. Then we create the first card inside using a <card> element. The content of that card is very close to HTML. In this case, we simply display a list control (<SELECT> tag), with four items in it. Each item points to another card using the syntax onpick = "#cardname." This causes the browser to navigate to the corresponding card when the user picks an item from the list. The rest of the deck creates four more cards that simply display the weather details for each city. A more efficient way to do this would be to use one card and some script to dynamically generate the weather data. One thing to note is that each information card implements a Back button, which is usually handled as a "soft" button by the device itself. Exhibit 4 shows what this display looks like on a WAP phone emulator.

WMLScript

The goal of WMLScript is to reduce the requirement for round trips back to the server. These scripts can be used to support advanced User Interface functions, add intelligence to the client, do input validation, provide access to the device's hardware and its peripheral functionality, and reduce the amount of bandwidth needed to send data between the server and the client. WMLScript closely resembles JavaScript, so Web developers can learn it with a minimum of effort.

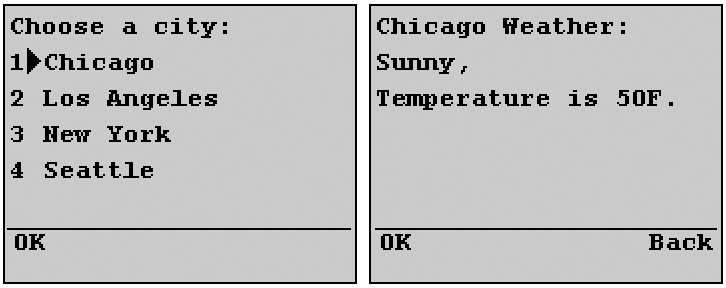
The WMLScript (calctax.wmls) shown in Exhibit 5 implements a simple tax calculation function.

This WMLS file starts by declaring an *extern* function. This is required to make the function available to the browser. Inside this function, we start by declaring three work variables. We then use the *getVar* function from the *WMLBrowser* library to retrieve the amount and tax rate values from the

Exhibit 3. Simple WML Deck

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
   "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
   <card id="selection" title="WML Example">
      <p>Choose a city:</p>
      <p>
      <select>
         <option onpick="#city1">Chicago</option>
         <option onpick="#city2">Los Angeles</option>
         <option onpick="#city3">New York</option>
         <option onpick="#city4">Seattle</option>
      </select>
      </p>
   </card>
   <card id="city1" title="Chicago">
      <do type="options" label="Back">
         <prev/>
      </do>
      <p>Chicago Weather:</p>
      <p/>
      <p>Sunny,</p>
      <p>Temperature is 50F.</p>
   </card>
   <card id="city2" title="Los Angeles">
      <do type="options" label="Back">
         <prev/>
      </do>
      <p>Los Angeles Weather:</p>
      <p/>
      <p>Sunny,</p>
      <p>Temperature is 76F.</p>
   </card>
   <card id="city3" title="New York">
      <do type="options" label="Back">
         <prev/>
      </do>
      <p>New York Weather:</p>
      <p/>
      <p>Rainy,</p>
      <p>Temperature is 60F.</p>
   </card>
   <card id="city4" title="Seattle">
      <do type="options" label="Back">
         <prev/>
      </do>
      <p>Seattle Weather:</p>
      <p/>
      <p>Cloudy,</p>
      <p>Temperature is 65F.</p>
   </card>
</wml>
```

browser context. (For more information on the available libraries, please consult the WML SDK.) Next, we need to make sure that these values can be interpreted as numbers and are greater than zero. We do this using the *isFloat* function from the *Lang* library. If either one of these tests fail, we simply return from the function with a code of "*invalid.*" We then use the *parseFloat* function to convert these text values into floating point numbers so we can manipulate them. We now perform a simple tax calculation and return the value into the result variable. Before returning this value, we must format it properly for display, which is done by the *format* function



Choose a city:
1▶Chicago
2 Los Angeles
3 New York
4 Seattle

Chicago Weather:
Sunny,
Temperature is 50F.

Exhibit 4. WML Weather Sample on a WAP Phone

Exhibit 5. WMLScript (calctax.wmls)

```
extern function calctax()
{
   // declare variables
   var amount, result, rate;

   // retrieve variables from browser context
   amount = WMLBrowser.getVar("amount");
   rate = WMLBrowser.getVar("rate");

   // validate variables
   if (!Lang.isFloat(amount) || amount <= 0)
   {
      Dialogs.alert("Amount must be a number and greater than zero!");
      return invalid;
   }
   if (!Lang.isFloat(rate) || rate <= 0)
   {
      Dialogs.alert("Rate must be a number and greater than zero!");
      return invalid;
   }

   // convert variables to floating point numbers so we can multiply them
   amount = Lang.parseFloat(amount);
   rate = Lang.parseFloat(rate);

   // calculate and format tax amount
   result = amount * rate / 100;
   result = String.format("$%.2f", result);

   WMLBrowser.setVar("tax", result);
   WMLBrowser.go("taxcalculator.wml#answer");
}
```

from the *String* library. This function works just like the *printf* function in C. Once the value is properly formatted, we set it in the browser's context using *setVar.* The last line returns control to the browser and navigates to the "answer" card which displays the result. As you can see, WMLScript is very similar to JavaScript except for a few idiosyncrasies specific to WML browsers and phone devices.

Exhibit 6 contains the WML deck (taxcalculator.wml) that uses the WMLScript file above. This is a standard WML deck. On entering the first

Exhibit 6. WML Deck (taxcalculator.wml)

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
   <card id="calc" title="Tax Calculator">
      <onevent type="onenterforward">
         <refresh>
            <setvar name="amount" value=""/>
            <setvar name="rate" value="8.0"/>
            <setvar name="tax" value=""/>
         </refresh>
      </onevent>
      <p>
         <big>Tax Calculator</big>
      </p>
      <p>
         <do type="accept" label="Calc">
            <go href="calctax.wmls#calctax()"/>
         </do>
         $(amount)? (Dollar Amount)
      <input type="text" name="amount" value="100" emptyok="false"
format="*N"/>
      </p>
   </card>

   <card id="answer" title="Tax Calculator">
      <do type="accept" label="new">
         <go href="#calc">
            <setvar name="amount" value=""/>
            <setvar name="tax" value=""/>
         </go>
      </do>
      <p>
         <big>Tax Calculator</big>
      </p>
      <p>Total tax amount is: $(tax) </p>
   </card>
</wml>
```

card, we use the <refresh> tag to initialize the three variables. The card then prompts the user for a dollar amount using an <input> field. The <do> tag sets up a “Calc” action button of type “accept” which will take the input value and call the *calctax* function in the *cacltax.wmls* file. At this point control is passed to the script until it returns to the “answer” card. This card simply displays the result and implements a “new” button that allows you to reset the values and return to the first card.

HDML

HDML stands for Handheld Device Markup Language. HDML is a North America-only standard that was developed by Phone.com. While HDML was widely accepted by North American phone companies, it is being rapidly supplanted by WML. So our recommendation is that you must at least support WML, even if you choose to support HDML; otherwise you will needlessly cut the potential size of your market. One nice advantage of HDML is that it is actually very close to HTML and includes some concepts shared with WML (like card decks), making the transition very easy.

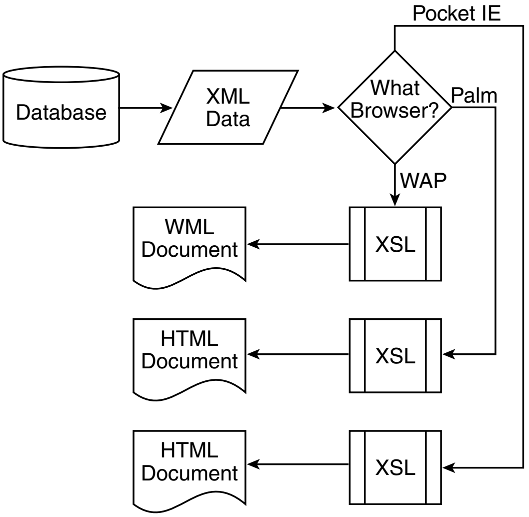


Exhibit 7. Publishing to Multiple Classes of Mobile Devices

Because of the facts noted above, and the limited amount of space we have for this text, we will not discuss HDML any further; we are confident that you can utilize the knowledge you gained in the HTML and WML sections to get started with HDML.

USING XML AND XSL TO TARGET MULTIPLE DEVICES

One problem with the previous scenario is that you would need to create a different ASP page for each class of devices you are targeting (sub-notebook, handheld PCs, phones). A more flexible approach would be to extract the data as XML, and use an XSL template to generate an HTML and/or WML page for each type of device you target. Exhibit 7 describes the process.

Explaining the intricacies of XML and XSL is beyond the scope of this chapter, but here is a simple example that uses an XSL style sheet to transform an XML document into a WML deck.

Exhibit 8 shows the source XML file, which would normally be generated from a database. It consists of a simple list of user names.

This XSL style sheet first outputs the <wml> and <card> elements, and the <do> action for the card, along with a table header. The <xsl:for-each select = “User”> then loops through the XML file and outputs a table row for each <User> element it. The WML output after transforming the XML

Exhibit 8. Source XML File with Simple List of User Names

```
<?xml version="1.0"?>
<UserList>
   <User Name="User1"/>
   <User Name="User2"/>
</UserList>
```

XSL file used to transform the document into WML

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:output
      method="xml"
      indent="yes"
      doctype-public="-//WAPFORUM//DTD WML 1.1//EN"
      doctype-system="http://www.wapforum.org/DTD/wml_1.1.xml"/>
   <xsl:template match="/UserList">
   <wml>
      <xsl:apply-templates/>
      <card>
         <do type="options" label="Back">
            <prev/>
         </do>
         <p>
            User List:<br/>
            <table columns="1">
            <tr>
               <td><i>Name</i></td>
            </tr>
         <xsl:for-each select="User">
            <tr>
               <td><xsl:value-of select="@Name"/></td>
            </tr>
         </xsl:for-each>
            </table>
         </p>
      </card>
   </wml>
   </xsl:template>
</xsl:stylesheet>
```

source document through this XSL style sheet looks like the one in Exhibit 9.

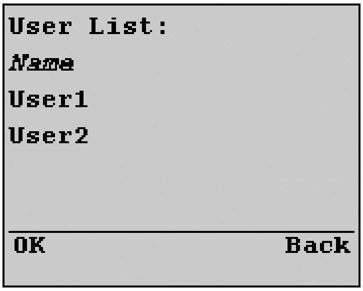
Exhibit 10 shows what the WML page would look like as displayed by a WAP phone emulator.

USING MICROSOFT'S MOBILE SDK FOR ASP.NET

As I write this (April 2001), Microsoft is working on a new technology called the .NET Mobile Web Software Development Kit (MW-SDK). This SDK extends ASP.NET and the .NET Framework to enable you to easily build mobile Web applications for cellular phones and Pocket PCs. The MW-SDK includes the Mobile Web Forms runtime and a set of mobile server controls that generate WML 1.1 and HTML 3.2. Consequently, you can create a single mobile ASP.NET page that intelligently formats content for different types of devices. The following devices are supported in the Beta 1 version, but many more are promised for Beta 2:

Exhibit 9. WML Output after Transforming XML Source Document through XSL Style Sheet

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
   <card id="users" title="User List">
      <do type="options" label="Back">
         <prev/>
      </do>
      <p>
         User List:<br/>
         <table columns="1">
            <tr>
               <td><i>Name</i></td>
            </tr>
            <tr>
               <td>User1</td>
            </tr>
            <tr>
               <td>User2</td>
            </tr>
         </table>
      </p>
   </card>
</wml>
```



User List:
Name
User1
User2

Exhibit 10. Simple WML Output from an XSL Transform

- Pocket PC with Microsoft Pocket Internet Explorer version 4.5
- Mitsubishi T250 phone
- Nokia 7110 phone
- Nokia WAP Toolkit 2.0 Beta simulator for the Nokia 7110
- Phone.com UP.SDK 3.2 for WML simulator with Ericsson R280LX skin
- Microsoft Internet Explorer 5.5
- Samsung Touchpoint phone
- Sony CMD-z5 phone
- Sprint Touchpoint phone
- Microsoft Mobile ExplorerT v 2.01 simulator
- Phone.com UP.SDK 4.0 simulator with the generic skin
- Phone.com UP.SDK 3.2 for WML simulator with Mitsubishi T250 skin

The Mobile Web SDK does not use XSL to transform the pages. Instead, it has an extensible control adapter-based rendering layer. Starting in Beta 2, developers will be able to write their own adapters to support devices not supported out-of-the-box, or to modify the existing ones.

What Are the Pros and Cons?

The big advantage of the MW-SDK is that it allows you to write a single set of ASP.NET pages and transparently support a large number of devices. This could save you a tremendous amount of work. In addition, the MW-SDK allows you to work with high-level controls like lists, tables, and even a calendar.

However, as of Beta 1, the MW-SDK is not integrated into the Visual Studio .NET development environment. This means that you cannot create pages in a visual development tool — you have to use a text editor like Notepad. Microsoft promises that as of the release of Visual Studio .NET Beta 2 and the MW-SDK Beta 2, the two will be fully integrated and new releases will be synchronized.

A .NET Mobile Web SDK Example

The MW-SDK example shown in Exhibit 11, written in C#, demonstrates the power and simplicity of this technology. It will display a simple pick-list of cities from which you could retrieve weather reports. It uses an array of objects to hold the city names and then it binds this array to the list control.

The first line imports the .NET class library for the Mobile UI controls, while the second line declares a namespace prefix that refers to the System.Mobile.UI namespace. This is done to save typing in the rest of the code.

Next we open a script block and declare that this script is to run on the server side and uses the C# language. This script block consists mostly of a class called WeatherCity. This class contains a public field, WeatherCity, along with two properties, CityName and Report, and it is simply used to hold the data in each element of the data array. Following that, you will find two event handlers. The first is Page_Load, which creates the array and binds it to the MenuList control every time this page is loaded. Each array element is assigned a new instance of the WeatherCity class to hold the two data fields. The second one is Menu_OnItemCommand, which runs every time the user selects an item in the list and switches to the appropriate weather report page.

The few lines following the script block show the real power of the MW-SDK. These eight lines of XML create two forms. FirstForm displays the pick list and attaches the Menu_OnItemCommand event to it. SecondForm

Exhibit 11. MW-SDK Example

```
<%@ Page Inherits="System.Mobile.UI.MobilePage" Language="C#" %>
<%@ Register TagPrefix="Mobile" Namespace="System.Mobile.UI" %>

<script runat="server" language="c#">
private class WeatherCity
{
   private String cityName, report;
   public WeatherCity(String cityName, String report)
   {
      this.cityName = cityName;
      this.report = report;
   }

   public String CityName
   {
      get
      {
         return this.cityName;
      }
   }
   public String Report
   {
      get
      {
         return this.report;
      }
   }
}

protected void Page_Load(Object sender, EventArgs e)
{
   if (!IsPostBack)
   {
      ArrayList array = new ArrayList();
      array.Add(new WeatherCity ("Chicago", "Sunny, 52F"));
      array.Add(new WeatherCity ("Los Angeles", "Sunny, 75F"));
      array.Add(new WeatherCity ("New York", "Rainy, 58F"));
      array.Add(new WeatherCity ("San Francisco", "Cloudy, 70F"));
      array.Add(new WeatherCity ("Seattle", "Cloudy, 65F"));
      MenuList.DataSource = array;
      MenuList.DataBind ();
   }
}

protected void Menu_OnItemCommand(Object sender, ListCommandEventArgs e)
{
   CityLabel.Text = "Weather for " + e.ListItem.Text;
   WeatherLabel.Text = e.ListItem.Value;
   ActiveForm = SecondForm;
}
</script>

<Mobile:Form id="FirstForm" runat="server">
   <Mobile:Label runat="server" StyleReference="title">City
Menu</Mobile:Label>
   <Mobile:List runat="server" id="MenuList"
OnItemCommand="Menu_OnItemCommand" DataTextField="CityName"
DataValueField="Report"/>
</Mobile:Form>

<Mobile:Form id="SecondForm" runat="server">
   <Mobile:Label runat="server" id="CityLabel" StyleReference="title" />
   <Mobile:Label runat="server" id="WeatherLabel" />
</Mobile:Form>
```

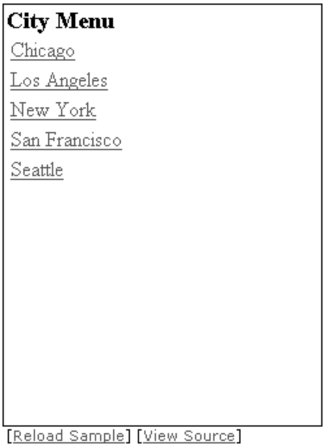


Exhibit 12. MW-SDK List Control as Rendered on a Pocket PC

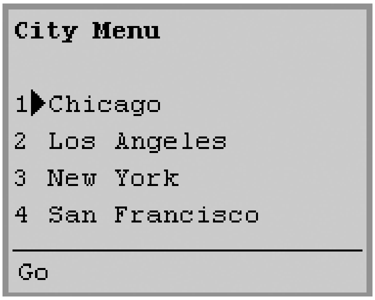


Exhibit 13. MW-SDK List Control as Rendered on a WAP Phone

simply displays a couple of text labels. That is all you need to do to display these forms on all the mobile devices supported by the SDK.

The output from this ASPX file would look something like Exhibit 12 on a Pocket PC and like Exhibit 13 on a WAP phone.

SUMMARY

Although we just skimmed the surface, we have covered a lot of wireless ground. You should now have a good general understanding of what the wireless Web is all about, and when it does or does not make sense to use for your applications. You should also understand what technologies are available for deploying data out to wireless devices, what each technology's strength and weaknesses are, and how they can be used.

To carry on from here, we would strongly encourage you to go out and try these technologies for yourself by building some simple prototypes. There is nothing like getting your hands dirty when you want to learn new technologies.

ABOUT THE AUTHOR

Sylvain Duford is Chief Architect at Cactus Internet. He has 12 years of experience in the software industry and specializes in designing eCommerce B2B systems using the Microsoft .NET platform. In the last five years, Sylvain has designed systems for customers like the Boeing Aircraft Company, Microsoft Corporation, Onyx Software, the Washington State Government, and the Federal Government of Canada. Sylvain is also a seasoned public speaker and technical writer.

Chapter 17
Putting Data in the Palm of Your Hand

Alex Lee

The proliferation of mobile computing in everyday life has made words like Palm and Blackberry commonplace among an increasingly tech-savvy population. No modern-day student would dare make a date without first checking her PDA and many executives spend their morning commute reading documents on their PocketPC.

Even as little as five years ago, the mobile computing landscape was much different. U.S. Robotics, a company known mostly for modems, was just introducing the original Palm Pilot devices. Critics were skeptical whether the product would succeed, given its small monochrome screen, awkward pen-stylus interface, proprietary operating system, and high cost for what was essentially billed as a "pocket organizer." But the skeptics were quickly silenced as users embraced the novelty of a computer in the palms of their hands.

Part of the challenge for the original Palm Pilot was providing users the ability to access or exchange data with their existing systems. This was achieved by allowing users to *synchronize* data with a local PC via a serial connection. The synchronization metaphor, which was already widely used in products like Lotus Notes, was remarkably successful, in part, due to its simplicity and effectiveness. Despite the limitations, such as the need to be physically connected to the PC and the potential for replication conflicts, most users were quite satisfied with the compromise of "syncing" with their PC to ensure their data was up-to-date. The open architecture of the Palm OS also allowed developers to create their own synchronization *conduits*, making it possible to synchronize almost any type of data.

Naturally, the next step in the evolution of mobile computing will be even broader data access capabilities, since it is difficult to have worthwhile mobile applications without some sort of real-time data access. As such, armies of solution vendors are battling for mobile data supremacy.

0-8493-1161-6/02/$0.00+$1.50

But, of course, there are many issues to consider. What devices to support? How fast? What kind of data source? How much data? How many users? Perhaps more so than with other information technology (IT), the success of implementing wireless data access is dependent on thoughtfully planned objectives and well-defined expectations. This is, in part, due to the inherent risks and high costs required for any sort of large-scale deployment of wireless data and devices.

In order to manage objectives and expectations, it may be prudent to begin by reviewing the advantages and disadvantages of wireless computing and the current infrastructure available to consumers. First, wireless computing allows increased productivity and convenience for people who normally cannot be constantly online through conventional means. For most organizations, constant connectivity means staff can improve customer service immensely. Also, with constant access, regardless of location, a company and staff can interact which helps streamline the workflow of data.

However, there are some limitations with the wireless approach. Current wireless networks are relatively slow and unreliable. Most wireless applications feature moderately difficult-to-use interfaces. For developers, numerous competing standards and platforms increase the complexities of creating applications. As well, for those who host their own internal applications, wireless applications mean additional IT infrastructure and support costs.

In terms of handheld hardware, the options are diverse. Clearly, the dominance of Palm devices (i.e., based on the Palm OS) makes them the primary contenders. These mobile computers, based on the architecture of the original Palm Pilot from U.S. Robotics (later to be purchased by 3COM and then spun off into its own company) can be found marketed under various brand names, including Palm, Handspring Visor, IBM WorkPad, and Sony CLIÉ. Most models offer long battery life, and feature 2 to 8 MB of memory and a touch stylus for data input. With regard to wireless connectivity, the Palm VII comes with a built-in wireless modem whereas the Palm V has several add-on options. Their popularity has spawned a wealth of development tools that make it relatively easy to develop custom applications.

Their primary competitors, PocketPCs, also continue to gain popularity. These devices, based on Microsoft's Windows CE operating system, tend to be more popular with business users who require compatibility with their desktop environment. PocketPCs are praised for their high-resolution color screens, Windows-like user interface, and PC compatibility. On the downside, they are more expensive, prone to shorter battery life, and applications tend to be more memory intensive. Development tools are readily available, since the platform is largely based on Windows. The most popular PocketPC models are Compaq's iPaq series and the HP Jornada.

Research In Motion's (RIM) Blackberry devices are becoming more ubiquitous. These pager-like units have rapidly garnered popularity due to their small size and real-time messaging capabilities. Using a miniature QWERTY keyboard, users can access e-mail from practically anywhere. As well, Blackberry devices are also equipped with standard Personal Information Management (PIM) functions such as an address book, calendar, and calculator. RIM provides a Blackberry SDK for developers.

In addition to the above, almost all personal communication system (PCS) phone providers now offer some sort of mobile Internet access. Essentially, this means riding a data stream along the digital signal normally used for PCS voice calls. A mini-browser is used on the phone to access Internet sites.

It is important to remember that your choice of hardware is really limited to what is supported by your mobile service provider. For example, Palm.Net is oriented primarily to Palm users. OmniSky offers support for both Palm and PocketPC PDAs. Go.Web can support Palm devices and Blackberry devices. AT&T and Sprint offer mobile Internet services through their PCS phones. In some cases, it is even possible to mix and match solutions. With the appropriate data cable, a Palm (or a laptop for that matter) can be connected to a PCS cell phone and use the phone's digital network as its Internet connection.

In terms of network bandwidth, the array of options is wide and growing. Most current wireless access depends on the infrastructure provided by existing second generation (2G) PCS networks. TDMA (Time Division Multiple Access), CDMA (Code Division Multiple Access), and GSM (Global System for Mobile) networks can provide data transfer rates varying between 9.6 and 19.2 Kbps, depending on a variety of factors. Some newer proprietary networks, like Metricom's Ricochet, can provide rates up to 128 Kbps. Third-generation (3G) PCS networks, which will start to be seen over the next five years, are promising data transfer rates up to 2 Mbps. While rates seem meager when compared to desktop standards, one should not be overly concerned with the current lack of bandwidth. Remember that most wireless applications are meant for convenience rather than stunning graphics or blazing speed. In the majority of cases, users must be willing to forgo the "bells and whistles" for the benefit of information, literally, at their fingertips.

Even in the current wireless infancy, the architecture of your wireless data access can vary greatly depending on your requirements and your current infrastructure. For those with existing Web-based applications, creating a WAP (Wireless Application Protocol) version of your site would likely make a lot of sense. Essentially, this means creating an interface to your existing back-end using one of the many wireless standards, such as HDML (Handheld Device Markup Language) or WML (Wireless Markup Language). By doing this, the content of the site is stripped to its essentials (usually text and simple graphics), which make it reasonable to transmit over smaller bandwidths. Mobile clients can access the site using a WAP browser which is available for almost all mobile platforms. If you have been clever enough to implement your site using XML (Extensible Markup Language; recall what was mentioned earlier about thoughtfully planned objectives), then the WAP approach makes even more sense, since the idea behind XML is the ability to provide different facades to a single source of content.

WAP applications require a mobile application server to manage the communication between the wireless device and back-end data source (see Exhibit 1). The mobile application server also *transcodes* the output from the application or data source into the proper format for your particular wireless device. While many developers will host their own application server on their premises, others may opt for the convenience of a Wireless Application Service Provider (WASP). Using a WASP reduces some of the initial costs while also avoiding some of the trouble of setting up the mobile application server.

Keep in mind that if your existing application is Web-based it is not absolutely necessary to create a WAP version of the site. For example, using a PocketPC equipped with Pocket Internet Explorer, it is possible to access an existing Web site, as is. However, screen limitations and relatively slow connection speeds usually do not make this a viable option. Also, this limits the solution to those mobile devices that can access standard Web sites (mostly PocketPCs and Palms).

Another way to WAP-enable your Web-based applications is to use one of the many emerging tools designed to adapt existing Web sites and applications for mobile platforms. One such tool is Aether Systems' ScoutWeb which offers a simple approach for interfacing existing Web sites to mobile devices. ScoutWeb allows developers to define rules for converting HTML via a simple Web-based interface and by using text files. ScoutWeb's application server runs alongside a company's existing Web or application servers and translates HTML content into wireless content on the fly. It can be hosted in-house or by Aether as the WASP. ScoutWeb can support approximately 30 different devices and developers can customize various settings for specific devices and/or Web pages.

2Roam uses a slightly different approach. Its Nomad publisher allows the developer to define elements of the Web content for mobile devices. Its Catalyst application server then publishes the content as defined by Nomad for the particular device making the request. The major advantage

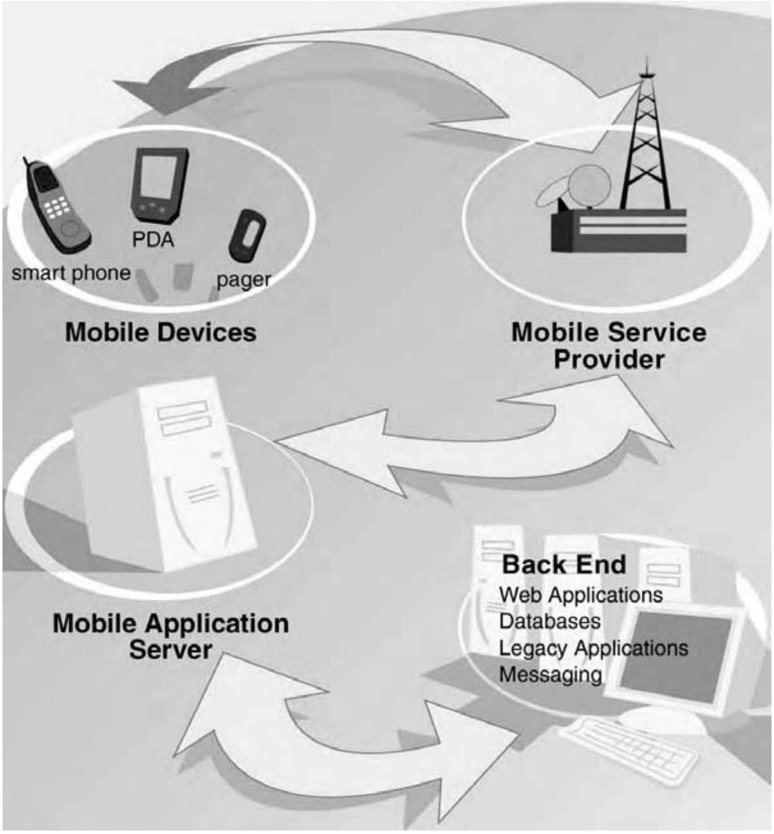


Exhibit 1. Wireless Application Model

of this method is that it minimizes the effort required to add additional device support. In addition to this, 2Roam offers several special features including support for secure transactions, SMS messaging (Short Message Service), alerts, and JavaScript, even for devices that do not support it. JavaScript support can be a significant issue, since many sites use it for user interaction and validation. Like Aether, 2Roam can host the wireless application or companies can install their own server in-house.

If your data does not have the benefit of a Web-based interface, then an explicit database tool is a better solution. The options for wireless database access are increasing every day. For the most part, it is possible to have wireless access to any major database. Currently, some of the more popular tools include Sybase SQL iAnywhere, IBM DB2 Everyplace, and Oracle's 9i Application Server Wireless Edition. Microsoft has also contributed to the field by recently introducing its Mobile Information Server and

SQL Server Lite products. Most of these products support several different handheld platforms and offer both the database component and application server. Hence, the entire wireless database solution can come from a single vendor.

One of the distinguishing features of these database solutions is the minimal size of the memory footprint required. For example, SQL iAnywhere is particularly popular because of its ability to produce databases as small as 50K. Similarly, IBM's DB2 Everyplace only requires 150K. On the contrary, Microsoft SQL Server CE (designed for Windows CE) has a 1-MB memory footprint, illustrating the additional overhead required by PocketPC applications.

For application development, many might prefer to do it the old-fashioned way and code their own solutions from scratch. Even here, the options are remarkably varied. Microsoft's eMbedded Visual Tools 3.0 offers C++ and Visual Basic developers an opportunity to exploit the Windows CE operating system (used on PocketPCs). Metrowerk's popular CodeWarrior tool provides a full-featured C++ development environment for the Palm OS (and other embedded systems). Other tools support many different languages, including Perl/CGI, Java, and Active Server Pages (ASP). In the end, the current options available for developers provide limitless options, although it is obvious that some tools offer several advantages over others. As such, it is important to choose a tool that addresses your specific requirements.

If you are interested in trying wireless data access, there are several easy ways to get started. First, and most obvious, a wireless device is required. If one is unavailable, there are several good emulators/simulators available which can be used by developers. For example, Openwave, which promotes the UP.Browser used in most WAP-enabled devices (like PCS phones), provides the UP.Simulator as part of its UP.SDK (Software Developer Kit). The UP.Simulator provides emulation of the UP.Browser on the Windows desktop platform to test WML and HDML sites. It can even emulate the look of many PCS phones, through the use of *skins*. This makes it simple for developers to see how their applications will run across several different devices. Palm also provides the Palm Operating System Emulator (POSE) which can run on several different operating systems (Windows, Mac OS, and Unix). It, too, has different skins to emulate the various Palm devices.

As far as development tools are concerned, one can download trial versions of most of the tools mentioned above from the respective vendors. Sybase offers most of its products for evaluation at www.sybase.com. Oracle and IBM are similarly generous. Using these evaluation tools, it is possible to get a very good idea of what it is like to develop data applications for wireless platforms.

It is also suggested that developers visit some existing wireless sites or experiment with existing wireless applications. You can find a portal to WAP sites at www.wap.com. After viewing a few sites, one quickly understands the user interface (UI) challenges for wireless applications, which can possibly mean the success or failure of an application. Limited screen real estate and data input capabilities make sophisticated interfaces unrealistic. Some wireless applications resemble something more akin to *circa* 1985 DOS programs. As more applications emerge, standardization should help developers in creating usable and consistent interfaces.

In fact, standardization will really be the key consideration in the ultimate acceptance of wireless applications. As such, it is prudent for developers to research any standards or guidelines promoted for the target platform. For example, Openwave provides usability guidelines for WAP applications. Similar guidelines can be found for other systems and protocols. By embracing standards, usability and consistency increase among all applications and benefit everyone.

One such standard, which currently is garnering a lot of recent discussion, is Bluetooth, the name given to a new protocol using short-range radio links. Bluetooth was designed to replace physical cables for all devices (portable and/or fixed). Compared to current standards, Bluetooth features robustness, simplicity, minimal power requirements, and low cost. The technology can potentially offer wireless access to LANs (Local Area Networks), PSTN (Public Switched Telephone Networks), the mobile phone network, and the Internet for home appliances and mobile handheld devices. In essence, Bluetooth, in a wireless sense, tries to be all things to all people.

However, recent developments have raised issues with Bluetooth, bringing into question its ultimate acceptance. After the initial hype more than a year ago, products which implement the technology have been scarce and relatively expensive. Other issues, such as disagreement over protocol specifications and concern regarding interference, also have hampered Bluetooth development efforts. Microsoft's recent decision to exclude native support for Bluetooth in its new Windows XP operating system may also reflect a growing lack of support from industry leaders. Thus, it will most likely be a few years until it is known if Bluetooth really delivers on its promise.

Clearly, this discussion only scratches the surface of mobile data access. There are numerous issues and emerging technology that are likely to change the mobile computing landscape significantly over the next five years. Hopefully, wireless technology will continue to mature and competing protocols will converge eventually into industry standards. The current wireless environment requires everyone (consumers and developers alike) to sort through the array of hardware and network options. Ultimately, the success and value of wireless data access depend on how quickly organizations can implement a productive system from the current myriad terminology.

ABOUT THE AUTHOR

Alex Lee is a co-founder and president of QUEUE Systems Inc. (www. QUEUESystems.net). Alex graduated from the University of Waterloo with a degree in systems design engineering. Established in 1989, QUEUE System is a multi-disciplinary consulting firm that provides complementary resources in its IT Consulting, Placement Agency, and New Media divisions.

Chapter 18
Programming Wireless Applications

Gilbert Held

Over the past few years, the global use of the Internet has risen at an exponential rate. Today, there are millions of servers providing billions of Web pages that are used to support electronic commerce, research, marketing, and other activities of organizations, as well as individual users. Accompanying the growth in the use of the Internet is a substantial increase in the use of wireless communications. Over the past few years, the population of cell phone subscribers has also grown exponentially.

Until recently, the use of cell phones to access data on the Internet was based on proprietary technology that had limited appeal for developers. During 1999, the Wireless Application Protocol (WAP) specification was announced. This protocol provides a standardized method for accessing and retrieving Web pages via cellular telephone. WAP includes several components, one of which is referred to as Wireless Markup Language (WML). WML can be viewed as an equivalent but not identical version of the Hypertext Markup Language (HTML) used to construct conventional Web pages that are viewed when surfing the Internet with a desktop or laptop computer.

Due to the limited display area on a cell phone, it would not be feasible to attempt to display conventional Web pages created via the use of HTTP. Recognizing the limitation associated with the cell phone display resulted in the development of the WML specification. This specification is designed to facilitate the display of Web pages, now referred to as cards. In this chapter, attention focuses on wireless application development in the form of obtaining a brief overview of the operation of the WML specification. A discussion and description of WML are used, creating an example of the specification. This chapter concludes with an examination of two

0-8493-1161-6/02/$0.00+$1.50

methods that can be considered to provide a Web site capable of supporting both wireless and non-wireless surfers.

WML OVERVIEW

Wireless Markup Language (WML) represents the language used for creating WAP documents. Assuming one has a degree of familiarity with the screen display capability of a conventional monitor, an LCD active matrix display built into most modern notebook computers, and the miniature LCD display contained in wireless phones, it is obvious that the amount of space available for displaying Web pages on a wireless phone is minimal in comparison to the other displays. Another significant difference between desktop and notebook computers and cell phones is the memory available for storage in each device. While computer memory is normally referenced in terms of megabytes of storage, more often than not one never even considers the amount of storage capability of a cellular phone. If one did, then one would note that most cellular phones at the maximum have available storage in terms of a few kilobytes of memory. This represents a small fraction of the amount of memory available with most laptop, notebook, and desktop computers.

The reduction in both display size and available memory means that the WAP protocol at best will turn a mobile telephone into a miniature Internet browser. Due to display and memory constraints associated with wireless phones, WML does not support graphics in the conventional sense. That is, one cannot add a .gif or .jpeg image to a page, and have that image displayed on a subscriber's cell phone. There is simply not enough memory in the current generation of cell phones to store such images.

In addition, the display capability of cell phones is currently based on the generation of pixels to produce text characters. This is another reason why it might be years, if ever, before one is able to view a streamlined version of a page filled with graphics such as one usually expects when surfing the Internet.

On the positive side, the elimination of graphics considerably facilitates the flow of data. For those who do not have a WAP-enabled cellular phone and are from Missouri, the "Show Me State," the difference in transmission time for a page with graphics removed is easily noted. To do this, first point the Web browser to any popular Web site, such as Amazon.com, Ford.com, or even Opm.gov. Time the period required to download the Web page until the activity light stops or the term "Done" appears at the bottom of the screen. Once this is accomplished, go to the browser's preference menu and select the appropriate entry to disable graphics.

Next, click on the page reload option and watch the display literally fly, since no graphics are involved in the download.

Thus, while WML will not provide the interesting graphics many admire or detest, the lack of transmission-intensive graphics support within WML is not necessarily bad. Because a general appreciation for WML has been established, attention can focus on some specifics of WML, including an examination of the use of some statements to create a WML document.

WML LANGUAGE

WML, like most languages, has its own set of terms and abbreviations used to facilitate a description of the language. Under WML, a document is referred to as a deck. A deck consists of one or more cards that can be linked to one another or to another deck in the same manner that one Web page can be linked to another location in the same page or to another Web page.

The rationale for the use of the terms "card" and "deck" is based on the fact that, unlike a conventional display, WAP-compatible devices can only display one card at a time. Although it may never have been tried, if one shrinks the size of the browser on a laptop, notebook, or desktop, and re-executes the browser icon, again one obtains the ability to view two Web pages simultaneously. While this author periodically has enough problems designing and viewing a single Web page at one time, the memory, display capability, and processing power of modern computers allow multiple pages in the form of multiple windows to be simultaneously displayed. The smaller screen display, small amount of memory, and relatively low processing capability of WAP-compatible devices preclude this capability from being considered for a WAP-compatible device.

A small deck consisting of related cards can be downloaded into a WAP-compatible device when a user requests a card in a deck. The user may then navigate among the various cards in a deck without requesting a new document from the WAP server. This minimizes required transmission from a mobile user whose data transmission rate is normally a small fraction of the transmission rate obtainable when conventional modems are used to connect to the Internet.

DOCUMENT CREATION

WML represents a variation of Extensible Markup Language (XML) which, while similar to HTML, is more stringent in its requirements concerning the use of closing tags and the use of quotes around attribute values. As a refresher, XML supports the tagging of information in a document that describes what the content represents, as well as identifies information contextually. In comparison, HTML, which is used for conventional Web page development, can only describe how to display content, but not what the content represents. While this difference may appear trivial, in actuality there is a considerable difference between the two. Under XML, the identification of information contextually permits queries to retrieve only relevant files, and makes online searches more efficient. In addition, because each piece of content is tagged, it becomes possible for a delivery system to pick out information from a database, and then repackage the information to dynamically construct multiple pages.

Recognizing the fact that the WAP devices have limited amounts of read-only memory (ROM) and random access memory (RAM), WML also requires a more pronounced data structure than HTML. The beginning of every WML document contains two lines of coding that identify the document as an XML document; and note that the document adheres to the WML standard. The following lines illustrate the code one should find at the beginning of a WML document:

```
<?xml version ‘1.0’?>
<!DOCTYPE wml PUBLIC “-//WAPFORUM// DTD WML 1.1//EN”
“http://www.wapforum.org./DTD/wml_1.1.xml”>
```

Note that the first line simply states that XML version 1.0 is being followed. The second line, which because of space constraints is continued as a third line, defines the document as a WML document and references the Web site of the WAP forum.

As previously mentioned, each WML document is called a deck, and a deck consists of one or more cards. The code in Exhibit 1 provides a simple example of a two-card WML deck. For simplicity, the two lines that are normally included at the beginning of every WML document are omitted from the listing shown in Exhibit 1.

If familiar with HTML, one will probably be familiar with the general code shown in Exhibit 1 without knowing WML code. For example, <wml> and </wml> represent the beginning and ending tags that define the WML document. Similarly, if one has knowledge of HTML, one would note that <p> and </p> tag pairs define a paragraph, and their use is carried over to WML. Some other elements within Exhibit 1 require a bit of elaboration. For example, the “id” element is required to provide the ability for the operator of a miniature browser system to navigate through various cards that make up a WML deck. The “do” element defines a menu of options that, in most cases, represent navigational tasks.

If it is assumed that a mobile cellular subscriber’s signal can be triangulated, then whenever that subscriber turns on the phone, his or her location can be determined. This makes it possible for a new class of Internet applications to be developed, one of which is explained next.

```
<wml>
<card id = “card 1” title = “Gotta Get Gas”>
    <p>
        <do type = “accept” label = “go to card 2”>
            <go href = “#card2”/>
        </do>
        Exxon 2 miles on RT.
    </p>
</card>
<card id = “card2” title = “Gotta Get Gas”>
    <p>
        Shell 3 miles on RT.
    </p>
</card>
</wml>
```

Exhibit 1. WML Two-Card Deck Coding Example

```
-- Gotta Get Gas --

Exxon 2 miles on RT.
```

Exhibit 2. The Display of the First Card Listed in Exhibit 1

Although probably not realized, the WML listing shown in Exhibit 1 could represent a portion of a dynamic gas location service for the mobile traveler. If a traveler is running out of gas, he or she could dial a special mobile number that would result in the vehicle’s position being noted. This information is relayed to a WAP server, which results in the retrieval and transmission of an applicable WML document based on the position of the subscriber.

Exhibit 2 illustrates the display of the first card from the WML example shown in Exhibit 1. Note that the title element in the listing results in the display of the title information contained in quotes across the top horizontal position of a wireless display, with underscore bars used to block each edge of the title to the corner of the display.

In examining the resulting display shown in Exhibit 2, note that “right” is purposely abbreviated as “RT.” Currently, the display capability of WAP-compatible cellular phones varies due to incorporation of WAP into some

-- Options --

Press > Next Station

Press < Prior Station

Press * Main Menu

Exhibit 3. Potential Options Display

older phones via a ROM upgrade. This action means that any WAP developer must carefully consider the minimum amount of cellular display capability available on different displays, and, when necessary, use abbreviations to ensure that information applicable to a card will fit on each type of display that could be used to access a WML document.

Concentrating once again on the WML example in Exhibit 1, for simplicity's sake, navigation statements in the listing were excluded. In an actual document, one would include one or more "go" elements that provide the mechanism to "flip" from one card to another. In addition, one would use applicable WML elements to input data from the subscriber's keypad, and display different options.

Because many WAP-compatible cellular phones will not include a full alphanumeric keypad, one cannot realistically expect users to enter multiple keystrokes to define letters, especially in a mobile environment. This means that many WML applications will be developed to allow a phone operator simply to press a numeric key, or perhaps even one or more of the four cursor keys popular on many cell phones, in order to respond to prompts. This cellular-browser interaction recognizes that for most Web surfing applications that occur via a cell phone, the subscriber will have a limited response menu available for use. Thus, simply flipping from one card to another through a deck, a poor design practice in a conventional Web application, would represent a suitable design for the cellular environment.

Exhibit 3 illustrates a potential display which informs the subscriber of some options available from the gas station example.

Note that the options display would more than likely be shown at the bottom of a card instead of as a separate card as shown in Exhibit 3.

Although the sample code previously listed in Exhibit 1 was for a two-card deck, assume for a moment that one has three cards, with the second providing navigation options back to the first, or forward to the next gas station. Then, if the subscriber pressed the greater than (>) symbol on his or her phone, the resulting display would appear as shown in Exhibit 4.

-- Gotta Get Gas --

Shell 3 miles on RT.

Exhibit 4. Scrolling through the Deck to the Next Card

Having discussed the general characteristics of WML and the constraints of cell phones associated with their use for Web browsing, the conclusion to this chapter focuses attention on the creation of WML servers.

WML SERVERS

There are basically two methods to consider in development of WML servers. Here, the term "WML server" references a WML document on a server and not necessarily a separate Web server dedicated for WML. Therefore, when discussing the creation of a WML server, one actually references the creation of WML documents on a server. With this in mind, there are two methods to consider for making WML documents: directly or via emulation. The creation of a WML document directly means the person uses one or more Web development tools tailored for WML document creation in the same manner that one makes an HTML document. The second method, which involves the use of an emulator, would be applicable if the organization already had a considerable number of Web pages and there was a desire to convert these pages to EML as easily as possible. In this situation, a WML emulator would retrieve HTML pages and, as best as possible, convert each page into a WML document consisting of one or more cards in a deck.

RECOMMENDED COURSE OF ACTION

Currently, WML emulators are in their infancy, because WAP is still in its developmental state. Thus, there is currently no sound foundation for selecting WML development or the use of an emulator. Perhaps the best course of action at the present time is for organizations to literally get their feet wet by creating prototype WML applications. Doing so will allow personnel to become familiarized with the technology, as well as understand the length of time necessary for the development of new applications. This information can be used to determine if the use of a WML emulator is both cost- and time-justified.

ABOUT THE AUTHOR

Gilbert Held, an award-winning author and lecturer, is the author of more than 40 books and 300 technical articles. Some recent titles include *Voice and Data Internetworking* and *Cisco Security Architecture* (co-authored with

Kent Hundley), both published by McGraw-Hill. Other recent titles include *Data Communications Networking Devices,* 4th ed., *Next Generation Modems*, *Ethernet Networks,* 3rd ed., and *Enhancing LAN Performance,* 3rd ed., all published by John Wiley & Sons. He can be reached via e-mail at gil_held@yahoo.com.

Chapter 19 Wireless Communications Data Center: Part I

John R. Vacca

Wireless technologies and systems are fairly new to the data center and are still emerging on the scene. Currently, wireless data center technologies are comprised of infrared, UHF radio, spread spectrum, and microwave radio. These technologies can range from frequencies in the MHz (United States), GHz (Europe), to infrared frequencies. The personal communication network (PCN) can either use code-division multiple access (CDMA) or time-division multiple access (TDMA).[1] There is a considerable controversy among experts in the field regarding the relative merits of spread spectrum (CDMA)[2] and narrow-band (TDMA) for private communication network (PCN). The preferred technique may actually vary with the specific PCN application scenario, and will be addressed later in the chapter.

As the deployment of wireless LANs grows in the data center, there is also a need for higher data rates. As a result, spectrum has been allocated for high- performance LANs (HIPERLAN) and SUPERNET activities at 5GHz, supporting connectivity of 20 to 25 Mbps. Moving to even higher frequencies (40 and 60 GHz) with connectivity of 100Mbps is the subject of current research, although these higher frequencies are more suited to fixed links applications.

Because of the wide range of services supported by Asynchronous Transfer Mode (ATM)[3] networks, ATM technology is expected to become the dominant networking technology for both public infrastructure networks and LANs. ATM infrastructure can support all types of data center services, from time-sensitive voice communications and multimedia conferencing to bursty transaction processing and LAN traffic. Extending the ATM infrastructure with wireless access meets the needs of data center users and customers who want a unified end-to-end networking infrastructure with

0-8493-1161-6/02/$0.00+$1.50

high performance and consistent service. Wireless ATM adds the advantages of mobility to the already great service advantages of ATM networks.

WIRELESS ATM: TECHNOLOGY AND APPLICATIONS

ATM has been advocated as an important technology for the wide area interconnection of heterogeneous networks in the data center. In ATM networks, the data is divided into small, fixed-length units called cells. The cell is 53 bytes. Each cell contains a five-byte header. This header contains identification, control priority, and routing information. The other 48 bytes are the actual data. ATM does not provide any error-detection operations on the user payload, inside the cell, nor does it offer any retransmission services.

ATM switches support two kinds of interfaces: User Network Interface (UNI) and Network Node Interface (NNI). UNI connects ATM end systems (hosts, routers, etc.) to an ATM switch, while an NNI can be imprecisely defined as an interface connection between two ATM switches. The International Telecommunication Union Telecommunication (ITU-T)[4] recommendation requires that an ATM connection be identified with connection identifiers that are assigned for each user connection in the ATM network.

At the UNI, the connection is identified by two values in the cell header: Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI). Both VPI and VCI can combine together to form a virtual circuit identifier. Exhibit 1 shows the UNI and NNI interface to a wireless ATM network.

In any event, there are two fundamental types of ATM connections: Permanent Virtual Connections (PVCs); and, Switched Virtual Connections (SVCs). First, a PVC is a connection set up by some external mechanism, typically network management. In this setup, switches between a source and destination ATM are programmed with the appropriate VPI/VCI values. PVCs always require some manual configuration. On the other hand, an SVC is a connection that is set up automatically through a signaling protocol. SVCs do not require the manual interaction needed to set up PVCs and, as such, are likely to be much more widely used. All higher layer protocols operating over ATM primarily use SVCs.

Reasons for Wireless ATM

From the beginning, the concept of ATM is for end-to-end communications (in a WAN environment). The communication protocol will be the same (ATM), and enterprises will no longer have to buy extra equipment (like routers or gateways) to interconnect their networks. Also, ATM is considered to reduce the complexity of the data center network and improve the flexibility while providing end-to-end consideration of traffic performance. That is why researchers have been pushing for an ATM cell-relay

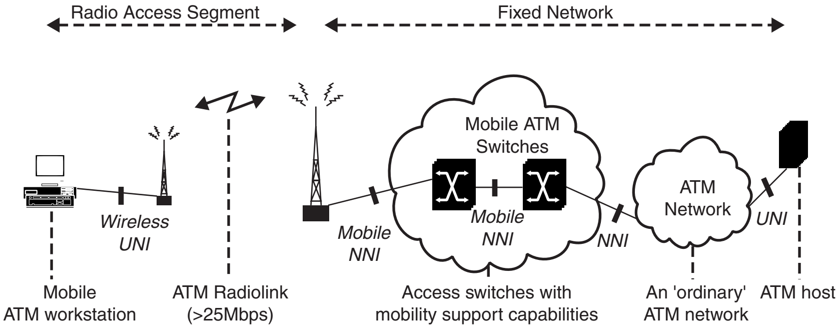


Exhibit 1. Wireless ATM Reference Architecture

paradigm to be adopted as the basis for next-generation wireless transport architectures for the data center.

There are several factors that tend to favor the use of ATM cell transport for a personal communication network. These are:

- Flexible bandwidth allocation and service type selection for a range of applications
- Efficient multiplexing of traffic from bursty data/multimedia sources
- End-to-end provisioning of broadband services over wireless and wired networks
- Suitability of available ATM switching equipment for inter-cell switching
- Improved service reliability with packet switching techniques
- Ease of interfacing with wired B-ISDN systems that will form the telecommunications backbone

In general, interworking can be seen as a solution to achieve wireless access to any popular backbone network, but the consequence, in this case, is a loss of the ATM quality-of-service characteristics and original bearer connections. The more interworking there is in a network, the less harmonized the services provided will be. Therefore, it is important to be able to offer appropriate wireless extension to the ATM network infrastructure.

One of the fundamental ideas of ATM is to provide bandwidth on demand. Bandwidth has traditionally been an expensive and scarce resource. This has affected application development and even user expectations. So far, application development has been constrained because data transmission pipes cannot support various quality-of-service parameters, and the maximum data transmission bandwidth that the applications have to interface with is relatively small. Finally, ATM has removed these constraints. Bandwidth has become truly cheap and there is good support for various traffic classes. A new way of thinking may evolve in application development.

The progress toward ATM transport in fixed networks has already started, and the market push is strong. It is expected that new applications will evolve that fully exploit all the capabilities of ATM transport technology. Users will become accostomed to this new service level and require that the same applications be able to run over wireless links. To make this possible, wireless access interface must be developed to support ATM quality-of-service parameters.

The benefits of wireless ATM access technology in the data center should be observed by a user as improved service and improved accessibility. By preserving the essential characteristics of ATM transmission, wireless ATM offers the promise of improved performance and quality-of-service, not attainable by other wireless communications systems like cellular systems, cordless networks, or wireless LANs. In addition, wireless ATM access provides location independence that removes a major limiting factor in the use of computers and powerful telecom equipment over wired networks. Exhibit 2 shows a typical ATM network.

Wireless ATM Architecture

The architecture proposed for wireless ATM communications for the data center is composed of a large number of small transmission cells called pico-cells. Each pico-cell is served by a base station. All the base stations in the network are connected via the wired ATM network. The use of ATM switching for intercell traffic also avoids the crucial problem of developing a new backbone network with sufficient throughput to support intercommunication among large number of small cells. To avoid hard boundaries between pico-cells, the base stations can operate on the same frequency.

Reducing the size of the pico-cells has major advantages in mitigating some of the major problems associated with in-building wireless LANs for the data center. The main difficulties encountered include delay due to multi-path effects and the lack of a line-of-sight path resulting in high attenuation. Pico-cells also have some drawbacks as compared to larger cells. There are a small number of mobiles, on average, within range of any base station; thus, base-station cost and connectivity are critical. As cell size is reduced, the hand-over rate increases. By using the same frequency, no hand-over will be required at the physical layer. The small cell sizes also provides the flexibility of re-using the same frequency, thus avoiding the problem of running out of bandwidth.

Mobile units in the cell communicate only with the base station serving that particular cell, and not with other mobile units. The basic role of the base station is to interconnect between the LAN (or WAN) and the wireless

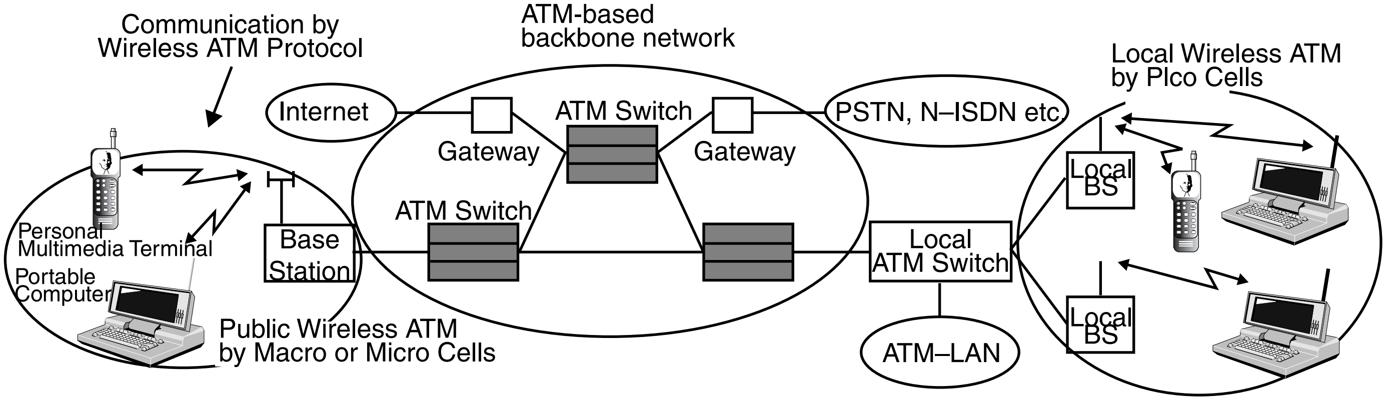


Exhibit 2. Normal ATM Network

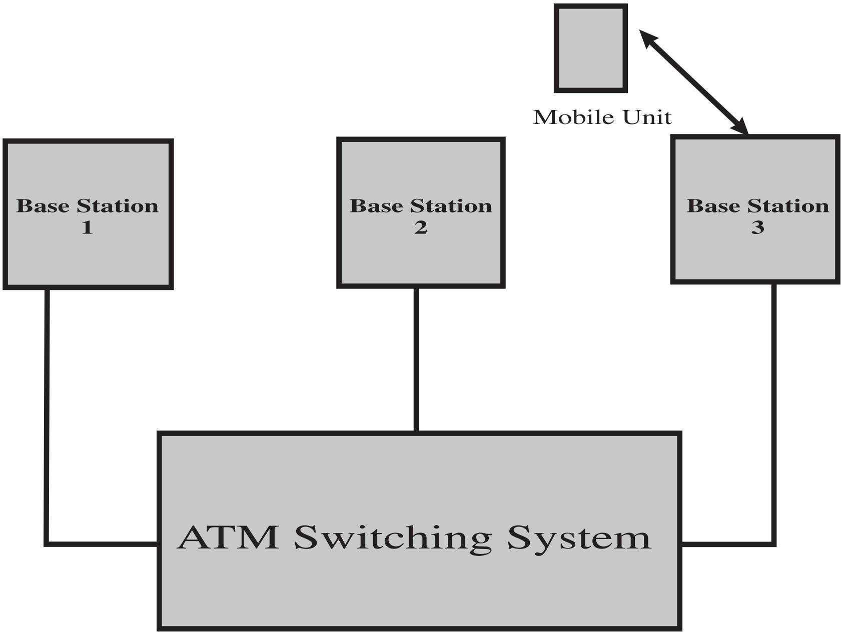


Exhibit 3. Normal ATM-to-Base Station Connection

subnets, and also to transfer packets and convert them to the wired ATM network from the mobile units.

In traditional mobile networks, transmission cells are *colored* using frequency division multiplexing or code division multiplexing to prevent interference between cells. Coloring is wasteful of bandwidth because, in order for it to be successful, there must be areas between re-use that are idle. These inactive areas could potentially be used for transmission. Exhibit 3 shows a typical ATM-to-base station connection.

STANDARDS

Wireless ATM research has been active for some time. There are many articles and books written on wireless ATM, and there are even announced wireless ATM prototypes, such as RATM (Radio ATM) by the Olivetti Research laboratory. Yet, the most important type of activity has been missing from wireless ATM scene. For enterprises with data center enterprise interests, the main objective is often to implement only equipment/systems conforming to standards. Thus, wireless ATM communications for the data center subject has been brought to two different standardization forums: namely, the European Telecommunications Standards Institute Society for

Technical Communications Remote Execution Service 10 (ETSI STC RES10) and the ATM Forum.

Currently, there are three standards bodies that have defined the physical layer in support of ATM: the American National Standards Institute (ANSI), the International Telecommunication Union's Telecommunications (ITU-T), and the ATM Forum. None of these bodies have considered the wireless ATM interface. The ETSI RES10 Subtechnical Committee is the first standardization body to start working on wireless multimedia, ATM compatibility, and standardization. The RES10 Committee has already engaged with the HIPERLAN (High Performance Radio Local Area Network) standardization, and the wireless ATM group is working on this subject. Initial work has concentrated on possible usage scenarios and specific requirements. Also, the search for available spectrum in the 5.2-GHz range for wireless ATM systems is crucial and therefore was one of the first tasks of the RES10.

The ATM Forum is not an official standards body, but it plays a significant role in the standardization arena because of its strong industrial participation and support. Wireless ATM activity has now been officially approved by the ATM Forum.

One wireless ATM activity solution that was approved divided the standardization of wireless ATM between the ATM Forum and RES10. Nevertheless, it would probably be wise to let the ATM Forum concentrate on the fixed network side and RES10 on the wireless interface. The main focus of the ATM Forum should be on the fact that the ATM physical layer is not necessarily always a reliable medium and that terminals may be mobile. Both of these facts are due to the fact that ATM/Broadband Integrated Services Digital Network (B-ISDN) connections may be stretched over the wireless links in the future and should be independent of the specific wireless interface.

Now, let us take a look at some ongoing projects in the area of wireless communications for the data center.

CURRENT PROJECTS

The following are some of the ongoing projects in the area of wireless ATM communications for the data center:

- Wireless ATM Network Demonstrator
- ATM Wireless Access Communication System
- International joint ventures

Wireless ATM Network Demonstrator

The objectives of this project are:

- To specify a wireless, customer premises access system for ATM data center networks that maintains the service characteristics and benefits of the ATM networks to the mobile user
- To promote the standardization of wireless ATM access for the data center
- To demonstrate and carry out user trials and test the feasibility of a radio-based ATM access system

For example, the Magic WAND project (Wireless ATM Network Demonstrator) covers the whole range of functionality from basic (wireless) data transmission to shared multimedia applications in Europe. The primary goal of the project is to demonstrate that wireless access to ATM (capable of providing real multimedia services to mobile users) is technically feasible. The project partners have chosen to use the 5-GHz frequency band for the demonstrator and to perform studies on higher bit rate operation >50 Mbps in the 17-GHz frequency band.

The aim of user trials is to verify a wireless access system for ATM data center networks that maintains the service characteristics and benefits of ATM networks in the 5-GHz range allocated to wireless high-speed data transmission. The feasibility of a radio-based ATM access system has also been demonstrated by user trials with selected end-user groups in hospital (medical consultation) and office environments.

The medical consultation trial shows an advanced scenario, fully exploiting the wireless ATM service capabilities in the hospital environment. The Joint Video Telecommunication Operating System (JVTOS) is being used with an X-ray viewing application, using both native audio and video services over ATM. In this scenario, doctors are equipped with a mobile terminal while visiting patients. With the help of a wireless ATM connection, doctors are able to retrieve patient information from the network, consult expert doctors, and share documents. The setup is shown in Exhibit 4.

Wireless ATM extends all the benefits of the ATM and therefore also the ATM signaling and virtual channels/paths into the mobile terminal, raising important issues that must be solved both in the wireless access interface and in the supporting customer premises ATM network. In the air interface, the wireless ATM transmission is subject to the problems associated with the radio medium and, therefore, special radio design measures are required in order to offer users an adequate level of service. These measures constitute some of the major technical challenges of this project.

The main result of the project is a Wireless ATM Access Network Demonstration system that serves as a proof-of-concept for the developed technology and helps the wireless ATM standardization work. The current achievements of the project include the complete functional system specification

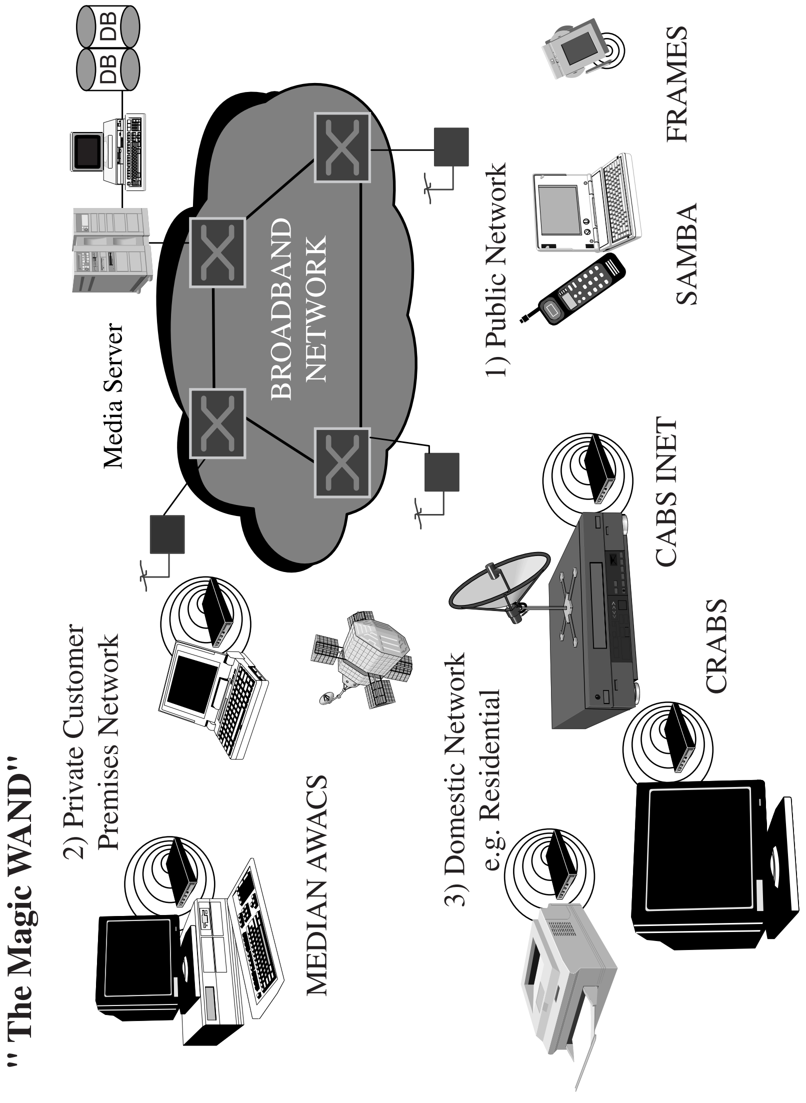


Exhibit 4. The Magic WAND Setup

on the demonstrator, which has been specified with the Specification and Description Language (SDL) and verified with the simulation model. In addition, the project has defined the exact demo platform setup and therefore enabled the basis for the implementation work that has been started on all parts of the system.

Besides demonstrator work, the project has been active in its liaison and standardization activities. The stochastical radio channel model for channel simulations was developed and verified by measurements on 5- and 17-GHz frequency bands. The model has been given as an input (for signal level 1 (SIG1) work). Furthermore, the project has been active in the standardization arena by contributing and harmonizing the work between the ATM Forum and ETSI RES10.

The Magic WAND project has continued the work of gaining knowledge on the wireless ATM radio design and its medium access control functions, as well as wireless ATM-specific control and signaling functions. These results have been and will continue to be contributed to ETSI and the ATM Forum in order to influence all of the relevant standards for wireless ATM systems.

ATM Wireless Access Communication System

The objectives or goals of the ATM Wireless Access Communication System (AWACS) project are the development of a system concept and testbed demonstration of public access to B-ISDN services. The system offers low-mobility terminals operating in the 19-GHz band with support of user bit rates up to 34 Mbps and radio transmission ranges of up to 100 m. The demonstrator of ATM Wireless Access (AWA) pre-prototype equipment provides immediate propagation data, basic encoding rules (BERs),[5] and ATM performance at 19 GHz. Based on this information, enhancement techniques for AWACS support cellular, as well as spectrum and power-efficient radio access technologies associated with HIPERLAN type 4 specifications.

Furthermore, the AWACS technical approach is centered around a testbed and associated trial campaign program. Trials are conducted using the existing ATM wireless access platform made available to the project by one of its partners. Associated program work is directed at enhancing this current state-of-art system toward the final target features of the emerging ATM wireless specifications; in particular, HIPERLAN type 4 is currently being defined by ETSI-RES10. These enhancements to the existing demonstrator are considered in the following areas:

- Application of source/channel coding and intelligent antennas
- Optimization of link layer protocols to match ATM bearer types

- Feasibility of 40-GHz radio frequency (RF)[6] technology for ATM wireless LAN applications
- Mobility management techniques, together with the impact on the radio bearer appropriate for high bit-rate communications

The AWACS field trial covers the concept of *virtual office* trials. This includes three potential cases, depending on the technical capabilities of the demonstrator:

1. Wireless multimedia communication link between an engineer at the production site and an expert at the office
2. Video communication in meetings between physically separated sites
3. Visual, wireless network access to virtual office facilities at the location of one of the partners

The objectives of these trials are summarized as follows:

- Improve communication between physically separated offices by telepresence technologies
- Reduce the need for traveling between geographically separated offices
- Improve the response time of expert advice in problem solving by visual communications
- Free staff from fixed office hours

Key Issues. The key issues to be considered include:

- The performance evaluation of a 19-GHz ATM-compatible modem
- Identification of the strengths and weaknesses of the existing ATM wireless experimental demonstrator
- Investigation of possible enhancement to the ATM-compatible modem
- AWACS field trials with the concept of *virtual office*, which aims to improve the communication between physically separated offices by telepresence technologies

Expected Results. The AWACS demonstrator based on ATM in packet transmission schemes supports limited, slow-speed mobility as it is in line with expected use of high data services. Therefore, the project generally covers the following directions, which are open to developers of mobile communication systems for the future: (1) construction of a wireless system providing seamless service in connections to hard-wired systems (quality oriented system) and (2) services making the most of the excellent mobility and portability of mobile communication systems (mobility-oriented system).

The AWACS trials indicate the capacity of the available system in a real user environment. The trial results contribute to the development of common specifications and standards such as ETSI-RES10 (for HIPERLAN type

4 specifications), ITU, Telecommunication Technology Committee (TTC),[7] and Association of Radio Industries and Businesses (ARIB) in Japan.

International Joint Ventures

Wireless ATM has started and there is a world-wide effort to unify and standardize its operation. The Public Communication Networks Group of Siemens AG, Newbridge Networks, and Broadband Networks Inc. (BNI) has begun an extensive joint research and development program to address the digital wireless broadband networks market. These three enterprises are focusing on integrating BNI's broadband wireless technology with the Siemens/Newbridge Alliance's MainStreetXpress™ family of ATM switching products to develop wireless network base stations that are fully compatible with wireline services.

BNI has already deployed terrestrial wireless networks that provide wireless cable in a digitally compressed MPEG2 (Motion Pictures Experts Group) format, delivering laser disk-quality transmissions with the capacity for hundreds of channels. The Siemens/Newbridge Alliance offers carriers the most comprehensive suite of ATM products and the largest ATM core infrastructure switch, scalable up to one terabyte and beyond. The introduction of ATM into the broadband wireless environment will enable network operators to cost-effectively deploy high-capacity access services such as high-speed data, broadcast (cable) distribution, and Internet access in the 28-GHz range. By incorporating both MPEG2 and ATM into the broadband wireless environment, the network solution provided by BNI and the Siemens/Newbridge Alliance ensures high-speed, high-quality, and high-capacity video, voice, and data transmissions. It also represents an effective bandwidth allocation that ensures sufficient capacity for additional innovative residential and commercial services as they evolve.

Finally, before moving on to Part II (Chapter 20), let us take a look at wireless communications hardware in the form of its functioning and applications for wireless communications in the data center. Diagrams (Exhibits 5 to 8) of various system configurations are included here.

WIRELESS COMMUNICATIONS HARDWARE AND APPLICATIONS

The following are wireless communications system configurations for the data center in the form of hardware and applications, as well as sample installation schematics as shown in Exhibits 5 to 8:

- Handheld communications terminal
- Wireless interface processor
- Remote data collection
- Example of an ArielNet wireless communications application

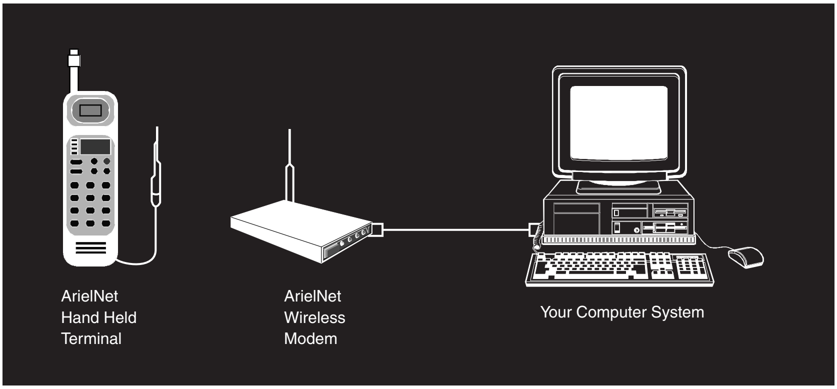


Exhibit 5. Single HHCT User Application

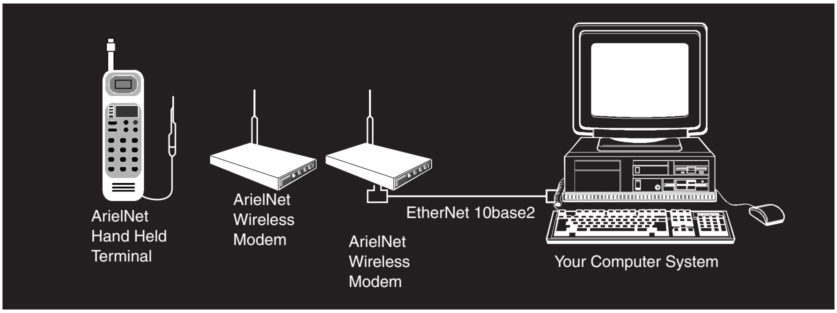


Exhibit 6. Using a Modem as a Message Repeater

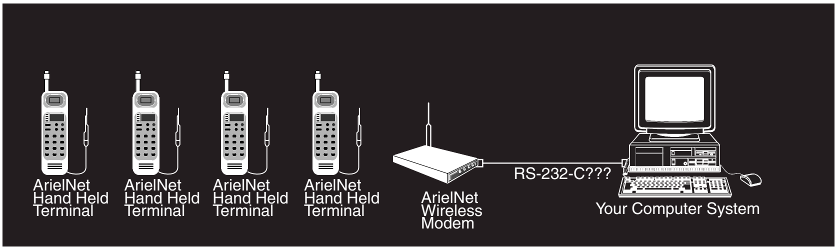


Exhibit 7. Multiple HHCT Users

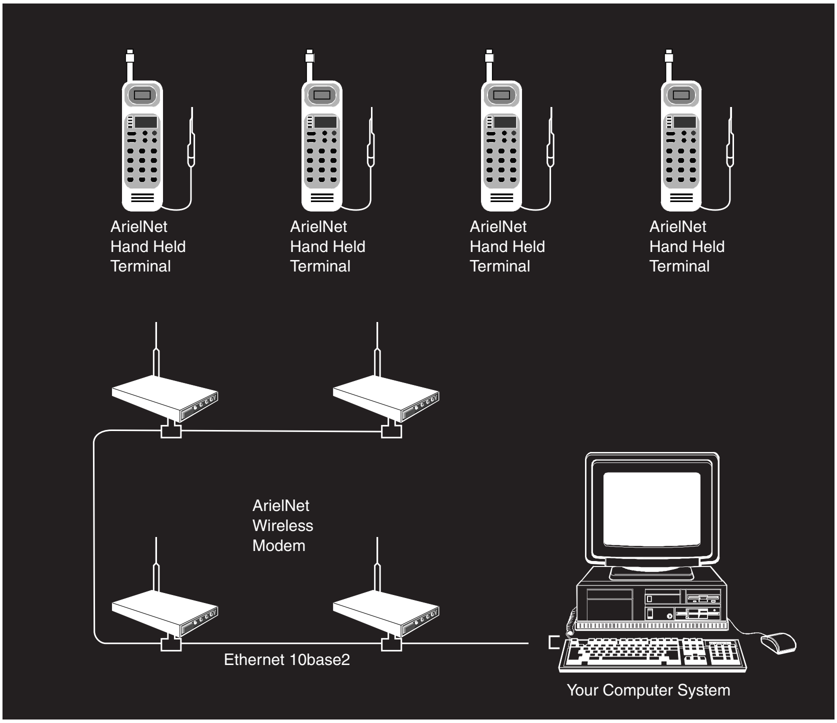

ArielNet Hand Held Terminal
ArielNet Hand Held Terminal
ArielNet Hand Held Terminal
ArielNet Hand Held Terminal
ArielNet Wireless Modem
Ethernet 10base2
Your Computer System

Exhibit 8. Multiple HHCT Users over a Wide Area Network (WAN)

Handheld Communications Terminal

A Hand Held Communications Terminal (HHCT) consists of a liquid crystal display (LCD), a 40-key keyboard, and an RF modem housed in a lightweight portable case. The low-power microprocessor in the HHCT provides the processing and communications functions. The HHCT unit provides most of the functions of an ANSII standard terminal. The HHCT is battery-operated for up to 12 hours on a charge and has provisions for connecting a bar code wand as an additional data collection device.

Communications between the wireless interface processor (WIP) and HHCT are carried over a narrow-band FM radio channel at a rate of 9600 baud. Any number of HHCTs can be addressed by the WIP, because each HHCT has a unique identification number. The communications process is transparent to the user.

Wireless Interface Processor

The wireless interface processor (WIP) is a small electronic enclosure that houses the microprocessor, radio transmitter, radio receiver, and the antenna and communications interface. A WIP provides a connection to fixed resources such as instruments, computers, machinery, inventory, and property. The WIP provides the communication path from a remote site to a host system, as well as to HHCTs. The host system provides access to information such as inventory databases, equipment status and scheduling, process status, and control. A WIP also allows the HHCT user to access networks such as the Internet and its global information services, including e-mail.

Remote Data Collection

A remote data collection system can be implemented using a WIP connected to the serial communications port of a computer system that has an inventory database application running. An HHCT with a bar code scanner attached can then be used to communicate inventory data over a large area. By connecting a bar code scanner, the HHCT can be used as an inventory control or data capture device.

The WIP is connected to the communications port (COM port) of the host computer. The host computer has its console assigned to the COM port (ctty) and executes an inventory or database application. The HHCT now can function as the computer console and provide data gathering and control of the host computer at up to 1000 ft away.

The wireless components of the ArielNet® wireless communications system operate in compliance with Federal Communication Commission allocations (part 15) for license-free operation. This puts the communication range for each element of the wireless network at 1000 ft. With an array of devices, much larger areas can be covered.

Example of an Arielnet® Wireless Communications Application

This application describes the use of wireless communications and the WIP in a product-delivery service enterprise. The enterprise is a bakery and the problem is inventory control. The bakery would like to sell all of its product while it is still fresh. Trucks are loaded each morning with the product, and each evening the trucks return with some product not sold that day. The product dispatcher would like a correct inventory of day-old products at the start of the day and have them placed on the truck so they are delivered to the correct customers. The delivery trucks each have a computer and an ArielNet® WIP. The truck computer could be a notebook computer system that provides customer order information, truck inventory, and route information to the driver. The truck computer also connects to a bar code reader that can read the bar codes on the product. Information is sent to the truck computer each morning as it is loaded, and updated information is sent back each evening by wireless to the bakery's main control computer.

Bakery Delivery and Product Control Sequence. The bakery computer sends customer information and the routing schedule to the truck computer. As the truck is loaded, each product is scanned with the bar code scanner, and the count is entered into the truck computer.

The truck computer communicates to the people loading the truck that the truck has the correct product loaded and sends this information to the bakery computer. The driver follows the computer route and scans each customer's product as it is delivered. The driver then returns to the terminal at the end of the day. As the truck is unloaded, any undelivered product is scanned. This information is sent to the bakery computer. If the information is complete, the driver is relieved; otherwise, corrective action is taken.

CONCLUSION AND SUMMARY

While wireless communication is experiencing rapid evolution, the fixed network has been going toward B-ISDN with ATM concept. ATM offers data rates that are considerably higher than current fixed network services. Interworking with ATM will set extremely hard requirements on the wireless air interface, but hopefully, continued development in technology will enable the industry to manufacture smaller and less power-consuming terminals with increased performance and functionality.

Predicting the future is always uncertain, but it can be assumed that frequencies under 2 GHz remain mainly for mobile communications where only low bit-rate services are offered (both data and speech). In this case, connections requiring close to 2 Mbps or more will need to be moved onto the higher frequencies. The possible choices at the moment seem to be around 5.2 GHz and 17.1 GHz.

The successful introduction of wireless ATM is strongly related to the success of ATM/B-ISDN in wired networks. If ATM/B-ISDN networks are to be a commercial success, wireless ATM should be seen not as today's technology but as inevitable development in the very near future.

Chapter 20, "Part II: Wireless Communications Data Center," takes a look at how the mobile nature of wireless communications in the data center provides consumers with the opportunity to access the data center from any place at any time. Today, as the Federal Communications Commission (FCC) makes available a new spectrum for wireless networks that will support a range of new services, both voice and data, wireless communications are poised on the brink of a new era.

However, new spectrum leads to new entrants, and wireless enterprises of the future will face a much more competitive marketplace. This competition will mean great things to the American consumer, who will benefit from the innovation and lower prices that the increased competitiveness will spark. Thus, with the introduction of more competition into the telecommunications marketplace, public policy decisions need to be crafted to ensure that this vision of a wireless future can be realized.

NOTES

1. TDMA divides the radio carriers into an endlessly repeated sequence of small time slots (channels). Each conversation occupies just one of these time slots. So, instead of just one conversation, each radio carrier carries a number of conversations at once. With the development of digital systems, TDMA is being more widely used.
2. The term "spread spectrum" defines a class of digital radio systems in which the occupied bandwidth is considerably greater than the information rate. The term "code division multiple access" (CDMA) is often used in reference to spread-spectrum systems and refers to the possibility of transmitting several such signals in the same portion of spectrum using pseudo-random codes for each one. This can be achieved either by frequency hopping (a series of pulses of carrier at different frequencies, in a predetermined pattern) or direct sequence (a pseudo-random modulating binary waveform whose symbol rate is a large multiple of the bit rate of the original bit stream) spread spectrum.
3. A cell-based data transfer technique in which channel demand determines packet allocation. ATM offers fast packet technology, real-time, and demand-led switching for efficient use of network resources. It is also the generic term adopted by ANSI and the ITU-T to classify cell relay technology within the realm of broadband WANs, specifically B-ISDN. In ATM, units of data are not time related to one another and, as part of the B-ISDN standard, are specified for digital transmission speeds from 34 Mbps to 622 Mbps. IBM currently offers ATM in a nonstandard 25 Mbps format. ATM will be the high-bandwidth networking standard of the decade.
4. The international body that develops worldwide standards for telecommunications technologies. The ITU-T carries out the functions of the former CCITT.
5. Basic encoding rules: rules for encoding data units described in the ISO ASN.1 standard.
6. Radio frequency: the generic term referring to frequencies that correspond to radio transmissions. Cable TV and broadband networks use RF technology.
7. The Telecommunication Technology Committee (TTC) was established as a private standardization organization in October 1985 to contribute to further activation of the field of telecommunications, in which the free-market principle was introduced based on implementation of the Telecommunication Business Law in 1985, and to respond to the Japan/U.S. Market Oriented Sector Service (MOSS) Conference, which was held that same year.

ABOUT THE AUTHOR

John Vacca is an information technology consultant and internationally known author based in Pomeroy, OH. Since 1982, John has authored 27 books and more than 340 articles in the areas of Internet and intranet security, programming, systems development, rapid application development, multimedia, and other Internet-related areas. John was also a configuration management specialist, computer specialist, and the computer security official for NASA's space station program (Freedom) and the International Space Station Program, from 1988 until his early retirement from NASA in 1995. John can be reached at jvacca@hti.net.

Chapter 20
Wireless Communications Data Center: Part II

John R. Vacca

For the wireless communications industry, 1994 was a banner year. The Federal Communications Commission (FCC) launched the first set of spectrum auctions for the narrowband and broadband personal communication service (PCS), giving birth to a whole new era — the era of personal communications. The vision of PCS is the concept of anytime, anywhere communications — whether that be data communications, voice communications, or both. But what is the real potential for this marketplace? How many individuals are likely to buy into the vision of anytime, anywhere communications?

In early 1995, the Personal Communications Industry Association (PCIA) completed a survey of its members to evaluate the growth, composition, and characteristics of the existing and future personal communications industry and published the results in a PCS Market Demand Forecast. The results indicated that by 2001, combined demand for new PCS, cellular, and paging and narrowband PCS will amount to almost 229 million subscriptions.

To meet this level of demand in the marketplace, the wireless industry must be assured that it will be able to deploy services in a timely fashion. Issues such as site acquisition and interconnection to the local exchange carriers are critical to timely deployment of developing wireless networks for the data center and competing effectively. Government must ensure that the industry has the opportunity to meet the anticipated demand outlined in the PCS Market Demand Forecast by ensuring a level playing field for all wireless telecommunications service providers and by allowing — where appropriate — competition (not regulation) to govern the marketplace.

0-8493-1161-6/02/$0.00+$1.50

THE PERSONAL COMMUNICATIONS INDUSTRY ASSOCIATION

Established in 1949, the PCIA has been instrumental in advancing regulatory policies, legislation, and technical standards that have helped launch the age of personal communications services. Through many vehicles (policy boards, market forecasts, publications, spectrum management programs, seminars, technician certification programs, and its industry trade show), the Personal Communications Showcase (PCIA) is committed to maintaining its position as the association for the PCS industry.

PCIA member enterprises include PCS licensees and those involved in the cellular, paging, enhanced specialized mobile radio (ESMR), specialized mobile radio (SMR), mobile data, cable, computer, manufacturing, and local and interexchange sectors of the industry, as well as private enterprise systems users, wireless system integrators, communication site owners, distributors and service professionals, and technicians.

PERSONAL COMMUNICATION SERVICE

Personal communication service (PCS) includes a broad range of telecommunications services that enable people and devices to communicate independent of location. PCS networks and devices operate over a wide range of frequencies assigned and authorized by the FCC. There are currently seven different air interface technologies proposed for standardization for the new PCS licensees that will be operating in the 1.8-GHz band. Service providers that will be operating at these frequencies are either new entrants with no established network or existing telecommunications service providers (such as cable, cellular, local exchange, and long-distance carriers). With the technology choices enterprises make over the next few months, there will need to be analysis of how and to what extent the various wireless and wireline networks will work together.

INTEROPERABILITY AND INTERWORKING

To facilitate roaming among PCS carriers, some degree of interoperability and interworking needs to be accomplished between the networks. PCIA defines interoperability and interworking as follows.

Interoperability

Interoperability is the ability to logically connect two or more functional network elements for the purpose of supporting shared processes such as call delivery. Service interoperability is defined as the assurance that a service invoked by a subscriber in a network will be performed by the other network in the same way from a user perspective. Network interoperability is defined as the direct one-to-one mapping of services and protocols between interconnected networks. For example, a subscriber may invoke call-waiting features exactly the same way in a Data Collection System (DCS) 1900 (Global System for Mobile-Communications (GSM)-based) network in New York City as in a DCS 1900 (GSM-based) network in San Francisco. In this scenario, call-waiting network protocol messages map between the two networks on a direct one-to-one basis.

Interworking

Interworking is the ability to translate between two or more dissimilar networks for the purpose of achieving effective interoperability. Service interworking is defined as the protocol translation that may or may not result in the service being performed in the receiving network in the same way from a user perspective. Network interworking is defined as functional mapping of services and protocols across networks (some services may not be delivered or may be delivered in a different way). For example, a subscriber with a PCS 2000 (Composite Code Division Multiple Access/Time Division Multiple Access [CDMA/TDMA[1]]) wireless personal terminal can register and authenticate on a San Francisco Interrupt Status-41 (IS-41)-based network, just as he or she could on a home-base DCS 1900 (GSM-based) network in New York City. Although the method of registering may not be identical between systems, the end result is effectively the same — the subscriber can be registered and authenticated on both networks, and location services work across both platforms.

Standards are being developed by domestic and international standards bodies to facilitate features and services delivered consistently and in similar fashion to an end user — regardless of the air interface or network implementation used. All networks do not necessarily need to interoperate or interwork with every other network. Those decisions will be made on a enterprise-by-enterprise basis. But the industry is working to ensure that if that choice is made, the technology will be available to support it.

Market Forecast

Since 1992, the PCIA has regularly surveyed wireless communications industry leaders to evaluate the growth, composition, and characteristics of the future of the personal communications industry; it has published these results in a PCS Market Demand Forecast. In its annual surveys, the PCIA has asked respondents to provide market-size predictions in terms of the number of anticipated subscriptions, not subscribers — anticipating that an individual would probably subscribe to more than one type of wireless service in the coming decade. As in previous years, the 1998 figures show that consumer demand for personal communications services is expected to grow at ever-increasing rates.

Demand growth for new broadband PCS[2] customers was expected to reach 30 million subscriptions by 2001. Total revenues were expected to reach $9.9 billion by the year 2001, with 8 percent of that revenue coming from data services. Average revenue per subscription was expected to be 30 percent less than that for cellular. Figures for 2006 indicate strong sustained growth — to almost 50 million subscriptions and total revenues reaching $28.6 billion, with 13 percent from data services.

Established voice services such as cellular are expected to grow as well. The 1998 year-end subscriber count of 46.4 million was expected to double to approximately 92.6 million subscriptions by 2001, with nearly 76 million cellular subscriptions expected by 2006. Some 40 percent of the total cellular subscriptions is expected to come from the enterprise segment, representing a presumed growth of the cellular market into households over the next 11 years. Total cellular revenues were forecast to be approximately $37 billion by 2001 and $42 billion by 2006.

In the narrowband PCS[3] arena, market size was expected to reach more than 60 million subscriptions by 2001; by 2006, 82 million one-way and 32 million two-way messaging subscriptions are anticipated. In addition, survey results forecast strong growth from new narrowband PCS and advanced one- and two-way messaging; and suggest that these will become established in the wireless world of the data center over the next decade. Customer segments will grow due to new narrowband applications and services. Survey results show that by the year 2001, more than 60 percent of one-way and about 76 percent of two-way subscribers were expected to be from enterprise segments. Assuming that enterprises will continue to upgrade services, they are expected to remain more than 60 percent of the total subscriber base through the next decade. Total revenues were expected to reach $5.8 billion for one-way paging and $2.1 billion for two-way paging by 2001, and $6.7 billion and $4 billion, respectively, by 2006.

Site Acquisition Issues

Acquiring PCS antenna and base station sites and gaining the appropriate zoning approvals vary by state and local jurisdictions and are important in wireless network deployment in the data center. Furthermore, there are issues regarding site acquisition (e.g., Federal Aviation Administration [FAA] tower regulations and the lack of a uniform policy regarding sites on federal property) that need to be addressed at the federal level.

Issues at the Local Level

There are more than 49,000 local jurisdictions throughout the nation, each with the authority to prevent antenna construction, establish standards that can result in site location degrading the quality of service, and prolong site selection, thereby making it unnecessarily expensive. With an estimated 200,000 new wireless antenna sites predicted over the next 11 years, any licensing obstacles present significant problems.

The U.S. Congress has recognized the need to remove state and local barriers to deploying Commercial Mobile Radio Service (CMRS) facilities by prohibiting state and local government regulation of matters relating to market entry and rates. The current draft of the Senate's Telecommunications Competition and Deregulation Act of 1995 states that no state or local statute may prohibit or have the effect of prohibiting the ability of any entity to provide interstate or intrastate telecommunications services. It further states that if, after notice and comment, the FCC determines that a state or local requirement is inconsistent with the legislation, the FCC shall immediately preempt enforcement of the requirement.

The Cellular Telecommunications Industry Association (CTIA) filed a Petition for Rule Making, requesting that the FCC initiate a rule-making proceeding to preempt state and local regulation of tower sites for CMRS. The petition states that the state preemption language in Section 332(c) of the Communications Act gives the Commission authority to exercise such preemption, since local zoning could constitute an *indirect* regulation of entry.

Comments on the Petition for Rule Making were pro and con. Predictably, service providers filed in support of the petition, while state and local governments and consumer groups filed in opposition. The challenge the wireless industry faces is balancing the recognized needs of the local community to have oversight and fee administration of zoning issues against attempts to meet the ever-increasing demand for new wireless services.

Additionally, the FCC has imposed build-out requirements on the new PCS licensees, which mandate that certain percentages of the licensees' markets be covered within set timeframes. Potential conflicts between state and federal regulations threaten to delay the entry of wireless services.

Site Acquisitions on Federal Property

Federal property could, in many situations, provide prime locations for PCS base stations. Unfortunately, many agencies of the U.S. federal government are unwilling or unable to entertain the prospect of such facilities because of perceived administrative burdens, lack of benefit to local agency staff, or lack of clear policy or regulations for leasing of federal property for such an installation. Additionally, all of the federal agencies that allow private communications facilities on their land have different regulations, lease documents, and processes for doing so. These are often difficult, time-consuming, and expensive for both the agency and the communications enterprises.

Making sure federal land resources continue to be available for efficient delivery of mobile communications services, and ensuring that taxpayers receive a fair price from every communications enterprise with transmitters on public lands, are goals shared by industry, the federal agencies, and the public. However, there needs to be a consistent, government-wide approach for managing the site-acquisition process on federal property.

The Executive Branch needs to set a clear directive in order to overcome the obstacles wireless licensees face when trying to acquire sites on federal property. The benefits to the federal government could include increased revenues from the installation of PCS networks above and beyond the auction proceeds, as well as the potential for improved communications on federal property.

FAA Tower Review Process

The PCIA has initiated discussions with the Federal Aviation Administration (FAA) to remove any possible FAA obstacles to efficient deployment of PCS systems. The FCC has established licensing rules that have streamlined the approval necessary to bring systems and PCS cell sites on line. However, due to administrative limitations, the FAA, which must review many requests for towers to ensure air safety, has experienced longer processing times that have delayed carriers' ability to activate certain transmitter sites. With approximately 30 to 35 percent of new wireless data center sites requiring FAA action and review, the PCIA fears that the FAA processing delays could significantly burden the industry. Working Groups at the national and local levels have been established as a forum to explore methods of educating the industry about FAA procedures and to explore ways to streamline the FAA tower review process.

The FAA, FCC, PCIA, and the Cellular Telecommunications Industry Association (CTIA) have all agreed to participate in this dialogue as part of an Antenna Work Group (AWG) in Washington, D.C. The PCIA has also participated in dialogues with the FAA Southern Region and is working on the local level in other working groups to identify ways to improve the FAA process.

Federal Radio Frequency Emissions Standard

As PCS, cellular, paging, and other wireless carriers build out networks for data centers, they are increasingly facing state and local laws and ordinances based on radio frequency (RF) exposure levels — often with conflicting scope and standards and resulting in compliance difficulties. Conflicting standards affect the range of wireless services and can greatly diminish the quality of service consumers receive. This adds greatly to the expense borne by the industry, not only in legal and other enterprise expenses, but also in lost revenue opportunities from long delays in providing services.

The FCC has the authority to preempt local jurisdictions on cell/antenna/ tower siting, but to date has approached this issue on a case-by-case basis. With as many as 200,000 new wireless sites to be installed (including new PCS sites, and additional sites that will be needed for the expansion and enhancement of service areas for paging, SMR, ESMR, and cellular service), a case-by-case approach to preemption is no longer realistic.

The FCC — on April 3, 1993 — issued its Notice of Proposed Rule Making, which proposed updating guidelines and methods for evaluating the environmental effects of electromagnetic exposure, and adopting the standard developed by the American National Standards Institute (ANSI) with the Institute of Electrical and Electronic Engineers (IEEE). In December 1994, the Spectrum Engineering Division of the Office of Engineering and Technology of the FCC issued information indicating that levels of exposure to RF at ground level below typical cellular towers are hundreds to thousands of times lower than the proposed standard.

On December 22, 1994, the Electromagnetic Energy Association (EEA) filed a petition with the FCC for a Further Notice of Proposed Rule Making. The petition requested that the FCC preempt state and local regulation of RF exposure levels found to be inconsistent with the FCC-proposed ANSI standard.

The PCIA favors the establishment of a single, national RF emissions standard that cannot be exceeded by local regulations. The PCIA encourages the relevant federal agencies to work cooperatively with industry on this issue to develop such a national standard.

INTERCONNECTION

Interconnection is composed of interconnection with local exchange carriers and mutual compensation.

Interconnection with Local Exchange Carriers

Negotiating reasonable rights, rates, and terms under which enterprises will interconnect with other networks is critical to the success of PCS. With many PCS hopefuls eyeing the local exchange market as a potentially lucrative area in which to compete, the terms of enterprise interconnection agreements as a co-carrier will become even more important as they strive to compete with local exchange carriers (LECs); therefore, they will need reasonable interconnection agreements so that they can offer customers low-cost exchange service.

As an example of current interconnection costs, type 2 interconnection charges for cellular carriers generally are measured on a per-minute basis, with costs ranging from $0.01 per minute to $0.05 per minute, and $0.02 per minute often being considered a *good* interconnection rate.

Interconnection charges have diminished cellular carriers' revenues since the first system came online, and they remain a high cost to carriers today. Take, for example, a cellular monthly bill of $48, which includes 97 minutes of air time, at the rate of $0.02-per minute for interconnection. Interconnection charges represent $1.94 of the bill, or 4.04 percent of revenue.

As air-time costs continue to decline in the wireless marketplace, interconnection costs will begin to reduce revenues even more. For example, Honolulu, Hawaii's Cybertel Cellular offers $0.04 per minute of air time in Kauai, Hawaii, to compete with the local exchange carrier. At an interconnection rate of $0.02 per minute, interconnection charges could consume 70 percent of the carrier's air-time revenue.

Obviously, those wishing to compete at the local loop must achieve lower interconnection costs to compete with established carriers on price. One solution to this problem is mutual compensation, where both telecommunications carriers are compensated for the traffic that terminates on their network.

Mutual Compensation

Mutual compensation is the concept that a carrier should be compensated for traffic that originates on another network but terminates on that carrier's network, and vice versa. Currently, wireless carriers must compensate wireline carriers for traffic that originates on a wireless data center network and terminates on a wireline network. Almost without exception, wireline carriers do not compensate wireless carriers for traffic originating on a wireline network and terminating on a wireless network.

The FCC has repeatedly stated that, for interstate traffic, wireline carriers must compensate wireless carriers for traffic originating on a wireline network and terminating on a wireless network. However, states have been reluctant to enforce mutual compensation on an intrastate basis, and therefore wireline carriers have refused to participate in mutual compensation on either an intra- or interstate basis.

Enforcement of mutual compensation rights of wireless carriers is considered to be key to full competition by wireless carriers in the telecommunications market and will have a significant, positive financial impact for the wireless industry. One potential solution to the high cost of interconnection would be mandating mutual compensation through reciprocal elimination of interconnection charges. One example of this solution is the agreement reached in New York state between Time Warner and Rochester Telephone (Rochester, New York), whereby Rochester Telephone will collect 1.1 cents per minute for traffic terminating on its network, and pay at the same rate for its own traffic terminating on other networks. According to the agreement, mutual compensation provisions are eliminated when the traffic flow differentials fall below 9 percent.

Numbering Issues

The issue of who controls numbers is key to the success of PCS carriers. Traditionally, most national numbering resources have been assigned by the North American Numbering Plan Administration sponsored by Bellcore,[4] which in turn is owned by the Bell operating enterprises. Generally, the dominant local exchange carrier ends up assigning numbers to wireless carriers in its local telephone market. Wireless carriers are usually charged for activating blocks of numbers in local exchange carrier networks, and the charges vary greatly.

Recently, Bellcore has come under scrutiny for its administration of numbering resources, and actions by wireline carriers have brought the issue to the forefront. For example, in Chicago, Ameritech proposed an *overlay* area code. This would require cellular and paging subscribers to give back their numbers and receive a new area code, thus freeing up numbers in the almost-exhausted code for new wireline subscribers. At a recent FCC open meeting, the FCC found this proposal to be "*unreasonably discriminatory*" against wireless carriers.

The FCC initiated a proceeding more than 2 years ago to examine whether an independent entity should oversee the assignment of numbers, and it appears as if the Senate telecommunications reform effort might mandate the formation of an independent entity to oversee the numbering assignment process.

Number Portability. Another key issue for those who want to compete with the local telephone enterprise is number portability, or the ability of an end user, such as an individual or enterprise, to retain its ten-digit geographic North American Numbering Plan (NANP) number — even if the end user changes its service provider, the telecommunications service with which the number is associated, or its permanent geographic location. With few exceptions, today end users may not retain their ten-digit NANP number if they:

- Switch service providers, referred to as *service provider portability* (a user switches from an incumbent LEC to a new competitive access provider)
- Change the service to which the number was originally assigned, referred to as *service portability* (a cellular telephone number becomes the wireline home telephone number)
- Change their permanent location, referred to as *geographic portability* (an end user moves to a different part of the city or state, and may be assigned either a new seven-digit phone number in the old area code or a new ten-digit number in a new area code).

Service provider portability, that is, moving a number from one service provider to another, is vital for those enterprises that wish to compete for customers at the local exchange level. It is much easier to gain market share if the customer an enterprise is trying to attract does not have to change his or her phone number when changing service providers.

Currently, 800-numbers are portable between 800-number service providers — an example of service provider portability. This portability allows the 800-service end user to retain his or her individual 800-number, even when switching 800-service providers. Portability of 800-numbers was ordered by the FCC and implemented in 1993.

Industry Efforts to Address Number Portability. The Industry Numbering Committee (INC), a consensus-based industry body sponsored by the Inter-Carriers Compatibility Forum (ICCF), has been actively addressing number portability issues since the Fall of 1993. The INC Number Portability Workshop has been addressing a range of issues associated with number portability, including a target portability architecture, network impacts of number portability, and high-level policy issues such as mandated interconnection.

Public Service Obligations

The advent of increased mobility is having an impact on telecommunications public policy. How does wireless technology fit into public policy initiatives such as universal service and access to enhanced 911 emergency calling services? Policies regarding universal service were developed to apply to a strictly wireline environment where competition at the local level was nonexistent. Additionally, wireless technologies present a challenge to the traditional wireline approach to providing enhanced 911 emergency calling. As wireless service providers begin to compete for the local loop, how wireless fits into such public policy provisions will need to be seriously considered.

Universal Service

Universal service, as a public policy concept, is the belief that access to basic telephone services by the widest possible cross-section of the American public is in the social and economic interests of the United States. Over a period of many years, Congress has mandated the creation of universal service programs to support universal service public policy goals. The FCC is charged with fulfilling these congressional mandates.

Within the telecommunications industry, universal service refers to a complex system of explicit and implicit charges and cost allocation mechanisms levied on particular carriers and customers in order to provide access to, and subsidize the rates of, basic wireline services for residential customers, high-cost customers and carriers, low-income customers, rural areas, and services for hearing- and speech-impaired consumers. Estimates of the current total costs of supporting universal service goals and policies range as high as $30 billion to $40 billion annually. Congress is intent upon reform of the universal service policy and funding mechanisms as part of its effort to reform existing telecommunications law. Any reforms could have a potentially huge economic impact on the wireless industry.

Universal service reform is a critical part of telecommunications reform and it appears inevitable if Congress passes a telecommunications reform bill. Although it is too early to tell what shape universal service will take, a number of issues need to be considered.

Wireless Access to Enhanced 911 Emergency Services

The FCC, on October 19, 1994, released a Notice of Proposed Rule Making (NPRM) regarding revision of the Commission's rules to ensure compatibility with enhanced 911 (E-911) emergency services. In many areas of the country, wireline subscribers are provided E-911 service by wireline carriers, which entails transmitting the address and phone number of the caller to the public safety answering point. The NPRM addresses Private Branch eXchange (PBX) issues and wireless service provider issues. The NPRM outlines proposed requirements on wireless services regarding:

- 911 availability
- Grade of service
- Privacy
- Re-ring/call back
- Liability
- Cost recovery
- Access to text telephone devices (TTY)
- Equipment manufacture, importation, and labeling
- User location information
- Compatibility with network services

- Common channel signaling
- Federal preemption

The proposed requirements have considerable technical and economic implications that need to be fully examined. The PCIA, in cooperation with representatives of the public safety community, drafted the Joint PCIA, Association of Public Safety Communications Officials (APCO), NASNA Emergency Access Position Paper, which was filed with the FCC in July 1994. This joint paper documented the first attempt of the PCS community to comprehensively address the needs of the public safety community. The FCC used the joint paper as a basis for its NPRM addressing enhanced 911 emergency calling systems.

The PCIA fully shares the FCC's important objective of maximizing compatibility between wireless services and enhanced 911 emergency calling systems. Specifically, it concurs that subscribers to real-time voice services interconnected with the public switched telephone network ultimately should enjoy the same access to advanced emergency response services as do wireline service subscribers, with due consideration for the unique characteristics of radio-based technology. At the same time, however, PCIA strongly disagrees with the approach toward achievement of the compatibility objective that is set forth in the NPRM.

The PCIA believes that full-scale regulatory intervention is not necessary at this time and that the profound technical issues raised by compatibility cannot be resolved through imposition of arbitrary deadlines as proposed in the NPRM. The PCIA proposes, as an alternative to arbitrary deadlines, that the industry work to develop technical solutions to the public safety community's requirements and that the FCC require periodic reports from the industry on its progress in meeting the ultimate goals that the FCC has set forth.

Now, one can look at wireless LANs for the data center by introducing the basics behind wireless LANs and giving an overview of how they work. It has been predicted that economically, wireless LANs will reach $2 billion in revenues by the year 2001. The cost of installing and maintaining a wireless LAN generally is lower than the cost of installing and maintaining a traditional wired LAN. Hence, more and more enterprises are implementing this LAN configuration.

WIRELESS LANS

A wireless LAN (WLAN) is a networking method that delivers all benefits of a local area network (LAN) to the data center with one very important advantage — no wires. No wires means that one now has the flexibility to immediately deploy workgroups wherever and whenever needed. WLANs allow different workstations to communicate and to access a network

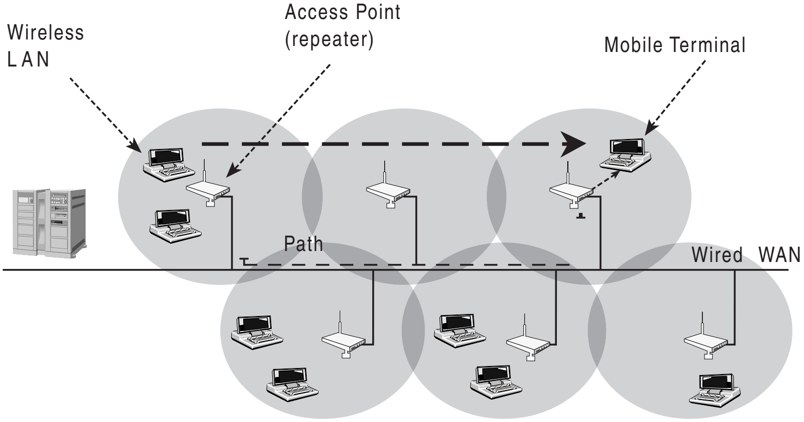


Exhibit 1. Configuration of Wireless LAN

using radio propagation as a transmission medium. The WLAN can then be connected to an existing wired LAN as an extension, or it can act as a stand-alone network. The advantage here is that WLAN combines data connectivity with user mobility and gives the user a movable LAN. WLANs are especially suited for indoor locations such as hospitals, universities, and office buildings.

Configuration of WLAN

The keystone to a wireless LAN is the cell. The cell is the area where all wireless communication takes place. In general, a cell covers a more-or-less circular area. Within each cell, there are radio traffic management units also known as access points (repeaters). The access point, in turn, interconnects cells of a WLAN and also connects to a wired Ethernet LAN through some sort of cable connection, as shown in Exhibit 1.

The number of wireless stations per cell depends on the amount (and type) of data traffic. Each cell can carry anywhere from 50 to 200 stations, depending on how busy the cell is. To allow continuous communication between cells, individual cells overlap. Cells can also be used in a stand-alone environment to accommodate traffic needs for a small- to medium-sized LAN between workstations or workgroups. A stand-alone cell would require no cabling. Another option is wired bridging. In a wired bridging configuration, each access point is wired to the backbone of a wired Ethernet LAN (see Exhibit 1). Once connected to a wired LAN, network management functions of the wired and the wireless LANs can be controlled. Wireless bridging is also an option; this allows cells to be connected to remote wireless LANs. In this situation, networking can stretch for miles if it is linked successively and effectively from access point to access point. Finally, by connecting several access points to external directional antennas instead of their built-in omni-directional antennas, access points can provide multi-cells. This is useful for areas of heavy network traffic, because with this configuration, they are able to automatically *choose* the best access point with which to communicate. Roaming can also be provided for portable stations. Roaming is seamless, and it allows a work session to be maintained when moving from cell to cell (there is a momentary break in data flow).

Pros and Cons

Now, let us take a look at some of the pros and cons of wireless LANs. They include:

- Range/coverage
- Throughput
- Integrity and reliability
- Interoperability
- Simplicity
- Security
- Cost
- Scalability
- Power consumption

Range/Coverage. Most wireless LANs use radio frequencies (RF) to function (normally in the range of 2.4 GHz). RF is used because of its ability to propagate through objects. In wireless LANs, objects blocking the path of communication between access points limit the range that a wireless LAN can cover. Typically, the radius of coverage is anywhere from 100 feet to more than 300 feet. Coverage can be extended via roaming, which was previously defined.

Throughput. Airwave congestion contributes to data rates for a wireless LAN. Typical rates range from 1 to 10 Mbps. As in wired Ethernet LANs, wireless LANs slow down as traffic intensifies. In traditional Ethernet LANs, users experience a minimal difference in performance when going from wired to wireless LANs.

Integrity and Reliability. Radio interference can cause degradation in throughput. Such interference is rare in the workplace, and existing robust designs of WLAN prove that such problems are nothing compared to similar problems in existence with cellular phone connections. After all, wireless data technology has been used by the military for more than 50 years.

Interoperability. Wireless and wired infrastructures are interoperable, yet dependent on technology choice and vendor implementation. Currently, vendors make only their products to be interchangeable (adapters access points, etc.). The IEEE 802.11[5] ensures compliant products that are able to interoperate between vendors.

Simplicity. Wireless LANs, due to their nature, are transparent to a user's networking operating system (OS). This allows excellent compatibility to existing OS and minimizes having to use any type of new OS. Also, because only the access points of wireless LANs require cabling, moving, adding, and setting up is much easier. Finally, the portable nature of wireless LANs allows networking managers to set up systems at remote locations.

Security. The military has been using wireless technology for a long time; hence, security has been a strong criterion when designing anything that is wireless. Components are built so that it is extremely difficult for "*eavesdroppers*" to listen in on wireless LAN traffic. Complex encryption makes unauthorized access to network traffic virtually impossible.

Cost. Infrastructure costs depend on the number of access points and the number of wireless LAN adapters. Typically, access points range anywhere from $2000 to $3000. Wireless LAN adapters for standard computer platforms range anywhere from $400 to $2000. Installation and maintenance costs vary, depending on the size of the LAN. The costs of installing and maintaining a wireless LAN are lower, in general, when compared to the costs of installing and maintaining a traditional wired LAN.

Scalability. The complexity of each network configuration varies, depending on the number of nodes and access points. The ability of wireless LANs to be used in a simple or complex manner is what makes them so attractive to offices, hospitals, and universities.

Power Consumption. Power consumption of a wireless LAN is very low when compared to that of a handheld cellular phone. Wireless LANs must meet very strict standards posed by government and industry regulations — hence making them safe devices to have around the workplace. Finally, no detrimental health effects have ever been attributed to wireless LANs.

Technology Options

There is a range of available technologies on the market from which manufacturers can choose. For each individual technology, there are individual advantages and limitations.

Narrowband Technology. Narrowband technology uses narrow frequency on the radio signal. Communications channels are apportioned to this signal, each with different channel frequencies. This technology works just like a radio station. Each channel in this technology could be similar to a radio station on one's FM stereo. Nevertheless, the frequencies used in narrowband technology are much higher (in the GHz range).

Spread Spectrum. Spread spectrum is the most commonly used technology among wireless LANs component manufacturers. This technology has been adopted from the military and provides secure and reliable communication. The disadvantage is that it consumes a large amount of bandwidth. The advantage is that it produces a louder and more detectable signal. Within the spread spectrum, two types of spread spectrum radio technology exist: frequency hopping and direct sequence.

Frequency Hopping. Frequency hopping (FHSS) uses frequency diversity to combat interference. Basically, what happens is that the incoming digital stream gets shifted in frequency by a certain amount (determined by a code that spreads the signal power over a wide bandwidth). If the signal is seen by an unintended receiver, it will appear as a short-duration impulse noise.

Direct-Sequence Spread Spectrum Technology. Direct-sequence spread spectrum (DSSS) technology generates a chipping code that encodes each data bit. Effectively, this produces a low-power, wideband noise in the frequency domain (thus rejected by narrowband receivers). The greater the number of chips in the chipping code, the less likely it will be that the original data will be lost. DSSS is the most commonly used spread spectrum technology.

Infrared Technology. An infrared (IR) system is another available technology for wireless LANs for the data center. This technology uses very high frequencies, just below visible light in the electromagnetic (EM) spectrum, to carry data. The disadvantage here is that IR cannot penetrate opaque objects, hence limiting its line of sight. Ranges of IR are approximately 3 ft, which makes it useless for most WLAN data center applications.

IEEE 802.11 STANDARD

The IEEE 802.11 standard, as shown in Exhibit 2, is the new IEEE standard for wireless LANs. The goal of this standard is to standardize wireless LAN development in the industrial, scientific, and medicine (ISM) frequency bands allocated by the Federal Communications Commission (FCC) in the mid-1980s. The bands allocated include the frequency ranges 902 to 928 MHz, 2400 to 2483.5 MHz, and 5725 to 5850 MHz. The advantage of these ISM bands is that they do not require a license. As long as the device operating in the ISM bands meets special FCC regulations, no license of

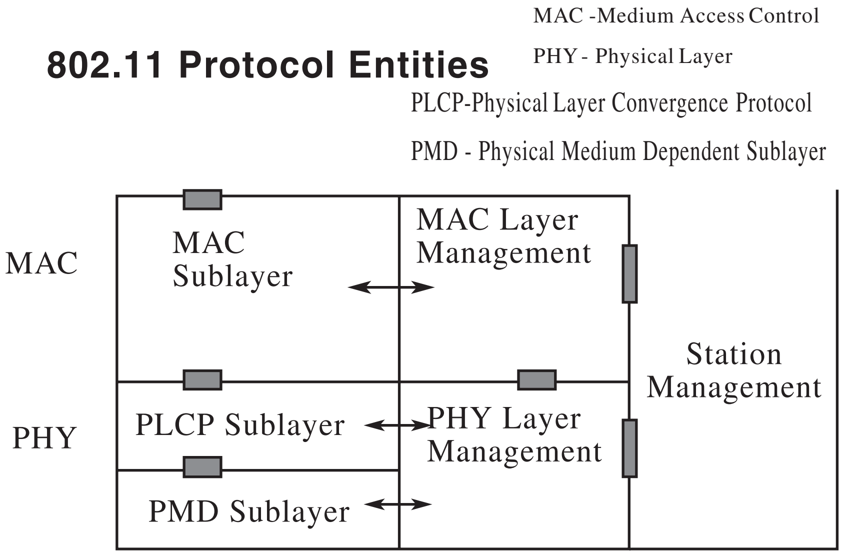


Exhibit 2. IEEE 802.11 Protocol

operation is necessary. The IEEE 802.11 standard focuses on media access control (MAC) and physical (PHY) protocol levels.

Medium Access Control

A medium access control (MAC) layer is built to allow overlapping of multiple networks in the same area and channel space. It has the ability to share mediums and to be robust for interference. The Distributed Coordination Function is used to provide efficient medium-sharing without any overlapping constrictions. Its frame formats are built to support the infrastructure and the ad hoc network support, as well as the wireless distribution system. The MAC layer provides the following services: authentication, deauthentication, privacy, MSDU delivery, association and disassociation, distribution, integration, and reassociation.

Physical Layer

A physical layer (PHY) is built to connect many stations together. Each station can transmit information to any other station in the network. As in other LANs, packets of the users' data are encoded according to the specific physical layer protocol and transmitted as a serial data stream over a physical medium to other stations on the LAN. Exhibit 2 shows a proposed configuration. Also, the decision to discard interpackets takes place at the physical layer as the result of an elasticity buffer overflow or underflow. As previously explained, within the physical layer are found the frequency hopping spread spectrum radio, direct-sequence spread spectrum radio, and infrared PHY Station Management (see Exhibit 2), which is used as a mediator between the MAC layer and the PHY layer.

IEEE 802.11 Future Development

Finally, a new specification known as the Internet-Access Point Protocol (IAPP) is now in existence. This specification goes beyond the work that was done by the IEEE 802.11 at the MAC and PHY (physical-layer specification) layers. This new standard works at higher OSI (open systems interconnection) layers to establish the way access points communicate across cells in the wired backbone. This new standard is backed by Aironet, Lucent Technologies, and Digital Ocean, Inc.

CONCLUSION AND SUMMARY

The fact that most data centers today have become a complex web of wireline and wireless service providers — providing both voice and data services to the end user at home, in the office, and walking or driving down the street — needs to be considered in any telecommunications policy initiative. The new wave of wireless service providers, while providing the consumer with more choices in services and features than ever before, presents a challenge to the public policymaker who tries to determine how to ensure that telecommunications services are made available to the broadest range of consumers. Competition will take care of that to a certain extent. However, where appropriate, government may need to step in on issues such as interconnection rights, mutual compensation, and numbering to ensure that new entrants are treated as equals by incumbent carriers. Furthermore, revision of universal service and enhanced 911 policies needs to take into consideration both the wireless and the wireline industries.

Additionally, the wireless industry is often faced with federal and state regulatory processes that can slow down the deployment of new networks. Federal guidelines regarding site acquisition and radio frequency emissions are necessary to ensure timely availability of new services. There continues to be a high demand for wireless services, and the industry is poised to meet that demand. However, public policy should be developed such that the promise of wireless services as an integral component of the data center is realized.

Finally, the wireless LAN can be very useful. To connect to a traditional wired LAN, a user must plug his or her computer into a wall or a floor LAN outlet. Portability and compatibility with all operating systems make wireless LAN an ideal choice for office intranets and the data center. The

IEE 802.11 standard promises to give data centers more control over wireless infrastructures — thus resulting in the blossoming of more and more wireless LANs in many offices.

END NOTES

1. The Composite CDMA/TDMA system is an air interface technology standardized for PCS in the 1.8-GHz band.
2. Broadband PCS refers to the family of mobile or portable radio services operating in the 1.8-GHz range and providing a wide variety of innovative digital voice and data services.
3. Narrowband PCS services include advanced voice paging, two-way acknowledgment paging, data messaging, and both one-way and two-way messaging.
4. Name ring a bell? The former research arm of AT&T, Telcordia Technologies is one of the world's top providers of telecom software (80 percent of the public telecom networks in the United States rely on its software) and consulting services. The company holds more than 800 patents and handles such tasks as doling out area codes. Created as the research unit for the seven Baby Bells during AT&T's 1984 breakup, Telcordia was then known as Bell Communications Research (Bellcore). After becoming a subsidiary of defense contractor Science Applications International, and with research only about 10 percent of its work, the company changed its name and began focusing on Internet-based technology. Clients include AT&T, GTE, and PCS Group.
5. IEEE 802.11 is the standard for wireless local area networks (WLANs) developed by the Institute of Electrical and Electronics Engineers (IEEE). It can be compared to the 802.3 standard for ethernet-wired LANs. The goal of this standard is to tailor a model of operation in order to resolve compatibility issues between manufacturers of WLAN equipment manufacturers. Thus far, the IEEE 802.11 standards committee is revising a version of a Media Access Control - Physical Level (MAC-PHY) level.

ABOUT THE AUTHOR

John R. Vacca is an information technology consultant and internationally known author based in Pomeroy, OH. Since 1982, John has authored 27 books and more than 330 articles in the areas of Internet and intranet security, programming, systems development, rapid application development, multimedia, and other Internet-related areas. John was also a configuration management specialist, computer specialist, and the computer security official for NASA's space station program (Freedom) and the International Space Station Program, from 1988 until his early retirement from NASA in 1995. John can be reached at jvacca@hti.net.

Chapter 21
Wireless Internet Security

Dennis Seymour Lee

Recalling the early days of the Internet, we can recount several reasons why the Internet came about. Some of these include:

- To provide a vast communication medium to share electronic information
- To create a multiple-path network which could survive localized outages
- To provide a means for computers from different manufacturers and different networks to talk to one another

Commerce and security, at that time, were not high on the agenda (with the exception of preserving network availability). The thought of commercializing the Internet in the early days was almost unheard of. In fact, it was considered improper etiquette to use the Internet to sell products and services. Commercial activity and its security needs are more recent developments on the Internet, having intensified in importance in the last few years.

Today, in contrast, the wireless Internet is being designed from the very beginning with commerce as its main driving force. Nations and organizations around the globe are spending millions, even billions of dollars, to buy infrastructure, transmission frequencies, technology, and applications in the hopes of drawing business. In some ways, this has become the "land rush" of the new millennium. It stands to reason then that security must play a critical role early on as well — where money changes hands, security will need to accompany this activity.

Although the wireless industry is still in its infancy, the devices, the infrastructure, and the application development for the wireless Internet are rapidly growing on a worldwide scale. Those with foresight will know that security must fit in early into these designs. The aim of this chapter is to highlight some of the significant security issues in this emerging industry that need addressing. These are concerns which any business wishing

0-8493-1161-6/02/$0.00+$1.50

to deploy a wireless Internet service or application will need to consider to protect its own businesses and its customers, and to safeguard its investments in this new frontier.

Incidentally, the focus of this chapter is not on accessing the Internet using laptops and wireless modems. That technology, which has been around for many years, in many cases is an extension of traditional wired Internet access. Neither will this chapter focus on wireless LANs and Bluetooth, which are not necessarily Internet-based, but deserve chapters of their own. Rather, the concentration will be on portable Internet devices, which inherently have far less computing resources than regular PCs, such as cell phones and PDAs (personal digital assistants). Therefore, these devices require different programming languages, protocols, encryption methods, and security perspectives to cope with the different technology. It is important to note, however, that despite their smaller sizes and limitations, these devices have a significant impact on information security mainly because of the electronic commerce and Intranet-related applications that are being designed for them.

WHO IS USING THE WIRELESS INTERNET?

Many studies and estimates are available today that suggest the number of wireless Internet users will soon surpass the millions of "wired" Internet users. The assumption is based on the many more millions of worldwide cell phone users that are already out there, a population that grows by the thousands every day. If every one of these mobile users chose to access the Internet through cell phones, indeed that population could easily exceed the number of wired Internet users by several times. It is this very enormous potential that has many businesses devoting substantial resources and investments in the hopes of capitalizing on this growing industry.

The wireless Internet is still very young, however. Many mobile phone users do not yet have access to the Internet through their cell phones. Many are taking a "wait and see" attitude to see what services will be available. Most who do have wireless Internet access are early adopters who are experimenting with the potential of what this service could provide. Because of the severe limitations in the wireless devices — the tiny screens, the extremely limited bandwidth, as well as other issues — most users who have both wired and wireless Internet access will admit that, for today, the wireless devices will not replace their desktop computers and notebooks anytime soon as their primary means of accessing the Internet. Many admit that "surfing the Net" using a wireless device today could become a disappointing exercise. Most of these wireless Internet users have expressed the following frustrations:

- It is too slow to connect to the Internet.
- Mobile users can be disconnected in the middle of a session when they are on the move.
- It is cumbersome to type out sentences using a numeric keypad.
- It is expensive using the wireless Internet, especially when billed on a "per-minute" basis.
- There are very few or no graphics display capabilities on the wireless devices.
- The screens are too small and users have to scroll constantly to read a long message.
- There are frequent errors when surfing Web sites (mainly because most Web sites today are not yet wireless Internet-compatible).

At the time of this writing, the one notable exception to these disappointments is found in Japan. The telecommunications provider NTT DoCoMo has experienced phenomenal growth in the number of wireless Internet subscribers, using a wireless application environment called i-Mode (as opposed to Wireless Application Protocol or WAP). For many in Japan, connection using a wireless phone is their only means of accessing the Internet. In many cases, wireless access to the Internet is far cheaper than wired access, especially in areas where the wired infrastructure is expensive to set up. I-Mode users have the benefit of "always online" wireless connections to the Internet, color displays on their cell phones, and even graphics, musical tones, and animation. Perhaps Japan's success with the wireless Internet offers an example of what can be achieved in the wireless arena, given the right elements.

WHAT TYPES OF APPLICATIONS ARE AVAILABLE?

Recognizing the frustrations and limitations of today's wireless technology, many businesses are designing their wireless devices and services, not necessarily as replacements for wired Internet access, but as specialized services that extend what the wired Internet could offer. Most of these services highlight the attractive convenience of portable informational access, anytime and anywhere, without having to sit in front of a computer — essentially, Internet services you can carry in your pocket. Clearly, the information would have to be concise, portable, useful, and easy to access. Examples of mobile services available or being designed today include:

- Shopping online using a mobile phone; comparing online prices with store prices while inside an actual store
- Getting current stock prices, trading price alerts, trade confirmations, and portfolio information anywhere
- Performing bank transactions and obtaining account information
- Obtaining travel schedules and booking reservations

- Obtaining personalized news stories and weather forecasts
- Receiving the latest lottery numbers
- Obtaining the current delivery status for express packages
- Reading and writing e-mail "on the go"
- Accessing internal corporate databases such as inventory, client lists, and so on
- Getting map directions
- Finding the nearest ATM machines, restaurants, theaters, and stores, based on the user's present location
- Dialing "911" and having emergency services quickly triangulate the caller's location
- Browsing a Web site and speaking live with the site's representative, all within the same session

Newer and more innovative services are in the works. As with any new and emerging technology, wireless services and applications are often surrounded by much hope and hype, as well as some healthy skepticism. But as the technology and services mature over time, yesterday's experiments can become tomorrow's standards. The Internet is a grand example of this evolving progress. Development of the wireless Internet will probably go through the same evolutionary cycle, although probably at an even faster pace.

Like any new technology, however, security and safety issues can damage its reputation and benefits if they are not included intelligently into the design from the very beginning. It is with this purpose in mind that this chapter is written.

Since the wireless Internet covers a lot of "territory," the same goes for its security as well. We will cover security issues as they relate to the wireless Internet in a few select categories, starting from the transmission methods to the wireless devices, and ending with some of the infrastructure components themselves.

HOW SECURE ARE THE TRANSMISSION METHODS?

For many years, it was public knowledge that analog cell phone transmissions are fairly easy to intercept. It has been a known problem for as long as analog cell phones have been available. They are easily intercepted using special radio scanning equipment. For this reason, as well as many others, many cell phone service providers have been promoting digital services to their subscribers and reducing analog to a legacy service.

Digital cell phone transmissions, on the other hand, are typically more difficult to intercept. It is on these very same digital transmissions that most of the new wireless Internet services are based.

However, there is no single method for digital cellular transmission. In fact, there are several different methods for wireless transmission available today. For example, in the United States, providers such as Verizon and Sprint use largely CDMA (Code Division Multiple Access), whereas AT&T uses largely TDMA (Time Division Multiple Access), and Voicestream uses GSM (Global Systems for Mobile Communications). Other providers like Cingular offer more than one method (TDMA and GSM), depending on the geographic location. All these methods differ in the way they use the radio frequencies and the way they allocate users on those frequencies. We will cover each of these in more detail.

Cell phone users are generally not concerned with choosing a particular transmission method if they want wireless Internet access, nor do they really care to. Instead, most users select their favorite wireless service provider when they sign up for service. It is generally transparent to the user which transmission method their provider has implemented. It is, however, an entirely different matter for the service provider. Whichever method the provider implements has significant bearing on its infrastructure. For example, the type of radio equipment the provider uses, the location and number of transmission towers to deploy, the amount of traffic it can handle, and the type of cell phones to sell to its subscribers are all directly related to the digital transmission method chosen.

FDMA

All cellular communications, analog or digital, are transmitted using radio frequencies that are purchased by, or allocated to, the wireless service provider. Each service provider typically purchases licenses from the respective government to operate a spectrum of radio frequencies.

Analog cellular communications typically operate on what is called Frequency Division Multiple Access (or FDMA) technology. With FDMA, each service provider divides its spectrum of radio frequencies into individual frequency channels. Each channel is a specific frequency that supports a one-way communication session, and each channel has a width of 10 to 30 kilohertz (kHz). For a regular two-way phone conversation, every cell phone caller would be assigned two frequency channels, one to send and one to receive.

Because each phone conversation occupies two channels (two frequencies), it is not too difficult for specialized radio scanning equipment to tap into a live analog phone conversation once the equipment has tuned into the right frequency channel. There is very little privacy protection in analog cellular communications if no encryption is added.

TDMA

Digital cellular signals, on the other hand, can operate on a variety of encoding techniques, most of which are resistant to analog radio frequency scanning. (Please note that the word "encoding" in wireless communications does not mean encryption. "Encoding" here usually refers to converting a signal from one format to another, for example, from a wired signal to a wireless signal.)

One such technique is called Time Division Multiple Access or TDMA. Like FDMA, TDMA divides the radio spectrum typically into multiple 30-kHz frequency channels (sometimes called frequency carriers). Every two-way communication requires two of these frequency channels, one to send and one to receive. But in addition, TDMA subdivides each frequency channel further into three to six time slots called voice/data channels, so that now up to six digital voice or data sessions can take place using the same frequency. With TDMA, a service provider can handle more calls at the same time, compared to FDMA. This is accomplished by assigning each of the six sessions a specific time slot within the same frequency. Each time slot (or voice/data channel) is about 7 ms in duration. The time slots are arranged and transmitted over and over again in rapid rotation. Voice or data for each caller is placed into the time slot assigned to that caller and then transmitted. Information from the corresponding time slot is quickly extracted and reassembled at the receiving cellular base station to piece together the conversation or session. Once that time slot, or voice/data channel, is assigned to a caller, it is dedicated to that caller for the duration of the session, until it terminates. In TDMA, a user is not assigned an entire frequency, but shares the frequency with other users, each with an assigned time slot.

As of the writing of this chapter, there have not been many publicized cases of eavesdropping of TDMA phone conversations and data streams as they travel across the wireless space. Access to special types of equipment or test equipment would probably be required to perform such a feat. It is possible that an illegally modified TDMA cell phone could also do the job.

However, this does not mean that eavesdropping is unfeasible. If we are talking about a wireless Internet session, consider the full path that such a session takes. For a mobile user to communicate with an Internet Web site, a wireless data signal from the cell phone will eventually be converted into a wired signal before traversing the Internet itself. As a wired signal, the information can travel across the Internet in clear-text until it reaches the Web site. Although the wireless signal itself may be difficult to intercept, once it becomes a wired signal, it is subject to the same interception vulnerabilities as all unencrypted communications traversing the Internet. Always as a precaution, if there is confidential information being transmitted over the Internet, regardless of the method, it is necessary to encrypt that session from end to end. We will discuss encryption in a later section.

GSM

Another method of digital transmission is Global Systems for Mobile Communications (or GSM). GSM is actually a term that covers more than just the transmission method alone. It covers the entire cellular system, from the assortment of GSM services to the actual GSM devices themselves. GSM is used largely in Europe.

As a digital transmission method, GSM uses a variation of TDMA. Like FDMA and TDMA, the GSM service provider divides the allotted radio-frequency spectrum into multiple frequency channels. This time, each frequency channel has a much larger width of 200 kHz. Again like FDMA and TDMA, each GSM cellular phone uses two frequency channels, one to send and one to receive.

Like TDMA, GSM further subdivides each frequency channel into time slots called voice/data channels. However, with GSM, there are eight time slots, so that now up to eight digital voice or data sessions can take place using the same frequency. Again like TDMA, once that time slot (or voice/data channel) is assigned to a caller, it is dedicated to that caller for the duration of the session, until it terminates.

GSM has additional features that enhance security. Each GSM phone uses a Subscriber Identity Module (or SIM). A SIM can look like a credit card-sized smart-card or a postage stamp-sized chip. This removable SIM is inserted into the GSM phone during usage. The smart-card or chip contains information pertaining to the subscriber, such as the cell phone number belonging to the subscriber, authentication information, encryption keys, directory of phone numbers, and short saved messages belonging to that subscriber. Since the SIM is removable, the subscriber can take this SIM out of one phone and insert it into another GSM phone. The new phone with the SIM will then take on the identity of the subscriber. The user's identity is not tied to a particular phone but to the removable SIM itself. This makes it possible for a subscriber to use or upgrade to different GSM phones, without changing phone numbers. It is also possible to rent a GSM phone in another country, even if that country uses phones that transmit on different GSM frequencies. This arrangement works, of course, only if the GSM service providers from the different countries have compatible arrangements with each other.

The SIM functions as an authentication tool because the GSM phones are useless without it. Once the SIM is inserted into a phone, the user is prompted to put in his or her personal identification number (PIN) associated with that SIM (if the SIM is PIN-enabled). Without the correct PIN number, the phone will not work.

Besides authenticating the user to the phone, the SIM is also used to authenticate the phone to the phone network itself during connection. By using the authentication (or Ki) key in the SIM, the phone authenticates to the service provider's Authentication Center during each call. The process employs a challenge-response technique, similar in some respects to using a token card to log a PC remotely onto a network.

The keys in the SIM have another purpose besides authentication. The encryption (or Kc) key generated by the SIM can be used to encrypt communications between the mobile phone and the service provider's transmission equipment for confidentiality. This encryption prevents eavesdropping, at least between these two points.

GSM transmissions, like TDMA, are difficult, but not impossible, to intercept using radio-frequency scanning equipment. A frequency can have up to eight users on it, making the digital signals difficult to extract. By adding encryption using the SIM card, GSM can add yet another layer of security against interception.

However, when it comes to wireless Internet sessions, this form of encryption does not provide end-to-end protection. Only part of the path is actually protected. This is similar to the problem we mentioned earlier with TDMA Internet sessions. A typical wireless Internet session takes both a wireless and a wired path. GSM encryption protects only the path between the cell phone and the service provider's transmission site — the wireless portion. The rest of the session through the wired Internet — from the service provider's site to the Internet Web site — can still travel in the clear. You would need to add end-to-end encryption if you need to keep the entire Internet session confidential.

CDMA

Another digital transmission method is called Code Division Multiple Access (or CDMA). CDMA is based on spread spectrum, a transmission technology that has been used by the U.S. military for many years to make radio communications more difficult to intercept and jam. Qualcomm is one of the main pioneers incorporating CDMA spread spectrum technology into the area of cellular phones.

Instead of dividing a spectrum of radio frequencies into narrow frequency bands or time slots, as we saw earlier, CDMA uses a very large portion of that radio spectrum, again called a frequency channel. The frequency channel has a wide width of 1.25 megahertz (MHz). For duplex communication, each cell phone uses two of these wide CDMA frequency channels, one to send and one to receive.

During communication, each voice or data session is first converted into a series of data signals. Next, the signals are marked with a unique code to indicate that they belong to a particular caller. This code is called a pseudo-random noise (PN) code. Each mobile phone is assigned a new PN code by the base station at the beginning of each session. These coded signals are then transmitted by spreading them out across a very wide radio-frequency spectrum. Because the channel width is very large, it has the capacity to handle many other user sessions at the same time, each session again tagged by unique PN codes to associate them to the appropriate caller.

A CDMA phone receives transmissions by using the appropriate pseudo-random noise code to pick out the data signals that are destined for it and ignores all the other encoded signals.

With CDMA, cell phones communicating with the base stations all share the same wide frequency channels. What distinguishes each caller is not the frequency used (as in FDMA), nor the time slot within a particular frequency (as in TDMA or GSM), but the pseudo-random noise code assigned to that caller. With CDMA, a voice/data channel is a data signal marked with a unique PN code.

Intercepting a single CDMA conversation would be difficult since its digital signals are spread out across a very large spectrum of radio frequencies. The conversation does not reside on just one frequency alone, making it hard to scan. Also, without knowledge of the pseudo-random noise code, an eavesdropper would not be able to extract the relevant session from the many frequencies used. To complicate interception even more, the entire channel width is populated by many other callers at the same time, creating a vast amount of noise for anyone trying to intercept the call.

However, as we saw earlier with the other digital transmission methods, Internet sessions using CDMA cell phones are not impossible to intercept. As before, although the CDMA digital signals themselves can be difficult to intercept, once these wireless signals are converted into wired signals, the latter signals can be intercepted as they travel across the Internet. Without using end-to-end encryption, wireless Internet sessions are as vulnerable as other unencrypted communications traveling over the Internet.

Other Methods

There are additional digital transmission methods, many of which are derivatives of the types we have already mentioned, and some of which are still under development. Some of these that are under development are called "third generation" or "3G" transmission methods. Second generation, or 2G technologies, like TDMA, GSM, and CDMA, offer transmission speeds of 9.6 to 14.4 kbps (kilobits per second), which is slower than today's typical modem speeds. 3G technologies, on the other hand, are designed to transmit much faster and carry larger amounts of data. Some will be capable of providing high-speed Internet access as well as video transmission. Below is a partial listing of other digital transmission methods, including those in the 3G category:

- *iDEN* (Integrated Digital Enhanced Network) — This is based on TDMA and is a 2G transmission method. Besides sending voice and data, it can also be used for two-way radio communications between two iDEN phones, much like "walkie-talkies."
- *PDC* (Personal Digital Communications) — This is based on TDMA and is a 2G transmission method used largely in Japan.
- *GPRS* (General Packet Radio Service) — This is a 2.5G (not quite 3G) technology based on GSM. It is a packet-switched data technology that provides "always online" connections, which means the subscriber can stay logged onto the phone network all day but uses it only if there is actual data to send or receive. Maximum data rates are estimated to be 115 kbps.
- *EDGE* (Enhanced Data Rates for Global Evolution) — This is a 3G technology based on TDMA and GSM. Like GPRS, it features "always online" connections using packet-switched data technologies. Maximum data rates are estimated to be 384 kbps.
- *UMTS* (Universal Mobile Telecommunications System) — This is a 3G technology based on GSM. Maximum data rates are estimated at 2 Mbps (megabits per second).
- *CDMA2000 and W-CDMA* (Wideband CDMA) — These are two 3G technologies based on CDMA. CDMA2000 is a more North American design, whereas W-CDMA is more European and Japanese oriented. Both provide maximum data rates estimated at 384 kbps for slow-moving mobile units, and at 2 Mbps for stationary units.

Regardless of the methods or the speeds, as we mentioned earlier, the need for end-to-end encryption will still be a requirement if confidentiality is needed between the mobile device and the Internet or Intranet site. Since wireless Internet communications encompass both wireless and wired-based transmissions, encryption features covering just the wireless portion of the communication are clearly not enough. For end-to-end privacy protection, the applications and the protocols have a role to play, as we will see later in this chapter.

HOW SECURE ARE THE WIRELESS DEVICES?

Internet security, as many of us have seen it applied to corporate networks today, can be difficult to implement on wireless phones and PDAs for a variety of reasons. Most of these devices have limited CPUs, memory, bandwidth, and storage abilities. As a result, many have disappointingly slow and limited computing power. Robust security features that can take less than a second to process on a typical workstation can take potentially many minutes on a wireless device, making them impractical or inconvenient for the mobile user. Since many of these devices have merely a fraction of the hardware capabilities found on typical workstations, the security features on portable devices are often lightweight or even nonexistent, from an Internet security perspective. Yet these same devices are now being used to log into sensitive corporate Intranets, or to conduct mobile commerce and banking. Although these wireless devices are smaller in every way, their security needs are just as significant as before. It would be a mistake for corporate IT and information security departments to ignore these devices as they start to populate the corporate network. After all, these devices do not discriminate; they can be designed to tap into the same corporate assets as any other node on a network. We will examine some of the security aspects as they relate to these devices.

Authentication

The process of authenticating wireless phone users has gone through many years of implementation and evolution. It is probably one of the most reliable security features of digital cell phones today, given the many years of experience service providers have had in trying to reduce the theft of wireless services. Since the service providers have a vested interest in knowing whom to charge for the use of their services, authenticating the mobile user is of utmost importance.

As we had mentioned earlier, GSM phones use SIM cards or chips, which contain authentication information about the user. SIMs typically carry authentication and encryption keys, authentication algorithms, identification information, phone numbers belonging to the subscriber, and so on. They allow the users to authenticate to their own phones and to the phone network to which they subscribe.

In North America, TDMA and CDMA phones use a similarly complex method of authentication as in GSM. Like GSM, the process incorporates keys, Authentication Centers, and challenge-response techniques. However, since TDMA and CDMA phones do not generally use removable SIM cards or chips, these phones instead rely on the authentication information embedded into the handset. The user's identity is therefore tied to the single mobile phone itself.

The obvious drawback is that for authentication purposes, TDMA and CDMA phones offer less flexibility when compared to GSM phones. To deploy a new authentication feature with a GSM phone, in many cases all that is needed is to update the SIM card or chip. On the other hand, with TDMA and CDMA, deploying new authentication features would probably require users to buy new cell phones — a more expensive way to go.

Because it is easier to update a removable chip than an entire cell phone, it is likely that you will find more security features and innovations offered for GSM as a result.

One important note, however, is that this form of authentication does not necessarily apply to Internet-related transactions. It merely authenticates the mobile user to the service provider's phone network, which is only one part of the transmission if we are speaking about Internet transactions. For securing end-to-end Internet transactions, mobile users still need to authenticate the Internet Web servers they are connecting to, to verify that indeed the servers are legitimate. Likewise, the Internet Web servers need to authenticate the mobile users that are connecting to it, to verify that they are legitimate users and not impostors. The wireless service providers, however, are seldom involved in providing full end-to-end authentication service, from mobile phone to Internet Web site. That responsibility usually falls to the owners of the Internet Web servers and applications.

Several methods for providing end-to-end authentication are being tried today at the application level. Most secure mobile commerce applications use IDs and passwords, old standbys, which, of course, have their limitations since they provide only single-factor authentication. Other organizations are experimenting with GSM SIMs by adding additional security ingredients such as public/private key pairs, digital certificates, and other public key infrastructure (PKI) components into the SIMs. However, since the use of digital certificates can be process-intensive, cell phones and handheld devices typically use lightweight versions of these security components. To accommodate the smaller processors in wireless devices, the digital certificates and their associated public keys may be smaller or weaker than those typically deployed on desktop Web browsers, depending on the resources available on the wireless device.

Still other organizations are experimenting with using elliptic-curve cryptography (ECC) for authentication, digital certificates, and public-key encryption on the wireless devices. ECC is an ideal tool for mobile devices because it can offer strong encryption capabilities, but it requires fewer computing resources than other popular forms of public-key encryption. Certicom is one of the main pioneers incorporating ECC for use on wireless devices.

As more and more developments take place with wireless Internet authentication, it becomes clear that, in time, these Internet mobile devices will become full-fledged authentication devices, much like tokens, smart cards, and bank ATM cards. If users begin conducting Internet commerce using these enhanced mobile devices, securing those devices from loss or theft now becomes a priority. With identity information embedded into the devices or the removable SIMs, losing these could mean that an impostor can now conduct electronic commerce transactions using that stolen identity. With a mobile device, the user, of course, plays the biggest role in maintaining its overall security. Losing a cell phone that has Internet access and an embedded public/private key pair can be potentially as disastrous as losing a bank ATM card with its associated PIN written on it, or worse. If a user loses such a device, it is essential to inform the service provider immediately about the loss and suspend its use.

Confidentiality

Preserving confidentiality on wireless devices poses several interesting challenges. Typically, when we access a Web site with a browser and enter a password to gain entry, the password we type is masked with asterisks or some other placeholder to prevent others from seeing your actual password on your screen. With cell phones and handheld devices, masking the password could create problems during typing. With cell phones, letters are often entered using the numeric keypad — a method that is cumbersome and tedious for many users. For example, to type the letter "R," you need to press the number 7 key three times to get to the right letter. If the result is masked, it is not clear to the user what letter was actually submitted. Because of this inconvenience, some mobile Internet applications do away with masking so that the entire password is displayed on the screen in the original letters. Other applications display each letter of the password for a few seconds first as they are being entered, before masking each with a placeholder afterward. This gives the user some positive indication that the correct letters were indeed entered, while still preserving the need to mask the password on the device's screen for privacy. The latter approach is probably the more sensible of the two and should be the one the application designers adopt.

Another challenge to preserving confidentiality is making sure confidential information such as passwords and credit card numbers are purged from the mobile device's memory after they are used. Many times, such sensitive information is stored as variables by the wireless Internet application and subsequently cached in the memory of the device. There have been documented cases where credit card numbers left in the memory of cell phones were reusable by other people who borrowed the same phones to access the same sites. Once again, the application designers are the chief architects to preserving the confidentiality here. It is important that programmers design an application to clear the mobile device's memory of sensitive information when the user finishes using that application. Although leaving such information in the memory of the device may spare the user of having to re-enter it the next time, it is, however, as risky as writing the associated PIN or password on a bank ATM card itself.

Still another challenge in preserving confidentiality is ensuring that sensitive information is kept private as it travels from the wireless device to its destination on the Internet, and back. Traditionally for the wired Internet, most Web sites use Secure Sockets Layer (SSL) or its successor, Transport Layer Security (TLS), to encrypt the entire path end-to-end, from the client to the Web server. However, many wireless devices, particularly cell phones, lack the computing power and bandwidth to run SSL efficiently. One of the main components of SSL is RSA public key encryption. Depending on the encryption strength applied at the Web site, this form of public key encryption can be processor- and bandwidth-intensive and can tax the mobile device to the point where the communication session itself becomes too slow to be practical.

Instead, wireless Internet applications that are developed using the Wireless Application Protocol (WAP) utilize a combination of security protocols. Secure WAP applications use both SSL and WTLS (Wireless Transport Layer Security) to protect different segments of a secure transmission. Typically, SSL protects the wired portion of the connection and WTLS protects largely the wireless portion. Both are needed to provide the equivalent of end-to-end encryption.

WTLS is similar to SSL in operation. However, although WTLS can support either RSA or elliptic-curve encryption, elliptic curve is probably preferred because the latter provides strong encryption capabilities but is more compact and faster than RSA.

WTLS differs from SSL in other ways. WTLS is built to provide encryption services for a slower and less resource-intensive environment, whereas SSL could tax such an environment. This is because SSL encryption requires a reliable transport protocol, particularly TCP (Transmission Control Protocol, a part of TCP/IP). TCP provides error detection, communication acknowledgments, and re-transmission features to ensure reliable network connections back and forth. But because of these features, TCP requires more bandwidth and resources than what typical wireless connections and devices can provide. Most mobile connections today are low bandwidth and slow, and not designed to handle the constant, back-and-forth error-detection traffic that TCP creates.

Realizing these limitations, the WAP Forum, the group responsible for putting together the standards for WAP, designed a supplementary protocol stack that is more suitable for the wireless environment. Since this environment typically has low connection speeds, low reliability, and low bandwidth in order to compensate, the protocol stack uses compressed binary data sessions and is more tolerant of intermittent coverage. The WAP protocol stack resides in layers 4, 5, 6, and 7 of the OSI reference model. The WAP protocol stack works with UDP (User Datagram Protocol) for IP-based networks and WDP (Wireless Datagram Protocol) for non-IP networks. WTLS, which is the security protocol from the WAP protocol stack, can be used to protect UDP or WDP traffic in the wireless environment.

Because of these differences between WTLS and SSL, as well as the different underlying environments that they work within, an intermediary device such as a gateway is needed to translate the traffic going from one environment into the next. This gateway is typically called a WAP gateway. We will discuss the WAP gateway in more detail in the infrastructure section below.

Malicious Code and Viruses

The number of security attacks on wireless devices has been small compared to the many attacks against workstations and servers. Part of the reason is due to the very simple fact that most mobile devices, particularly cell phones, lack sufficient processors, memory, or storage that malicious code and viruses could exploit. For example, a popular method for spreading viruses today is by hiding them in file attachments to e-mail. However, many mobile devices, particularly cell phones, lack the ability to store or open e-mail attachments. This makes mobile devices relatively unattractive as targets since the damage potential is relatively small.

However, mobile devices are still vulnerable to attack and will become increasingly more so as they evolve with greater computing, memory, and storage capabilities. With greater speeds, faster downloading abilities, and better processing, mobile devices can soon become the equivalent of today's workstations, with all its exploitable vulnerabilities. As of the writing of this chapter, cell phone manufacturers were already announcing that the next generation of mobile phones will support languages such as Java so that users can download software programs such as organizers, calculators, and games onto their Web-enabled phones. However, on the negative side, this also opens up more opportunities for users to unwittingly download malicious programs (or "malware") onto their own devices. The following adage applies to mobile devices: "The more brains they have, the more attractive they become as targets."

HOW SECURE ARE THE NETWORK INFRASTRUCTURE COMPONENTS?

As many of us who have worked in the information security field know, security is usually assembled using many components, but its overall strength is only as good as its weakest link. Sometimes it does not matter if you are using the strongest encryption available over the network and the strongest authentication at the devices. If there is a weak link anywhere along the chain, attackers will focus on this vulnerability and may eventually exploit it, choosing a path that requires the least effort and the least number of resources.

Since the wireless Internet world is still relatively young and a work in progress, vulnerabilities abound, depending on the technology you have implemented. We will focus in this section on some infrastructure vulnerabilities for those who are using WAP.

The "Gap in WAP"

Encryption has been an invaluable tool in the world of E-commerce. Many online businesses use SSL (Secure Sockets Layer) or TLS (Transport Layer Security) to provide end-to-end encryption to protect Internet transactions between the client and the Web server.

When using WAP, however, if encryption is activated for the session, there are usually two zones of encryption applied, each protecting the two different halves of the transmission. SSL or TLS is generally used to protect the first path, between the Web server and an important network device called the WAP gateway that we mentioned earlier. WTLS (Wireless Transport Layer Security) is used to protect the second path, between the WAP gateway and the wireless mobile device.

The WAP gateway is an infrastructure component needed to convert wired signals into a less bandwidth-intensive and compressed binary format, compatible for wireless transmissions. If encryption such as SSL is used during a session, the WAP gateway will need to translate the SSL-protected transmission by decrypting this SSL traffic and re-encrypting it with WTLS, and vice versa in the other direction. This translation can take just a few seconds, but during this brief period, the data sits in the memory of the WAP gateway decrypted and in the clear, before it is re-encrypted using the second protocol. This brief period in the WAP gateway — some have called it the "Gap in WAP" — is an exploitable vulnerability. It depends on where the WAP gateway is located, how well it is secured, and who is in charge of protecting it.

Clearly, the WAP gateway should be placed in a secure environment. Otherwise, an intruder attempting to access the gateway can steal sensitive data while it transitions in clear-text. The intruder can also sabotage the encryption at the gateway, or even initiate a denial of service or other malicious attack on this critical network component. Besides securing the WAP gateway from unauthorized access, proper operating procedures should also be applied to enhance its security. For example, it is wise not to save any of the clear-text data onto disk storage during the decryption and re-encryption process. Saving this data onto log files, for example, could create an unnecessarily tempting target for intruders. In addition, the decryption and re-encryption should operate in memory only and proceed as quickly as possible. Furthermore, to prevent accidental disclosure, the memory should be properly overwritten, thereby purging any sensitive data before that memory is reused.

WAP Gateway Architectures

Depending on the sensitivity of the data and the liability for its unauthorized disclosure, businesses offering secure wireless applications (as well as their customers) may have concerns about where the WAP gateway is situated, how it is protected, and who is protecting it. We will examine three possible architectures and discuss the security implications behind each.

WAP Gateway at the Service Provider

In most cases, the WAP gateways are owned and operated by the wireless service providers. Many businesses that deploy secure wireless applications today rely on the service provider's WAP gateway to perform the SSL-to-WTLS encryption translation. This implies that the business owners of the sensitive wireless applications, as well as their users, are entrusting the wireless service providers to keep the WAP gateway and the sensitive data that passes through it safe and secure. Exhibit 1 below shows an example of such a setup, where the WAP gateway resides within the service provider's secure environment. If encryption is applied in a session between the user's cell phone and the application server behind the business' firewall, the path between the cell phone to the service provider's WAP gateway is typically encrypted using WTLS. The path between the WAP gateway to the business host's application server is encrypted using SSL or TLS.

A business deploying secure WAP applications using this setup should realize, however, that it cannot guarantee end-to-end security for the data since it is decrypted, exposed in clear-text for a brief moment, and then re-encrypted, all at an outside gateway that is away from its control. The WAP gateway is generally housed in the wireless service provider's data center and attended by those who are not directly accountable to the businesses. Of course, it is in the best interest of the service provider to keep the WAP gateway in a secure manner and location.

Sometimes, to help reinforce that trust, the businesses may wish to conduct periodic security audits on the service provider's operation of the WAP gateways to ensure that the risks are minimized. Bear in mind, however, that by choosing this path, the business may need to inspect many WAP gateways from many different service providers. A service provider sets up the WAP gateway primarily to provide Internet access to its own wireless phone subscribers. If users are dialing into a business' secure Web site, for example, from 20 different wireless service providers around the world, then the business may need to audit the WAP gateways belonging to these 20. This, unfortunately, is a formidable task and an impractical

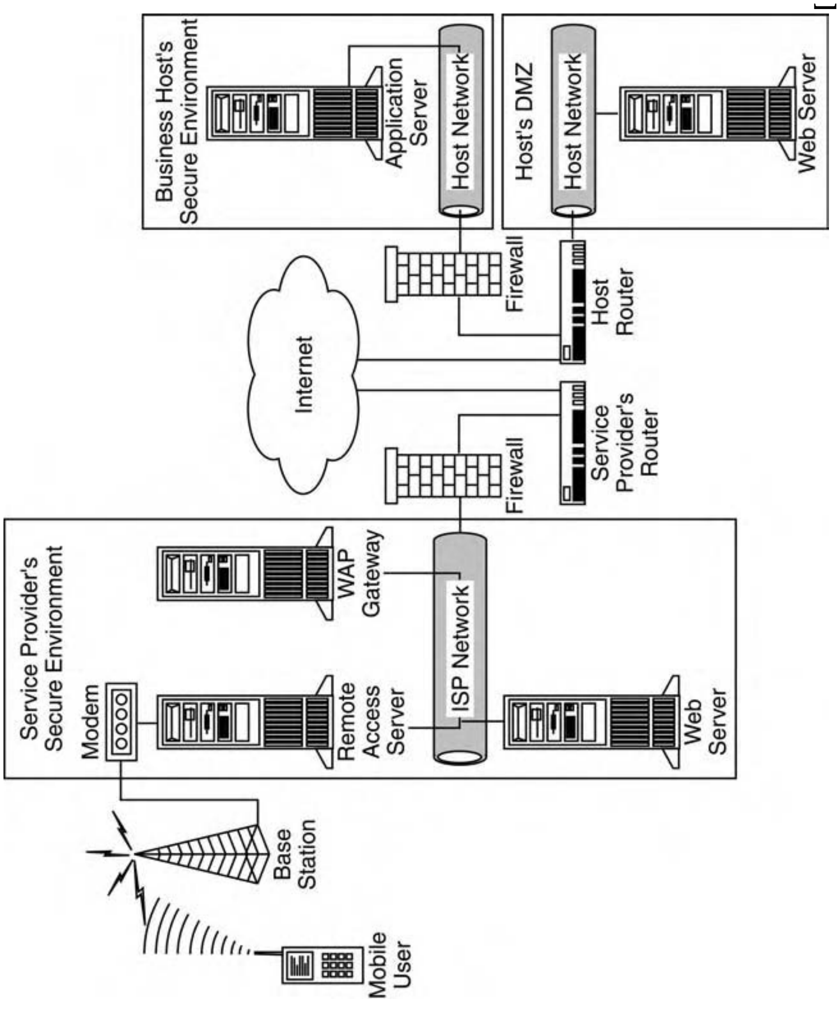


Exhibit 1. WAP Gateway at the Service Provider

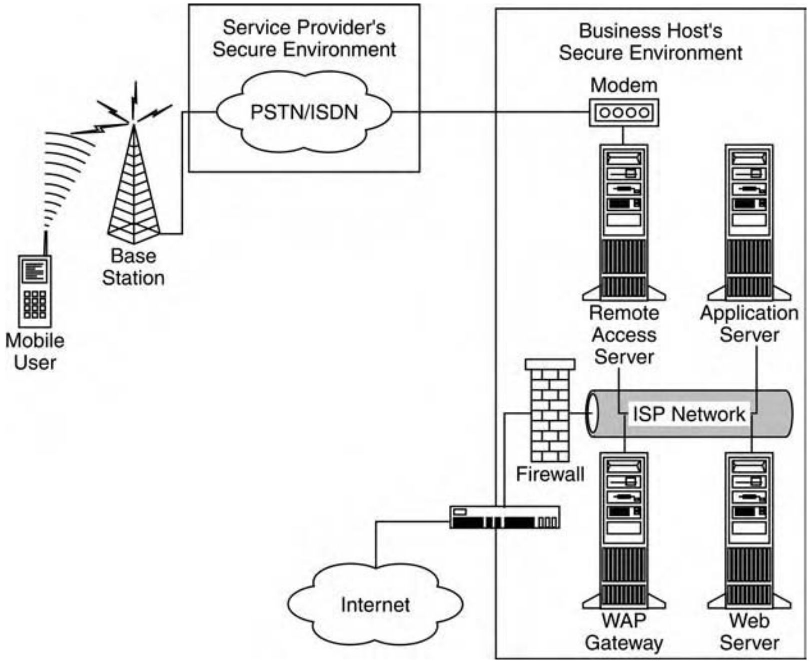


Exhibit 2. WAP Gateway at the Host

method of ensuring security. Each service provider may apply a different method for protecting its own WAP gateway, if that provider protects it at all. Furthermore, in many cases, the wireless service providers are accountable to their own cell phone subscribers, not necessarily to the countless businesses that are hosting secure Internet applications, unless there is a contractual arrangement to do so.

WAP Gateway at the Host

Some businesses and organizations, particularly in the financial, health care, or government sectors, may have legal requirements to keep their customers' sensitive data protected. Having such sensitive data exposed outside of the organization's internal control may pose an unnecessary risk and liability. To some, the "Gap in WAP" presents a broken pipeline, an obvious breach of confidentiality that is just waiting to be exploited. For those who find such a breach unacceptable, one possible solution is to place the WAP gateway at the business host's own protected network, bypassing the wireless service provider's WAP gateway entirely. Exhibit 2 shows an example of such a setup. Nokia, Ericsson, and Ariel Communications are just a few of the vendors offering such a solution.

This approach has the benefit of keeping the WAP gateway and its WTLS-SSL translation process in a trusted location, within the confines of the same organization that is providing the secure Web applications. Using this setup, users are typically dialing directly from their wireless devices, through their service provider's Public Switched Telephone Network (PSTN), and into the business' own Remote Access Servers (RAS). Once they reach the RAS, the transmission continues onto the WAP gateway, and then onward to the application or Web server, all of these devices within the business host's own secure environment.

Although it provides better end-to-end security, the drawback to this approach is that the business host will need to set up banks of modems and Remote Access Servers so users have enough access points to dial in. The business will also need to reconfigure the users' cell phones and PDAs to point directly to the business' own WAP gateway instead of typically to the service provider's. Not all cell phones allow this reconfiguration by the user, however. Furthermore, some cell phones can point to only one WAP gateway, while others are fortunate enough to point to more than one. In either case, reconfiguring all those wireless devices individually to point to the business' own WAP gateway may take significant time and effort.

For users whose cell phones can point to only a single WAP gateway, this reconfiguration introduces yet another issue. If these users now want to access other WAP sites across the Internet, they still must go through the business host's WAP gateway first. If the host allows outgoing traffic to the Internet, the host then becomes an Internet Service Provider to these users that are newly configured to point to the host's own WAP gateway. Acting as a makeshift Internet Service Provider, the host will inevitably need to attend to service and user-related issues, which to many businesses can be an unwanted burden because of the significant resources required.

Pass-Through from Service Provider's WAP Gateway to Host's WAP Proxy. For those businesses who want to provide secure end-to-end encrypted transactions, yet want to avoid the administrative headaches of setting up their own WAP gateways, there are still other approaches. One such approach, as shown in Exhibit 3, is to keep the WTLS-encrypted data unchanged as it goes from the user's mobile device and through the service provider's WAP gateway. The WTLS-SSL encryption translation will not occur until the encrypted data reaches a second WAP gateway-like device residing within the business host's own secure network. One vendor developing such a solution is Openwave Systems (a combination of Phone.com and Software.com). Openwave calls this second WAP gateway-like device the Secure Enterprise Proxy. During an encrypted session, the service provider's WAP gateway and the business' Secure Enterprise Proxy negotiate

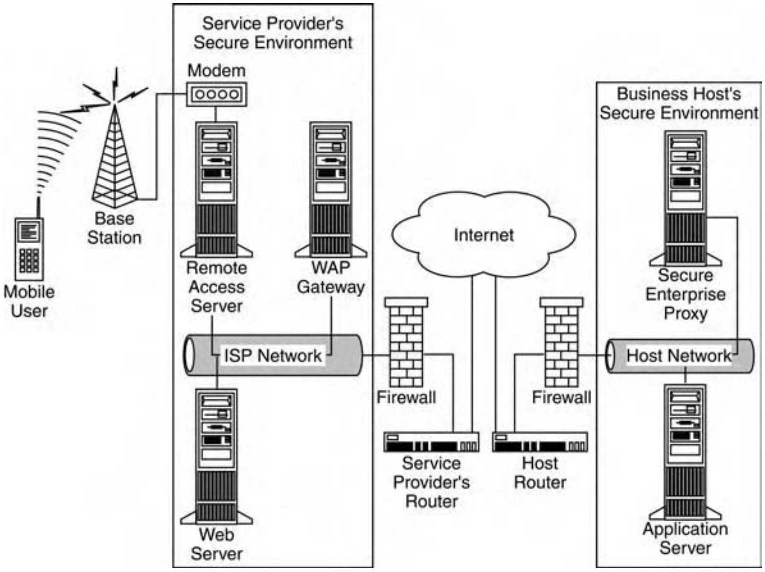


Exhibit 3. Pass-Through from Service Provider's WAP Gateway to Host's WAP Proxy

with each other, so that the service provider essentially passes the encrypted data unchanged onto the business that is using this Proxy. This solution utilizes the service provider's WAP gateway since it is still needed to provide proper Internet access for the mobile users, but it does not perform the WTLS-SSL encryption translation there, so it is not exposing confidential data. The decryption is passed on and occurs instead, within the confines of the business' own secure network, either at the Secure Enterprise Proxy or at the application server.

One drawback to this approach, however, is its proprietary nature. At the time of this writing, to make the Openwave solution work, three parties would need to implement components exclusively from Openwave. The wireless service providers would need to use Openwave's latest WAP gateway. Likewise, the business hosting the secure applications would need to use Openwave's Secure Enterprise Proxy to negotiate the encryption pass-through with that gateway. In addition, the mobile devices themselves would need to use Openwave's latest Web browser, at least Micro-browser version 5. Although about 70 percent of WAP-enabled phones throughout the world are using some version of Openwave Micro-browser, most of these phones are using either version 3 or 4. Unfortunately, most of these existing browsers are not upgradable by the user, so most users may need to buy new cell phones to incorporate this solution. It may take some time before this solution comes to fruition and becomes popular.

These are not the only solutions for providing end-to-end encryption for wireless Internet devices. Other methods in the works include applying encryption at the applications, adding encryption keys and algorithms to cell phone SIM cards, and adding stronger encryption techniques to the next revisions of the WAP specifications, perhaps eliminating the “gap in WAP” entirely.

CONCLUSION

Two sound recommendations for the many practitioners in the information security profession are:

1. Stay abreast of the wireless security issues and solutions.
2. Do not ignore the wireless devices.

Many in the IT and information security professions regard the new wireless Internet devices as personal gadgets or executive toys. Many are so busy grappling with the issues of protecting their corporate PCs, servers, and networks, that they could not imagine worrying about yet another class of devices. Many corporate security policies make no mention about securing mobile hand-held devices and cell phones, even though some of these same corporations are already using these devices to access their own internal e-mail. The common fallacy heard is, because these devices are so small, “What harm can such a tiny device create?”

Security departments have had to wrestle with the migration of information assets from the mainframe world to distributed PC computing. Many corporate attitudes have had to change during that evolution regarding where to apply security. It is no exaggeration to say that corporate computing is undergoing yet another significant phase of migration. It is not so much that corporate information assets can be accessed through wireless means, since wireless notebook computers have been doing that for years. Rather, the means of access will become ever cheaper and, hence, greater in volume. Instead of using a $3000 notebook computer, users (or intruders) may now tap into a sensitive corporate network from anywhere, using just a $40 Internet-enabled cell phone. Over time, these mobile devices will have increasing processing power, memory, bandwidth, storage, ease of use, and, finally, popularity. It is this last item that will inevitably draw upon the corporate resources.

Small as these devices may be, once they access the sensitive assets of an organization, they can do as much good or harm as any other computer. Ignoring or disallowing these devices from an information security perspective can have two probable consequences. First, the business units or executives within the organization will push, and often successfully, to deploy wireless devices and services anyway, but shutting out any involvement or guidance from the information security department. Inevitably, information security will be involved at a much later date, but reactively, and often too late to have any significant impact on proper design and planning.

Second, by ignoring the wireless devices and their capabilities, the information security department will give attackers just what they need — a neglected and unprotected window into an otherwise fortified environment. Such an organization will be caught unprepared when an attack using wireless devices surfaces.

Wireless devices should not be treated as mere gadgets or annoyances. Once they tap into the valued assets of an organization, they are indiscriminate and equal to any other node on the network. To stay truly informed and prepared, information security practitioners should stay abreast of the news developments and security issues regarding wireless technology. In addition, they need to work with the application designers in an alliance to ensure that applications designed for wireless take into consideration the many points covered in this chapter. And finally, organizations need to expand the categories of devices protected under their information security policies to include wireless devices, since they are, effectively, yet another infrastructure component to the organization.

References

Books

1. Blake, Roy, Wireless Communication Technology, Delmar Thomson Learning, 2001.
2. Harte, Lawrence et al., Cellular and PCS: The Big Picture, McGraw-Hill, New York, 1997.
3. Howell, Ric et al., Professional WAP, Wrox Press Ltd., 2000.
4. Muller, Nathan J., “Desktop Encyclopedia of Telecommunications,” Second Edition, McGraw-Hill, New York, 2000.
5. Tulloch, Mitch, Microsoft Encyclopedia of Networking, Microsoft Press, 2000.
6. Van der Heijden, Marcel and Taylor, Marcus, Understanding WAP: Wireless Applications, Devices, and Services, Artech House Publishers, 2000.

Articles and White Papers

1. Saarinen, Markku-Juhani, Attacks against the WAP WTLS Protocol, University of Jyy, Askyl, Finland.
2. Anne, Saita, Case Study: Securing Thin Air, Academia Seeks Better Security Solutions for Handheld Wireless Devices, April 2001, http://www.infosecuritymag.com
3. Complete WAP Security from Certicom. http://www.certicom.com
4. Radding, Alan, Crossing the Wireless Security Gap, January 1, 2001. http://www.computerworld.com
5. Does Java Solve Worldwide WAP Wait? April 9, 2001. http://www.unstrung.com
6. DeJesus, Edmund X., Locking Down the…Wireless Devices Are Flooding the Airwaves with Millions of Bits of Information. Securing Those Transmissions Is the Next Challenge Facing E-Commerce, October 2000. http://www.infosecuritymag.com
7. Izarek, Stephanie, Next-Gen Cell Phones Could Be Targets for Viruses, June 1, 2000. http://www.foxnews.com

8. Nobel, Carmen, Phone.com Plugs WAP Security Hole, eWEEK, September 25, 2000.
9. Secure Corporate WAP Services: Nokia Activ Server. http://www.nokia.com
10. Schwartz, Ephraim, Two-Zone Wireless Security System Creates a Big Hole in Your Communications, November 6, 2000. http://www.infoworld.com
11. Appleby, Timothy P., WAP — The Wireless Application Protocol (White Paper), Global Integrity.
12. Wireless Devices Present New Security Challenges — Growth in Wireless Internet Access Means Handhelds Will Be Targets of More Attacks, CMP Media, Inc., October 21, 2000.

Section IV
Project Approaches and Life Cycle

Despite the evolution of technology, the basic success of IT projects is still dependent on the overall project approach and the development life cycle. This section examines some frameworks, methodologies, and approaches for building Internet and wireless business solutions through the following chapters:

"Prototyping Methods for Building Web Applications" (Chapter 22) shows how existing prototyping methods can be adapted to build Web-based applications. It presents the Modified Prototyping Method, which treats Web applications as living entities, constantly adjusting to a changing business environment.

"Component-Based Development" (Chapter 23) examines a framework for building applications through an assembly of components. This framework is applicable to development on the Web through a variety of platforms, such as COM and CORBA.

"User Interface Rejuvenation Methodologies Available with Web-to-Host Integration Solutions" (Chapter 24) presents the alternatives for rejuvenating host application interfaces that are generally characterized by dated, character-based user interfaces. These approaches can be translated for building wireless solutions as well.

"Determining Whether To Buy or Build an E-Commerce Infrastructure" (Chapter 25) explains how organizations can meet the demands of E-commerce initiatives through a strong architecture that is either built, purchased, or a combination of the two approaches.

"An Implementor's Guide to E-Commerce" (Chapter 26) explores many of the complexities involved in setting up an E-Commerce site outside the basic technology itself. The concept of an E-team (extended) approach to the entire E-life cycle is also proposed.

Chapter 22
Prototyping Methods for Building Web Applications

Jim Q. Chen
Richard D. Heath

Internet technology is changing the way people live, the way they compute, and the way they conduct business. For information systems (IS) developers, however, nothing has changed more than the way applications are designed, built, and distributed. Web technology has brought us to a new world of software engineering, with new techniques, new tools, and a new design and deployment environment. The technology enables IS developers to deliver applications easily and quickly, and provides more efficient methods to do maintenance and updates of applications. As a result, IS developers are more responsive to user needs and quicker to customize applications for specific users.

The technology also brings challenges, including competing architectures, platforms, and tools, most of which are still evolving. Developers are being challenged to explore new methodologies and best practices to address World Wide Web-specific development issues, such as maintenance of content-rich Web applications, security, application scalability, and an ever-increasing demand for fast system deployment by customers. For many developers, building Web applications is a "mad science."[1] The traditional system development methods such as prototyping methods can still be effective, but they need to be adapted and enriched in the new development environment.

This chapter discusses Web-based business applications and their challenges, and describes how existing prototyping methods are adapted to build Web-based applications. For ease of reference, the adapted method is referred as the Modified Prototyping Method (MPM). The intent of this chapter is to provide some useful information for technology managers and developers of Web software applications.

0-8493-1161-6/02/$0.00+$1.50

The chapter is organized as follows. Web applications and their major components are addressed first, followed by the challenges of Web application development. Finally, the Modified Prototype Method is discussed. The chapter concludes with a summary and discussion of the advantages and disadvantages of deploying Web applications.

WEB APPLICATIONS

In recent literature, a Web application is defined as "any application program that runs on the Internet or corporate intranets and extranets." The user of a Web application uses a Web browser on a client computer to run the program residing on the server. The entire processing is done on the server as if it were done at the user's local machine. In this chapter, the term is used in a broader context to include any application that is Web browser based.

There are three types of Web applications: static Web documents; simple, interactive Web applications; and complex Web-based database systems. Static Web applications do not interact or exchange information with their viewers. Their purpose is to share and distribute information to the public. Most personal Web sites can be classified as static.

The next level of sophistication is the interactive Web application, where visitors to the sites can exchange information with the site owners. Many such Web sites use response forms to collect feedback or customer evaluation on their products or services. Complex Web applications handle sophisticated business transactions online, such as online banking, stock trading, and interactive database queries. They may be full-blown Java applications running on the client side, but their codes are automatically downloaded from the server, with multi-tiered client/server architecture. Alternatively, they could be applications based on ActiveX technology and Active Forms, which executes on both the client and the server.

Complex Web database systems are the cornerstone technology for E-commerce. This chapter focuses on the development of such industrial-strength Web applications.

A Web application is based on individual Web pages, whether they are static or dynamic. This enables one to divide the application into clearly demarcated sections, allowing or denying access as needed. For example, human resource divisions might be allowed access to certain areas of the application when performing their human resource duties, whereas sales departments might want to look at the inventory part of the application while placing a customer order.

As shown in Exhibit 1, each portion of the application can have its own Web page. Each page can include an appropriate user interface for gathering

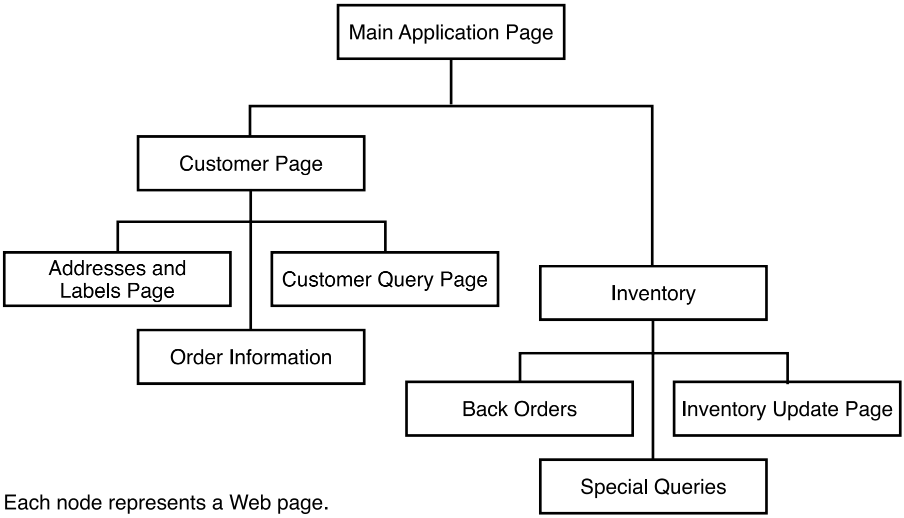


Exhibit 1. Layout of a Web Application

and displaying data to the user. Each page can include help, right alongside the application's interface, and can contain links to almost any other part of the application.

The application can be broken down as finely as the developer desires. Each page can do several functions, or merely one function. Special pages for specific users can be added and accessed based on the user's identity, which can easily be determined and managed by standard HyperText Transfer Protocol (HTTP) and Web techniques such as authentication and the use of cookies. New functionality can easily be added, merely by adding additional Web pages and the appropriate links. Functionality can easily be updated or fixed by changing existing pages.

The use of Web-based technology means that the application can be managed from one central location. The developer can maintain total control over the content at the server, rather than having to worry about delivering binary content to each individual user.

WEB APPLICATION COMPONENTS

An industrial-strength Web database application may consist of five major components, as shown in Exhibit 2. The Web server runs specialized Web server software that supports HTTP to handle multiple Web requests, and is responsible for user authentication in case of intranet and extranet applications.

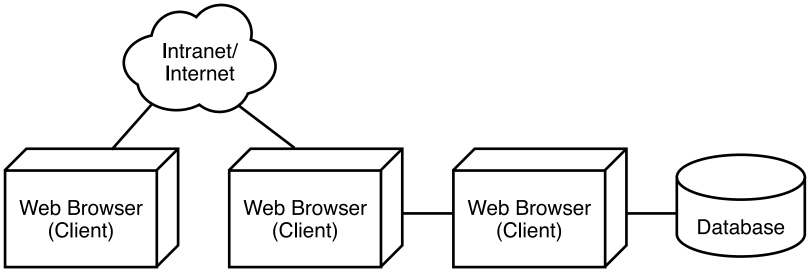


Exhibit 2. Web Database Application Components

An application server performs most of the application processing logic and enforces business rules. It is also in charge of maintaining the state management and session control logic required for an online transactional system. The database server hosts the database management system (DBMS) and provides data access and management capabilities. In a typical session, the Web server processes client requests and sends HTML pages back to the client. When needed, a Web server connects to an application server to process business logic (e.g., for credit authorization or checking inventory status).

The database server performs database query and sends the result back to the Web server. Such multi-tier architecture provides high system scalability. When the system demand increases, workload can be distributed on additional application or database servers. However, this layout does not mean that one must have an application server for the Web applications, nor does it imply that the Web server and application server or database server cannot be located on the same machine. The decision on architectural components is affected by the requirements of the application, the business strategy, and the existing and future technology infrastructure.

CLIENT-SIDE PROCESSING

Client-side processing has grown very popular because it improves the overall application's responsiveness and frees some of the Web server resource for other tasks. Java applets and ActiveX components are the two main technologies that allow developers to create and maintain code that runs on client workstations. ActiveX controls and Java code reside on the server and are delivered to the client on demand. Both provide means for automatically ensuring that the latest version of the code is available to the client. Version updating is done almost transparently, so that the user need not even know that any changes have been made. Both can be delivered to a user's browser via a simple HyperText Transfer Protocol (HTTP) request.

Java applets and ActiveX differ mainly in the means of execution. ActiveXs are compiled binary code delivered as OCXs, and are stored on the client. In most cases, the downloaded code remains on the client machine for subsequent use. ActiveXs require that Internet Explorer be the client, or that a special plug-in be used in Netscape Navigator. Java code can be run on any machine that has a Java virtual machine installed, and thus is cross-platform in nature. OCXs are less secure, in that they can contain code that does damage things, such as deleting and altering local files. While powerful OCXs should not be used except in a closed system such as an intranet/extranet, Java applets are well controlled by the browser, and thus are better suited for an open system such as the Internet.

SERVER-SIDE PROCESSING

Any Web application will do at least some server-side processing. In its most strict form, applications that use server-side processing do all of the application's processing on the server, and then send only HTML back to the client. In the case of Web database applications, the Web browser sends a database request to the Web server. The Web server passes the request using the Common Gateway Interface (CGI) or Internet Server Application Programming Interface (ISAPI) to the application server, where the Web-to-database middleware may be located. The application server then uses database middleware such as Open Database Connectivity (ODBC) to connect to the database. The application server receives the query result, creates the HTML-formatted page, and sends the page back to the Web server, using the CGI or ISAPI transmission standard. The Web server sends the page to the browser. Server-side programming options include Java, JavaScript, VBScript, ActiveX, and CGI-script (Perl, C, and C++).

Challenges of Web Application Development

Web application development, whether Internet based or intranet and extranet based, presents unique challenges for developers. The major challenges include security, content-rich maintenance, integration with legacy systems, and fast application deployment. For Internet-based applications, there are two additional two challenges: scalability and load balancing.

Scalability. An Internet application runs in a different operating environment from a non-Internet-based application. Non-Internet-based systems operate in a well-confined environment. The system users and workload are well understood. Internet applications, however, run in an open environment, where workload and user profiles are less understood and less predictable. Therefore, Internet applications can encounter highly variable and potentially huge peak transaction loads. The system must be designed to handle dramatic fluctuations of user demand, and to have additional upgrades to boost the system performance and to support additional users.

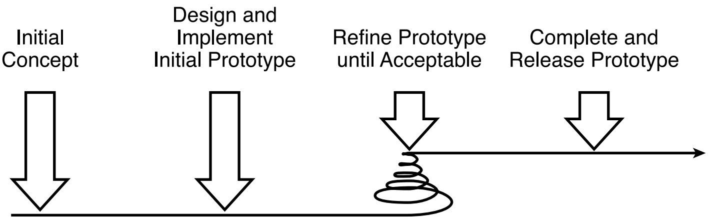
Load Balancing. In a multi-server Internet application, an unbalanced workload on the servers reduces system performance, reliability, and availability. Balancing the system's load requires careful selection from an array of tools and techniques. There is no single silver bullet that can be applied to all application systems. Load-balancing techniques include application partitioning and service replication.

Security. Security is a major concern for Internet applications because of the open operating environment. Even for intranet and extranet applications, security should be a concern. No one product that is bought can guarantee a secure application. Security must be designed into an application, and must be maintained in that application. Furthermore, an organization-wide security policy and procedure must be in place. The following security issues must be addressed:[2]

- *Privacy:* how to ensure that confidential data are safeguarded
- *Integrity:* how to ensure that data consistency and accuracy are maintained when data is traveling on the network
- *Authentication:* how to verify the true identity of the parties involved in a business transaction
- *Access control:* how to allow authorized users to access only the information they are allowed to access: how to prevent unauthorized access
- *Non-repudiation:* how to prevent denial of transaction submissions, either from the sending or receiving ends of the communication process

Content-Rich Maintenance. Most Web applications are content rich. Content-rich applications require frequent updates and maintenance. A less frequently updated Web site quickly casts doubt in its visitors' minds about its accuracy and usefulness. For Web applications, the notion of maintenance takes on a different meaning, where the lines between development and maintenance blur to the point where they are really the same thing.

Integrating Legacy Systems. More and more organizations are linking their legacy systems, which may run on different computing platforms, to their Web applications. Many Web middleware solutions are available to bridge Web technology to relational databases or legacy systems. For example, Oracle Corporation, Informix Software, and Sybase Corporation offer Web database middleware; and IBM's MQSeries and Talarian's SmartSockets


Initial Concept
Design and Implement Initial Prototype
Refine Prototype until Acceptable
Complete and Release Prototype

Adapted from McConnell, Steve, "Rapid Development," Microsoft Press, 1996.

Exhibit 3. The Evolutionary Prototyping Method

are message-oriented middleware tools. The challenge is to find the proper tools that fit the organization's needs.

Fast Development. A well-designed, high-quality Web application can bring quick investment return. With the increasing global competition in E-commerce, customers are demanding that their systems be delivered at an accelerated pace.

Development Methodology

The prototyping method was formally introduced to the information systems community in the early 1980s to combat the weakness of the traditional waterfall model.[3] It is an iterative process of system development.

Working closely with users, developers design and build a scaled-down functional model of a desired system. They demonstrate the working model to the user, and then continue to develop the prototype based on the feedback they receive until they and the user agree that the prototype is "good enough." At that point, they either throw away the prototype and start building the real system (because a throwaway prototype is used solely to understand user requirements), or complete any remaining work on the prototype, and release the prototype as final product (an evolutionary prototype). Exhibit 3 illustrates the evolutionary prototyping process.[4] Note that the maintenance phase begins only after the final system is formally deployed.

The prototyping method has gained popularity because of its ability to capture user requirements in concrete form. In fact, the method is often used for designing decision support systems, when neither the decision-maker nor the system designer understands the information requirements well. It is often used along with traditional system development methods to speed up the system development process.

Another related method is staged delivery, in which the product is divided into several stages. Each stage consists of detailed design, code, debug, and delivery for a component of a desired system. Like the evolutionary prototyping method, there is a distinctive boundary between development and maintenance.

These methods have proven very successful when customized to specific development environments for non-Web-based applications development. They are also applicable to Web-based applications. In fact, prototyping methods are especially suitable for Web-based applications because of the ease of system delivery and updates afforded by Web technology. However, the unique requirements of Web applications require the designers to consider additional factors when using these models.

Exhibit 4 outlines the modified prototype method (MPM) for Web application development. MPM allows for basic functionality of a desired system or a component of it to be formally deployed right away. The maintenance phase is set to begin right after deployment. The method is flexible enough not to force an application to be based on the state of an organization at a given time. As the organization grows and the environment changes, the application changes with it, rather than being frozen in place.

Basic System Analysis and Design. The basic system analysis and design involves studying general user requirements, as well as the underlying data model, the user interface, and the architecture requirement. Understanding user requirements really means understanding the requirements of two things: Web content and system behavior.

Web content refers to information and its organization on a Web site. Designers need to decide what information is to be included, what level of details there should be, and how the information should be organized on the Web site. Traditional techniques such as survey and interview can still be used for Web content requirement analysis, especially for intranet/extranet applications. System behavior refers to the system's intended functionality. A powerful method to design system functionality is to develop use cases. A typical use case consists of a group of scenarios tied together by a common user goal. It serves as an easy-to-use communication tool to express the interaction and dialog between system users and the system itself.

The data model is one of the most important part of an application, so getting this right is crucial. While changes and additions can be made later, such changes are costly. Although there is no way to determine all of the data needs right at the beginning, doing a good analysis and design on the data that is known will go a long way toward application success. The data model should be flexible enough to adapt to changing needs. By adhering to the strict database normalization rules, designers can minimize the problems that might arise from the need to change the data model.

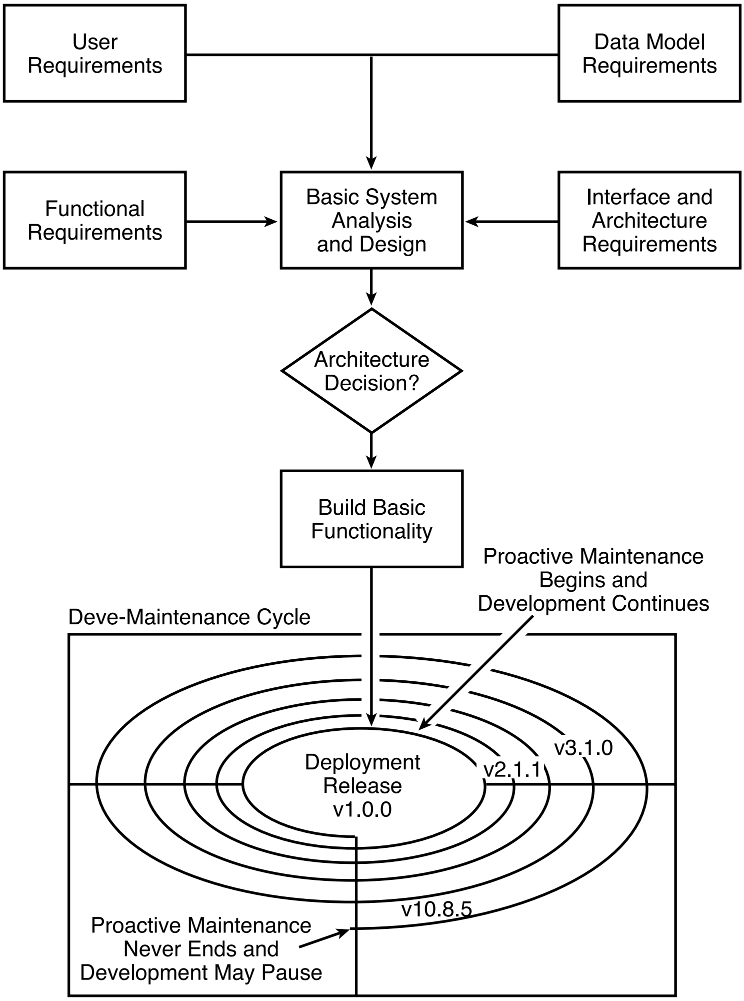

User Requirements
Data Model Requirements
Functional Requirements
Basic System Analysis and Design
Interface and Architecture Requirements
Architecture Decision?
Build Basic Functionality
Proactive Maintenance Begins and Development Continues
Deve-Maintenance Cycle
Deployment Release v1.0.0
v2.1.1
v3.1.0
v10.8.5
Proactive Maintenance Never Ends and Development May Pause

Exhibit 4. The Web Development Methodology

Finally, basic interface and architectural decisions must be made, based on the organization's existing technology infrastructure and user needs. Designers should determine if they will do server-side processing, where the data will reside, and if they will use Java or ActiveX. They must also determine if they need to allow salespeople access to the data while they are away, if customers need access to the application, and which part of the application will reside on the application server. Choosing a proper architecture has a long-lasting impact on the organization. It will determine how flexible the organization will be, technology-wise, in adapting to constantly changing business needs.

Architecture Decision. After the system requirements and use cases have been carefully analyzed, the decision on system architecture must be made. This decision must be made based on both the current needs and future development. For a simple static Web site, the clients and Web server are the only two components needed. However, for Web applications that are dynamic and process business logic, at least three significant architectural components are needed: clients, a Web server, and an application server. It is also very common for most Web applications to have a database server. There are many ways to lay out a Web application architecturally, but there are three major models: (1) a thin client, (2) a fat client, and (3) distributed and component based.

1. *Thin client.* A thin client has minimal computing power because all of the business logic and rules are processed at the server. The client is a standard Web browser. This model is primarily used for Internet-based and some extranet-based applications, because there need not be any control over the client's configuration. The model gives developers greater freedom in system deployment and maintenance. However, application performance may be a bit slow, due to the fact that all processes are done at the server side.
2. *Fat client.* In a fat client, a fair amount of business logic and rules is executed on the client machine. Fat clients typically use dynamic HTML, Java applets, or ActiveX controls. Fat clients are used for some intranet applications that must provide customized services to certain user groups; for example, special reports tailored to top executives. System performance speed is expected to be faster because some business logic is done locally.
3. *Distributed and component based.* Distributed and component-based architectures are used to support distributed object-oriented systems. In the previous two models (thin and fat clients), a business system is deployed at one location, and the business logic for the application is implemented in a tightly coupled proprietary system. A distributed object system, however, allows parts of the system to be located on separate computers, possibly in many different locations.

The object system itself is an assembly of reusable business software components. Business components are self-contained units of code designed to perform specific business functions. A major benefit of a distributed object system is its adaptability to a changing environment. As a business' products, processes, and objectives evolve over time, new business software solutions can be easily assembled using reusable business components. Another benefit is the elimination of the vendor "lock-in" problem.

There are three major competing distributed and component-based architectures for Web applications: the Distributed Component Object Model (DCOM), the Common Object Request Broker Architecture (CORBA), and Enterprise JavaBeans (EJB). DCOM is a Microsoft networking standard that permits different software components to interact with one another as integrated Web applications.

The ActiveX technologies are built primarily on DCOM, and can be divided into the following major components: ActiveX controls, ActiveX scripting, ActiveX documents, and the ActiveX server framework. ActiveX controls are interactive objects (units of code into which properties and methods are encapsulated) that are embedded in a Web page to perform various functions of a Web application. ActiveX controls can be created using a variety of languages and executed across the Internet or corporate intranets. ActiveX scripting allows developers to control intercommunication between software components, including ActiveX controls, ActiveX documents, and Java applets. ActiveX documents allow users to view different documents inside other applications. For example, a Microsoft Excel document can be viewed and edited inside a Word document. The ActiveX server framework provides features that support database access, security, and session management. For example, Active server pages provide state management and session control for a Web application. The disadvantage in using DCOM is the client requirement of running Windows.

CORBA is a set of standards that addresses the need for interoperability among the rapidly proliferating number of hardware and software products available today. The CORBA model allows applications to communicate with one another, no matter where they are stored. The object request broker (ORB) is the middleware that establishes the client/server relationships between objects. Using an ORB, a client can transparently invoke a method on a server object, which can be on the same machine or across a network. The central protocol of the CORBA distributed component model is the Internet Interoperable ORB Protocol (IIOP). CORBA was proposed by the Object Management Group (http://www.omg.org). It is an important step on the road to object-oriented standardization and interoperability.

The Enterprise JavaBeans (EJB) model, defined by Sun Microsystems, is an application programming interface (API) specification for building scalable, distributed, component-based, multi-tier applications.[5] EJB is different from the original JavaBeans model, which provides a standard specification for developing reusable, prefabricated Java components that are mainly used on the client side of a business application. EJB, however, is defined as a server-side model for component-based, transaction-oriented, distributed enterprise computing. The model defines four key components: (1) the server, (2) the container, (3) the remote method invocation (RMI), and (4) the interface to back-end system and databases; that is, Java Database Connectivity (JDBC).

The server provides a standard set of services for transaction management, security, and resource sharing. The container is where JavaBeans execute. The container provides life-cycle management (from object creation to destruction), persistence management, and transparent distribution services. The remote method invocation API allows JavaBeans components running on one machine to invoke methods on remote JavaBeans as if they were local. The JDBC API provides relational database connectivity for Java applications. EJB is an alternative or a complement to the DCOM and CORBA models. Major software vendors supporting EJB include IBM, Netscape Communications, Oracle, and Borland International.

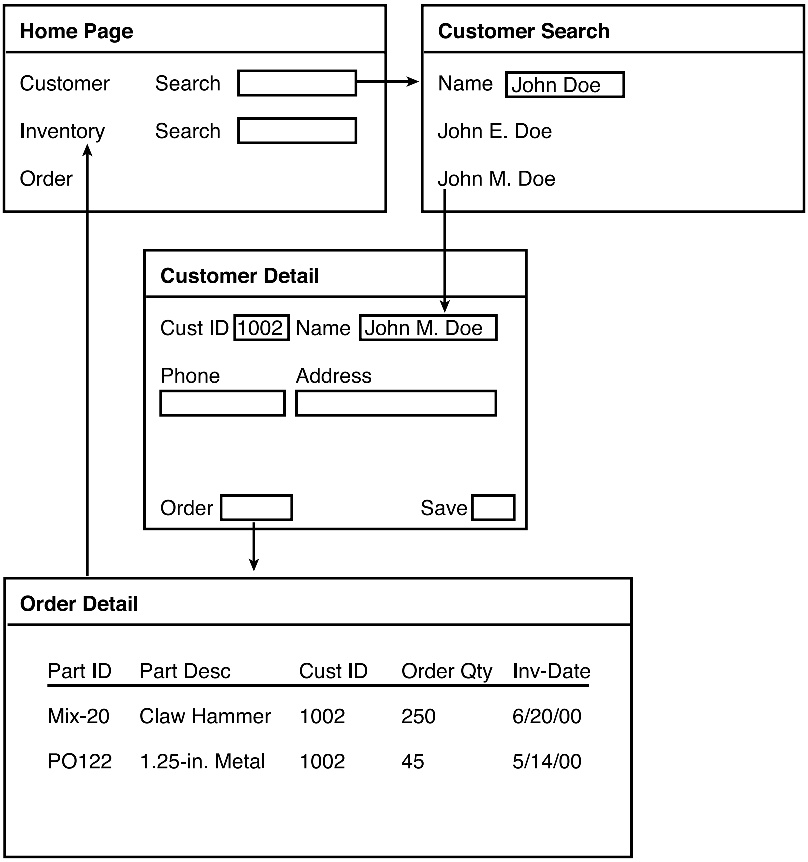
The difference between CORBA and EJB or DCOM is that since CORBA is just a specification, it relies on individual vendors to provide implementations. CORBA and the EJB approaches are merging and are interoperable in some of today's implementations. The decision on which architecture to adopt depends on several factors:[2]

- Size, complexity, and level of scalability required by the application
- Existing hardware/software
- Level of compatibility of the different components that are assembled to create the application
- Type of development tools available

In general, for Intranet applications, the organization should consider the mainstream software in use within the organization. If the organization is primarily Windows-centric, DCOM/ActiveX might be the choice. If the organization is UNIX based or running under a multi-platform environment, then OMG's CORBA might be a proper way to go. If the organization is committed to Java, and plans to use it extensively in the future, then Sun's EJB might be the choice. For Internet applications, it is difficult to predict the client-side environment; potential users may use any type of browser in any version. Therefore, designers need to determine their targeted user groups and try to accommodate their needs first.

By the end of this step, developers should have an idea of how their application will be structured, what each tier of the application will be doing, what the data model will look like, and what basic functions can first be deployed.

Building and Deploying Initial Version. This step begins with laying out the application as a series of connected Web pages, each page performing a specific function. Perhaps the initial version will provide nothing more than a collection of form and report pages that allow users to query, update, and report on customer address information. Developers might



Part ID	Part Desc	Cust ID	Order Qty	Inv-Date
Mix-20	Claw Hammer	1002	250	6/20/00
PO122	1.25-in. Metal	1002	45	5/14/00

Exhibit 5. Storyboards: Flow of Web Page Sketches

need to build only a few HTML pages and a few reports to at least let users become aware of the application and get used to the basic functions provided. A helpful tool that can be used to lay out the user interface is storyboarding, in which storyboards are used to capture Web pages and the flow among them (see Exhibit 5 for an example).

Deve-Maintenance Cycle. Once the initial version is deployed, developers enter a deve-maintenance (*deve*lopment and *maintenance*) cycle, characterized by incremental enhancements and proactive maintenance.

- *Incremental enhancement.* After a basic functionality is deployed, developers can provide incremental enhancements. Perhaps they can add a section of the application that allows users to access inventory data, which could be built and deployed without worrying about integrating it with customer order data. Once the customer and inventory data is deployed, developers might build the part of the application that connects the two areas, allowing users to check orders against both customers and inventory. The really powerful part is that integrating the new into the existing functionality might be as simple as adding HTML code to the existing pages in the customer and inventory sections. These changes, of course, will take effect immediately, as they are placed in the code base for the Web server. New functionality can be highlighted and explained right on each Web page. Developers can easily direct users to help screens, and point out new items and functions. Bug fixes can be transparently updated without a single change needing to be made on users' machines. New users who ask for specific reports can be given access to particular pages that meet their particular needs. All of this can be done with the code under the complete control of the development team because all of the code will sit on the server, waiting to be accessed by the user.

- *Maintenance.* Maintenance takes on a different meaning for Web applications. The distinctive maintenance phase in the traditional system development cycle no longer exists. The maintenance phase is interleaved into the development phase. The application is constantly evolving and changing, with old features going away and new features being added. Thus, it is not really clear where maintenance begins and development ends. Maintenance may become bug-fixing, while development means adding new features. However, there will no doubt be much overlap between the two as new features are integrated into old portions of the application. Therefore, in the end, differentiating between a maintenance programmer and a development programmer may be difficult, at least in small IS departments.

The method also means that traditional reactive maintenance practices must be replaced by routine, systematic, and proactive maintenance. For intranet and extranet applications, developers will know very quickly if the application is malfunctioning or if it contains outdated information, because their colleagues (intranet users) and business partners (extranet users) would love to let them know they have made a mistake. However, for Internet applications, the general public is very unlikely to take the trouble to let organizations become aware of the problem. The easiest thing for them to do is to leave and go to competitors. Therefore, routine maintenance and frequent updates are essential for content-rich Web applications.

What This Does Not Mean

The method does not mean that testing is no longer needed. Developers can still test changes and additions as much or as little as they do now.

They should simply not deploy a new part of an application until they feel confident that it is ready for use by the users.

Nor does this mean that the developer will become a slave to user requests, or that the application will become a mishmash of different, special applications. Using good application management techniques will still be necessary. Developers will still need to apply sound configuration management, and only build those new features that are thoroughly thought out and planned. Web technology allows developers to seamlessly integrate new features into the application. Often, this will take nothing more than making a Web page accessible by updating a link on an existing page. Users will see the change the very next time they go to the application.

COMPARING MPM AND TRADITIONAL PROTOTYPING METHODS

The differences between MPM and existing prototyping methods are more differences in emphasis and content rather than in fundamental approach. First, MPM calls for a formal maintenance phase to begin right after the initial version is deployed, while in traditional prototyping methods, formal maintenance begins only after the final system is deployed. Second, MPM maintenance activities are interleaved with development activities, while in traditional prototyping methods, there is a distinctive boundary between development and maintenance.

Third, there is no definite end of the system development process in MPM. At a certain point, the application may reach a stable state and development may pause. However, as the business grows and the environment changes, development activities will resume. It may seem like the application does not have a boundary or defined scope. In a sense, this is both true and false. It is true because Web technology affords the platform of an open application design, which allows one to easily expand the scope of an application. In fact, distributed and component-based Web application designs have become a new trend. These new architectures offer Web applications great flexibility and adaptability to changes. It is also false because Web applications should be developed like any other system. Developers need to plan the project and define an initial scope of the system. As the business grows and practices change, developers revise the scope of the application. The key, however, is to maintain forward thinking and to adopt an open technology architecture. Finally, MPM calls for proactive maintenance to replace reactive bug-fixing maintenance.

COMPARING MPM AND EXTREME PROGRAMMING

Extreme programming (XP) is a lightweight software development methodology that is claimed to be successful in reducing cost, meeting customer requirements, improving program quality, and increasing programming productivity. XP is aimed at small-sized development teams working in problem domains whose requirements are less understood and are changing. It is based on four core values: communication, simplicity, feedback, and courage. Development personnel should c*ommunicate* effectively among team members, users, and management. They should not waste time on a complete system analysis and design, but design as they go. In addition, they should keep it *simple*. They should also gain frequent *feedback* by coding in small iterations and working toward fast release cycles. Finally, they should have the *courage* to rewrite and improve code (refactor) when the code does not adequately meet new requirements.

The main tenets of the XP methodology are a collection of programming practices that are practiced to extremes. XP practices include iterative planning, pair programming, collective code ownership, tests several times a day, continuous integration, a 40-hour week, and on-site customers.

XP is not a solution for every project in every organization; rather, it has its limitations. It works only for small teams (of two to ten members). The ideas of XP are nothing more than common-sense practices that are as old as programming. The difficulty in successful adoption of an XP approach is not to learn the pieces, but to put them together and keep them in balance. This is not an easy task to accomplish. In fact, XP requires a new programming culture that may be at odds with most corporate cultures. For example, existing programming cultures are likely to resist such XP practices as programming a large project without a complete system specification or analysis and design, writing testing code first, or working more than 40 hours a week. Furthermore, XP does not account for different personality types and work styles. Its success rests on the assumption that every player in the game has the necessary skills and will to do his or her best to be an unsung hero.

If XP is an extreme step away from traditional, "heavyweight" software development methods, then MPM is a step in between the two extremes. MPM and XP share many common practices:

- Both methods seek maximum programming productivity, system reliability, and adaptability to changing business requirements.
- Both methods call for small iterations and short release cycles.
- Both methods advocate incremental changes. This means starting out with a minimal design and letting the program expand in directions that provide the best business values.
- Both methods emphasize testing and customer involvement in the development process.

XP is a code-centric (or bottom-up), practice-oriented approach. It prescribes a set of precise practices that the development team members must follow. For example, if developers are doing everything except pair

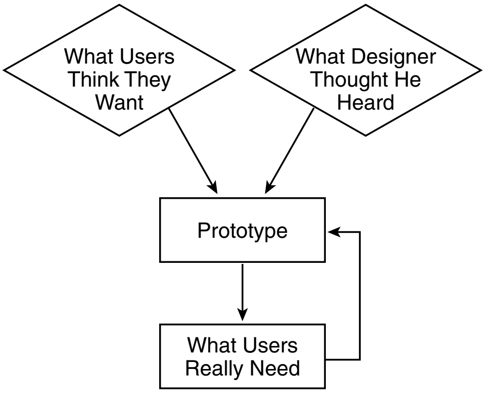


Exhibit 6. User and Developer Input

programming, they are not practicing XP. In contrast, MPM is a process-oriented and top-down approach. MPM places more emphasis on the overall process of application development and less emphasis on specific techniques. It requires little more formal up-front analysis than XP.

Furthermore, MPM is proposed for Web application development. It addresses some of the issues specific to Web applications, for example, the issues of maintenance, system scalability, and Web technology architectures.

User Involvement

Some developers disdain users, believing that they do not know what they want and are too ignorant to know a good application when they see one. Some developers are slaves to users and do whatever a user asks. Obviously, the correct path lies somewhere in between.

Exhibit 6 illustrates the important role played by users and developers in the process. Users have unique knowledge that is crucial to meeting their needs. Developers have technical knowledge and skills that can bring new ideas and features to an application that the users might never think of. Each new feature requested by the user should be carefully evaluated. For example, when the user requests adding a new link from the order page to the catalog page in a Web application, the developer should analyze the impact of the new feature on the existing use cases. Would this change create additional scenarios to consider? Alternatively, is there a different solution to the user request?

For Internet applications, getting users' involvement in the design process is difficult, but not impossible. The prototyping method provides designers with a unique way to collect user input. Once a prototype is deployed on the Web, its users' online actions can be monitored using the Web server's logging facilities. The subsequent analysis determines which parts of the application are being used the most and which are being used the least. This information can be used to plan future development priorities and make application development more efficient.

THE ADVANTAGES AND DISADVANTAGES

Advantages

Web-based applications promise a number of advantages over traditional non-Web-based applications.

1. *Control over application.* As the Web application developer, one can control the application on the server side for all users. One can easily control the code base and gain access to any part of the application. The application can become truly dynamic, from the binary execution to the available help. One can provide instant updates and customization.
2. *Cross-platform capability.* An HTML solution provides the ability to run an application on any Web browser on any operating system. Having cross-platform capability relieves one from worrying about a client's configuration. If a client has a browser that can run Java code, one might not even need to know which operating system users have. This can be a particular advantage if an organization wants to give its customers access to part of the application. Telling a Macintosh shop that they cannot get to one's customer service application because they are not able to run one's special client software is probably not good customer service. Giving the client a URL and a password that allow them access from almost any machine they have will build a lot more goodwill.
3. *Control over versioning.* Instead of worrying about whether a particular user has the right version of a DLL, EXE, or database file, one can control this at the server. There is no longer the need to get the latest version of any part of the application out to the user. One can always be sure that the client has the right code at the right time.
4. *User input.* The prototyping method allows user inputs to be quickly and easily integrated into an existing application. This can often be nothing more than a hyperlink to a new Web page. Users who need access to specific or limited areas of the application can be given access merely by being added to the password list, instead of having their client machines updated.

Disadvantages

Web applications are not the silver bullet that everyone has been dreaming about for so long. Depending on how a Web application is built and which technologies are chosen, some things must be sacrificed.

1. *Speed loss.* Web applications do not run as fast as those running on a local machine because of the downloading time and network traffic. This may become less of a problem as computer hardware and software improve.
2. *Data presentation limit.* If one chooses to go with server-side Java scripting, or a total HTML solution such as is available via a tool like Intrabuilder, one may be limited to the interface defined by HTML. In other words, one may be unable to provide the users with the latest in the widgets and gadgets that the modern user interface can provide. For example, tools such as datagrids and their capabilities are currently not available. This may limit one's ability to clearly lay out an application and present data to the user. However, forthcoming advances in HTML technology will reduce this limitation as the HTML interface becomes more sophisticated.
3. *Security vulnerability.* Web applications are inherently vulnerable to malicious Internet attacks. These attacks can be classified as vandalism and sabotage, breach of privacy, theft and fraud, violations of data integrity, and denial of service. As E-commerce technologies become more sophisticated, however, these threats will be minimized.

CONCLUSION

Web applications are an essential element in E-commerce. They offer system developers many challenges and opportunities. Design and implementation of a successful Web application require a disciplined approach that takes the organization's long-term development into consideration. The MPM discussed here requires a new mindset. Instead of viewing an application as having a start and a finish, developers should treat Web applications as living entities, constantly adjusting to the changing business environment. This may mean a radical change not only in an organization's development processes, but also in its management techniques, and even its hiring and training methods. It might no longer put its newest hires on maintenance to get them up to speed. In fact, maintenance might not even exist anymore.

References

1. Callaway, Erin, "Method from the Madness," *PC Week*, February 3, 1997, Vol. 14, No. 5, pp. 99–100.
2. Fournier, Roger, *A Methodology for Client/Server and Web Application Development*, Yourdon Press Computing Series, Prentice-Hall PTR, Upper Saddle River, NJ, 1999.

3. Naumann, J. D. and A. M. Jenkins, "Prototyping: The New Paradigm for Systems Development," *MIS Quarterly,* 6(3), pp. 29–44.
4. McConnell, Steve, *Rapid Development*, Microsoft Press, 1996.
5. Jubin, Henri and Jurgen Friedricks, *Enterprise JavaBeans by Example*, Prentice-Hall PTR, Upper Saddle River, NJ, 2000.

ABOUT THE AUTHORS

Richard D. Heath is president and CEO of Royal Oaks Information Systems, St. Cloud, MN.

Jim Q. Chen is an assistant professor of Business Computer Information Systems, St. Cloud State University, St. Cloud, MN.

Chapter 23 Component-Based Development

Nancy Stonelake

Component-based development is being touted as the solution to the latest software crisis. What is it and how true is the hype? The objectives of this chapter are:

- To define component-based development;
- To describe its benefits and weaknesses;
- To examine the basic architecture and popular component models;
- To examine alternatives and component-based developments in conjunction with current technology and data management; and
- To examine some of the challenges facing IT shops that want to move to a component approach.

DEFINITION

Component-based development differs from traditional development in that the application is not developed completely from scratch. A component-based application is assembled from a set of preexisting components. A component is a software bundle that performs a predefined set of functionality with a predefined API. At its simplest level, a component could be a class library or GUI widget, or it may be as complex as a small application, like a text editor, or an application subsystem, like a help system. These components may be developed in-house, reused from project to project, and passed between departments. They may be purchased from outside vendors who specialize in component development, or bartered between other companies in similar lines of business.

Components can be divided into two broad "types," namely, business components and framework components. Business components encapsulate knowledge of business processes. They may be applied in a vertical

0-8493-1161-6/02/$0.00+$1.50

industry sector, such as banking, or in a cross-industry standard business function like accounting or E-commerce. Framework components address specific software architecture issues like the user interface, security, or reporting functions.

BENEFITS

How is component-based development better than traditional development practices? If we compare developing enterprise-wide applications to auto manufacturing, current application development is like machining each part from scratch pretty much for every automobile being assembled. This is time consuming and expensive, when most of the parts are the same or similar in configuration. Henry Ford revolutionized manufacturing by standardizing parts and having workers specialize in small aspects of construction. Component-based development works on the same principles and reaps similar benefits.

Due to the similarity between all software applications, using components can reduce design time. Almost all applications have some security system, error handling, and user help functionality. Why are we wasting our time deciding how to provide help to users when the real question is what level of help users need? Components can provide framework solutions that can be tuned to our business requirements. This has the additional benefit of allowing us time to focus on the business logic, which is the key to fulfilling requirements.

Implementation time is reduced because components are already built. Additional coding may be required to integrate the component into the system, but the required functionality is already there.

These two main facts have additional implications. Since components are prebuilt, testing time is reduced. Components are already unit tested; they only require integration testing within the application. Overall, with components we require less design, development, and testing resources. This means we need fewer people with highly specialized and hard-to-find skill sets and we can leverage the people we have to do the things needed in the application.

Additionally, the cost of developing the components can be leveraged over many buyers. Since we acquire components from other departments in the company and pass our components on to them, they share in the costs. Vendors sell to multiple users and they all share in the development and maintenance costs. Component developers can afford to have designer/developers devoted to each component over its life cycle and these resources can become specialists in the component piece.

WEAKNESSES

Component development is an immature industry. This has several effects and implications, primarily limited vendors, products, and skilled human resources.

At this time there are limited choices in component vendors and company stability may be an issue. While there is a stable set of GUI component vendors, the offerings in true business components are limited. The lack of availability also limits competitive advantage. If all our competitors are using the same business logic, are we doing anything better than they are or are we just matching the pace? Product stability is also an issue. It is important that the API of a component remain constant otherwise we may incur heavy maintenance costs when integrating new product releases.

Component-based development requires a different approach to software development and there are few people who have actually done it. It requires people who can discern what parts of an application may be useful to other applications and what changes may need to be made to support future applications. In other words, you need a good designer/architect with a good crystal ball. To successfully reuse components you must have designers/implementers who are familiar with the component library, so they don't spend all their time looking for components that don't exist. They must also be able to adapt components to fulfill the requirements.

In addition, there must be supporting corporate strategies to promote reuse. No benefit is gained by crafting components that remain unused. Designers and developers need some impetus to change and the resources to support component development must be provided. This means taking the time to develop and locate components and promoting awareness of their availability.

BASIC ARCHITECTURE

Component architecture is based on locating components where they can best serve the needs of the user. This must account for several factors. Speed, processing power, and accessibility. One possible architecture is shown in Exhibit 1.

GUI widget components sit on the client; however, business logic may be required locally for complex applications, or on the server for transactional applications. These components can be placed wherever the architect sees fit. Component-based development does not provide an architecture as much as it permits good architectural choices to be implemented.

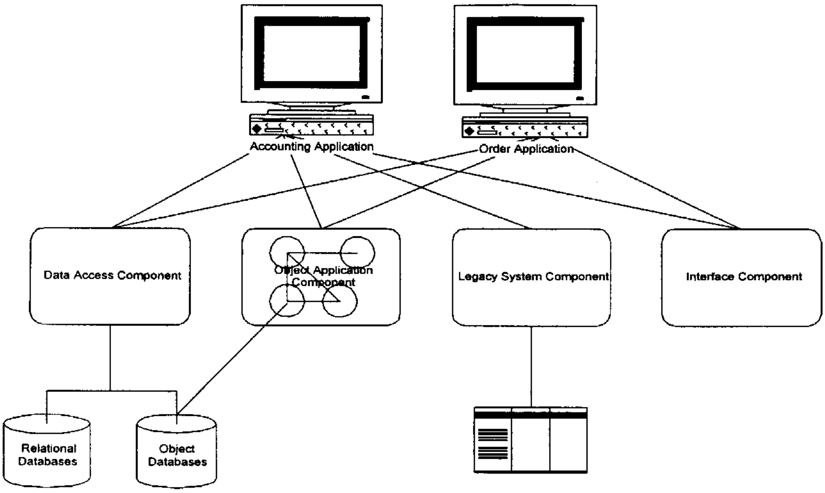


Exhibit 1. Component Architecture

In distributed environments components can follow the CORBA or DCOM models. Components can be wrapped as CORBA objects or have a CORBA object interface. This makes them accessible through an ORB, permitting ready distribution. Alternatively, components can conform to the COM model and be distributed using the DCOM specifications. These two distribution models can interwork, but such a discussion is beyond the scope of this chapter.

A component architecture can be described as a service-based architecture, as in the SELECT Perspective, where components act as the interface to a "service" that is a black box encapsulation of a collection of related functionality which is accessed through a consistent interface. Services are shared among applications to provide the application functionality. The system is then distributed according to business requirements, rather than software limitations.

COMPONENT TYPES

As mentioned previously, different components address different areas of functionality. These can be divided into framework components and business components. Framework components can be further broken down into data access components, user interface components, and subsystem components. Business components include business logic components and application components. These groupings allow us to place components based on the best configuration for our environment.

Data access components handle database interaction, including creation, deletion, query, and update. While data access components generally access a relational database, components can also access flat files, object databases, or any other persistent storage mechanism. This allows us a mechanism to change the back-end data storage without impacting the delivered applications. It also allows us to deliver the same application using different databases with minimal change. The data component is replaced to suit the new environment.

User interface components handle user interaction and define the look and feel of the application. Separation of this component allows us to change the interface so that applications can take on the look and feel of the deployment environment. This helps to reduce training time by offering the user a consistent paradigm across applications.

Subsystem components provide functionality like error handling, security, or user help. They allow for standardization across applications.

Business logic components encapsulate the policies of a business. By separating them from general application or data logic, we can easily change applications to reflect changing business policies, such as offering discounts to large customers, or recommending complementary products.

Application components are small applications that contribute to the functionality of a larger piece, for example, text editors. Application components include legacy applications that are wrapped to provide a standard interface for interaction with other components or use within applications.

COMPONENT MODELS

Currently there are two primary component models: JavaBeans and ActiveX. These two models can interact over bridges or by wrapping one as the other.

JavaBeans is the component model for the Java programming language. Because Beans are written in Java they run on any platform. A Bean may implement any functionality but it must support the following features: introspection, customization, events, properties, and persistence. JavaBeans are intended to be integrated with visual development environments and they should have some visually customizable properties, although they may not have a visual representation themselves.

ActiveX is based on the Component Object Model (COM) and was developed by Microsoft. While ActiveX has been a proprietary technology, Microsoft plans to transition it to an industry standards body. ActiveX enables developers to embed event-driven controls into Web sites by optimizing the COM model for size and speed. While ActiveX components implemented in different languages can interact, ActiveX components are compiled into platform-specific formats. The most common ActiveX implementation is for "Windows-Intel," limiting ActiveX to a Microsoft environment.

The OMG is currently in the process of defining a distributed component model based upon the Object Management Architecture. This will define a CORBA component and make integrating CORBA Components significantly easier.

COMPONENTS, OO, CLIENT SERVER, AND NETWORK COMPUTING

Component-based development has its roots in object-oriented technology and client/server development and can act as an enabler for network computing.

Components extend object technology and object methodologies. Like objects, components should be developed in an iterative, incremental fashion. Components must be identified from existing applications and reworked to apply to new applications. Components and objects incorporate the idea of encapsulation and black box accessibility. With components, as with objects, we are not concerned with how a service is performed internally, only that it is performed correctly. Components are refinements of objects in that the API is defined in a language-independent standard. Components, like objects, communicate through industry standard middleware: CORBA or DCOM. This middleware acts as a layer of abstraction, so that components can be called in the same fashion, regardless of their function. This further hides the component's implementation, whereas direct object communication can rely on implementation-specific calls.

Components act as service bundles, relying on tightly coupled object or legacy applications to implement their functionality. Components are then loosely coupled to form applications.

Components can be used to develop stand-alone applications or assembled in a traditional client/server fashion, with components providing server functionality like database access or client functionality on the user interface. Additionally, components allow us to move one step beyond. Components are designed to provide a limited service, and so allow for a true separation of the interface, business logic, and persistence. This allows them to be assembled in a multi-tier relationship and locate the components/tiers in the best place to run them.

Components can also enable network computing. Network computing allows for dynamic deployment, execution, and management of applications. Network computing architectures feature cacheable dynamic propagation, cross-platform capabilities, automatic platform adjustment, and runtime context storage. Since components are small units of work they are easily cacheable. Components written in Java using the JavaBeans specification are cross-platform and ActiveX components can run on any Microsoft-friendly platform. Components can be dynamically managed, running on whatever server is appropriate given the current load.

ALTERNATIVES

As shown, component technology can work with client/server and network computing architectures, as well as object-oriented development.

The primary alternatives to component-based development are traditional "from scratch" development and package implementation.

Component-based development is superior to "from scratch" in that we anticipate reduced design, development, and testing time, with a lower bug ratio, since components are prebuilt and pretested. We spend our time developing new components that are missing from our library, and crafting and testing the links between components.

Component-based development is superior to package implementations in its flexibility. We have more control over what features are included based on our needs, and we can change those features as our needs change.

CHALLENGES

It looks as if component-based development is a good thing. It saves time and money. How can we use components effectively in our own development environments? We will examine several areas: design, component acquisition, implementation, and maintenance.

Remember the idea behind component-based development is to free up our resources to concentrate on finding solutions for business problems. This can take us down several alleys. We may have to make a paradigm choice. If the application needs to be distributed or if the components are developed in multiple languages, we will have to decide whether to use DCOM or CORBA. The environment the application will run in and the available components will influence this choice. When working in an all-Microsoft environment DCOM is the obvious choice. Where heterogeneous operating systems are used CORBA is a better choice.

Acquiring components has its own challenges. If they are to be acquired from internal sources, channels for reuse have to be set up. This means components have to be described in a way that other departments can use them easily. There must be a mechanism for publishing their availability; and accounting systems must reflect the costs of component development and recapture on reuse. In short, the whole corporate structure may have to change.

Purchasing components presents other problems that may be influenced by corporate culture. While the ideal is that components can be replaced at will, the reality is that an application may become dependent on a component. Corporations may not desire this dependence. When purchasing components the financial stability of the provider company and the product stability must be considered. Will the product be supported in the future, and will its functionality and API remain consistent? Resources must be allocated to identify suitable components and evaluate the risk and future considerations that may impact their use.

Integrating components into an application presents challenges to developers and project managers. If the component is a class library, the object model will be affected. Library considerations can affect the subclass relationship. There may be conflicts between releases if you override methods in a subclass or make extensions to the purchased library.

Components also require good configuration management. You may not just be dealing with your own code releases, but also with the code releases of vendors, and your releases will impact users of your components. Code releases should be scheduled so downstream users can schedule regression testing. Vendor releases should be integrated into the application release and should undergo full regression testing. While there will be a lag time, efforts to keep everyone on the same release should be made, otherwise the releases may diverge into separate products. This will lead to confusion about what the component should do and require additional maintenance resources.

Another issue for project managers is developer resentment. Many developers feel that code that is not developed in-house is not as good as their own code. In addition, there is the old hacker mentality of trying to get into the guts of the component, instead of using the interface. This will make integration of vendor-supplied software updates more difficult because the component has lost its "black box" functionality. Staff that can act as advisors on component use are required. The advisors will work with the development teams to recommend components for use on specific projects and harvest new components. Rotating development staff through the advisory positions will build knowledge about the component library and development process and help in identifying functionality that is used across development teams.

Finally, there are long-term maintenance considerations. If a component is developed in-house, who is responsible for maintaining it? Is it the developer of the component, or the user who may require modifications to apply it? Organizational change may again be necessary. A software library, with its own dedicated staff, is a good solution to this problem. The library staff is responsible for maintaining the components and managing potential conflict between multiple users. For purchased components, maintenance is also an issue. Vendors may be slow to correct problems, and you may find yourself maintaining the component and feeding the fixes back to the vendor. Even with prompt vendor response, the component must be regression tested and fed into the release schedule.

CONCLUSION

In conclusion, component-based development offers an environment that can facilitate multi-tier architecture and allow a true separation of data, business logic, and user interface. It has the potential to increase developer productivity and lower costs but it is not an approach without risk. The advantages must be weighed against the risks over the entire software life cycle.

References

ActiveX FAQ, Microsoft Corporation, 1996.

Allen, Paul and Frost, Stuart, *Component-Based Development For Enterprise System: Applying the SELECT Perspective,* Cambridge University Press and SIGS Books, 1998.

Austin, T., *Is Network Computing Just a Slogan?* Gartner Group, 1997.

Hamilton, Graham (Ed.), *JavaBeans API specification Version 1.01,* Sun Microsystems, 1997.

Natis, Y., *Component Models Move to the Server,* Gartner Group, 1997.

Smith, D., *Microsoft Bolsters ActiveX: Developers Should Use Caution,* Gartner Group, 1996.

ABOUT THE AUTHOR

Nancy Stonelake is a senior manager with the Deloitte & Touche Consulting Group and has strong expertise in Object Technology.

Chapter 24
User Interface Rejuvenation Methodologies Available with Web-to-Host Integration Solutions

Carlson Colomb
Anura Gurugé

The ability to easily, economically, and very dramatically rejuvenate the anachronistic user interface of mainframe and AS/400 applications is an integral, valuable, and widely publicized feature of most contemporary Web-to-host integration solutions. Schemes to facilitate user interface rejuvenation when it came to PC/workstation-based host access have also been readily available with traditional screen emulation solutions (e.g., Eicon's Aviva for Desktops) for nearly 15 years. All leading 3270/5250 emulators offer high-level language application program interface (HLLAPI) or equivalent, at a minimum, as one possible means whereby the harsh and dated "green-on-black" screens of SNA applications can be intercepted and totally revamped before they are presented to a user. However, it is sobering to realize that not even 25 percent of the millions of mainframe and AS/400 screens that are regularly displayed around the world with these traditional access solutions have been rejuvenated. "Green-on-black" is still the norm when it comes to IBM host access.

A kind of cultural inertia has prevailed, up until now, that believed that "green-on-black" was acceptable and furthermore was in keeping with the notion of legacy access to data center systems. There was also the inevitable

0-8493-1161-6/02/$0.00+$1.50

issue of cost, with prior rejuvenation schemes invariably requiring some heavy-duty programming effort. This, however, is now changing, and changing rapidly with corporations across the world beginning to standardize on Web pages, within the context of intranets, Internet access, and extranets, as their preferred and strategic means for presenting and soliciting information. "Green-on-black" screens look highly incongruous next to Web pages; hence the importance of user interface rejuvenation options vis-à-vis Web-to-host integration products — especially with those offering either Web browser-based (e.g., 3270-to-HTML conversion) or browser-invoked (e.g., Java applet-based emulation) access to mainframe and AS/400 applications. It is also important to note that these rejuvenation options offered by Web browser-oriented host access solutions are significantly easier and less costly to implement than previous solutions.

Extensive and compelling user interface rejuvenation options are now available with both 3270/5250-to-HTML conversion and with applet-based thin-client terminal emulation. Implementations of both of these disparate access schemes now also offer straight-out-of-the-box AutoGUI schemes that automatically and on-the-fly apply a series of default transformations to "green-on-black" screens to make them more user friendly and modern without the need for any programming, scripting, or even customization. Today, 3270/5250-to-HTML conversion (e.g., Eicon's Aviva Web-to-Host Server or Novell's HostPublisher) is used more often for rejuvenation than applet schemes, due to two interrelated reasons. The first has to do with its popularity as a true thin-client solution for easily and quickly Internet enabling SNA applications, such as those for travel reservation, electronic investing, banking, and parcel/cargo tracking, in order to open them up for online access by the public over the Internet. The other has to do with the fact that most of the HTML conversion solutions offer some type of relatively easy-to-grasp and straightforward mechanism to facilitate extensive rejuvenation, in addition to built-in AutoGUI capabilities.

DIFFERENT LEVELS OF REJUVENATION

With 3270/5250-to-HTML conversion there is always some amount of automatic rejuvenation brought about by the fact that the host screens are now being rendered in HTML, and in addition are being displayed within a browser window, as *bona fide* Web pages. For a start, the HTML converted output, even if it is still devoid of graphical elements and mainly textual, is unlikely to have the trademark black background of 3270/5250 screens. On the other hand, there is usually no automatic rejuvenation with many applet-based emulation schemes, although more and more products are now beginning to offer some type of AutoGUI capability as an option (e.g., IBM's Host On-Demand). Applet-based emulation, true to its claim of being a *bona fide* tn3270(E)/tn5250 emulation, tends to still opt for "green-on-black" emulation windows — displayed alongside Web browser windows. There is, however, some valid justification for this. User interface rejuvenation may not be conducive to certain applications, in particular those that involve high-volume data entry or very high-speed, real-time transaction processing. User interface rejuvenation could slow down and get in the way of such applications. Applet-based 3270/5250 emulation, without rejuvenation, is thus the optimum way to cater to these applications within the context of Web-to-host integration.

3270/5250-to-HTML and the applet-based schemes thus offer two very different levels of rejuvenation:

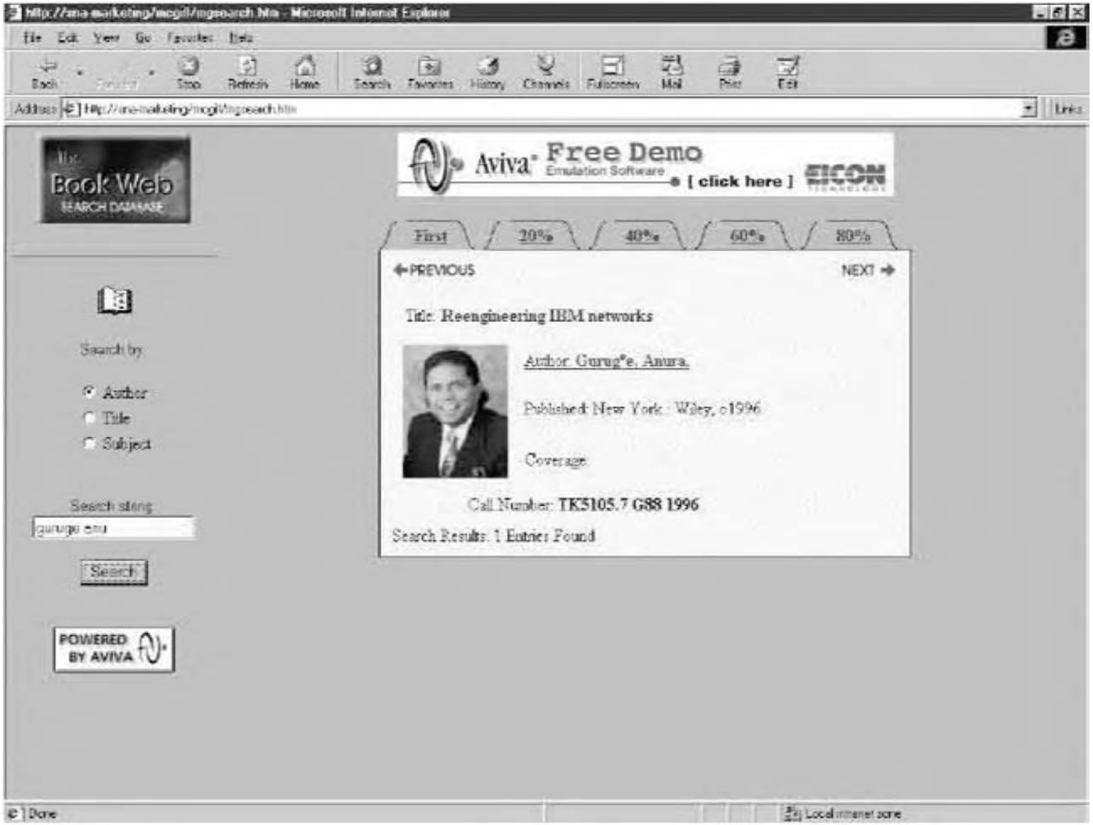
1. Simple default AutoGUI transformations, such as the inclusion of a colored background, substitution of Web page like input "trenches," and some screen color remapping. These transformations are, in effect, "screen-neutral," and apply to all the 3270 or 5250 screens of a given application that are using the rejuvenation process.
2. Application-specific, highly customized facelifts, replete with many graphical and possibly even multimedia elements, akin to the library access screen shown in Exhibit 1.

The two side-by-side screens in Exhibit 2 illustrate the notion of screen content-independent, default transformations achieved with a 3270-to-HTML conversion product. Note that the content and overall appearance of the rejuvenated HTML-based screen is much the same as that of the original screen. This type of minimal intervention and marginal reconstitution is the hallmark and goal of AutoGUI rejuvenation. Note, nonetheless, that the rejuvenated screen, even with this modicum of modernization, is considerably more appealing to the eye and more contemporary looking than the original. The bottom of the rejuvenated screen includes a command field to compensate for the absence of the tool bar found in the emulator window, as well as a set of buttons and an action key input box to emulate 3270 program function (PF) key actions.

Examples of 3270-to-HTML conversion solutions that offer an AutoGUI capability include Eicon's Aviva Web-to-Host Server, Novell's HostPublisher, Attachmate's HostSurfer for the HostPublishing System (HPS), and Intelligent Environments' ScreenSurfer. Examples of applet-based thin-client emulators that now have an AutoGUI function include IBM's Host On-Demand, OpenConnect's AutoVista, and WRQ's Reflection EnterView 2.5.

REJUVENATION OPTIONS WITH APPLET-BASED THIN-CLIENT EMULATORS

At present, there is really no commonality, consensus, or market-leading approach, let alone any standards, when it comes to Web-to-host related user interface rejuvenation. This, unfortunately, is unlikely to change in



http://ana-marketing/mcgill/mgsearch.htm - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Edit
Address http://ana-marketing/mcgill/mgsearch.htm
Links
The Book Web SEARCH DATABASE
Aviva Free Demo Emulation Software [click here] EICON
First 20% 40% 60% 80%
PREVIOUS
NEXT
POWERED BY AVIVA
Done
Local intranet zone

Search by

● Author
○ Title
○ Subject

Search string: gurugo anu

Search

Title: Reengineering IBM networks

Author: Gurug*e, Anura.

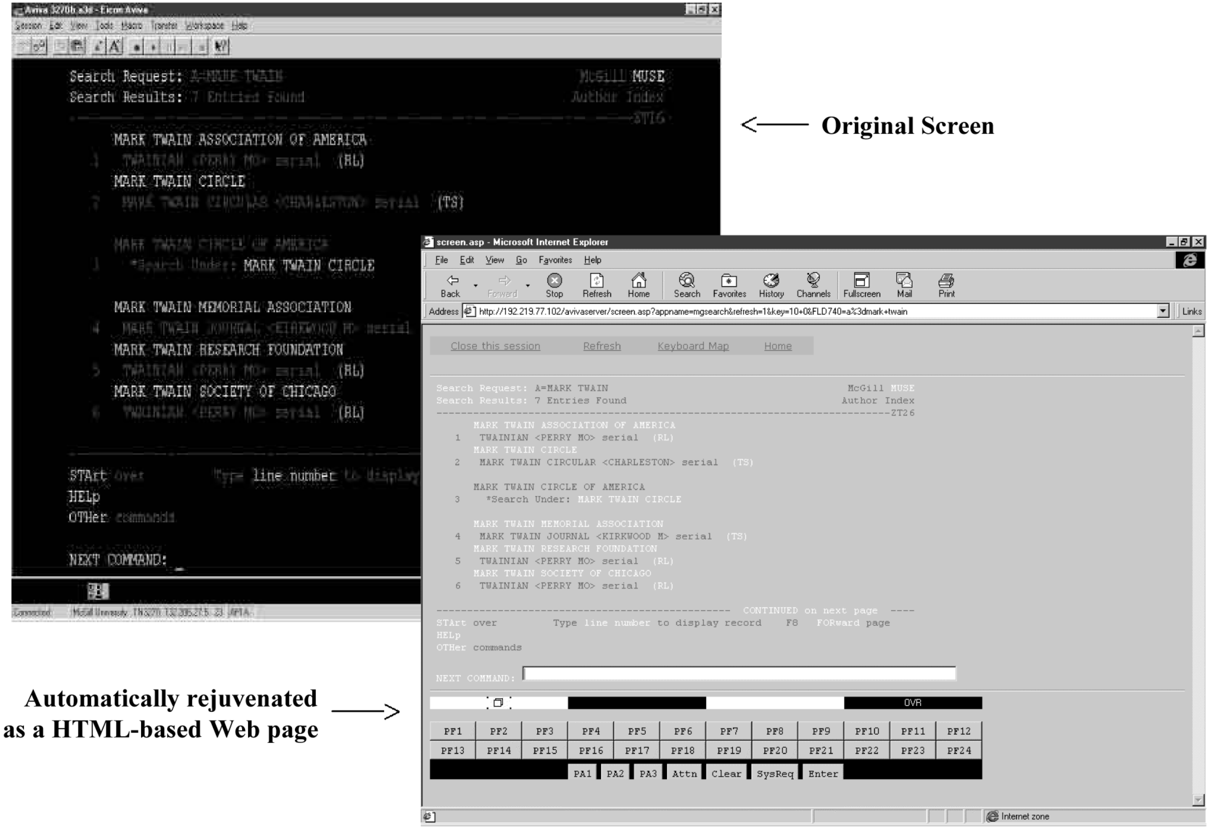
Published: New York : Wiley, c1996

Coverage:

Call Number: TK5105.7 G88 1996

Search Results: 1 Entries Found

Exhibit 1. Library Access Screen


Aviva 3270th.a3d - Eicon Aviva
Session Edit View Tools Macro Transfer Workspace Help

```
Search Request: A=MARK TWAIN                                   McGill MUSE
Search Results: 7 Entries Found                                Author Index
---------------------------------------------------------------------ZT16
      MARK TWAIN ASSOCIATION OF AMERICA
  1    TWAINIAN <PERRY MO> serial  (RL)
      MARK TWAIN CIRCLE
  2    MARK TWAIN CIRCULAR <CHARLESTON> serial  (TS)

      MARK TWAIN CIRCLE OF AMERICA
  3     *Search Under: MARK TWAIN CIRCLE

      MARK TWAIN MEMORIAL ASSOCIATION
  4    MARK TWAIN JOURNAL <KIRKWOOD M> serial
      MARK TWAIN RESEARCH FOUNDATION
  5    TWAINIAN <PERRY MO> serial  (RL)
      MARK TWAIN SOCIETY OF CHICAGO
  6    TWAINIAN <PERRY MO> serial  (RL)

START over         Type line number to display
HELp
OTHer commands

NEXT COMMAND: _
```

http://192.219.77.102/avivaserver/screen.asp?appname=mgsearch&refresh=1&key=10+0&FLD740=a%3dmark+twain

```
Search Request: A=MARK TWAIN                                   McGill MUSE
Search Results: 7 Entries Found                                Author Index
---------------------------------------------------------------------ZT26
      MARK TWAIN ASSOCIATION OF AMERICA
  1    TWAINIAN <PERRY MO> serial  (RL)
      MARK TWAIN CIRCLE
  2    MARK TWAIN CIRCULAR <CHARLESTON> serial  (TS)

      MARK TWAIN CIRCLE OF AMERICA
  3     *Search Under: MARK TWAIN CIRCLE

      MARK TWAIN MEMORIAL ASSOCIATION
  4    MARK TWAIN JOURNAL <KIRKWOOD M> serial  (TS)
      MARK TWAIN RESEARCH FOUNDATION
  5    TWAINIAN <PERRY MO> serial  (RL)
      MARK TWAIN SOCIETY OF CHICAGO
  6    TWAINIAN <PERRY MO> serial  (RL)

------------------------------------------------  CONTINUED on next page  ----
START over         Type line number to display record    F8   FORward page
HELp
OTHer commands

NEXT COMMAND:
```

OVR
PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10 PF11 PF12
PF13 PF14 PF15 PF16 PF17 PF18 PF19 PF20 PF21 PF22 PF23 PF24
PA1 PA2 PA3 Attn Clear SysReq Enter
Internet zone

<---- Original Screen

Automatically rejuvenated as a HTML-based Web page ---->

Exhibit 2. Screen Content-Independent, Default Transformations

the short to mid-term. All of the current rejuvenation techniques are, thus, vendor and product specific — each with its own foibles and nuances. Obviously, they all have to provide mechanisms for realizing the basic operations such as session establishment, screen identification, screen-to-screen and intra-screen navigation, input and output field selection via either a row-and-column position designation or an indexing (e.g., third field on screen) scheme, and function key management.

There are essentially four very different ways to realize complex user interface rejuvenation, as opposed to AutoGUI transformations, with applet-based schemes. These options are:

1. **API-oriented schemes:** With this approach, there is an API or a Java (or ActiveX) object class library associated with the applet that provides total access to the underlying host access sessions, screens, data fields, and attributes. Java, C++ or Visual Basic (or comparable) development tools are then used with this API or object class library to intercept the host screens and then perform the necessary user interface rejuvenation functions — including data and graphical elements from auxiliary sources. IBM, Eicon, and Attachmate are promoting the Java-oriented Open Host Interface Objects (OHIO) specification as a strategic, object-oriented scheme for this type of rejuvenation — with Eicon's Aviva for Java V2.0 being the first to offer an implementation of this specification that is likely to become an industry standard. The Java-based Host Access Class Library (HACL), as implemented on IBM's Host On-Demand applet, is another example of this type of API approach. OpenConnect promotes an API known as JHLLAPI, which is a Java version of the HLLAPI found on traditional emulators. Some products also offer their host access functionality in the form of JavaBeans to facilitate client-side object-oriented software development.
2. **Applet-based client:** This method relies on an intelligent, customizable, applet-based, front-end client that executes on a PC/workstation and performs the transformations necessary to convert the 3270/5250 data stream into a rejuvenated interface, on-the-fly, and without any ongoing assistance or intervention from an intermediary server component. The applet is programmed, offline, with the requisite transformations via a companion design tool — usually referred to as a design studio. ResQNet technology (as available with IBM's Host On-Demand) is an example of this approach.
3. **Server augmented applet:** With this approach, an applet, using a heuristic, rules-based scheme, automatically performs many of the 3270/5250 structure-to-GUI element conversions. The applet will typically contain screen images that are roughly 3 kb in size for each screen that has to be rejuvenated. The applet then works in

conjunction with a server component that intercepts and preprocesses both the outbound and inbound data streams. Client/Server Technology's Jacada for Java is the quintessential example of this approach. (Client/Server Technology is now called Jacada, Inc.)

4. **Integrated Development Environment (IDE):** With this approach, a visual, drag-and-drop programming environment à la that associated with Visual Basic is used to extend a host access applet, on an application-specific basis, so that it displays a rejuvenated user interface. This methodology was pioneered by OpenConnect's OpenVista product. With this approach, all rejuvenated screen images that are likely to be used by the application are appended to the applet and downloaded "in-bloc," each time the applet is downloaded.

All of these rejuvenation techniques, as well as those for 3270/5250-to-HTML conversion, provide a mechanism through which the developer can record and capture the screen-by-screen dialog associated with a given application — along with all of the data entry requirements and the navigation through the various screens required to complete a dialog or intercept error conditions. Typically, the developer will have an open and active window that displays, online, the original "green-on-black" 3270/5250 screen(s) being rejuvenated. The rejuvenated interface, with any required graphical and multimedia components such as sound or animation, is formulated within another window. Data fields from the "green-on-black" window can be dragged across or "cut-and-pasted" into the new window.

IBM's Host On-Demand provides the best-known example of API-based user interface rejuvenation. Host On-Demand offers two separate means for realizing user interface reengineering — in addition to providing a complete set of JavaBeans, with Ver. 3.0 onward, to facilitate programmatic access. The first of these methods is the provision of a Host Access Class Library API that can be used with any Java development tool, as well as with C/C++, Visual Basic, PowerBuilder, and LotusScript, to create a reengineered user interface or realize programmatic access. Other products, such as Eicon's Aviva for Java V2.0, offer similar Java-centric class libraries (e.g., the Aviva Class Library [ACL]). The second option offered by IBM is the technology from ResQNet.com. ResQNet realizes its integration with Host On-Demand to perform its rejuvenation functions using the Host Access Class Library API.

The Host Access Class Library (HACL) includes a set of classes and methods that provides object-oriented abstractions to perform the following tasks on a tn3270(E) or tn5250 connection with an SNA application:

- Read a screen image in terms of its 3270/5250 data stream
- Send input to the application, in a 3270/5250 data stream, as if it was coming from a 3270/5250 screen
- Specify a specific field relative to a display image through a numerical indexing scheme (e.g., third unprotected field on the screen)
- Read and update the 3270/5250 status line that appears at the bottom of 3270/520 screens in an area designated as the operator information area (OIA)
- Transfer files
- Receive and post notifications, asynchronously (i.e., not in real-time), of designated events such as the arrival of a new screen

The OHIO interface, as implemented by Eicon's Aviva for Java V2.0, can be thought of as an extended version of HACL targeted at becoming a vendor-neutral industry standard in the future.

ResQNet's rejuvenation technology revolves around the premise of dynamic pattern recognition and substitution. In this respect, ResQNet technology bears some resemblance to that available with CST's Jacada. ResQNet does the bulk of its processing at the client without continual, ongoing assistance of an intermediate server component. To do this, ResQNet relies on an intelligent Java client that is typically 300 to 500 kb in size. Within the context of Host On-Demand, this applet relies on Host On-Demand functionality, accessed via the Host Access Class Library API, to establish communications with an SNA application and to interchange data with it. Extensive customization is achieved using the separately priced Customization Studio and Administrator options of the product. The Administrator capability permits the capture of the screens that are to be further customized by the Customization Studio. Through the Customization Studio, one can rearrange fields, insert graphical images, include check boxes, add new fonts, and perform any kind of text string translation — including translating from one language or character set to another.

CST's Jacada for Java is another popular and powerful applet-based means for realizing user interface reengineering. Much of the user interface conversions performed by Jacada revolves around a potent, rules-based system known as CST KnowledgeBase. CST claims that the KnowledgeBase, at present, contains over 700 3270/5250-centric pattern definitions that permit the dynamic recognition of oft-found 3270/5250 screen elements — for example, the F8-Forward, F7-Backward, and F3-Exit PF-key definition designations that invariably litter the bottom of most 3270/5250 applications. Each pattern definition included in the KnowledgeBase has a substitution string that may involve graphical elements associated with it. The conversions specified in the KnowledgeBase can be automatically applied in offline mode to an application's screens through CST's Jacada Automated Conversion Environment (ACE). The Java applet that will render and manage the rejuvenated interface, as well as host communications, albeit with ongoing support from a Jacada server, will be generated by ACE without the need for developer intervention. There is a facility whereby the conversions specified within the KnowledgeBase can be overridden, for a particular rejuvenation process, by ACE. Extensive customization is also possible using ACE, where a developer, aided by easy-to-follow wizards, can capture, online, the screens that need to be reengineered and then perform the necessary conversions using a combination of the transformations included in the KnowledgeBase and bespoke alterations. The average size of a Jacada Java applet required to render a typical rejuvenated screen is around 3 kb. Often-used Java classes and screen layouts can be cached on a PC's hard drive or RAM memory to minimize the amount of data that has to be downloaded from the Jacada server.

With OpenVista's IDE approach, the IDE, following drag-and-drop instructions from a developer, will generate the Java code to create a single applet that will display and handle the new rejuvenated interface — along with all of the underlying tn3270(E)-oriented interactions necessary to communicate with the host application. With the OpenVista approach, rejuvenation-related transformations are not done on-the-fly by the applet, at the client, as it receives 3270/5250 data stream from the application. Instead, the required transformations are designed into the applet. If so desired, a Java applet produced by OpenVista can be modified, augmented, or refined using any of the popular Java development tools. There is even an OpenVista-provided API to facilitate quick access into the Java classes that appear in the applet. The finished applet is then stored at the appropriate Web server so that it can be dynamically downloaded to a browser — and, where applicable, cached on a PC/workstation hard-drive.

REJUVENATION OPTIONS WITH 3270/5250-TO-HTML CONVERSION

3270/5250-to-HTML conversion products typically offer two different techniques for facilitating user interface rejuvenation. These options are:

1. **Scripting:** With this approach, a script-based mechanism that either leverages popular scripting schemes (such as JavaScript) or is vendor specific and proprietary is used to reengineer the user interface. Novell's HostPublisher, for example, allocates three HTML templates to each application: one template for the data transporting LU-LU session, another for the SSCP-LU control session, and the third for the bitmap Web page that is used to support light-pen and cursor positioning operations. The template associated with the LU-LU session can be customized, typically with a JavaScript, to provide a set

of conversions that apply to all the 3270 screens displayed by that application.

2. **API-based rejuvenation:** With this approach, as with its counterpart vis-à-vis applet-based emulators, one or more APIs are provided to enable developers to easily access the output being produced by the HTML conversion process. These APIs can be accessed from programming language such as C, C++, Visual Basic, Visual J++, or Microsoft's Visual InterDev.

Scripting typically is the easiest, most expeditious, and consequently highly popular way to realize interface reengineering with 3270/5250-to-HTML. Scripts enable dynamic content to be added to Web pages. The scripting scheme may be client- or server-centric. Both schemes enable the browser image seen by the user to be made up of intermingled HTML code, scripting code, Java applet code, as well as objects such as Enterprise JavaBeans. With the client-centric approach, the script code (e.g., JavaScript code) necessary to handle the new presentation elements is embedded within the HTML page representing one or more 3270 screens. This code is then downloaded to the client PC/workstation along with the rest of the Web page. The code will execute on the client, typically within the Java Virtual Machine within the browser, to handle various components of the new interface. Most of the server-centric approaches, such as that offered by Eicon's Aviva Web-to-Host Server, revolve around Microsoft's Active Server Page (ASP) methodology for browser-neutral, server-based scripting. Support for ASP-based scripting came to be with Microsoft's Windows NT-based Internet Information Server (IIS) 3.0 Web server. ASP is also now supported by Microsoft Peer Web Services Version 3.0 on Windows NT Workstation and Microsoft Personal Web Server on Windows 95/98.

ASP permits the creation of dynamic, highly interactive, high-performance Web server applications. ASP works with any ActiveX scripting language and has integrated support for VBScript, JScript, and InterDev. Support for other popular scripting languages such as IBM's REXX and Perl (i.e., Practical Extraction and Report Language) is available via widely available plug-ins from third parties. ASP, in addition, permits multiple scripting languages to be used interchangeably to create a single Web server application. ASP relies on scripting engines developed using Microsoft's Component Object Model (COM) architecture to process scripts. Irrespective of the scripting scheme used, the output of an ASP application is always HTML. It is thus ideally suited for creating and manipulating HTML within the context of interface reengineering based on 3270/5250-to-HTML conversion. ASP, at least at present, can only be used with Microsoft Web servers. To support other Web servers, most vendors that offer ASP-based interface rejuvenation typically also offer a non-ASP scheme. Attachmate, for example, provides a Visual Basic-to-HTML capability in addition to its ASP support.

Interface rejuvenation via an API as supported by many of the 3270/5250-to-HTML products is in essence a logical equivalent of the HLLAPI and EHLLAPI interfaces that have been universally available with all major 3270/5250 fat-client emulators and tn3270(E) clients for many years — for the development of so-called screen-scraping-based client applications. Screen scraping, as graphically alluded to in the phrase, refers to the notion of dynamically capturing, via the API, what would be the 3270/5250 screen images displayed by an SNA application by intercepting and interpreting the data streams transmitted by that application. Screen scraping, which is performed at the client, permits the content of the 3270/5250 data stream to be used as input to a client-resident application driving a GUI or performing a different business task to that of the SNA application whose screens are being scraped. 3270/5250-to-HTML is in reality also a screen-scraping technique — albeit server, as opposed to client, based and HTML centric.

The APIs, which can typically be invoked from any contemporary programming language, permit a server component to be developed that talks 3270/5250 on the host side, and generates HTML on the Web server side. To be fair, this API approach, which is obviously highly flexible and extensible but at the same time significantly more involved than scripting, is better suited for developing sophisticated new applications that extract data from SNA applications, than for just realizing user interface rejuvenation. The exact machinations involved in reengineering a user interface via an API-based scheme, as is to be expected, are product as well as application language specific.

BOTTOM LINE

The bottom line when it comes to Web-to-host related user interface rejuvenation is that there is plenty of choice, whether it be for the 3270-to-HTML conversion solutions or the applet-based emulation approaches, spanning the entire spectrum of possibilities ranging from out-of-the-box AutoGUI schemes to those involving *bona fide* Java applet programming with a tool such as Visual Café or Microsoft's InterDev. The key frustration is the lack of any commonality among the approaches advocated by the market-leading vendors, although OHIO may eventually become an industry standard. With scripting schemes for 3270/5250-to-HTML conversion, ASP, although server specific, is highly popular and widely used to the extent that it can be viewed as a *de facto* standard. With all of the options now readily available, including the powerful and compelling AutoGUI schemes, there is really no excuse for not rejuvenating the dated and hostile user interfaces of mainframe and AS/400 applications.

ABOUT THE AUTHORS

Carlson Colomb is director of marketing for the Aviva Business Unit at Eicon Technology. He has worked for a number of IT vendors in the enterprise solutions market, including EDI, E-commerce, Fax Servers, and IBM Host Access. Colomb has a Bachelor of Commerce, with a major in marketing, from AACSB-accredited Concordia University.

Anura Gurugé is the editor of Auerbach's *Data Communications Management* and an independent technical consultant who specializes in all aspects of contemporary IBM networking. He has first-hand, in-depth experience in SNA-capable i•nets, SNA/APPN/HPR/AnyNet, Frame Relay, Token-Ring switching, ATM, system management, and the nascent xDSL technologies. He was actively involved with the Token-Ring switching pioneer Nashoba Networks, which was acquired by Cisco Systems in 1996.

In a career spanning 24 years, he has held senior technical and marketing roles at IBM, ITT, Northern Telecom, Wang, and BBN. He can be contacted at (603) 293-5855 or guruge@cyberportal.net.

Chapter 25
Determining Whether To Buy or Build an E-Commerce Infrastructure

Carol A. Siegel

According to International Data Corporation, the Internet commerce market will top the U.S.$425 billion mark by 2002. A recent Pricewaterhouse-Coopers survey of Global 100 companies found that 85 percent see E-business as an investment priority over the next three years. Enabling this unprecedented demand for E-services will require business managers to increasingly become technically savvy so that they may make informed decisions on where the IT dollar is spent. As E-commerce business requirements become more challenging to implement from both a cost and technical perspective, companies are looking to outside providers to supply expertise in application development, as well as to offer connectivity options and application hosting facilities. By outsourcing a portion of their infrastructure, a company will not only reduce its fixed costs because it will not have to build additional infrastructure, but it can also contain its E-business variable costs by purchasing or canceling incremental services as directly dictated by business demand.

To describe E-commerce infrastructure as a subset of all infrastructure may be a useful logical tool, but as networks continue toward global interconnection and E-commerce products run end-to-end applications, there will ultimately be no division between E-business and business from an infrastructure perspective. E-business links intranets, extranets, the Internet, as well as legacy systems — all utilize a company's entire infrastructure. But what is E-commerce infrastructure exactly? Broadly described, E-commerce infrastructure is the plumbing behind the façade that enables E-business to work. Think of it as all of the hardware, software,

0-8493-1161-6/02/$0.00+$1.50

networking, security, and support that are behind the screen that the end user sees on his computer. This is the client view. From an infrastructure viewpoint, in order to deliver this information to the client, E-services must be provided. The goal of these E-services is to enable multiple E-businesses — in fact, the objective of E-service offerings is to enable as many E-businesses as effectively as possible. More formally, infrastructure E-services can be decomposed into at least four categories: application, network, security, and support. These E-services form the building blocks that lead to rapid deployment of new applications and sustain efficient operation of those applications going forward. An E-service capability must have the flexibility to expand on demand as new businesses are launched, and contract just as quickly in those periods of nonexpansion. In many cases, mission-critical E-commerce applications may require an array of software applications, server hardware, networking bandwidth, and support staff that the current infrastructure simply cannot deliver. This is the classic reason to look to outsourcing as a means of addressing those variable costs and reducing fixed costs. By outsourcing, a company can gear up quickly to meet demand, and pare down just as quickly as requirements change — while minimizing long-term impact on standard operating costs.

How well an organization develops, implements, and maintains its E-services can make the difference between its ultimate success and failure. A loss of service for a day or even an hour can result in a loss of consumer confidence, decline in stock value, potential loss of funding — not to mention a direct loss of revenue. E-services that do not facilitate the E-business strategy of the company or those are just plain too expensive can lead to arrested delivery capability. Companies whose individual businesses build their own independent infrastructures will ultimately be saddled with redundant efforts, as many of the businesses will spend dollars on research and implementation of the same technologies and products. As all company networks will inevitably be connected, the cost of integration efforts may exceed any initial cost savings to the business. In the present world of internetworking, it makes sense for a company to work as an integrated unit and build an infrastructure that works for everyone (ideally). The company's central infrastructure should be built in such a way that it satisfies both the tactical and strategic needs of the business, being based on technologies that stand the test of time. Defining the correct array of E-services should be an integral part of the IT tactical and strategic planning process. The actual provision of individual E-services can be achieved either by building them in-house or outsourcing part or all of them to service providers such as Internet service providers (ISP), telecommunications and networking providers, Web hosting facilities, etc. Web hosting, the most commonly outsourced E-service, is a term that is used to describe content publishing, 24 × 7 support, application and transaction processing, and secure access. ISPs can provide secure backbone connectivity and access between the client(s), the hosting facility, and the company. Certain ISPs do offer hosting capabilities as well, as service companies expand their E-offerings and business models are in a state of constant evolution. When considering outsourcing alternatives, ultimately, the decision comes down to one: whether it makes sense to build E-services, or would it be more cost effective to buy them as needed, integrating them into the current IT environment. In order to arrive at the point where such a decision can be made, significant analysis regarding the E-service needs of the organization in question must be performed.

GETTING TO THE BUY/BUILD DECISION POINT

In order have all the information necessary to be able to make an informed decision as to whether to build or purchase E-services, there are (at a minimum) eight specific tasks that need to be performed. If this process were flowcharted and made a project that ran over a four-month period (for example), the resultant chart might look like Exhibit 1. Tasks are labeled for discussion purposes.

Task 1: Collect all the E-commerce information in the company by interviewing all of the business units (BUs). This will involve taking an inventory of all existing and future E-commerce applications and understanding each BU's individual E-commerce requirements. Key areas of information to assemble are the business model, the business drivers, industry trends, competitor analyses, risk analysis, budget constraints, rollout requirements, and service support levels. The purpose of this task is to gather the raw business requirements that are to be fed into the requirements analysis process.

Task 2: Consolidate and interpret all the E-commerce information gathered from both a business and technical perspective. Identify the business progression of E-commerce from a technical complexity viewpoint, starting with view-only functionality like research and market information (this requires pushing out information onto a Web site). Proceed in ascending order, ending with online trade execution as the most difficult to implement (this is a fully interactive application with real-time updating of back-end databases). Chart the BU information over time to understand when each line of business will need that specific business functionality. Also indicate implementation dates in terms of planning stages and go-live. It is this process that attempts to bridge the gap between business terms and technical functionality, as each of the business endeavors will have to be decomposed into technical components to understand the complexity of implementation. Business managers must work together with technical architects to identify this list.

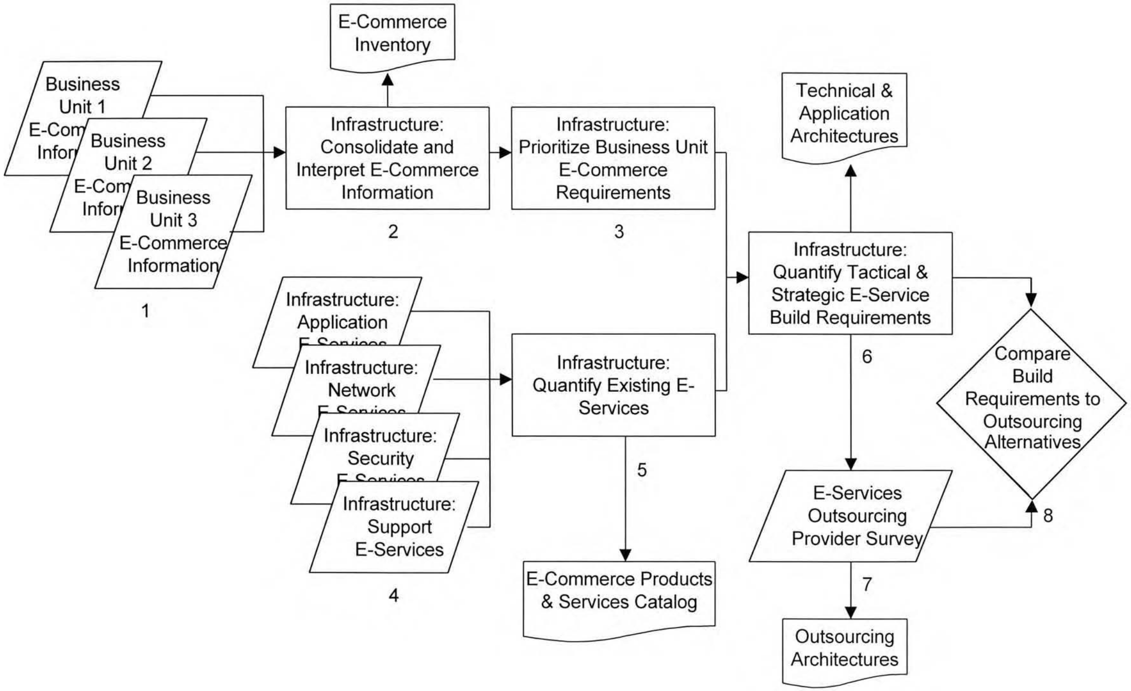

Business Unit 1 E-Com Infor
Business Unit 2 E-Com Infor
Business Unit 3 E-Commerce Information
1
E-Commerce Inventory
Infrastructure: Consolidate and Interpret E-Commerce Information
2
Infrastructure: Prioritize Business Unit E-Commerce Requirements
3
Infrastructure: Application E-Services
Infrastructure: Network E-Services
Infrastructure: Security E-Services
Infrastructure: Support E-Services
4
Infrastructure: Quantify Existing E-Services
5
E-Commerce Products & Services Catalog
Technical & Application Architectures
Infrastructure: Quantify Tactical & Strategic E-Service Build Requirements
6
E-Services Outsourcing Provider Survey
7
Outsourcing Architectures
Compare Build Requirements to Outsourcing Alternatives
8

Exhibit 1. The E-Services Buy or Build Decision Flow

Task 3: Prioritize the E-commerce service needs of the organization. Decide which projects are critical in the near term to the organization as a whole, because it is for them that the central infrastructure will be built. Understand the scope and depth of infrastructure that must be available (bought or built) in order to support the "collective good" of the business requirements. This prioritization may be grounded in financial analysis, but may have political ingredients as well. As each BU feels that it is the most important, infrastructure will have to make some hard decisions as to where the spend is allocated. Task 3 will serve as input into Task 6 the purpose of which is to quantify the architecture build requirements.

Note: Tasks 1 to 3 can be executed in parallel with Tasks 4 and 5.

Task 4: Identify the detailed E-services for each individual infrastructure discipline: application, network, security, and support. Examples of these would be static/dynamic content, connectivity options, cryptographic technologies, and service availability, respectively. Collectively, these are the E-commerce-related infrastructure services that enable all the business applications. The four infrastructure disciplines work in conjunction with one another to provide these services to the BUs. After identifying each of the detailed services, they must be quantified, which leads to Task 5.

Task 5: Quantify all existing E-services by determining a way to measure them. For example, for disk space, an application E-service could be measured in mega or gigabytes. Bandwidth, a networking E-service, could be measured by a range of 56K to 155 Mbits. Cryptographic technologies, a security E-service, could be measured by the number of digital certificates issued or processed. Service reliability, a support E-service, could be guaranteed at 99.9 percent. Then, for each unit of measurement, a price per unit must be assigned. It is here that the true cost of these E-services to the company will be determined. Only by knowing the component costs of the four categories of E-services, can a price tag (and therefore profit margin) be placed on them for sale to the BUs. The last step in this process is to commit to deployment availability per geographical location for each component service. For example, there may be a worldwide rollout starting with the United States, then Europe, followed by Asia. As for Task 3, this task will serve as input into Task 6.

Task 6: Quantify the tactical and strategic E-service build requirements. This task will take the prioritized, collective BU requirements and represent them in the same manner as the company's existing E-services. This will show how the company's technical infrastructure capability can deliver the required level of E-services that permit the businesses to do the E-commerce they have planned. As each E-service has been priced, infrastructure can see what the total cost will be to deliver these services. There will, undoubtedly, be a gap between what infrastructure can offer the businesses today and what it needs to build both in the short and long term. The outcome of this process is to develop a tactical and strategic E-service build requirement by developing the technical and application architectures. The technical architecture is a description of technologies and their integration for building services, consistent with the service and design characteristics defined in the business requirements and the application architecture. The application architecture is a definition of the application services that need to be built to support the strategic business model. This includes a definition of what the service will do, the application service interfaces, and performance.

Task 7: Survey and analyze potential outsourcing partners to understand alternative service options. Task 6, the build requirements, will serve as input here. The chosen vendor must be capable of providing a reliable, high-performance, redundant, and secure environment, as well as be financially stable. If applicable, the outsourcers should propose technical and application architectures, as well as network diagrams, showing how their facilities can be seamlessly integrated with end users and the company network.

Task 8: Compare the company's strategic E-service build requirements to the outsourcers' proposals. The result of this comparison will indicate whether to buy or build E-services from a technical and financial perspective. There will be other, more intangible factors to be considered as well; these are discussed in a later section.

The elapsed time of this entire process can take anywhere from two to six months, depending on various factors: the size of the organization (number of BUs), the disparity in architectures, the depth of the outsourcing survey, or simply the resources (or lack thereof) of dollars and staff needed to perform this study. Clearly, due to the time sensitivity of the subject at hand, the more resources applied, the better.

INTERVIEWING THE BUS FOR E-COMMERCE APPLICATIONS

To obtain the information necessary to make an informed decision, each of the BUs will be interviewed as to their existing and future E-commerce plans. This may be a time-consuming process for several reasons. First, there might not be an existing inventory or current list of all E-commerce applications in the organization. In medium- to large-sized companies, it is often the practice of BUs to finance and launch their individual E-commerce products with minimal or no involvement of any central or "corporate" department (at least initially). Generally frowned upon by the central departments, the BUs operate in this manner for a number of reasons. By building and maintaining their own infrastructures, most BUs feel that the application development process is faster and cheaper, as they only have to pay for their separate piece without allocating to a general funding pool. This logic becomes increasingly flawed as front-end systems need to be fed by data from back-end systems. End-to-end processing dictates connectivity between all of the infrastructure, which means that technologies and products must be compatible (refer to Exhibit 2 under architectural principles). Only in those organizations where development and implementation standards regarding network connectivity, security, customer communications, and software applications/tools play a more dominant role, are central departments included at the planning stages of the product life cycle. It is always advisable, however, to integrate centralized standards at product conception time, as this will prove to be the most cost efficient and have the most growth potential in the long term. Standardization of technologies like Web servers and browsers, network, database, security, middleware, messaging, etc. across all business lines will lead to control of full lifecycle costs of investment (procurement, deployment, and maintenance), as well as increased re-use of processes, applications, and technology components. Adherence to firmwide standards leads to increased inter-business sharing of information, which is an integral part of the Web business model. Another added benefit of using firmwide standards is the potential for inherent compliance with regulatory requirements (critical for financial institutions). If a company has done its due diligence with respect to incorporating compliance mechanisms into its standards, BUs will automatically be in compliance with the regulations.

It is recommended that a standard set of questions be developed before the BU interview process commences. The purpose of these questions is to give the infrastructure technical architecture team the necessary information so it they can identify and quantify the current E-service levels and estimate future demands. Exhibit 2 offers a sample questionnaire that can be used for this interview process. For purposes of illustration, all of the questions in this table (and article) refer to the financial services sector. In general, E-commerce can be broken down into two main categories of business types: consumer-to-business and business-to-business. In the financial services sector, examples of consumer-to-business applications are retail online banking and online brokerage. Examples of financial services business-to-business applications are online trading of fixed-income securities for dealers and online order execution of equity index options and futures.

For the information gathering process, it is recommended that intense resources be applied. The reasons for this are due to rapidly changing technology cycles and the time to market urgency for applications. It makes sense to have the most comprehensive information available as soon as possible so that the collective good of the business requirements

Exhibit 2. Sample Business Unit (BU) Questionnaire

Questions for BU	Points to Consider
What is the name and business description of the E-commerce product?	• Business model description • Description should be in business terms not technical terms • All key features should be identified
What are the main business drivers for this product?	• Increase revenues with new customers and services • Decrease cost of operation • Open new markets • Customer retention • Gain new customers in existing markets • Sell more products and services to existing customers • Improve speed of delivery and service levels • Channel integration
Who are your main competitors in this business space, and what are they doing?	• Scope, functionality, and capabilities of Web site presence and business-to-business application • Information regarding future rollout plans • Strategies regarding delivery of E-services • General industry trends
What are your greatest concerns with respect to implementing E-commerce applications?	• Competitive efforts • Security • Business case viability • Deployment costs • Transaction costs • Processing volumes • Customer acceptance • Response time • Speed to market • Customer support
Who are the users of your application, and where are they physically located?	• Is the product a consumer-to-business or a business-to-business application (e.g., retail online banking or online trading of fixed income securities for dealers, respectively), or are the users internal business units? • Location of the users: domestic and international (i.e., U.S., Europe, Asia, Latin America, etc.) • Growth requirements • Rollout timetable
What are the timing and budgetary constraints?	• Date by which the product must go live • Financial information regarding product creation, rollout, and maintenance

Exhibit 2. Sample Business Unit (BU) Questionnaire (Continued)

Questions for BU	Points to Consider
Which technologies, software/hardware products, and technical architectures is the product based on?	• Present and future deployment • Architectural principles: — Interoperability — Scalability — Extensibility — Reliability — Portability — Consistency
What are the E-service level requirements for the product?	• Sensitivity and criticality of the business information manipulated by the application • Security (i.e., authentication, authorization, integrity confidentiality, privacy, nonrepudiation) • Performance (i.e., position update within x seconds, 200+ concurrent requests) • Capacity (i.e., position updates/day, number of clients, number of transactions/day) • Availability (i.e., 24 × 7 × 365, 7–7 for business days) • Monitoring (i.e., network, security state and event monitoring, intrusion detection, access log reporting) • Audit (i.e., audit trails, access violations, regulatory requirements) • Help desk and technical support • Redundancy (i.e., servers, hosts, network components) • Dependability (i.e., fault evasion, tolerance, and prevention) • Disaster recovery (i.e., recovery time objectives)
What is the preferred user fee model?	• No charge • No charge — bundled with other offerings • Monthly use • Per transaction basis • Tiered pricing tied to balances • Combination of flat fee and transactional charge

can be determined. This information forms the basis for all subsequent decisions and, as such, needs to be comprehensive, accurate, and timely.

INTERPRETING BUSINESS REQUIREMENTS

The purpose of gathering business requirements is to understand what is and will be needed in terms of infrastructure to enable the organization's E-commerce efforts to go forward. Once the raw business requirements have been gathered, they need to be consolidated and categorized in business terms in increasing order of technical complexity. This will show which technologies will be needed to satisfy business needs and when. This, in and of itself, is a difficult task, as it requires a relatively in-depth understanding of how each application works. An example of such a consolidation and representation is shown in Exhibit 3. Typical BUs for financial services are equities, fixed income, retail banking and brokerage, derivatives, futures and options, foreign exchange, private banking, and cash management. Target customers of these BUs might be retail clients, hedge funds, pension funds, corporates, banks, insurance companies, or internal employees or other internal businesses.

In Exhibit 3, the *y*-axis shows the natural progression of financial services E-commerce functionality from a technical perspective, starting with the most simple on the bottom (Research/Market Information) and finishing with the most complex to implement (Trade Execution/Order Routing) at the top. When evaluating the implementation level of difficulty, consider the classic Web model represented by a three-tier architectural model: client, server, and back-end database or resource manager. At the more simple levels, technical implementations may involve only static content, while at the higher levels, dynamic and interactive content are implicit. For example, to view market research data, information need only be pushed out from the back end to the client or presented on the Web server for client access from the Internet. Information flows out of the organization, with no query of back-end internal data by an external client. From an application and security perspective, this is relatively easy to do. More complex business functionality such as pricing and ultimately trade execution involves real-time or near-real-time interactive data update where the data resides on multiple back-end resources. The data may involve market feeds or other data from sources external to the company. This data must then be integrated and presented back through the Web server to the client with performance and throughput guarantees. Security here becomes much more complicated, as strong authentication and nonrepudiation play a mandatory role in the transaction by guaranteeing identity and providing the legal framework for a binding electronic contract. These types of applications push the limits of today's technology, as there is practically no tolerance for error.

By charting the business requirements over time (the *x*-axis) in terms of planning and go-live dates, the infrastructure team of architects can understand which E-services will be needed and when. It is then their task to translate these business requirements into architectural requirements that, in turn, will specify the following:

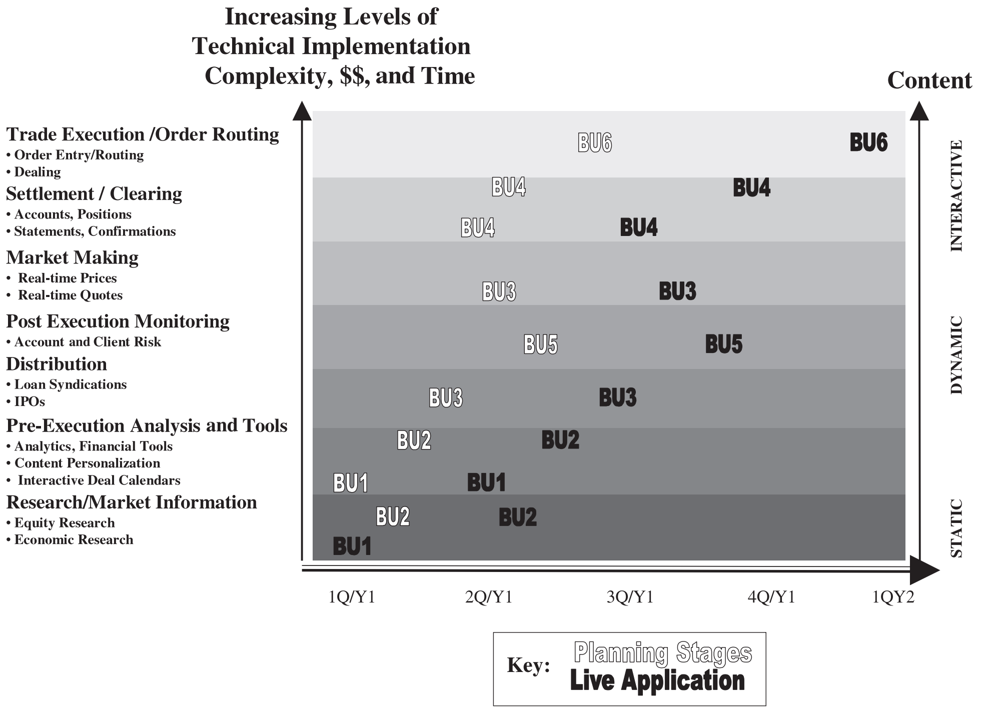


Exhibit 3. E-Commerce BU Requirements Consolidation, Financial Services Example

- The detailed E-services needed to support the business requirements, broken down by top-level E-service discipline (application, network, security, and support)
- Technical and application architectures that define and diagram standardized constructs to be used to deliver the desired E-services
- Integration and migration roadmaps, complete with timetables and costs, that show how the present infrastructure evolves into the strategic objective
- Architectural models designating components such as clients or servers that support the desired tactical and strategic architectures; the architecture itself is product independent; during the construction phase, specific product choices can be selected to best implement the architecture

All the above tasks and deliverables can conceivably be built and supported in-house by an infrastructure team — provided the required resources (dollars and staff) can be allocated. The ideal goal is to satisfy all requirements, but realistically, this is almost never possible. Before deciding what to build (or to outsource), other factors must be considered to narrow the scope. Although unpopular, one must ask the following questions: Are all BUs created equal? Each BU's requirements need to be prioritized and weighted. Whose needs should be satisfied first, given limited resources (as is always the case)? Is it the BU that produces that most revenue today? Or is it the BU that has the most strategic business value going forward? Or is it the BU whose technical architecture most closely fits the firmwide tactical and strategic architecture, lending itself to minimal expenditure for migration and implementation. Clearly, it will be some intersection of these types of decision vectors that will yield the elusive "collective business good." Only when the collective requirements have been prioritized and weighted, can the tactical and strategic architectures be developed.

IDENTIFYING AND QUANTIFYING E-SERVICES

Identifying and quantifying E-services is probably the most difficult task in this entire process. All areas of the technical organization play a role in providing these services and must coordinate with one another in packaging them as one service to the client. In this example, infrastructure's clients are the BUs that have planned or are doing E-commerce.

The most common E-commerce service provided to clients by an organization's technical infrastructure is hosting services. Web and application hosting provide the necessary hardware, software, network, and support organization that enable BUs to offer Web-based (Internet, intranet, extranet, etc.) business to their clients. Their clients are external clients, external business partners, or outside vendors, as well as (potentially) other areas within the organization itself.

Hosting services require considerable detailed component E-services from the four main categories: application, network, security, and support. Exhibit 4 shows a detailed list of typical E-services required for hosting E-commerce applications. This kind of breakdown must be performed for any E-service presented to the client or consumer of services (in this case, the BU).

The list presented in Exhibit 4 is divided into three sections: fees, basic services by default, and add-on services that can be purchased on-demand. It is presented in a common format for hosting services. This list provides the detailed E-services that, in combination, can satisfy consolidated business needs. Each detailed E-service falls into a top-level category, and will need to have a unit of measurement associated with it, a price per unit, and an availability timetable. Service level requirements (SLAs) needed to support each of the E-services will also need to be identified. Many of these component E-services will be difficult to quantify, as their true cost to the organization is nebulous and difficult to calculate. In many cases, the tools needed to measure the service are not available, and may need to be built. Painful as it might be, this exercise is mandatory to the task of arriving at the buy or build decision point. If an organization does not know the true cost of its E-services, how can it price any product or application composed of them or decide whether it makes sense or not to build more infrastructure?

ADDING VALUE BACK TO THE BUSINESS UNITS

During this entire exercise, a wealth of information is generated that merits notation. Reexamining Exhibit 1, note that there are four output documents that provide value to other areas within the organization. After Task 2, E-commerce efforts from all of the individual BUs have been consolidated, and a document labeled the “E-Commerce Inventory” can be generated. This document, when redistributed back to the BUs has great value. First, it is informative by showing each BU what everyone else is doing, and indicating their position on the general development curve. Consequently, the BU can start discussions with other BUs that are doing or will do the same kinds of applications. By doing so, they can share R&D information, product information, and learn from one another’s mistakes. Second, opportunities for cross-selling products between businesses may become apparent, as data from multiple back-end databases can be retrieved to form new customer bases and feed into multiple front ends.

	E-Service Category	Unit Measurement	Price/Unit
HOSTING SERVICES			
Initial Setup Fee	n/a		
Monthly fee	n/a		
Basic Services by Default			
URL	Application		
Static Content (e.g. HTTP)	Application		
Standard Dynamic Content (e.g. , CGI, Java and Flatfiles)	Application		
Server Log Files with Analysis and Reporting and Usage Statistic	Application		
DNS	Networking		
Security Services:			
Secure Internet Connectivity (e.g. SSL)	Security		
Hardened Demilitarized Zone OS	Security		
Secure External Connectivity	Security		
Security Alerts	Security		
Demilitarized Zone File Integrity	Security		
System & Network Intrusion Detection	Security		
Cryptographic Technologies	Security		
FTP Upload/Download or Drop Box	Application		
Basic Software Change Staging	Application		
Daily Backup (7 day rolling)	Support		
System Monitoring (24/7/365)	Support		
Network Monitoring (24/7/365)	Support		
Secure Physical Environment (Datacenter)	Support		
Quality of Service:			
Service Availability (7/24)	Support		
Service Reliability (99%)	Support		
Technical Support	Application		
Disk Space (1GB)	Application		
Bandwidth (56K to T3)	Networking		

Add-on Services			
Test Environment	Application		
Server Customization	Application		
Software Change Control Customization	Application		
Analysis & Reporting:			
Integrity Checking	Application		
Intrusion Detection	Application		
Customer Profiles	Application		
Audit Logs	Application		
Security Services	Security		
Non-Repudiation (Certificate-Based Signing)	Security		
Authentication: Client, Server Certificate Infrastructure	Security		
Integrity/Confidentiality	Security		
Entitlement/Authorization	Security		
Job Management	Support		
Transactional Services:			
Performance Guarantee (response time; throughput)	Application		
Transactional Guarantee (consistency)	Application		
Dynamic Content linked to database: Same Box	Application		
Dynamic Content linked to database: Internal Network	Application		
Server Customization	Application		
Software Change Control Customization	Application		
Test Environment	Application		
Connectivity Options:			
Dialup	Networking		
Dedicated Line	Networking		
Virtual Private Network (VPN)	Networking		
News	Application		
Online Chat	Application		
Market Data	Application		

Exhibit 4. E-Service Category Breakdown Worksheet

After Task 5 (quantifying existing E-services), an E-commerce product and services catalog can be developed that can be used as the basis for selling infrastructure services back to the BUs. This is the way the infrastructure spend is recouped directly from the businesses. After the cost of the E-service product is known, appropriate profit margins can be embodied as well. An example of such a catalog is shown in Exhibit 5. Note the correspondence of the items to those in Exhibit 4.

MAKING THE BUY OR BUILD DECISION

Using the completed worksheet, a comparative service survey can be conducted. Internet service providers, telecommunications/networking providers, and E-commerce service providers can be contacted and an apples-to-apples comparison can be made in terms of price, functionality, and service levels. Upon completing this exercise for the Web hosting example, it will become apparent as to why the E-services were listed in this manner. The E-services were constructed not only to parallel internal technical/application infrastructure offerings, but also to mimic the fee structures currently used by third-party Web hosting facilities.

In addition to fee structures, the outsourcer should be supplying a network diagram and architectural model of what exactly is to be outsourced. Typically, the decision to outsource involves allocating some or all of what is called "the middle tier" in architectural terms. A three-tiered architecture is generally the implementation model that is used for Web-based applications. Tier 1 refers to the client tier located on the Internet, but it can also refer to a business partner or extranet network. Note that this tier offers no protection to the systems located in it, as it is a public network. Tier 2, the middle tier, is called the server tier, and is normally where the Web and application servers and programs reside. In terms of security, this middle tier can be used as a demilitarized zone or DMZ (taken from the military term). The purpose of a DMZ is to protect the inner company network by not permitting direct access to internal information from the public Internet. Commonly, company information is replicated into the DMZ, pushed out to it from the internal network in a controlled manner. Access to the DMZ from the public side is limited and controlled by the use of firewalls, which can consist of bastion hosts and choke routers. Both sides of the DMZ are typically protected by these firewalls, which can be configured according to many parameters (e.g., IP address, IP service, host name, etc.). Tier 3, the back-end tier, is where the company's more sensitive information resides. This information is proprietary and confidential in nature and must be accessed only by those authorized to do so. Generally, unless there are significant security mechanisms in place, direct access to internal databases from an external client is not permitted, but always passes through a firewalled middle tier.

BASIC SERVICES BY DEFAULT
Initial Setup Fee, Monthly Fee

- URL
- Static Content (e.g., HTTP)
- Standard Dynamic Content (e.g., CGI, Java and Flatfiles)
- Server Log Files (analysis, reporting and usage statistics)
- DNS
- Firewalled Secure Internet Connectivity
- Encrypted FTP Upload/Download
- Software Change Staging
- Daily Backup (7 day rolling)
- System Monitoring (24/7/365)
- Network Monitoring (24/7/365)
- Secure Physical Environment (Datacenter)
- Service Availability (7/24 with prescheduled agreed on maintenance windows)
- Service Reliability (99 percent uptime)
- Technical Support (24/7/365)
- Disk Space (1 GB)
- Data Transfer Volume (2 GB)
- High Bandwidth (Fractional T3)

ADDITIONAL SERVICES ON DEMAND

- Dynamic Content Linked to Database on Same Box
- Dynamic Content Linked to Databases in Internal Network
- Secure Sockets Layer (SSL)
- Test Environment
- Connectivity Options: Dialup, Dedicated Line, VPN
- Server Customization
- Software Change Control Customization (Fixes and Version Control)
- Analysis & Reporting
    - Integrity Checking
    - Customer Profiles
    - Audit Logs
- Security Services
    - Strong Authentication
    - Intrusion Detection
    - Integrity/Confidentiality
    - Entitlement
    - Non-Repudiation
- Transactional Services
    - Performance Guarantees
    - Transactional Guarantees
- Job Management
- External Helpdesk
- Search Engine/Knowledge Management
- News Groups and Services
- Market Data
- Online Chat

Exhibit 5. E-Service Products and Services Catalog: Web Hosting

An architectural representation of the three tiers is shown on the left side of Exhibit 6. Listed under each tier are some typical technologies and products that correspond to them. On the lower right of Exhibit 6 is a network configuration diagram that is divided into three parts: the Internet, an E-commerce service provider, and a company network. Typical users and suppliers for these three network clouds are listed below each of them. The arrow that connects the two diagrams, which runs from Tier 2 of the architectural diagram to the E-commerce service provider cloud of the network diagram, shows a potential physical configuration of outsourcing the "middle" architectural tier.

Note that the outsourcer must provide full documentation on how it plans to achieve this effort, including, at a minimum, implementation roadmaps, migration plans, front- and back-end migration strategies, product choices, E-service support levels, timetables, costs, and growth projections.

This kind of comparison will yield hard numbers with which to draw conclusions based on current and future IT spends. There are other more intangible factors to consider, however, before arriving at the final decision. Some of these are the implicit benefits derived from building infrastructure in-house, including:

- The in-house staff is familiar with the IT environment; there will be a learning curve involved for consultants to bring them up to speed.
- Management will not have to deal with cultural integration issues between external consultants and internal employees; there can be resentment and a lack of cooperation.
- By developing infrastructure services in-house, management is making an investment in the technical capital of its staff; by outsourcing a key part of the operation, there may be minimal "transfer-of-technology" knowledge from the consultants to the employees.
- As a matter of course for in-house developed applications, there may be a series of reviews of the proposed application that have been integrated into the process without fanfare or chargeback. Reviews such as application security assessments, architectural compliance with corporate standards and policies, and a corporate communications review as to conformity with the official corporate look and feel of public presence may be performed in a perfunctory manner as part of any go-live process. There is also the inherent compliance with regulatory requirements. If a significant chunk of the operation was outsourced, these issues might require more formal efforts on the part of the corresponding internal entities, together with associated costs.

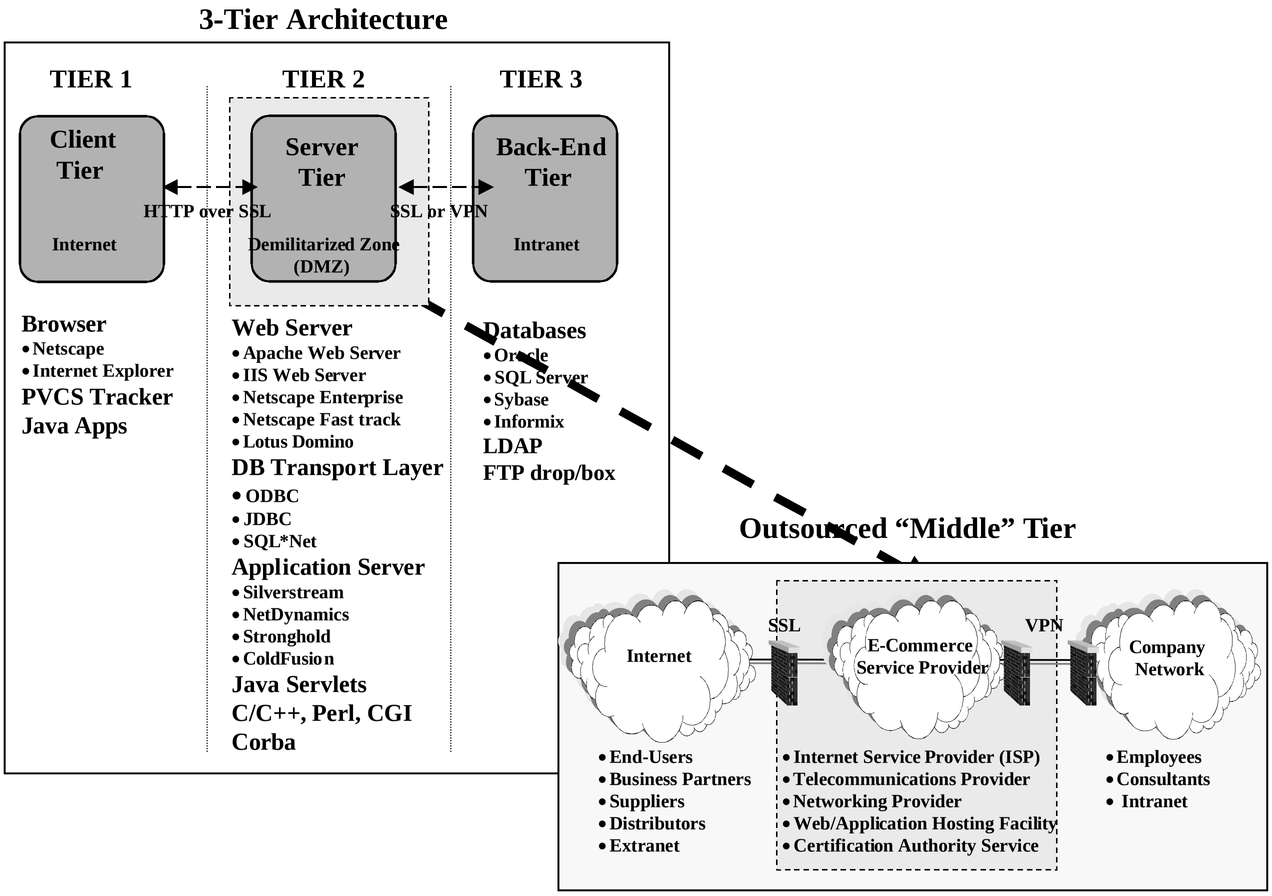


Exhibit 6. Outsourcing the "Middle" Tier

The decision whether to outsource will be based on both a cost/benefit analysis of the hard cost savings (fixed plus variable), as well as the softer pros and cons mentioned above. This so-called final decision here is really a fluid one — one that can and should be revisited on a quarterly basis. Looking back at Exhibit 1, it is important to note that the entire process (Tasks 1 through 8) should be an iterative one, repeating perhaps twice per year. Individual tasks, such as the gathering of business requirements, should be happening on a continuous basis, as it is critical for infrastructure to remain in constant contact with customer needs and adjust services accordingly. Re-prioritized business requirements will lead to updated tactical and strategic directives for build requirements. On the other side of the equation, outsourcing alternatives will broaden and integration will become simpler. One thing is certain: technologies will improve and new products will emerge, offering more advanced E-services and service levels to clients. Whether outsourcing is chosen or not as a solution, it will remain the responsibility of infrastructure to take ownership in delivering quality E-services to their clients.

ABOUT THE AUTHOR

Carol A. Siegel is a leading information systems security expert and information risk manager, specializing in technologies that define the E-commerce space. She was the product manager for E-commerce infrastructure for Deutsche Bank AG, and is currently the chief security officer of American International Group, Inc. She can be e-mailed at siegelc@interactive.net.

Chapter 26
An Implementor's Guide to E-Commerce

John Care

The paradigm of Internet Time is now starting to stretch Internet projects after four years of compression. Internet systems are becoming increasingly complex as tools and languages gain in complexity. In 1996, an IT organization, or even a power user in sales and marketing, only needed to cope with setting up a Web server and a relatively small subset of HTML commands.

Since that time, a complete Internet architecture (see Exhibit 1) has evolved, and it is a rare project that now gets built in weeks instead of months. Compounding this effect is a labor shortage of developers capable of exploiting this infrastructure, where six months of experience commands sizable salary demands. Increasingly, "Net" projects need to look outward, capitalizing on tried and trusted user requirements documents and QA procedures, while simultaneously capturing the technological zest and frenzy of the Web.

Meanwhile, business-to-consumer Internet commerce will grow to $93 billion in 2002 from $13 billion in 1998. More impressively, business-to-business Internet commerce will reach $1.3 trillion in 2003 from a mere $43 billion in 1998.[1] In late 1999, the value of goods and services traded over the Internet was doubling every three to four months. *The Industry Standard* reports that from 1998 to 1999, the number of Web users increased by 55 percent worldwide, the number of hosts grew by 46 percent, the number of Web servers by 128 percent and the number of new Web addresses by 137 percent.[2] The Federal Reserve Board's chairman, Alan Greenspan, made this observation about the development of E-commerce:

0-8493-1161-6/02/$0.00+$1.50

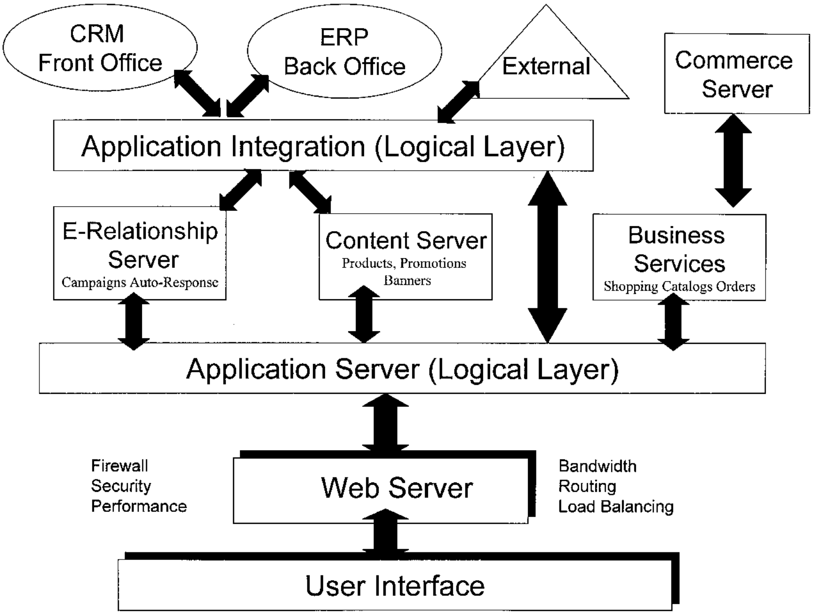


Exhibit 1. E-Commerce Architecture

> The newest innovations, which we label information technologies, have begun to alter the manner in which we do business and create value, often in ways not readily foreseeable even five years ago.

Against this backdrop, a corporation's senior officials have determined that an electronic commerce site is now critical to profitability and revenue growth. The information technology (IT) group is mandated to make it happen. Implementation of anything but the simplest site requires the input and work of multiple departments and project stakeholders and can now rival the complexity of an ERP installation. So where to start?

A BROAD DEFINITION OF E-BUSINESS

The Center for Electronic Commerce broadly categorizes E-business into five distinct types:[3]

1. *Self-services* provide 24 × 7 access to important business and personnel data. The most common examples in this category include online employee benefit enrollment, updates of personnel or 401(k) records, access to shipping status of customer orders, and some forms of online banking.
2. *Information access* provides search and retrieval capabilities for both public and proprietary domain data archives. Examples would

be credit agencies such as Equifax and financial sites such as Edgar Online and Hoovers.

3. *Shopping services* allow customers to seek and purchase goods and services though electronic networks. Although retail sales sites such as Lands End and The Gap are usually associated with this category, the classification can be extended to Business-to-Business (B2B) sites for purchasing used industrial equipment and commodities. Online auction sites such as eBay are variations within the category.
4. *Interpersonal communication services* provide a mechanism for individuals, groups, or corporations with mutual interests to exchange information and ideas in a cooperative environment. Examples include online interactive helpdesks, updated files sent to a printer by a publisher, joint supplier/customer groups building product specifications, and even e-mail communications between suppliers and purchasing agents.
5. *Virtual enterprises* are business arrangements such that trading partners are able to join in complex business activities, although separated by geography, as if they were a single enterprise. This includes true supply chain integration, transmitting forecast and planning data throughout a multi-tier chain. Another more common example is a grouping of allied suppliers allowing a common customer to do business with them ("one-stop shopping") via a single transaction.

The purpose of such broad classifications is actually one of focus. The fewer categories covered by an initial E-commerce project, the greater the likelihood of both business and technological success. This is in reality nothing more than the time-honored "contain the scope" mantra preached by project managers for generations.

More Than Taking Orders

Designing customer-facing applications instead of inward (internal) applications is still new to most organizations. Not only does the interface need to be functional, it also needs to be visually appealing. The E-customer, whether in Business-to-Business (B2B) or Business-to-Consumer (B2C) will only repeatedly visit a Web site for a limited number of reasons:

- The E-customer saves time
- The E-customer saves money
- Products and services are only available via the Web site
- Comparison shopping

A recent *USA Today* report showed that 60 percent of purchase transactions are abandoned prior to pressing the "Buy" button. Major reasons cited included incorrect pricing, special offers, customer service questions, and general uncertainty. This is hardly the standard fare of software bugs and slow performance that typically concern technologists.

WEB SITE DESIGN

Aside from some basic infrastructure decisions, Web site design represents the most contentious area between the technologists and the business people. The fundamental issue is who owns the site and the content. The usual King Solomon-style answer is to have the technology group be responsible for the day-to-day operation of the site and to keep it running, but to charge either sales, marketing, or in more forward-looking organizations, the online content group, with the appearance and structure of the site. The Web site represents the 'brand' of the corporation and as such needs to be designed by marketing — or with heavy input from it, as opposed to several HTML/Java programmers building the outward face of the company. Web developers know the difference between Java and JavaScript, and they like downloading plug-ins. Visitors to the site say, "So, how can I buy this?"

The four most common Web site operational requirements cited by users in numerous surveys are:

1. *Content.* The content of a site needs to be fresh and interesting. Matching this requirement is frequently the prime operational cost once a site goes live.
2. *Ease-of-use.* An intuitive interface that guides a user from screen to screen, with complete online help. Additionally, provide an option to remain on the Web page and contact customer support to resolve any questions.
3. *Fast downloads.* Filling a page with large .gif or .jpeg graphics, animations, or Java applets dramatically slows download times. The longer a page takes to load, the higher the chance a transaction will be abandoned. As a datapoint, Zona Research estimated that slow download times cost U.S.-based sites $4.6 billion in 1999.
4. *A search engine.* Many sites operate with search engines that are not up to the task. Either they cannot handle long queries like "Black Tommy Hilfiger Sweater" or they produce multiple pages of useless information based on a single keyword ("Tylenol").

Above all, it is key to be consistent. Once a corporate style has been decided upon, all pages within the system need to follow the style and the interface.

It is also worthwhile to note several things *not* to do.[4] *Do not*:

1. Use frames. Although a marvelous piece of programming technology, frames tend to be confusing and can radically cut down on the

amount of free space available on a page for display. In particular, double-frames (top and bottom or top and side) receive extremely poor ratings.

2. Require scrolling. If a user needs to scroll left and right, or up and down to get to vital information, the page immediately loses its impact. The keep-it-simple philosophy is an excellent guideline.
3. Forget to confirm. A user needs to know that a requested action has actually taken place. These range from something as simplistic as an order acknowledgment to more complex error trapping.
4. Ignore the 12-second rule. If an action has not occurred in 12 seconds, a user has a higher probability of canceling or refreshing a screen. (*Note:* Various Web pundits place different values on the wait factor, but the principle is that a measurable limit needs to be set.)
5. Design for T-1 speeds if dealing with B2C systems, international systems or remote dial-in. 56k access is a good guideline.
6. Provide an English-only system if dealing with international trade. Although used by 57 percent of the world's surfers, English is only spoken by 8 percent of the population.[5]

PRIVACY

On March 7, 1999, *The New York Times* revealed that the world's most powerful software company was collecting data about its customers with an apparent total disregard for privacy. This incident, although quickly corrected by Microsoft, thrust the concept of privacy squarely under the nose of the American public.

A privacy violation occurs when an organization uses information collected from its customers, or prospects, in ways the end user did not explicitly agree to when providing the data. This information can be anything from online behavior patterns to phone and fax numbers to demographic data. Although the buying and selling of consumer information has long been a staple of marketers worldwide, the speed of collection, collation, and dissemination via the Internet is an order of magnitude faster.

The establishment and publishing of a privacy policy is essential for any commercial Web site dealing in B2B or B2C transactions. Online privacy statement generators are available via the Direct Marketing Association.[6] Several third-party organizations such as TrustE and BBB Online (a unit of the Better Business Bureau) also provide audits and seals of approvals. For a high-volume, major site, the American Institute of Certified Public Accountants CPAWebTrust program supplies the most vigorous trustmark.

Exhibit 2. Basic Definitions of Web-Centric Marketing Terms

Term	Definition
Click-through rate	Percentage of viewers who click on a banner ad
Conversion rate	Percentage of visitors who purchase
Abandon rate	Percentage of visitors who abandon a transaction before purchase
Up-sell rate	Percentage of transactions where a higher cost item was sold measured against the original intended purchase
Eyeballs	A net-centric term indicating the number of unique views each item or site obtains
Impression	Number of views (per link or page) by product or feature category during entire visit
Cross-sell rate	Percentage of completed transactions that included an additional product outside of the original intended purchase

For an IT project manager, this is a task to pass off to the legal department in terms of the wording and certification. However, ensuring that no illicit data collection is actually performed behind the scenes and that no consumer's data is visible to anyone else is the responsibility of the project team. The most common avenues for "accidental" collection are typically remnants of debug code or stub CGI scripts left by the programming team.

The counterpoint of privacy is that the more a vendor can learn about a customer, the more value the vendor can provide by customizing for that customer. This is the basic tenet of one-to-one marketing. Eventually, as the relationship evolves, it becomes a matter of trust — more data can be voluntarily collected and put to good use.

ADVERTISING AND MARKETING

At first glance, this is hardly the realm of information technology, but through the extended project team it is a critical function of any E-commerce site. This reaches further than simple Web site design, which is covered elsewhere. First, the prime consideration is if advertising (banner) space will be featured on the site. If so, then both inbound and outbound traffic need to be monitored.

Web traffic analysis by slicing Web server log usage data is a common adjunct to any commercial site and many tools exist[8] to extract and examine such data. Standard analysis details the total hits, ranks popular pages, and shows the entry and exit points of the site by user, and where that user came from. The breakdown from the E-commerce viewpoint is that such tools do not provide linkage to the end result (i.e., a sale or order).

The marketing department will be interested in various Internet metrics (see Exhibit 2).

A newer form of analysis, microconversion, with more of a focus on ROI has been proposed by IBM researchers[7] and is gaining credence in the marketplace. While the conversion rate of an online store is the percentage of visitors who purchase from a store, it does not indicate the possible factors affecting sales performance. The notion of microconversion extends the traditional measures by considering the four general shopping steps in an online store, product impression, click-through, basket insertion, and purchase. Microconversion rates are computed for each adjacent pair of measures resulting in three additional functional rates.

1. Look-to-Click rate: product impressions converted to click-throughs
2. Click-to-Basket rate: click-throughs converted to basket placement
3. Basket-to-Buy rate: basket placements converted to purchases

The end result for IT is that measurement, both inbound to one's site and outbound (potentially to partners paying royalties), needs to be included in the initial project scope in order to generate any form of meaningful ROI for the site and its marketing programs.

SALES CHANNEL CONFLICT

Opening up the corporate sales process to one's customers will undoubtedly save the company money and streamline delivery for customers. However, depending on the sales model used, there may be considerable conflict between the different sales channels within an organization. (An example of channel conflict might be a full-service brokerage offering $20 trades to its clientele and not compensating its field brokerage force.)

Although not an issue for IT to directly resolve, requirements from the sales and partner organizations are likely to revolve around lead management, compensation, and quota recognition — all of which add to the complexity of the operation and will involve batching data to payroll or external systems.

INTEGRATING EXISTING AND NEW SYSTEMS

Barring the most recent "dot.com" companies, every corporation has a collection of legacy systems that contains useful information for the enterprise. In addition, there are invariably operational systems running on a current technology set that also require linkages to the E-commerce site. In numerous implementations witnessed by this author, the critical path to bringing a site live actually relies on the speed of integration as opposed to the base commerce technology itself.

Although not an exhaustive list, the types of systems shown in Exhibit 3 need to be examined for integration to the site.

Exhibit 3. Potential Integration Points

IVR/VRU	Interactive voice recognition for customer service
Inventory	Is the product or service actually in stock?
Pricing tables	Current pricelists (commercial, retail, government, etc.)
Shipping/Fulfillment	To provide status updates for orders
External shipping	Realtime linkage to UPS, Fedex, or chosen vendor
Financials	Credit check, E-cash
Billing and invoicing	To ensure payment is made for the transaction
Credit card	Credit card authorization
External	Trading partners supply chain systems
Marketing	Lead and campaign management, personalization
eCRM	Electronic customer relationship management

Taking Care of the E-Customer

A much quoted Jupiter research report[9] showed that over 40 percent of the top 120 Internet commerce stores either did not respond to customer e-mail or took longer than five days to do so. A number of these sites did not even provide an 800-number or an e-mail address for queries. There are also many individual stories, including several experienced by this author, about the lack of integration of sales, service, and Web site. Unless a corporation provides a truly unique and monopolistic service, the maxim "the competition is only a click away" is one to live by.

Based on a corporation's desire to have customer service as a differentiator, linkage to a CRM (customer relationship management) system becomes an added requirement. Despite the proliferation of Web self-service sites, at some point customers and prospects may need to speak to a live customer service person. Even Amazon.com, which bombards the consumer with status and update messages, still has live bodies at the end of the phone. The benefits of a CRM package are numerous, including the ability to:

- Attract, acquire, and then retain customers
- Determine who are the most profitable customers
- Allow anyone in the corporation access to appropriate customer data
- Provide base data for marketing campaigns
- Obtain eventual linkage to field sales force automation system

How Much Does It Cost?

Exhibit 4 sets out an approximate cost schedule to build an industrial-strength E-commerce site, based on estimates from various suppliers.[10]

These are undoubtedly daunting figures, with a wide degree of variance, but nevertheless bolster the assumption that building an E-commerce site

Exhibit 4. Approximate Cost Estimates for "Industrial Strength" E-Commerce Sites

Cost of Technology	Low ($)	High ($)
E-commerce applications	1,000,000	2,000,000
Integration to existing systems	50,000	500,000
eCRM software	500,000	1,000,000
Chat, discussion, bulletin boards	200,000	200,000
Operations	1,000,000	10,000,000
Site Hosting and network infrastructure	250,000	2,000,000
Total Technology Cost	**$ 3,000,000**	**$ 15,700,000**
Cost of Implementation	**Low ($)**	**High ($)**
E-commerce applications	2,000,000	8,000,000
Integration to existing systems	300,000	1,000,000
eCRM software	1,000,000	2,000,000
Chat, discussion, bulletin boards	50,000	50,000
Operations	1,000,000	10,000,000
Site hosting and network infrastructure	250,000	2,000,000
Total Implementation Cost	**$ 4,600,000**	**$ 23,050,000**
Overall Project Cost	**$ 7,600,000**	**$ 38,750,000**

is neither a trivial nor cheap project for any reasonably sized corporation. Naturally, a significant part of the implementation costs can be absorbed internally if both the manpower and budget dictate.

SUMMARY

Internet application development and deployment has gained the reputation of being fast and not tied to prior systems and methodologies. Unfortunately, as the tools and requirements imposed on such E-systems have become more complex, the project scope, reach, and budget have risen dramatically. The needs of an E-commerce site cross multiple departmental boundaries — from sales and marketing to legal to shipping to product management and finance — and all of these groups need to be stakeholders in the E-commerce project and be part of the E-team constructing the site.

A simple four-step process, remarkably similar to methodologies that have existed since the COBOL era, forms the basis for development.

1. Perform a needs analysis to determine how to leverage existing legacy and operational systems. Obtain input from all departments that will integrate their business to the site. Nominate at least one member from each department to the E-team.

2. Choose an open-standards-based platform solution. Search for the optimum combination of the scalability and reliability for the company's requirements.
3. Connect the E-commerce application to the corporate ERP (enterprise resource planning), CRM (customer relationship management), and legacy systems.
4. Test and test again.

Notes

1. Forrester Research Papers, 1999, www.forrester.com.
2. My How We've Grown, Maryann Jones Thompson, *The Industry Standard,* April 26th, 1999, www.thestandard.com.
3. The Environmental Research Institute of Michigan, www.erim.com.
4. A detailed study of aesthetic Web design is located at www.creativegood.com.
5. Reference www.emarketer.com.
6. Found at www.the-dma.org.
7. E-Commerce Intelligence: Measuring, Analyzing, and Reporting on Merchandising Effectiveness of Online Stores, Gomory, Hoch et al., IBM Research Center, Yorktown Heights, NY.
8. Doubleclick www.doubleclick.com, Media Matrix www.mediamatrix.com, Marketwave www.marketwave.com.
9. Jupiter Research (www.jup.com).
10. Estimates provided by USWeb/CKS (www.usWeb.com), Interactive Week (www.zdnet.com/intweek), Clarify Inc. (www.clarify.com), and Eloyalty (www.eloyalty.com).

ABOUT THE AUTHOR

John Care is Director of Technical Services for a large customer relationship management software provider. He may be contacted at jcare@email.msn.com.

Section V
E-Enabled Architecture and Design

This section examines methods for architecting and designing Internet-based business solutions. Some of the chapters contained in this section are fairly technical in focusing on the different tiers of the overall solution, including server components, middleware, and the user interface.

"Usability Design for Web-Based Applications" (Chapter 27) provides a set of recommendations for attaining usable Web-based application interfaces aiming to help users successfully apply Web-based software to accomplish their work and save organizations time and money in lost productivity, training, and technology rejection or abandonment.

"Component Architectures: COM and CORBA" (Chapter 28) defines components, classifies them, and reviews some of the popular standards in the industry. Elements of the design cycle are also discussed in this chapter.

"Distributed Computing: Objects and Components in the Enterprise" (Chapter 29) compares and contrasts four different models of distributed computing, namely: CORBA, RMI, COM, and Web-based application servers.

"Developing Internet Solutions with SOAP" (Chapter 30) examines the Simple Object Access Protocol, an Internet specification that provides a method to invoke programming and to pass data (both parameters and results) from clients to servers and to get a response from those servers.

"Elements of a Well-Designed Corporate Web Site" (Chapter 31) provides insight into how a Web site should be designed, developed, and deployed. This chapter can be used as an audit reference in evaluating a Web site.

"Developing a Trusted Infrastructure for Electronic Commerce Services" (Chapter 32) offers design methods for confirming sender and recipient identities, protecting confidentiality, and date and time stamping to develop a trusted network infrastructure for electronic commerce.

"Evaluating and Selecting E-Commerce Software Solutions" (Chapter 33) discusses several different vendor solutions that support E-commerce. These are the ones that have survived the Internet crunch that the rest of the industry has endured.

Chapter 27
Usability Design for Web-Based Applications

Chris Forsythe

Until recently, most Web-based development was conducted in an atmosphere largely forgiving and tolerant of the shortcomings associated with World Wide Web technologies and their interfaces. Most organizations had only one or two knowledgeable Web developers who were rarely challenged to meet standards comparable to those for traditional software applications. Web sites providing unique information or services encountered little or no competition from similar sites. Although there was no shortage of enthusiasts, expectations were generally low and interest was sustained merely by the Web's novelty and potential. Similarly, the relatively small number of total potential users meant that any gains or losses resulting from a high-quality or poorly designed Web application were mostly small, if not nonexistent.

As Web technologies have matured, much has changed. Today, the business operations of many corporations hinge on the effectiveness and efficiency of their intranets, and the development of Web marketing strategies is a must. In more and more cases, Web technology is being used to provide resources that greatly enhance the capacity to conduct work. Furthermore, the availability of immediate cross-platform solutions and the relative ease of applications development make Web-based applications the most practical software solution for many organizations.

The evolution of Web technologies from a plaything to a legitimate business tool has generated expectations that Web applications will perform equally to, if not better than, common commercial software. Over the past ten years, the commercial software business has been driven by usability and the need to provide users with an intuitive, error-tolerant, efficient, and productive user interface. This should not have come as any surprise,

0-8493-1161-6/02/$0.00+$1.50

because it is fairly obvious that software that allows more work to be done, better and faster, is far preferable to less usable software. The same relationship between usability and business success also exists regarding Web user interfaces.

USABILITY BASICS

In the late 1980s, an interesting phenomenon began to occur in the commercial software business as marketing people and materials started to tout software as user friendly, based on implementation of a graphical user interface (GUI). Because a GUI was thought to be the singular ingredient necessary for attaining user friendliness, often little else was done for the sake of usability.

This phenomenon has left the term "user friendly" so overused and misused that it no longer has much meaning. Just as GUIs were once touted as the one element needed to achieve usability, today a Web front end is often cited as the basis for asserting the usability of Web-based products. As was the case for GUIs in the 1980s, these claims typically go unchallenged, and organizations purchase too many products without requesting proof of their usability.

How is usability attained? There are many approaches, and the one chosen depends on the importance of the software. If software is to be frequently used by large numbers of people or for critical activities or both, the approach followed should provide a high level of assurance that usability is attained. Such an approach may include job analysis and task analysis, followed by rapid prototyping interspersed with frequent usability testing. In contrast, if there will be limited numbers of users and the software and its functions are noncritical, it is often sufficient for the developer simply to adhere to design guidelines and obtain informal peer reviews. The importance of software usually varies between these two extremes, so the level of effort devoted to ensuring usability is varied as well.

COST JUSTIFICATION

Common Misconceptions

The most common misconception about usability among a broad range of software developers and engineers from various disciplines is that usability is largely common sense. Were common sense and good intentions all that is necessary to consistently produce usable software products, the term computer anxiety would never have become part of today's common vernacular, companies would not have to commit substantial sums to training, help desks, and other user support services, and there would not be such a high incidence of companies investing capital to computerize their business processes only to achieve marginal improvements in productivity.

The misconception that usability is only common sense not only promotes the delusion that usability can be attained at no expense, but also perpetuates the misapplication of talented software developers to the job of user interface design. This task distracts these individuals from the challenges of code development and without the necessary know-how is only rarely a source of great rewards.

Usability comes at a cost. Most corporate software development projects involve significant technical challenges that consume the bulk of both financial and human resources. To commit to usable design requires that resources be diverted from the core code development to various activities associated with the definition, prototyping, testing, and refinement of the user interface. Furthermore, a usable interface often imposes requirements on the supporting software that are inconvenient and met only by overcoming certain difficulty. The net result is that software, Web based or otherwise, with a highly intuitive user interface is likely to cost more to develop than if usability were neglected.

The trade-off that results from weighing the importance of the software against the level of effort devoted to development of the user interface increases in importance as corporations automate their business practices and pursue information-driven processes. Usability costs must be paid whether they are incurred through up-front development costs, training costs, lost productivity, or technology rejection and abandonment. The most cost-effective approach for applications that will be used frequently by a large number of users and involve operations for which errors may have costly and otherwise undesirable consequences is almost always to shift costs to development and make the investments necessary to ensure usability.

Case Study

The case of a hypothetical Web-based time-card application demonstrates this costing logic. The application is to be used once per week (a conservative estimate given that most employers encourage staff to make time-card entries daily, if not more frequently) by 8000 employees. Experience with many such Web-based applications makes it reasonable to assert that thorough involvement of usability specialists in the development of the time card will reduce the average transaction time from four to three minutes.

This reduction results from improving the efficiency of transactions and reducing the likelihood of errors. Such improvements may be gained in many different ways. For example, integration of two pages into one avoids lost time as users wait for pages to download across busy networks. Similarly, providing immediate access to supporting information such as time-charging codes could yield equivalent savings by avoiding the need for users to go to another location and search for necessary information. Finally, presenting text and data in a readable format can easily prevent errors in data entry, reducing the likelihood of an erroneous time card from five to less than one percent.

Presuming an average cost for employees of $50 per hour, the following savings are calculated:

$$\begin{aligned}
\text{1 min/employee} \times \text{8,000 employees} &= \text{8,000 min} \\
\frac{\text{\$50/hour}}{\text{60 min/hour}} &= \text{\$0.83/min} \\
\text{\$0.83/min} \times \text{8,000 min} &= \text{\$6,640/week} \\
\text{\$6,640/week} \times \text{52 weeks/year} &= \text{\$345,280/year} \\
\text{\$345,280/year} \times \text{5-year life span} &= \text{\$1,726,400 savings}
\end{aligned}$$

These savings only address the cumulative sum of lost productivity. These same improvements to the user interface would also serve to reduce the time required for training, regardless of whether training occurs formally or informally. Although it should be possible to develop an interface that is sufficiently intuitive to obviate the need for training, for the present purposes a reduction in average training time from 30 to 15 min is assumed. Such a reduction leads to the following additional savings:

$$\begin{aligned}
\text{15 min/employee} \times \text{8,000 employees} &= \text{120,000 min} \\
\text{\$0.83/min} \times \text{120,000 min} &= \text{\$99, 600 savings}
\end{aligned}$$

Still, the savings do not stop here. Typically, this type of application necessitates some type of user support, whether formally through a help desk or informally through co-workers answering each other's questions. Once again, it is reasonable to assert that improvements to usability would result in a reduction in the number of employees who seek assistance on at least one occasion from fifteen to five percent. Without even considering that some employees may not only seek assistance, but seek assistance on multiple occasions, further savings are calculated as follows:

$$\begin{aligned}
\text{Ten percent fewer employees seeking assistance} &= \text{800 employees} \\
\text{Average time spent on assistance} &= \text{15 min} \\
\text{15 min} \times \text{2 employees} &= \text{30 min} \\
\text{30 min} \times \text{800 employees} &= \text{24,000 min} \\
\text{\$0.83} \times \text{24,000 min} &= \text{\$19,920 savings}
\end{aligned}$$

In this example, the savings realized from a ten percent reduction in the number of first-time requests for assistance should cover a good portion, if not all, of the costs incurred from including a usability specialist on the interface design team and conducting usability testing. Granted, accommodating usability within the software design may also increase software development costs because the most usable design is often not the easiest to develop. These costs, however, should also be dwarfed by the sum of the potential savings.

The decision to design for usability is always a matter of cost transfer. Designing for usability transfers cost to development. When usability is neglected, either intentionally or inadvertently, a decision is made to transfer costs to training and to the overhead charged to nearly every account to which employees allocate their time. The example of the Web-based time-card application illustrates that by paying a little more up front to ensure usability and transferring costs to development, substantial savings are realized over the life span of the software. By paying a little extra for usability, a buyer purchases quality and the long-range savings that accrue from it.

THE WEB DESIGN CHALLENGE

A plethora of sources offers guidance on interface design for Web-based applications. These sources include Web sites offering style guides or other design recommendations, books addressing Web page design, and Web developers willing to share their opinions. Given the newness of the Web and the relatively rapid evolution in the capabilities afforded by HTML and Web browsers, the amount of so-called expertise being offered on good and bad design practices is astounding.

A recent study comparing Web design guidelines to traditional human-computer interface (HCI) guidelines suggests the need for caution in adopting Web guidelines regardless of their source. The study of 21 Web style guides found that the time devoted to Web design guidelines was less than 1/25 that devoted to traditional HCI guidelines. Furthermore, 75 percent of the more than 350 distinct design recommendations found appeared in only one style guide. This suggests that most of the advice being offered is merely personal opinion. The fact that only 20 percent of the 270 Web-relevant recommendations found in traditional HCI style guides appeared in any of the Web style guides suggests that existing, readily available knowledge concerning user interface design is largely ignored.

The study also asked a group of human factors practitioners to rate the importance of each of the 270 Web-relevant recommendations found in the traditional design guides to the usability of an interface. Of the 20 recommendations rated most essential to usability, only one was found in any of the Web design guides. Thus, Web design guides do not only fail to address much of what has traditionally been accepted as effective user interface design practice, they also fail to consider those facets of design most essential to usability.

It is generally recommended that developers of Web-based applications approach various sources of design guidance with due skepticism and follow instead the rich body of knowledge relating to user interface design, including guidelines, evaluation, and testing. This well-researched, widely accepted, and generally reliable source of Web-relevant guidance contains the collective knowledge of hundreds of practitioners derived from countless hours of laboratory and field research concerning facets of user interface design that either contribute or distract from usability.

ATTAINING USABLE WEB-BASED APPLICATION INTERFACES

The surest formula for attaining usable Web-based application interfaces is to follow a process that incorporates the identification and resolution of usability concerns into every phase of the interface design. This process involves up-front analysis to gain an understanding of the user and the job.

There is no more frequently cited heuristic within human factors than "Know thy user and you are not thy user." In essence, this heuristic instructs interface designers to make no assumptions regarding what the user needs, prefers, understands, and can use. Most important, just because software may seem easy and make sense to the development team does not mean that the user will be able to understand and use it.

As the interface is developed, designers should follow established guidelines regarding usability design. Last, nothing should be taken for granted. Thorough usability testing should be conducted with representative users applying the software, or reasonable prototypes, to perform tasks representative of those for which the software is intended.

Although usability is largely achieved through the process followed in developing the user interface, a great deal of knowledge regarding the facets of design contributes or distracts from an interface's usability. The following sections present some of the HCI guidelines judged most essential to Web usability by the group of human factor experts in the previously discussed study. The guidelines illustrate the effects that specific user interface design decisions may have on the ability of users to successfully apply software to accomplish their work.

Direct Usability of Information

Information should be presented in a directly usable format that does not require decoding, interpretation, or calculation. This means that users should not be given an answer that requires them to seek other references for its interpretation. Although incorporating various reference documents into an application may greatly expand the scope of the application, such additional labor is typically paid for many times over through the resulting increased productivity. Furthermore, not incorporating these references assumes users both possess and know how to use them.

Similarly, users should not be required to use such items as calculators for operations than an application can be programmed to perform. This is because of the probability of human error resulting from reading values from a screen and entering those values by a keypad or keyboard.

Ease of Navigation through Data Displays

When displayed data exceeds a single display, users must have an easy mechanism for moving back and forth through material. The most common violation of this design principle regarding Web-based applications occurs when reports are written from a database directly into an HTML file and the resulting table extends beyond the right border of the browser window. The likelihood of error is introduced because users must awkwardly scroll back and forth and may lose track of the row from which they are reading. In these cases, the costs of thousands of users scrolling back and forth and losing their places within data reports must be weighed against the additional effort required to reformat the reports for display with minimal or no scrolling.

Concise Instruction Levels

Users should not be expected to read more than three help displays or remember more than five points. This statement from traditional HCI sources may be too lenient for Web-based applications because, strangely, users exhibit a much lower tolerance for written instructions presented through Web interfaces than through traditional computer text displays. Thus, for Web-based applications, the requirement for concise, to-the-point help is considerably more stringent and requires greater attention to the content of help systems than might be expected.

Consistent Use of Color

Color should be used consistently within a display or across a set of displays because misinterpretations result from the fact that users both intentionally and unintentionally attribute meaning to colors. Unfortunately, developers often choose colors for aesthetic reasons and fail to consider the unintended meanings users assign to them. Throughout the design of an interface, therefore, the developer must assume that users will assign meaning to colors and exercise care in their selection and use.

Distinct Naming

Similar names for different functions should be avoided. This is because in choosing items from a menu, pull-down list, or table, especially with familiar interfaces, users often discriminate solely on the basis of the general outline of the selection without actually reading the words. For this reason, the labels placed on buttons, in menus, or as list items should be distinct, both semantically and with regard to the visual appearance of the words themselves.

Indication of Function Actuation

A positive indication of function actuation should be provided. This guideline has particular ramifications for Web-based applications. In many respects, Web browsers provide the subtlest of cues that user actions are having an effect. The user cannot be assumed to recognize that a selection has been made on the basis of flying meteorites or spinning electrons that appear against a small graphic within the browser window.

For these reasons, other cues that a user action is having an effect should be provided. For example, most Web users are familiar with the concept of pages wherein text loads before images. Similarly, for back-end processes requiring that the user wait, it is often possible to provide a message informing the user that a request has been received and is being processed. Likewise, for long, perhaps graphically intense downloads, the user may be provided with a confirmation message that the requested download is about to commence.

Confirmation of Destructive Actions

Users should be required to confirm potentially destructive actions before the action is executed. Although such confirmation requests are a standard feature in most non-Web user interfaces, this design practice has often been neglected with software developed for the Web. This situation is exacerbated by the ease with which Web developers may insert a Clear Form-button that all too frequently is placed immediately adjacent to buttons used for frequent operations such as Submit.

The common assertion that confirmation messages are a nuisance is usually made by experienced, frequent computer users and rarely by novice or infrequent users. Unless designing for a highly competent user population that has expressed its disfavor with confirmation messages, the developer should always err conservatively and include them.

Efficient Data Input

Data input actions should minimize user actions and memory load on the user. In short, any transformations, formatting, or similar modifications of user input should be done by the computer. It is unproductive for users to expend their resources performing various data manipulations to prepare data to be inputted into the application when these manipulations could be done by the machine.

Logical Data Entry Formats

Data entry formats should match source document formats. When a Web interface is used in transferring data from paper forms to electronic data files, the interface should mimic the paper form in sequence and layout as closely as possible.

Automated Data Entry

Data should be automatically entered in a field when it is known or can be computed. Once again, overall productivity is served when writing a few extra lines of code to perform an operation or filling in data accessible from one or more databases results in the transfer of work from the user to the machine. Similarly, if the user has entered data once, it should be filled in later and not require reentry by the user.

Simplified Data Entry Rules

Complex rules for entering data should be avoided. In general, flexibility should be the rule. For a social security number, for example, the user should be allowed, but not required, to use dashes. For a phone number, users should be allowed to enter the area code, but if they do not it should be determined by the machine based on the three-number prefix.

For every data input, therefore, consideration should be given to the various formats by which a user might naturally enter the data. To the extent allowed, all such formats should be accommodated by stripping away excess punctuation and referring to translation tables or other similar mechanisms.

Required and Optional Field Differentiation

Cues should be given to distinguish required from optional fields. Many applications request information that is useful, but not necessary. Because users often assume every space must be filled, they devote inordinate amounts of time to searching for information, the benefit of which is far outweighed by the cost of its retrieval. Thus, unless it is truly necessary, users should have the option of skipping fields for which they do not readily know or cannot readily obtain the desired information.

Feedback on Data Input

The user should be provided positive feedback regarding the acceptance or rejection of data input. Many applications allow users to submit requests that will not be filled for several minutes or hours. In such cases, the user should be provided feedback regarding the acceptance or rejection of the request on submission and not when the results are returned. In particular, this calls for routines that validate the user request prior to its submission for processing.

Error Correction

An easy mechanism should be provided for correcting erroneous entries. This mechanism should make it easy for the user to identify and correct any errors made. Should a request be rejected, the user should not be required to reformulate the request from the beginning, but be allowed instead to correct only the erroneous entry and resubmit the request. Similarly, the application should provide sufficient details regarding the location and nature of errors so that correction is intuitive and easily accomplished.

RECOMMENDED COURSE OF ACTION

No one intentionally designs interfaces to be nonusable. At the same time, usability does not come without some expenditure. Consequently, in environments in which development costs and schedules drive design decisions, it is far too easy to neglect usability or assign it low priority. Too often user interfaces are slapped together after all other functional elements are essentially completed. Similarly, design issues that are critical to the usability of the product are all too often relegated to the status of "nice to have."

It is interesting that a software bug that causes a one percent failure rate will receive endless hours of attention and result in the software ultimately being rejected. Yet an interface feature that causes users to fail to successfully complete ten percent of their transactions with the software is considered trivial and not worthy of the precious time and resources required to correct the problem. Similarly, days of analysis, research, and testing are devoted to issues related to functional elements of the software code, but decisions regarding interface features that could be critical to the users' success or failure are made off-the-cuff, with little or no discussion and rarely any analysis or testing.

There is no shortage of excuses for failing to give usability its due consideration. In the end, however, the costs of this failure are paid through incremental drains on the overall productivity of the enterprise. Just because these costs are largely hidden does not mean they do not warrant correction. By following the guidelines presented in this chapter, organizations take the first basic steps toward attaining productive and usable Web-based application interfaces that ultimately aid, not hinder, business success.

ABOUT THE AUTHOR

Chris Forsythe is senior member of the technical staff in the Statistics and Human Factors Department at Sandia National Laboratories in Albuquerque, NM.

Chapter 28
Component Architectures: COM and CORBA

T.M. Rajkumar
David K. Holthaus

Database technologies have evolved from the hierarchical databases in the 1970s to relational database management systems in the 1980s and object databases and client/server systems in the 1990s. Although the shift from central processing to client/server did not fully leverage object technology, Internet-based technologies promise to provide the infrastructure for objects. Web-based browsers are poised to become the universal clients for all types of applications. These applications increasingly depend on components, automation, and object layers linking systems.

During the same period, it became less and less possible for software developers to quickly, efficiently, and inexpensively develop all of the functions and modules demanded by customers. Therefore, software development methodologies for Internet and Web applications are increasingly focused on component technologies. Component technology breaks the application into intrinsic components and then glues them to create an application. Using components, an application is easier to build, robust, and delivered quickly.

WHAT IS A COMPONENT?

A component is an independently delivered package of software services. A component is language independent and allows reuse in different language settings. Software components can be either bought from outside or developed in-house. Implementation requires that it must be possible to integrate them with other applications using standardized interfaces.

0-8493-1161-6/02/$0.00+$1.50

They must efficiently implement the functionality specified in the interface. Components may be upgraded with new interfaces.

A component encapsulates methods (i.e., behavior) and data (i.e., attributes). Components must provide encapsulation, but inheritance is not as rigid a requirement. Components may include other components. Components do not necessarily have to be object oriented, though a large majority of them are because it provides mechanisms to hide the data structure (i.e., encapsulation). Using objects makes components easier to understand and easier to create.

Components may be classified in many different ways. One such classification is based on their function within applications: business or technical components.

Business components usually include the logic that supports a business function or area. These must be developed in-house because it forms part of the core knowledge of an organization. In addition, the business knowledge required to create them generally does not exist outside. They are also difficult to develop because organizations must standardize in some manner. There must be a common vision for the organization, and a common architecture must be present to develop business components.

Technical components are represented by elements that are generic and can be used in a wide variety of business areas. These typically come in the form of GUI components, charting, or interapplication communication components.

A second classification is based on granularity of components. Fine-grained components such as class libraries and encapsulated components are typically small in size and are applicable in a wide range of applications. Although they have large reuse across multiple applications, they are close to code and provide limited productivity to a developer in large-scale applications.

Large-grained components provide broader functionality, but they must be customized for use. A framework is an example of a large-grained component. Frameworks provide two benefits: flow of control and object orientation. A framework is basically groupings of components packages or components that belong to a logically related set and together provide a service. They provide a substrate or lattice for other functional components, and a framework can be composed of other frameworks. They also provide the flow of control within components. This helps in the scale of the solution developed.

Object orientation of frameworks helps with the granularity of the components. Ideally, during the assembly stage, a small number of large components is optimal. However, to increase the generality of the solution

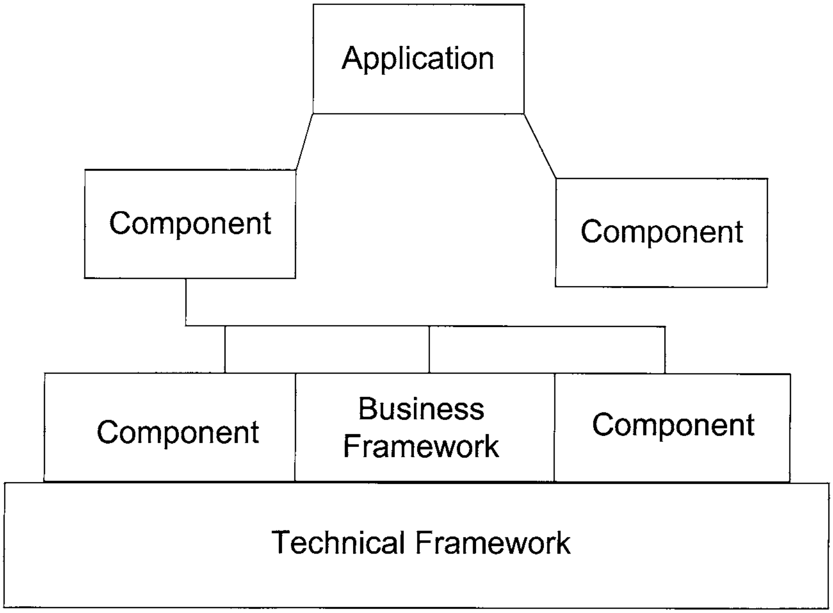


Exhibit 1. Application Integration with Components

created, a large number of small components is the ideal. Large components must be customized prior to delivering needed functionality. Frameworks allow developers to modify and reuse components at various levels of granularity. Frameworks are examples of “white-box” components (i.e., one can look inside the components to reuse them). With inheritance, the internals of parent classes are visible to subclasses in a framework. This provides a developer with flexibility to modify the behavior of a component. Thus, frameworks enable customization, allowing developers to build systems quickly using specialized routines.

Frameworks come in two categories: technical and business. Technical frameworks encapsulate software infrastructure such as the operating system, graphical user interface (GUI), object request broker (ORB), and transaction processing (TP) monitor. Microsoft Foundation Class (MFC) is an example of such a framework. Business frameworks contain the knowledge of the objects in a business model and the relationships between objects. Typically, they are used to build many different components or applications for a single industry. Technically, while not based on components, Enterprise Resource Planning (ERP) and software such as SAP are examples of business frameworks. An application is generally built with both technical and business frameworks (see Exhibit 1).

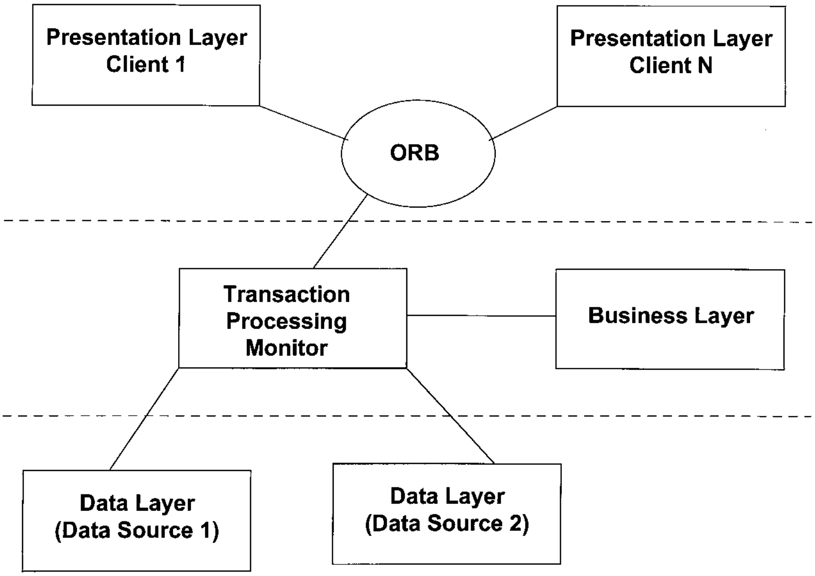


Exhibit 2. Three-Layer Client/Server Architecture

CLIENT/SERVER COMPONENTS

Client/server systems typically use three tiers: presentation layer, business layer, and data or server layer (see Exhibit 2). The objective behind the three tiers is to separate the business layer from the presentation and data layers. Changes in one layer are isolated within that layer and do not affect others. Within component technologies, the business layer communicates to the presentation layer and the data layer through an object bus, which is typically an object request broker (ORB). This layering makes the system very scalable.

An ORB is a standard mechanism through which distributed software objects and their clients may interact. Using an ORB, an object and its clients can reside on the same process or in a different process, which they can execute on different hosts connected by a network. The ORB provides the software necessary to convey the requests from clients to objects and responses from object to client. Since the ORB mechanism hides the details of specific locations, hosts, and conversion of data representation, and hides the underlying communication mechanism, objects and clients can interact freely without having to worry about many details. Thus, distributed applications can incorporate components written in different languages and are executable on different host and operating system platforms. This flexibility allows the data layer to be composed of legacy software and relational and object databases.

Business logic may reside on multiple server computers and data may reside on multiple servers. A TP monitor must be used to manage the business logic and to provide centralized control. A TP monitor also manages the logic on the servers by providing an array of mission-critical services such as concurrency, transactions and security, load balancing, transactional queues, and nested transactions. A TP monitor can prestart components, manage their persistent state, and coordinate their interactions across networks. TP monitors thus become the tool to manage smart components in a client/server system with components.

The real benefit of components in client/server applications is the ability to use the divide-and-conquer approach, which enables clients to scale through distribution. In this approach, an application is built as a series of ORBs. Since an ORB is accessible by any application running on a network, logic is centrally located. Developers can change the ORB to change the functionality of the application. If an ORB runs remotely, it can truly reflect a thin client. ORBs are portable and can be moved from platform to platform without adverse side effects to interoperability and provide for load balancing.

COMPONENT STANDARDS

Object models such as ActiveX, which is based on COM, CORBA, and Java Beans, define binary standards so that each individual component can be assembled independently. All component standards share the following common characteristics:

- A component interface publishing and directory system
- Methods or actions invocable at runtime by a program
- Events or notifications to a program in response to a change of state in an object
- Support for object persistence (to store such information as the state of a component)
- Support for linking components into an application

The following paragraphs describe each standard.

ActiveX, COM, and DCOM

ActiveX is based on COM technology, which formally separates interfaces and implementation. COM clients and objects speak through predefined interfaces. COM interfaces define a contract between a COM and its client. It defines the behavior or capabilities of a software component as a set of methods or properties. Each COM object may contain several different interfaces but must support at least one unknown. COM classes contain the bodies of code that implement interfaces. Each interface and COM class has a unique ID, IID, and CLSID that are used by a client to instantiate an object in a COM server. There are two types of object invocations:

- In-process memory (DLLs), where a client and object share the same process space
- Out-of-process model, where a client and object live in different processes

Clients can easily call either. A remoting layer makes the actual call invisible to a client. An ActiveX component is typically an in-process server. An actual object is downloaded to a client's machine. DCOM is COM extended for supporting objects across a network. DCOM allows objects to be freely distributed over several machines and allows a client to instantiate objects on remote machines.

CORBA

The Common Object Request Broker Architecture (CORBA) is a set of distributed system standards promoted by the Object Management Group, an industry standards organization. CORBA defines the ORB, a standard mechanism through which distributed software and their clients may interact. It specifies an extensive set of bus-related services for creating and deleting objects, accessing them by name, storing them in a persistent store, externalizing their states, and defining ad hoc relationships between them.

As illustrated in Exhibit 3, the four main elements of CORBA are:

1. *ORBs.* This defines the object bus.
2. *Services.* These define the system-level object frameworks that extend the bus. Services include security, transaction management, and data exchange.
3. *Facilities.* These define horizontal and vertical application frameworks that are used directly by business objects.
4. *Application objects.* Also known as business objects or applications, these objects are created by software developers to solve business problems.

A Comparison of CORBA and DCOM

Both CORBA and DCOM use an interface mechanism to expose object functionalities. Interfaces contain methods and attributes as a common means of placing requests to an object. CORBA uses standard models of inheritance from object-oriented languages. DCOM/ActiveX uses the concept of multiple interfaces supported by a single object. DCOM requires that multiple inheritance be emulated through aggregation and containment of interfaces.

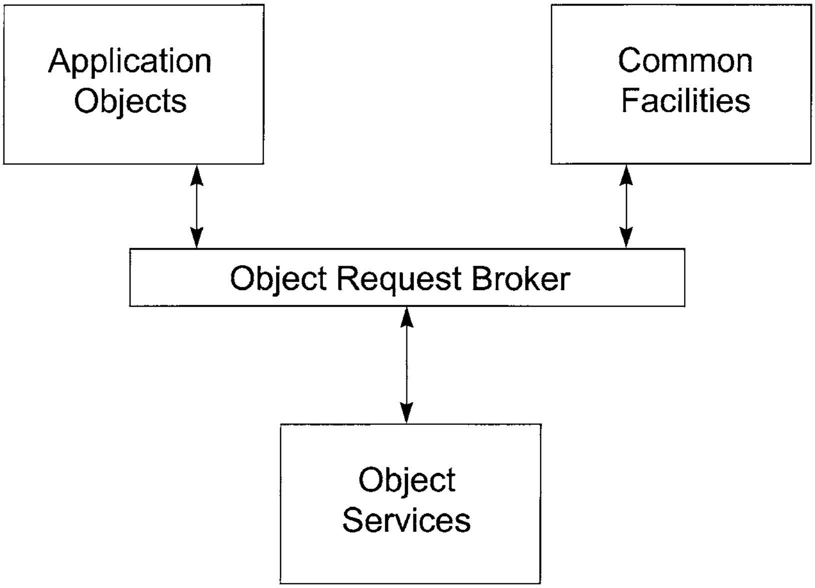


Exhibit 3. CORBA Architecture

Another difference is the notion of object identity. CORBA defines the identity of an object in an object reference, which is unique and persistent. If the object is not in memory, it can be reconstructed based on the reference. DCOM, in contrast, defines the identity in the interface; the reference to the object itself is transient. This can lead to problems when reconnecting because the previously used object cannot be directly accessed.

Reference counting is also different in both. A DCOM object maintains a reference count of all connected clients. It uses pinging of clients to ensure that all clients are alive. CORBA does not need to do remote reference because its object reference model allows the recreation of an object if it had been prematurely deleted.

CORBA uses two application program interfaces (APIs) and one protocol for object requests. It provides the generated stubs for both static and dynamic invocation. In addition, a dynamic skeleton interface allows changes during runtime.

DCOM provides two APIs and two protocols. The standard interface is based on a binary interface that uses method pointer tables called *vtables.* The second API, object linking and embedding (OLE) automation, is used to support dynamic requests through scripting languages. OLE automation uses the IDispatch method to call the server.

CORBA is typically viewed as the middleware of choice for encapsulating legacy systems with new object-oriented interfaces, since it provides support for languages such as COBOL and mainframe systems. DCOM has its roots in desktop computing and is well supported there.

Java Beans

Java Beans enables the creation of portable Java objects that can interoperate with non-Java object systems. Unlike ActiveX, which predominately operates in Windows environments, Java Beans is intended to run in diverse environments as long as there exists a Java Virtual Machine that supports the Java Bean API. Java Beans provides the standard mechanisms present in all the component technologies. This standard is still continuing to evolve.

Comparison of Java and ActiveX

Java Beans has all the capabilities of a Java application. However, if one runs a version of Java Beans that has not been signed by a digital source, its capabilities are limited like any other applet. Java also has limited multimedia support. In contrast, ActiveX objects cannot run from the Web unless they are trusted and have access to all of the capabilities of Windows. Hence, ActiveX supports multimedia.

ActiveX and Java both use digitally signed certificates to protect against malicious attacks. In addition, Java Beans is available for a large number of machines and has cross-platform capability. ActiveX is most widely available on the Windows desktop.

Irrespective of the technology standard, bridges, available from different vendors, can translate between standards. Hence, organizations should choose a standard in which they have the greatest expertise for analysis, design, and development.

HOW TO DESIGN AND USE COMPONENTS

As shown in Exhibit 1, applications are built from the composition and aggregation of other simpler components, which may build on frameworks. Application design is broken into component and application development. Component development is divided into component design and implementation. A strong knowledge of an application's domain is necessary to develop frameworks and components. In general, the steps of domain definition, specification, design, verification, implementation, and validation must be done prior to application. The following sections explain these steps.

Domain Definition

Domain definition defines the scope, extent, feasibility, and cost justification for a domain. An organization must define the product it plans to build as well as the different business and technical areas that must be satisfied through the use of software.

Domain Specification

Domain specification defines the product family (i.e., framework) used for application engineering. It includes a decision model, framework requirements, and a hierarchy of component requirements. The decision model specifies how components will be selected, adapted, and reused to create complete application systems. Product requirements are assessed by analyzing similarities in functions, capabilities, and characteristics, as well as variances among them. The component part of the product family is represented hierarchically. When an organization considers components, it must consider not only what the component can do for the domain now, but also what it will do in the future.

Domain Design

A domain expert must work with the component designer to use a modeling methodology and extract the design patterns that occur in that domain. Design patterns are repeatable designs used in the construction of an application. The architecture, component design, and generation design are specified here. Architecture depicts a set of relationships among the components such as hierarchical, communication, and database. Component design describes the internal logic flow, data flows, and dependencies. Generation design is a procedure that describes how to select, adapt, and compose application systems using the decision model and architecture.

Domain Verification

Domain verification is a process that evaluates the consistency of a domain's requirements, specification, and design.

Domain Implementation

During this procedure, components are either developed or acquired off the shelf to fit the architecture. Each component must be tested within the common architecture it supports as well as any potential architecture. Certification of components must be acquired when necessary. Determinations as to how to store it in repositories, the implementation of application generation procedures, and how to transition to an assembly mode must also be made.

Domain Validation

Domain validation evaluates the quality and effectiveness of the application engineering support. Application engineering consists of the following:

- *Defining requirements.* In this process, an application model that defines a customer's requirements is defined. This model uses the notation specified in the domain engineering steps. Typically, a use-case model can be used to identify requirements. Use cases are behaviorally related sequences of transactions that a user of a system will perform in a dialog.
- *Selecting components and design.* Using rules in the decision model, reusable components are selected based on the component specification (capabilities and interfaces) and design (component logic and parameters).
- *Generating software.* The application is then generated by aggregating components and writing any custom software.
- *Testing.* Testing involves the testing of use cases, components, integration testing, and load testing.
- *Generating documentation.* The application documentation is created.

MANAGING THE COMPONENT LIFE CYCLE PROCESS

Developing with components means an organization must move from doing one-of-a-kind development to a reuse-driven approach. The aim is to reorganize the resources to meet users' needs with greater efficiency. The steps in this process are discussed in the following sections.

Establishing a Sponsor

This involves identifying component reuse opportunities and shows how their exploitation can contribute to an organization's IS goals. Sponsors must be identified and sold on the various ideas.

Developing a Plan

This plan should guide the management of the component development process. The plan includes the following:

1. *Reuse assessment.* This assessment should evaluate the potential opportunity for reuse, identify where the organization stands with respect to reuse, and evaluate the organization's reuse capability. Exhibits 4, 5, and 6 can be used to conduct the assessment.
2. *Development of alternative strategies.* On the basis of the assessment, an organization can develop a strategy to implement and align the process as well as choose the appropriate methodologies and tools.
3. *Development of metrics.* In the planning stage for implementation, metrics must be used to measure success.

Exhibit 4. Assessment of Component Potential for Reuse

Concern	What to Ask
Domain potential	In the given domain, are there applications that could benefit from reuse?
Existing domain components	Are expertise and components available?
Commonalities and variables	Is there a sufficient fit between need and available components? Can they be customized?
Domain stability	Is the technology stable? Do the components meet stable standards? Are the components portable across environments?

Exhibit 5. Assessment of an Organization's Capability to Reuse Components Columns

Application Development	Component Development	Management	Process and Technology
Component identification for use in application	Needs identification, interface, and architecture definition	Organizational commitment, planning	Process definition and integration
Component evaluation and verification	Component needs and solutions	Managing security of components	Measurements and continuous process improvement
Application integrity	Component quality, value, security, and reusability determination	Intergroup (component and application coordination)	Repository tool support and training

Implementation

The organization finally implements the plan. Incentives can be used to promote reuse by individuals and the organization.

CONCLUSION

Component technology is changing the way client/server applications are being developed. Supporting tools for this software environment are rapidly emerging to make the transition from regular application development to component-based development. With proper training of staff, planning, and implementation, organizations can make a smooth transfer to this new mode of development and rapidly develop and efficiently deliver client/server applications.

Exhibit 6. Organizational Reuse Capability Model

Stage	Key Characteristics
Opportunistic	• Projects individually develop reuse plan • Existing components are reused • Throughout project life cycle, reuse of components is identified • Components under configuration and repository control
Integrated	• Reuse activities integrated into standard development process • Components are designed for current and anticipated needs • Common architectures and frameworks used for applications • Tools tailored for components and reuse
Leveraged	• An application-line reuse strategy is developed to maximize component reuse over a set of related applications • Components are developed to allow reuse early in the life cycle • Process performance is measured and analyzed • Tools supporting reuse are integrated with the organization's software development efforts
Anticipating	• New opportunities for reuse of components build on the organization's reuse capability • Effectiveness of reuse is measured • Organizations reuse method is flexible and can adapt to new process and product environment

ABOUT THE AUTHORS

T.M. Rajkumar is an associate professor in the department of decision sciences and MIS at Miami University, Oxford, OH.

David K. Holthaus is a software specialist for Nationwide Insurance Enterprise, Columbus, OH.

Chapter 29
Distributed Computing: Objects and Components in the Enterprise

David Russo

It has become abundantly clear in this era of electronic commerce that monolithic solutions and islands of corporate information are no longer sufficient to maintain the enterprise in the new millennium. Instead of devoted client/server applications, the information system's focus has become one of managing and assembling the interactions of the distributed objects and components that comprise the modern environment. Further complexity is added to this shift in the application development paradigm by the competing and variously exclusive distributed computing solutions available: CORBA, RMI, MTS/COM, and the generic application server.

HISTORY

A brief history of application development is in order to fully appreciate the distributed computing paradigm and the resulting issues posed.

In the late 1980s several prominent computer scientists, including Bertrand Meyer (1987, *Object-Oriented Software Construction*) and Grady Booch, independently arrived at the concept of an "object" or "class" as the base component in the development of an application. At that time, almost all applications were monolithic and purpose-built; the advent of widespread communication between computing machinery was still in the future. In fact, the highest bandwidth network was driven by "sneakerware"; for example, the New York Stock Exchange employed large numbers

0-8493-1161-6/02/$0.00+$1.50

of young people who moved roles of tape and disks among isolated computing environments.

Meyer proposed his basic concepts of an object as a method by which the development of applications would be better managed and communicated. In Meyer's view, the "class" or object was centered about a data store or unique data element. This data store would have clearly defined "public" interfaces or methods and would thereby eliminate the concept of "global" or shared data. The putative user of such a class or object would not have to understand anything about the internal operations of the object. Instead, only the definition of the public interface would be required to use the object in any application. In Meyer's view, this "data-centric" approach provided not only more efficiently built applications but also held out the promise of reusability (i.e., components developed for one application could easily be used to assemble other completely disparate applications).

Ten years of hindsight have proven that the work of Meyer, Booch, and a host of others was correct. Today, most applications are designed and built using the concept of an "object" model (although this development is approached through a host of quasi-religious and wildly hostile object methodologies). A number of varying "object-oriented" languages have sprung up to supply the basis for this development, most prominently C++ and Java.

In the early 1990s with the arrival of the nascent Internet and the widespread availability of inexpensive and powerful Wintel machines, the computing environment began to move to one of inter-platform computing. The first step in this direction was the advent of the client/server paradigm wherein a "client" machine interacted with a "server" platform. The data or information resided on the server, while the client platform was the basis for input or information display; thus began the concept of "distributed" computing.

The client/server model was hindered by a number of issues: There were problems with synchronization of the data on the server, the scalability of an application across large numbers of users, and the increasing size of the client-side software leading to what was known as "fat" clients. More importantly, most client/server applications did not incorporate in their design the ever-more powerful and dominant Internet and its associated architecture of Web-based servers, HyperText Markup Language (HTML), and HyperText Transfer Protocol (HTTP) protocols. This led to a situation where Larry Ellison, CEO of Oracle and one of the earliest and loudest adherents to the client/server paradigm, declared the "death of client/server" in 1998.

In parallel with the rise and subsequent fall of the client/server paradigm was the formation of the Object Management Group (OMG), a consortium of nine companies that existed to form a distributed computing inter-operability standard. The work of the OMG led to the publication and dissemination of the CORBA (Common Object Request Broker Architecture) standard in October 1989. The purpose of the OMG and its CORBA standard was to define a "component-based software inter-operability and communication standard." This standard would be based on software development that models the real world through a representation of objects where these objects "are the encapsulation of the attributes, relationships and methods of software identifiable program components" (OMG, 1999).

The problem with the earliest CORBA release was that the OMG produced a standard but not a specification. This led to a situation where the "CORBA standard" product from one vendor would not work with the "CORBA standard" product from another vendor. The lack of a specification meant that there was no single test suite or compliance board that would verify that a given CORBA implementation actually met the CORBA standard.

In addition, and not unlike the early client/server architectures, CORBA failed to address a specific inter-platform communication protocol. Similarly, CORBA, again not unlike the early client/server, did not take into consideration the Internet and its coming dominance of the distributed computing environment.

CORBA and the OMG attempted to recover from this situation with the publication of CORBA 2.0 and the Internet Inter-ORB Protocol (IIOP) standards and attendant specifications. Unfortunately for the OMG, other approaches to handle distributed computing had become available during this period.

Among these competing and/or complementary products were the Component Object Model (COM) and its networked relative, Distributed COM (DCOM), from Microsoft. Sun Microsystems, with its Java language standard, provided a technique referred to as Remote Method Invocation (RMI), which allowed any Java class (object) to be transformed into a distributed component or Java Bean. The RMI effectively makes Java a distributed computing environment, although Sun was one of the founders of the OMG.

In addition to the various flavors of vendor-supplied distributed computing approaches, the latter part of this decade has seen the arrival of application servers, which are an architectural approach for binding a variety of machines, operating systems, and disparate applications to a Web-based solution.

In addition to the above approaches, EDS/MCI has forwarded an Open Distributed Computing (ODP) standard and specification that is being worked through the International Standards Organization. At the same time, Carnegie Mellon University has proposed a COTS (Complete Off The Shelf) Based System (CBS) initiative. This approach focuses on improving the techniques and practices by which previously existing components are assembled into large, distributed computing systems. While CBS recognizes CORBA and COM as component integration technologies, it is clear that the CBS initiative will, if widely accepted, modify the way these and other distributed computing technologies are employed.

At this time, these various standards are readily available and, while each has its supporters, no clear solution for the management of distributed components exists. It is clear, however, that the concept of distributed components, their creation, assembly, and management is one of the hottest topics in today's information systems world.

The unifying principle behind this thumbnail history is that the concept of an *object* has moved from being a technique for determining and building the components of an application to a method for describing, encapsulating, assembling, and managing entire applications in a distributed computing environment. The original concept of an object could have been used to define a Binary Tree Abstract Data Type (ADT) and its associated methods and how it would perform in a given application. The distributed computing paradigm view of an object incorporates all of the original definition as well as requiring the interfaces and techniques by which the Binary Tree ADT can have its methods and interfaces managed remotely, whether it be via a Java Bean, an Object Request Broker (ORB), or as a COM component.

The Case for Distributed Objects and Components

Distributed object computing is an architecture, a paradigm, and a set of technologies whereby objects can be distributed across a heterogeneous network in such a manner that arbitrary assemblies of these objects can operate as a single entity. Effectively, distributed objects extend the object-oriented analysis and design methodology to the network as a whole. The concept of an object refers to the nonincorporated software entity that is capable of being merged with other objects to form an assembly. Once integrated into the assembly, the *object becomes a component.*

The gains from realizing an effective means of creating, assembling, and managing a distributed set of objects are many. Extant software (i.e., legacy systems) can nonintrusively be incorporated into greater entities, thereby protecting existing investment. Order-of-magnitude gains in productivity can be realized as the developer moves away from the line-at-a-time paradigm and into the assembly, or industrial era of software development. For those objects that must be created using the techniques of the earlier era, the possibilities of reuse are manifold, thus protecting new investment. Efficiencies in deployment and execution are also realized because the individual objects comprising an assembly can be hosted on those machines that can best be utilized by that particular component. Last, and probably most important to the enterprise information area, is that the various "islands" of technology can be bridged to and effectively and selectively used to further the efforts of the corporation. Now the accounts receivable, inventory, order, and accounts payable systems can be merged at the data flow level without replacement or costly rework of the individual applications.

These five reasons — protection of extant investment, gains in productivity, protection of future investment, enhanced utilization of assets, and effective inter-departmental data integration — provide the basis for the movement to the distributed computing paradigm.

The Approaches to Realizing Distributed Computing in the Enterprise

In the commercial world, there are three primary approaches to effecting the distributed computing paradigm: CORBA, COM, and Java. A fourth approach, the application server, constitutes a form of application "metaware" wherein distributed computing is realized through a defined set of machines and "bound" applications. The application server approach is something of a superset and a subset of distributed computing approaches in that its focus is on "Webifying" extant applications via a set of proprietary interfaces. Interestingly, the more powerful application servers offer interfaces to COM/DCOM, MTS, CORBA, and Java Beans. In the following sections, the salient characteristics of each of these approaches will be discussed and compared.

CORBA

The oldest of the distributed computing standards, CORBA is the product of the OMG which at this time comprises more than 800 contributing companies and organizations. CORBA, with its vendor-neutral position and set of "open" standards, has always provided the most hope for a truly unified way to incorporate any software object into a distributed assembly.

However, the very "openness" of CORBA and its associated loose specifications has contrived to be the cause of its undoing. The CORBA products from most vendors — while compliant with the standards — in fact differ somewhat from one another and hence rely on vendor-specific protocols. This means that the key issue of standardization is illusory. The lack of a tight specification, while appealing to potential vendors in that it allows them to build their "own" CORBA implementation, means that the third-party market for CORBA-based applications does not exist because there is no standard.

This is not meant to imply that CORBA has no value in the distributed computing world. If a department or business is willing to employ the CORBA solution of a single vendor (such as BEA and its ObjectBroker system or Inprise with its Windows ORB VisiBroker), then a wide range of machines and many of their applications can be used to form a distributed computing environment.

How CORBA Works

CORBA uses an Object Request Broker (or ORB) for implementing its inter-object invocations. The latest specification for the ORB specifies the Internet Inter-ORB Protocol (IIOP) as the protocol by which objects are "remoted." The ORB acts much like a bus or backplane in a hardware device through which each CORBA object interacts with other CORBA objects.

For a CORBA client object to request service from a CORBA server object, the client must acquire a reference to the server component. The ORB will parse the available services (methods) of the server object and connect them with the request from the client object. The available methods are accessed by the ORB from Interface Definition Language (IDL) skeletons programmatically created for each server object. These IDL stubs are effectively the distributed object's interface made available for reading, distribution, and connection by the ORB. The IDL compiler provides type and method exposition information for each skeleton that is then stored in an Interface Repository. Effectively, the ORB is a message router and object invocation device that relies on IDL stubs and skeletons to resolve the various service requests.

The server-side skeleton will receive the invocation from the ORB and execute the requested method. Results from the method, as well as input arguments from the client proxy, will be routed via IIOP to the ORB that is responsible for providing them to the server stub. Hence, CORBA requires mulitple ORBs if a multi-platform solution is being deployed.

The essence of CORBA is the ORBs used to connect the various distributed objects. With the advent of IIOP, some of the vendor specificity problems have been eased, allowing ORBs from a variety of sources to interact successfully.

Java language facilities exist that allow native Java applications to access ORB capabilities.

COM/COM+/DCOM: THE MICROSOFT SOLUTION FOR DISTRIBUTED COMPUTING

As usual, Microsoft has provided a complete homogenous solution regarding the distributed computing paradigm. This solution is in the form of its COM and DCOM standards and products. It must be remembered, however, that with nearly 90 percent of all workstations being Wintel and with some corporations employing 100 percent Wintel solutions, the single-platform perspective may not represent a significant limitation.

In addition, the use of the COM/DCOM approach avoids many of the pitfalls encountered during a distributed computing deployment in that the support is already built into the great majority of Wintel platforms. COM is already installed on millions of machines with DCOM standard on NT 4.0 (DCOM is available via download for Windows 95 and 98). And, unlike CORBA, there is a huge third-party application market ready and willing to provide products and services for any Wintel initiative.

The two solutions represented by COM/DCOM and CORBA are not totally at odds. If the potential buyer needs only to distribute computing objects in a pure Wintel environment, then COM/DCOM is clearly the path to pursue. This is true for a number of reasons, the most salient being that this solution path is free. If, however, given the current situation where DCOM is not yet available for non-Wintel platforms, a prospective user desires to deploy a distributed computing solution in a heterogeneous environment, then only CORBA or the Java/RMI solution will suffice.

How DCOM works

DCOM is sometimes referred to as *COM on the wire.* DCOM works similarly to CORBA in that a client-side DCOM object creates a message and invokes a wire protocol similar to IIOP called ORPC (Object Remote Procedure Call) to communicate with the server DCOM object. Instead of an ORB, DCOM relies on a service control manager (SCM) to perform the various services of locating and activating an object implementation. As in CORBA, the server is responsible for invoking the method requested via interaction with the SCM. Once the client-side device has received a reference to the server-side object, it can access the exposed methods of the server.

In DCOM, the client side is called the proxy, while the server side is referenced as the stub. These stubs and proxies use an Interface Definition Language similar in purpose to that of the CORBA IDL. Instead of maintaining and using an "interface repository" as CORBA does, DCOM avails itself of the Microsoft registry services, thereby enhancing flexibility at the cost of increased complexity. Interestingly, DCOM server components can be built in a number of languages including C++, Java, and COBOL.

Because DCOM interacts with low-level COM services to access objects, any platform that supports COM can successfully interact with DCOM. COM is available for various UNIX flavors as well as some mainframe facilities from Software AG.

In terms of performance, there is no exact information by which to compare CORBA and DCOM. Because DCOM is primarily used with homogeneous environments while CORBA works in mixed OS/platform environments, any comparison would be less than meaningful. In general, the two approaches are very similar at a high level, differing only slightly in programmatic details such as whether parameters are passed by reference or by value (DCOM passes either by reference or value depending on the IDL, while CORBA passes by value only).

JAVA RMI

Java has proceeded to evolve from a Web-based applet development language to a serious contender in all forms of computation. With the advent of Java 2.0, there is almost no corner of the computing environment that does not have an available Java solution. While Microsoft has done everything in its power to create OS-dependent Java applications, Sun Microsystems carries on with its standards of a Java that runs anywhere and everywhere.

Because of Java's extreme flexibility and its ease of use, the Java RMI has become much more widely adapted within the distributed computing environment than either CORBA or DCOM. Unfortunately, this popularity has not gone unnoticed. Microsoft's Internet Explorer does not provide any support for RMI, which forces RMI users to use the Netscape browser, which in turn requires the installation of any necessary libraries on all potential clients.

RMI

The Java Remote Method Invocation (RMI) is significantly different from DCOM and CORBA in that it is almost completely a language-based capability that does not rely on the complexities of ORBs or SCMs and associated IDLs to implement distributed computing. Of course, the downside of this approach is that if one's distributed objects are not written in Java, then they cannot be accessed via RMI.

Sun Microsystems has recently announced that support for both Java RMI and Java IDL (designed so that Java applications can function in CORBA) will be provided for the indefinite future. It appears that Sun is going to simultaneously support both approaches to distributed computing and not allow RMI to "die on the vine." This announcement has ended months of industry speculation concerning which way Sun was going to go with regard to its supported distributed computing model.

How RMI Works

Effectively, the RMI uses Java language extensions to extend the Java Virtual Machine (JVM) address space so that it appears to include other virtual machines independent of where they might actually be hosted. RMI is effectively a JVM-to-JVM communication protocol allowing objects to be "passed." Unlike IDL-based distributed models, the RMI requires no mapping to common interface definition languages (IDLs). The syntax of RMI is such that it appears almost identical to that used for local invocations.

Because the inter-object communication relies on the JVM executable, objects can be distributed dynamically, thereby removing the requirement to provide installation on the client prior to implementation. This, of course, greatly eases the burden of distribution and maintenance. In addition, the RMI benefits from the built-in security features of the JVM, thereby guaranteeing a secure distributed object environment.

JAVA BEANS

A Java Bean is effectively a technique and an instantiation of a software component model for Java. The Java Bean technique relies on the *serialization* of an object provided by the JVM and language constructs. Serialization allows the state of an object (and any objects it refers to) to be written to an output stream. Later, the serialized object can be recreated by reading from an input stream. This technique is used to transfer objects between a client and a server for RMI. Many Beans are provided not as class files but rather as pre-initialized, serialized objects.

The Java Bean was designed for visual manipulation in a builder tool, much like an ActiveX component. Java Beans are not an architecture for distributed computing, but rather a technique for building and distributing the components of a distributed computing environment. Effectively, the Bean technology provides a way through Java *introspection* and *reflection* whereby a builder/developer can determine the properties, methods, and events of a Java Bean.

APPLICATION SERVERS

Application servers are both an architectural approach and a vendor-specific offering for a specific set of services and tools for implementing the architecture. The basic premise of the application server paradigm is to allow selected components, or objects, within the enterprise computing environment to be "exported" to the Web via links across Web servers. While application servers are not normally thought of as part of the distributed computing world, they effectively perform the same services as a CORBA, DCOM, or RMI, with the most significant difference being their Web-based orientation.

The more powerful application servers, such as the Open Connect Systems EIS suite, allow for the development of "objects," which are effectively applications or portions of applications. The binding of inputs to and outputs from these components and the attachment of these objects to specific Web pages accessed via a Web server are performed by the development and deployment services provided by the tooling.

The advent of application servers radically changes the distributed computing environment. The application server architecture provides for an arbitrary number of machines to actually host the components. In addition to this cross-platform capability, various load-balancing and state-server facilities are offered. In a very real way, while the industry has vacillated regarding which distributed computing protocol to use, vendors have offered an ultimate "glue" option whereby virtually any object, application, or component can be identified and bound to a specific Web enabler.

In addition to the ability of application servers to bind virtually any application to a Web page (including mainframe applications such as CICS and IMS), they also provide interfaces into CORBA and DCOM.

The main difference between application servers and the traditional distributed computing solution is that the application server architecture does not make any provision for dynamic or on-the-fly access to embedded objects. In fact, in most application server offerings, the components have to be "built" (i.e., identified and wrapped within the tool suite). This is not radically different from CORBA or DCOM, but both of these do provide significant capability in terms of dynamic object access. The application server, while platform independent in its ability to access applications, does generally require that all machines hosting the applications be identified to the various controllers and Load Balance Brokers that make up the deployed architecture. Again, this is unlike CORBA and DCOM, where the actual machine being used is transparent to the client or the server.

CONCLUSION

Distributed computing, with its roots in object-oriented technologies, has been around for a number of years. The OMG has attempted to standardize distributed computing with CORBA. Microsoft is pushing its own platform-specific distributed computing model — COM/DCOM. Sun's Java RMI is the latest distributed computing approach, and is more widely adapted than either CORBA or COM/DCOM.

A fourth option exists — the Web-based application server — which is particularly well suited for organizations that are building new applications for Web servers and wish to bind existing applications into the new Web-based applications.

Which of the four approaches an organization should adopt depends on the environment and the application needs. IT organizations evaluating the possibilities should answer the following questions before making a selection:

1. What operating systems and platforms will be utilized by the clients and servers?
2. Are the new applications to be delivered via Web technologies (i.e., intranets, Internet)?
3. What off-the-shelf or custom applications will use the distributed computing infrastructure? Which of the models do the applications support?

ABOUT THE AUTHOR

David Russo is a Senior Architect at Open Connect Systems in Dallas, TX. David has worked as a software designer, developer, and process engineer for more than 20 years at a number of firms, including Texas Instruments, Raytheon Systems, and Open Connect Systems. David's undergraduate degree is from the University of North Florida, and he holds a Master of Science degree in Computer Science from Southern Methodist University. David also serves as an Adjunct Professor of Computer Science at SMU. David would enjoy your comments regarding this chapter and can be reached at drusso@oc.com.

Chapter 30
Developing Internet Solutions with SOAP

Charles Dow

What is SOAP? SOAP is an Internet specification that provides a method to invoke programming and to pass data (both parameters and results) from clients to servers and to get a response from those servers.

SOAP was designed to allow multiple random computers (i.e., computer programs) to successfully communicate, similar to the way most browsers will communicate with a Web server, where the server does not need to know the client; but unlike a browser, instead of just passively reading files from the remote computer (pardon the oversimplification), SOAP actually allows these computers to work together to solve a problem, i.e., work together as pieces of an application.

TECHNICAL UNDERPINNINGS OF SOAP

SOAP uses HTTP as the transport protocol. (*Note:* For this chapter we are not considering any of the other transport protocols that SOAP can also leverage.) To most people, HTTP and surfing the World Wide Web are synonymous. Using HTTP as the transport protocol avoids the problem of configuring Web servers, firewalls, proxies, or ports. Most experienced programmers can write SOAP-compliant programs with no special training or tools. A SOAP method is simply an HTTP request and response that conforms to the SOAP rules.

But SOAP requires more than HTTP. HTTP provides an agreed-upon method to send and receive messages between services. SOAP also requires an agreed-upon common language for the message and its structure. For that purpose the other de facto Internet standard, XML, was chosen to operate with SOAP.

HTTP has already been used to invoke code across the Internet, as browsers have been launching ActiveX, Java, and CGI programs for a long time. But by adding the discipline and flexibility that XML allows, the SOAP

0-8493-1161-6/02/$0.00+$1.50

specification has documented, improved, and organized what was becoming an increasingly common practice into a standard specification that is well suited for a broad range of applications. Coupled with the improvements that are occurring simultaneously with HTTP, we have a formidable new weapon with which to tackle our new challenges.

SOAP is platform- and language-independent by design.

WHAT DID THE SOAP DESIGNERS IGNORE?

SOAP does not provide nor is it a substitute for a programming language or model. SOAP is trying to simplify the plumbing of Internet applications and achieve the ubiquity and success of HTTP. The designers in effect appear to have said, "HTTP works; it is commonly available and accepted; let's use it. XML is also becoming ubiquitous; let's use that too."

The authors of SOAP sidestepped the DCOM, CORBA, and other wars (see the section entitled "Other Competing Protocols") and created a simple application protocol for the Internet.

Other competing protocols, e.g., DCOM and CORBA, are viewed as best suited to a controlled group, where the sender and receiver are likely to be known to each other, e.g., server to server within an organization.

While SOAP lacks the elegance and richness of those competing technologies, it makes up for that in simplicity by allowing applications to be built now. SOAP is an attempt to provide a common foundation, not an attempt to resolve those different opinions.

WHY SHOULD YOU CARE?

You can now create applications from smaller working mini-apps, services, or components — call them what you will — that could be scattered all over the Internet. You can use your favorite programming language or object model, or choose to not use any of those, and yet collaborate and interact with other applications using SOAP TODAY! You can send messages merrily on their way to any number of partners, collaborators, customers, or others, if you so choose, or provide a service to the world.

These clients or servers could use a vast array of Web-ready cell phones, Palm Pilots, IBM mainframes, Internet devices, etc. Any device that can access the Internet should easily be able to use SOAP.

Web Services are becoming the new Holy Grail for application developers. As the Internet becomes more and more ubiquitous, we are now beginning to ask once again: What is an application? We had grown to accept that pieces of an application might be spread all over a computer's memory space. A very common example is a PC's operating system with its many pieces and optional components. Now that surfing the Web is popular, we have grown to accept that applications will be able to use the Internet as "plumbing." But we still do not have many applications where different portions are hosted by different organizations.

Web Services is the ability to provide application services over the Internet usually for a commercial purpose or for some other form of collaboration. This is seen as the next wave of computing. Microsoft, IBM, Sun, and others are all charging hard in this direction.

Many financial services companies are also pushing very hard to accomplish this on their own, not content to wait on the software vendors.[1] One vision in the United States has you providing all your passwords to Vanguard Group (Vanguard is the second-largest mutual fund company in the United States); once you have done that, you will be able to move your funds from an account in a financial institution to Vanguard, and vice versa.

This sort of collaboration is facilitated by SOAP. There are some occasions today where application hosting is used, e.g., when you search on a Web site, that search engine may be hosted elsewhere by a company that specializes in that technology; when you virus-check files on the Internet, that virus-check application may be hosted by a different organization. That application collaboration is totally transparent to the user — as it should be.

We seem to be moving inexorably on a path to the future where a great many computers all over the world will be interconnected. These computers are already working on specific applications at the request of computer users including upgrading their software on the fly with little or no input from you or me. Later they will work autonomously for the average person to facilitate business that can be referred to as "everyday life." They collate, organize, and eventually make decisions on our behalf. There are many obstacles that need to be overcome, but in time we will likely discover that the technical obstacles were the easiest to solve.

Finally, SOAP is also well suited to allow interaction with devices, such as refrigerators and heating and cooling systems. These labor-saving devices will become increasingly automated and remotely controlled in the near future.

THE KISS RULE

SOAP is likely to be successful because the authors applied the KISS (Keep it Simple Stupid) rule. Several attempts have been made to achieve widespread interoperation of computer applications over the years with little success. SOAP's timing is fortuitous because XML is also riding a wave of popularity, with most of its tools freely available.

The rise of the Internet has changed the global technology arena by essentially wiring everyone together. SOAP is an attempt to piggyback application invocation on the popularity of the ubiquitous HTTP protocol and increasingly popular XML. XML allows programmers to get complex and sophisticated, if they so desire.

LET US EXAMINE THE POPULAR HTTP

HTTP can perform request/response communications over TCP/IP. TCP/IP has, to all intents and purposes, won the Internet protocol war. An HTTP client connects to an HTTP server using TCP/IP. HTTP provides the ability to package information and to connect dissimilar computers or devices. HTTP is text-based, which supports simplified debugging capability. It is based on a standard. The HTTP 1.1 specification has some wonderful new features if you are an application programmer, e.g., persistent connections are the default (where the TCP connection remains open between consecutive operations); request/responses can be pipelined (where multiple requests and responses occur over a single HTTP connection); when reliable caching under user control is available there is a reduction in latency, i.e., faster response; better warnings etc. See section on "Further Reading" for more details.

HTTP Headers

SOAP/1.0 mandates HTTP headers that facilitate firewall or proxy filtering. The headers allow network administrators to deal with SOAP requests intelligently. The spec also defines how parameters, return values, and exceptions should be declared. These are all needed to do proper programming.

HTTP Request Methods

A SOAP request can be used with a variety of HTTP request methods but is primarily used with an HTTP POST request. The examples below show that the SOAP requests are similar to most HTTP requests used for everyday surfing of the Web. SOAP also allows M-POST if one is using the HTTP Extension Framework.

HTTP Content Type

A rule of the SOAP request is that it must use the text/xml content type. Further, it must contain a request URI. The SOAP request also indicates the method to be invoked through the use of the SOAPMethodName HTTP header. The SOAP response is similar.

HTTP Extension Framework

You will also come across the HTTP Extension Framework, a generic extension mechanism for HTTP to permit the coordination of extensions. SOAP is able to use an HTTP Extension Framework by using "M-" HTTP method name prefix, e.g., M-POST. The "M-" essentially forces the server to respond, an essential feature when an application is using the request/response pattern and needs a response to continue processing. *Note:* The specification allows either the client or the server to force the use of the HTTP Extension Framework.

WHO IS SUPPORTING THIS TECHNOLOGY?

DevelopMentor has made reference implementations available on several platforms using Java and Perl. IBM and its subsidiary Lotus Development, Microsoft, Ariba, Commerce One, Compaq, Intel, Software AG, and SUN are some of the familiar names that are supporting this technology standard.

OTHER COMPETING PROTOCOLS

Competing technologies include:

- DCOM
- CORBA's Internet Inter-ORB Protocol (IIOP)
- RMI

These have typically attempted to do more but were complex to implement and usually required close coordination between sender and receiver. Some of these competing technologies have achieved widespread adoption in communities but not the critical mass that would make them the prevailing standard for the Internet. These earlier technologies usually assumed that we would be working with a network and collaborating with close associates, not the Internet.

Their notable weaknesses are:

- DCOM is still Microsoft-centric; it often requires some version of Windows.
- CORBA usually requires ORBs that are known to work together.
- RMI typically requires both client and server to be Java applications.

SOAP'S APPEAL

SOAP will allow any class of device to communicate with any other. It is not influenced by the operating system used by these devices. It does not require domain authentication. It is easy to use even if firewalls are in place or proxy servers are used. It does not require firewall tunnelling or any special software to operate effectively.

SOAP UNWRAPPED

SOAP (also known as XMLP/SOAP[2]) is considered a transport protocol for XML messaging. The specification consists of three parts:

- An envelope that defines the contents of the message being exchanged
- Encoding rules for different data types
- A convention for representing remote procedure calls and responses

SOAP uses SOAP messages, as described in the next section.

A SOAP MESSAGE

A SOAP message is fundamentally a one-way message that can be combined using exchange patterns to perform request/response, multicast, or other functions. *Note:* Version 1.1 includes support for Simple Mail Transfer Protocol, FTP, and TCP/IP. There are two types of SOAP messages: call and response.

Every SOAP message is an XML document that consists of:

- *A mandatory SOAP envelope.* The envelope mainly contains references to namespaces as well as additional attributes such as the encodingStyle attribute that specifies the serialization rules[3] used in a SOAP message.
- *An optional SOAP header.* The optional header is often used to specify the following attributes:
  1. encodingStyle: this allows the author to specify the encoding style used by the header entries.
  2. mustUnderstand: this allows the author to specify how the message must be processed and force the application to recognize how the application should process the message.
  3. actor: this specifies by whom the message should be processed.
- *A mandatory SOAP body.* The body is used to package the information that needs to be sent to the application.

If the HTTP request uses namespaces, it must use the prescribed two SOAP namespaces:[4]

1. An envelope namespace
2. An encoding namespace

The server is required to discard SOAP messages that use incorrect namespaces. The application will assume that the author of the message meant to use the correct namespaces if they are missing.

SOAP Fault Element

SOAP provides a facility for reporting what error occurred and who reported it.

SOAP Fault Codes

These are a set of predefined faults that can occur in an interaction, as shown in Exhibit 1. All lines before SOAPAction are standard HTTP.[5] The

Exhibit 1. SOAP Request Using HTTP

```
POST /timeServer HTTP/1.1
Host: www.timeserver.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "http://this.author.net/timeServer"

<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
   <SOAP-ENV:Body>
      <m:GetTimeInCountry xmlns:m="http://this.author.net/timeServer">
         <country>Mexico</country>
      </m: GetTimeInCountry >
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note: If you are using the HTTP extensions, you would need to use the M-POST verb.

SOAPAction header indicates that this is a SOAP request allowing the server to do special handling, if so desired.

The SOAP envelope follows. The envelope contains the SOAP body. Notice the xmlns attribute which specifies the XML namespace (which could be assumed). And the SOAP-ENV: encodingStyle, which points to the SOAP serialization specification for the body (authors are free to choose another standard specification or design a custom XML structure).

The SOAP body is enclosed. The first attribute points to the method that executes. The second parameter provides some data to the program (additional parameters are of course allowed, but not required for the example shown in Exhibit 2).

An excellent example[6] that shows the simplicity of SOAP applications may be found (at the time of writing) at http://www.skonnard.com/soap/. Try it.[7] It works for IE5 and IE6. Point your browser to http://www.skonnard.com/soap/. Use the drop-down entitled: Test-Endpoints and Select the http://soap.develop.com/sd2/vbasperl.soap (reverse_string) option. Using this example, you are able to send a SOAP message of your choice to the server over the Internet; the server will take any string you send and reverse it; the result is returned.

Notice that in the payload, there is the string !dlrow olleH, which is Hello World reversed. Feel free to edit the string and send a different SOAP message.

MORE SOAP?

SOAP is not finished as yet. Several areas are being worked on — versioning, authentication, etc. — so keep watching for more details. See the section on “Further Reading.”

Exhibit 2. SOAP Response Using HTTP

```
HTTP/1.1 200 OK
Content-Type: text/xml;
charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope
   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/>
   <SOAP-ENV:Body>
      <m:GetTimeInCountryResponse
xmlns:m="http://this.author.net/timeServer">
         <Time>17.30</Time>
      </m: GetTimeInCountryResponse >
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note: The response element name is the request element with "Response" appended. This is a useful convention. The XML child element in the body, in this instance named Time, has the response (there can be multiple child elements).

SOAP IMPROVED

One of the first steps that you may now be tempted to do is to add to your environment of choice a set of wrappers that would allow you to use a method in your language of choice and have all the SOAP mechanisms invoked in the background transparently, e.g., the example above could be coded in Java as:

```
string timeServer.getTimeInCountry(nameOfCountry)
```

This would further simplify programming and not clutter up the business logic of the application.

Notes

1. Say It Loud, Their Average and Proud, *Fortune,* April 30, 2001, 115.
2. Editors' Copy: XMLP/SOAP/1.1 http://www.w3.org/2000/xp/Group/1/04/17/xmlp-soap-01
3. Serialization is the mechanism for packaging the information in the XML.
4. An XML namespace is a collection of names, which are used in XML documents as element types and attribute names. The purpose of an XML namespace is essentially to agree on a common set of names.
5. Hypertext Transfer Protocol: HTTP/1.1, http://www.w3.org/Protocols/rfc2068/rfc2068
6. This code is courtesy of Aaron Skonnard, http://www.skonnard.com
7. The only error you may get is permission denied. If so, try it again after a few days; the Web site might be undergoing further development.

Further Reading

1. SOAP 1.1. Specification SOAP: Simple Object Access Protocol, April 18, 2000. http://www-106.ibm.com/developerworks/library/soap/soapv11.html; Don Box, DevelopMentor; David Ehnebuske, IBM; Gopal Kakivaya, Microsoft; Andrew Layman, Microsoft; Noah Mendelsohn, Lotus Development Corp.; Henrik Frystyk Nielsen, Microsoft; Satish Thatte, Microsoft; Dave Winer, UserLand Software, Inc.
2. SOAP: Simple Object Access Protocol. msdn.microsoft.com/xml/general/soapspec.asp

3. The Next Servlet: The Request and the Response. http://wdvl.internet.com/Authoring/Java/Servlets/next_servlet.html
4. Morris, Charlie, Why is SOAP Causing Such a Lather? November 6, 2000. http://www.wdvl.com/Authoring/Languages/XML/Soap/lather.html
5. Namespaces in XML World Wide Web Consortium, January 14, 1999, http://www.w3.org/TR/REC-xml-names/
6. Nielsen et al., An HTTP Extension Framework, February 2000. http://www.normos.org/ietf/rfc/rfc2774.txt
7. Yergeau, F., UTF-8, a Transformation Format of Unicode and ISO 10646. http://www.faqs.org/rfcs/rfc2044.html
8. UTF-8 and Unicode Standards. http://www.utf-8.org/
9. Aldrich, Jonathan, Dooley, James, Mandelsohn, Scott, and Rifkin, Adam, Providing Easier Access to Remote Objects in Distributed Systems, California Institute of Technology 256-80, Pasadena, CA. http://www.cs.caltech.edu/~adam/jedi/paper/
10. HTTP Made Really Easy: A Practical Guide to Writing Clients and Servers. http://www.jmarshall.com/easy/http/
11. HTTP — Hypertext Transfer Protocol. http://www.w3.org/Protocols/
12. Gettys, Jim, Overview of New HTTP/1.1 Functionality and Changes from HTTP/1.0. http://www.w3.org/Talks/9608HTTP/http_1_1.ppt

ABOUT THE AUTHOR

Charles Dow is vice president of Product Engineering–Banking Systems for SLMsoft.com Inc.

Chapter 31
Elements of a Well-Designed Corporate Web Site

Mark G. Irving

Most organizations that create Web sites intend them to be well designed. They want the sites to explain the company and its activities in a way that is accurate, interesting, and easy to follow. However, many organizations have not planned how to accomplish this objective. This chapter details what constitutes a well-designed Web site.

WEB SITE DEVELOPMENT

Web sites can be developed in-house or by outside developers. Each method has its own set of risks, as well as a few risks that are common to both. The following sections outline these risks.

Web Pages Developed In-House

Many organizations economize on their sites by developing them internally. After all, they reason, there is the talent in-house, so why pay a premium for it? In addition, it can be up and running much faster than if it were developed outside.

Such organizations, however, run some significant risks. If one can answer *no* to any of the following questions, one's site could be improved, whether or not it was quickly or systematically developed:

1. Did your Web site take more than a day to develop?
2. Is there is a dedicated full-time Web master to administer the site?
3. Has the Web master been trained in HTML, Java, Active X programming?
4. Is the Web master experienced in this vocation?

0-8493-1161-6/02/$0.00+$1.50

5. Does the Web master attend training seminars on a regular basis?
6. Are Web pages updated in a timely fashion?
7. Is there a Web page policy, or are there established guidelines for page layouts and content?
8. Are pages reviewed and approved before being published on the Web?
9. Is your Web site isolated from your corporate network?
10. Do you have a staging area or test Web server where changes are tested before being published?
11. Is your Web server secured?
12. Do you have a firewall in place to prevent update access to your Web site?
13. Is your Web site backed up on a scheduled basis?
14. Are your backups tested on a regular basis?
15. Do you have a graphic artist employed who works with your Web master?
16. Do your pages have a common theme?
17. Are common graphics and navigation elements incorporated in them?
18. Are the pages logically linked to each other, or are they confusing and difficult to follow?
19. If you have a large Web site, do you have reference to a site map on the home page?

Web Pages Developed Outside the Organization

Web sites developed outside the organization also involve risks. Changes should be made if one can answer *no* to any of the following questions:

1. (All of the questions listed in "In-House Developed Web pages")
2. Is your Web developer well established and respected in the field?
3. Do you have a nondisclosure agreement with your developer?
4. Has a contract between your company and developer been prepared and reviewed by your legal department, before signing?
5. If your Web site resides on the developer's server, is it properly secured from other customers' sites?
6. Are modifications tested and approved by your company before being published?
7. Is there a good working relationship between your company and the developer?
8. Was the Web site developed jointly by the developer and the company?
9. During the audit, do you have access to the developer for questions?

A good Web site can be developed in-house, but sufficient time and resources must be put toward such an effort. Generally, a site that has been prepared with the aid of an outside developer will cost more (initially), take more time to develop, have more thought put into it, and use the expert's experience. Consequently, better design elements will be incorporated into the site. This leads into the next section.

DESIGN ELEMENTS

Web sites that are updated as necessary and contain good design elements are usually well designed. Web sites that are poorly constructed usually contain flaws that surface when they are audited.

Exhibit 1 shows a poorly constructed Web site. It is unorganized and cluttered. If the site has a theme, it is not evident. Awkward fonts, poor formatting, and a busy background make the site unattractive and confusing to follow. Poor spelling and grammar are used. Not all the links work. The site has no counter, no guest book, and no contacts list.

Exhibits 2 and 3, in contrast, show a well-constructed site. It is organized, attractive, and uncluttered. Its theme is evident. Spelling and grammar are correct and it uses graphics well. All its links work because it is maintained. It contains a counter, a guest book, and contacts list.

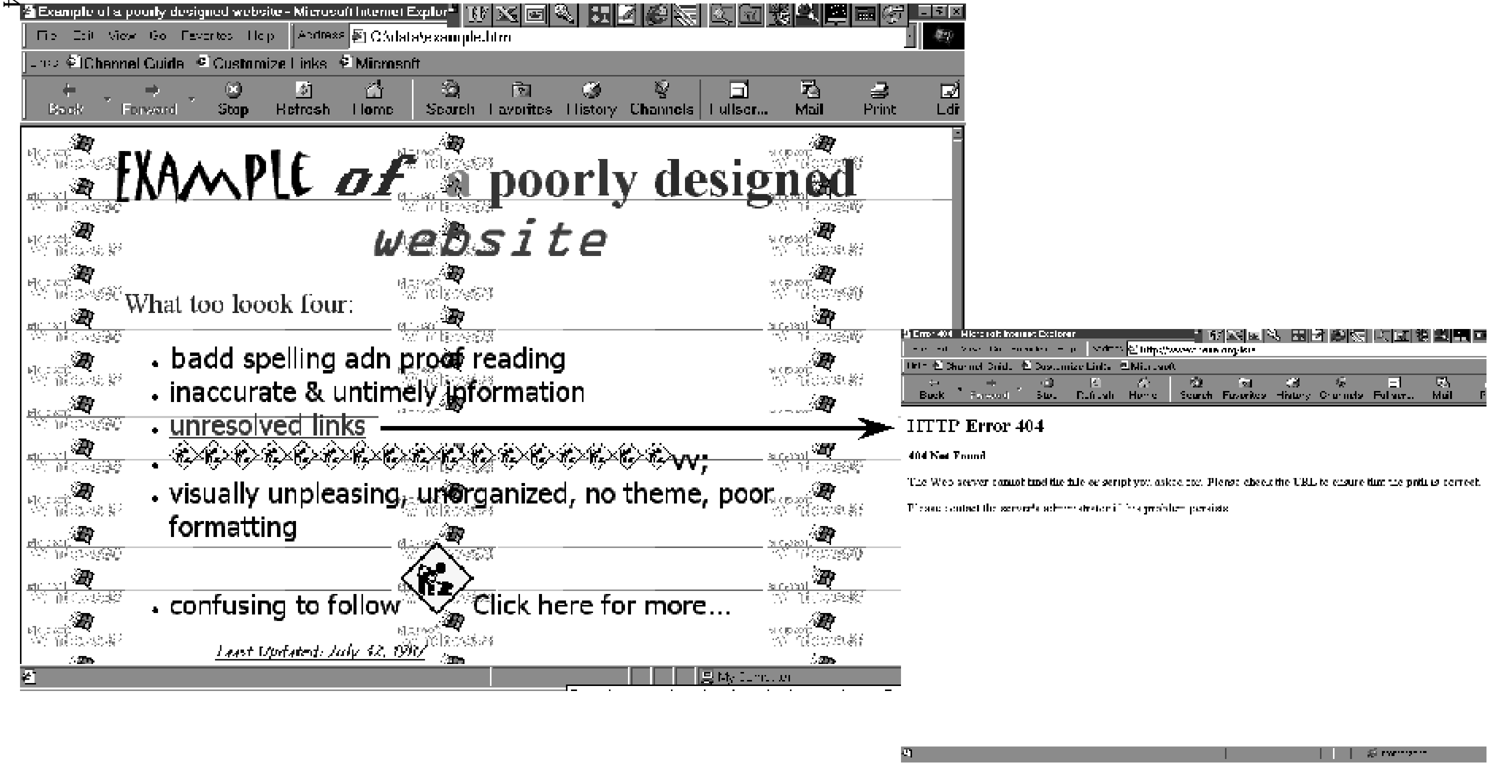
The following sections describe the basic elements of good Web site design.

Web Page Security. Make sure update access to the Web pages is properly secured from unauthorized access. Review where source code for Web pages is stored, and check who has update access to this area.

Ease of Navigation Between Web Site Pages. It should be easy to go back and forth between different areas on the Web site. Logical layouts with common elements should be grouped together. The use of multiple pages that are linked is preferable to one long page that requires a lot of scrolling. Having a common menu bar in the same location on each screen is helpful.

Visually Pleasing Layout. A Web site should have visual appeal. Incorporating the right balance of colors, pictures, text, animations, and white space is a difficult task. This takes the talents of an artist and a magazine editor rolled into one.

Graphical Animations. The use of eye-catching graphics in the forms of banners, movie clips, cartoons, etc. can put the final touches on a Web site. Overuse, by the same token, can make a Web site look messy and cluttered. Many times, animations are used for advertising for sponsors of a Web page.


Example of a poorly designed website - Microsoft Internet Explorer
File Edit View Go Favorites Help
Address C:\data\example.htm
Links Channel Guide Customize Links Microsoft
Back Forward Stop Refresh Home Search Favorites History Channels Fullscr... Mail Print Edit
My Computer
Error 404 - Microsoft Internet Explorer
Back Forward Stop Refresh Home Search Favorites History Channels Fullscr... Mail

EXAMPLE of a poorly designed website

What too loook four:

- badd spelling adn proof reading
- inaccurate & untimely information
- unresolved links
- vv;
- visually unpleasing, unorganized, no theme, poor formatting
- confusing to follow Click here for more...

Last Updated: July 42, 1967

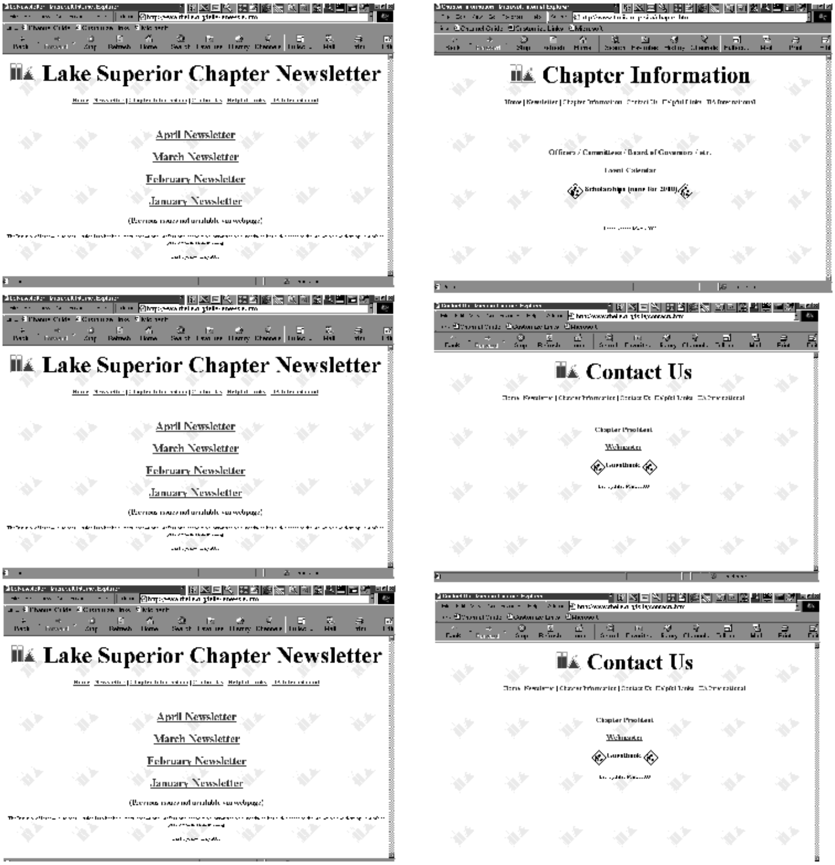
HTTP Error 404

404 Not Found

The Web server cannot find the file or script you asked for. Please check the URL to ensure that the path is correct.

Please contact the server's administrator if this problem persists.

Exhibit 1. Example of a Poorly Constructed Web Site



Lake Superior Chapter Newsletter

April Newsletter

March Newsletter

February Newsletter

January Newsletter

Chapter Information

Officers / Committees / Board of Governors / etc.

Event Calendar

Scholarships (new for 2000)

Contact Us

Chapter President

Webmaster

Guestbook

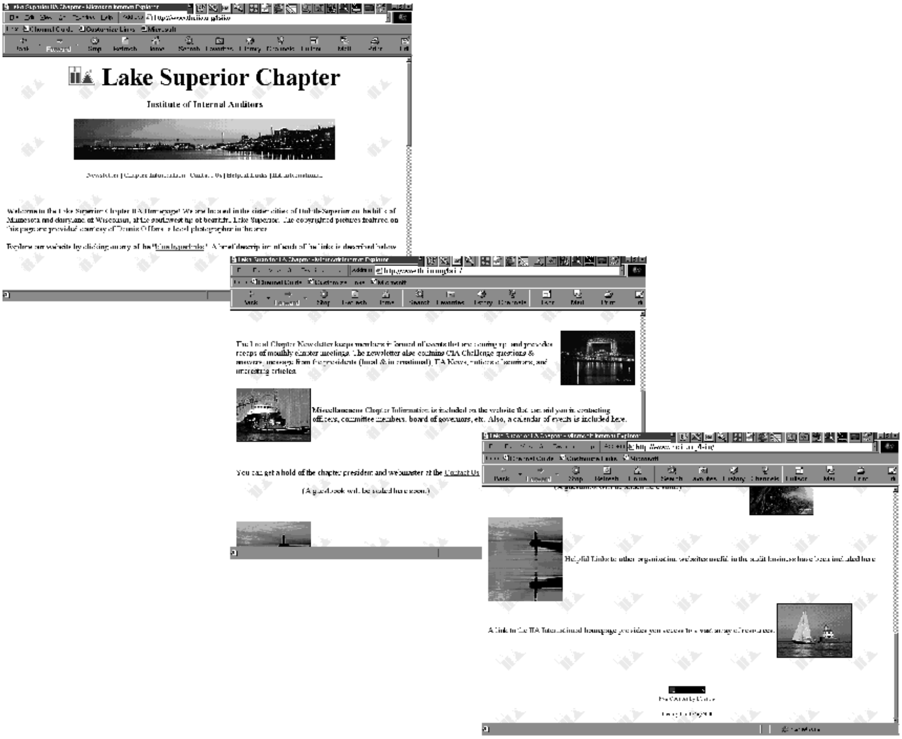
Exhibit 2. Home Page of a Well-Constructed Web Site

Timely, Accurate, and Interesting Information. Information that is not all of these does not belong on a Web page. Outdated, misspelled, boring content is the quickest way to ensure that visitors will not return a second time to a Web site. Proofreading and editing are essential to any successful Web site.

Statistical Information Tracking. This is the use of software to track how many visitors visit a Web site, where they are from, and what time they access the Web site. Guest books fall in this category to solicit what people think of the site and to get new ideas for improving a site.

TYPE OF SITE

What type of Web site is being reviewed? Web sites fall into one of the following classifications:



Lake Superior Chapter

Institute of Internal Auditors

Newsletter | Chapter Information | Contact Us | Helpful Links | IIA International

Welcome to the Lake Superior Chapter IIA Homepage! We are located in the sister cities of Duluth-Superior on the hills of Minnesota and dairyland of Wisconsin, at the southwest tip of beautiful Lake Superior. The copyrighted pictures featured on this page are provided courtesy of Dennis O'Hara, a local photographer in the area.

Explore our website by clicking on any of the "Photo hyperlinks". A brief description of each of the links is described below.

The Local Chapter Newsletter keeps members informed of events that are coming up, and provides recaps of monthly chapter meetings. The newsletter also contains CIA Challenge questions & answers, messages from the presidents (local & international), IIA News, notices of seminars, and interesting articles.

Miscellaneous Chapter Information is included on the website that can aid you in contacting officers, committee members, board of governors, etc. Also, a calendar of events is included here.

You can get a hold of the chapter president and webmaster at the Contact Us

(A guestbook will be added here soon.)

Helpful Links to other organizations' websites useful in the audit business have been included here.

A link to the IIA International homepage provides you access to a vast array of resources.

Exhibit 3. Site Pages of a Well-Constructed Web Site

- *Informational*: includes an overview about the company, the officers, financial information, site locations, whom to contact, etc. The majority of corporate Web sites fall into this category. Can be on the Internet or an internal intranet.
- *E-commerce*: selling a product or service. Used as an avenue to exchange currency for a product via credit card over the Internet, or via phone or fax. Usually will find some type of security feature employed to ensure financial transactions are secure. Most often on the Internet, but some variations are present on intranets; employee sales and auctions of merchandise.
- *Combination*: elements of informational and E-commerce are both present. More and more companies are heading this way. Many traditional businesses have expanded from department stores to catalogs, mail orders, into E-commerce. This is a natural progression.

Keep in mind that there are more risks in E-commerce and combination Web sites. Databases of customer information (credit cards, phone numbers, addresses) and product inventory (items on hand, dealer costs, etc.)

must be linked to provide E-commerce. Security between corporate assets and what the outside customer can access must be adequately safeguarded. A hacker who gains access here could do a lot of damage.

SUMMARY

This chapter has outlined the elements of a well-designed Web site.

Books and Articles

1. Holzschlag, Molly E. *Laura Lemay's Guide to Sizzling Web Site Design.* Software edition; Lemay, Laura, Series Editor, Sam.net Publishing, 1997. ISBN 1-57521-221-8.
2. Holzschlag, Molly E. and Oliver, Dick. *Teach Yourself HTML 4.0 in 24 Hours,* 2nd edition; Sam.net Publishing 1997. ISBN 1-57521-366-4.
3. Menkus, Belden. Auditing the Effectiveness of the Design of a Web Page. *EDPACS* (June 2000); 2000 CRC Press LLC.
4. Dallas, Dennis A. Perl and CGI for Auditors (74-15-03). *EDP Auditing,* Auerbach Publications, 2000, CRC Press LLC (February–March 2000).

Tools

1. Microsoft Frontpage Express; Version 2.02.1118; Copyright 1995-97; Microsoft Corporation.
2. Claris Homepage; Claris Corporation.
3. Internet Explorer, Microsoft Corporation.
4. Netscape Communicator, Netscape/AOL, Inc.

ABOUT THE AUTHOR

Mark G. Irving, CISA, is an information systems auditor at Minnesota Power (MP). Mark has been with MP for more than 20 years, holding positions in engineering computer applications, information systems, load research, and internal audit. Mark is also the president of the computer consulting and Web design firm Irving M&D Family Computer Practice. He is also founder and co-owner of Teleport Enterprises, a hand-held and Web training institute located in and operating out of Duluth, MN.

Chapter 32
Developing a Trusted Infrastructure for Electronic Commerce Services

David Litwack

The use of Internetworking applications for electronic commerce has been limited by issues of security and trust and by the lack of universality of products and services supporting robust and trustworthy electronic commerce services. Specific service attributes must be addressed to overcome the hesitation of users and business owners to exploit open systems — such as the Internet — for commercial exchanges. These service attributes include:

- *Confirmation of identity (non-repudiation).* This indicates proof that only intended participants (i.e., creators and recipients) are party to communications.
- *Confidentiality and content security.* Documents can be neither read nor modified by an uninvited third party.
- *Time certainty.* Proof of date and time of communication is provided through time stamps and return receipts.
- *Legal protection.* Electronic documents should be legally binding and protected by tort law and fraud statutes.

SERVICE ATTRIBUTE AUTHORITY

To support these service attributes, an organization or entity would need to provide:

- Certificate authority services, including the registration and issuance of certificates for public keys as well as the distribution of certificate revocation and compromised key lists to participating individuals and organizations

0-8493-1161-6/02/$0.00+$1.50

- A repository for public key certificates that can provide such keys and certificates to authorized requesters on demand
- Electronic postmarking for date and time stamps, and for providing the digital signature of the issuer for added assurance
- Return receipts that provide service confirmation
- Storage and retrieval services, including a transaction archive log and an archive of bonded documents

These service attributes could be offered singly or in various combinations. The service attribute provider would have to be recognized as a certificate and postmark authority. The following sections describe how a service attribute provider should work.

Certificate Authority

Although public key encryption technology provides confidentiality and confirmation of identity, a true trusted infrastructure requires that a trusted authority certify a person or organization as the owner of the key pair. Certificates are special data structures used to register and protectively encapsulate the public key users and prevent their forgery. A certificate contains the name of a user and its public key. An electronic certificate binds the identity of the person or organization to the key pair.

Certificates also contain the name of the issuer — a certificate authority — that vouches that the public key in a certificate belongs to the named user. This data, along with a time interval specifying the certificate's validity, is cryptography signed by the issuer using the issuer's private key. The subject and issuer names in certificates are distinguished names, as defined in the International Telecommunications Union-Telecommunications Standards Sector (ITU-TSS) recommendation X.500 directory services. Such certificates are also called X.509 certificates after the ITU-TSS recommendation in which they were defined.

The key certificate acts like a kind of electronic identity card. When a recipient uses a sender's public key to authenticate the sender's signature(or when the originator uses the recipient's PKS to encrypt a message or document), the recipient wants to be sure that the sender is who he or she claims to be. The certificate provides that assurance.

A certificate could be tied to one individual or represent an organizational authority that in turn represents the entire organization. Certificates also could represent various levels of assurance — from those dispensed by a machine to those registered with a personally signed application. Additional assurance could be provided by the personal presentation of a signed application along with proof of identity or by the verification of a biometric test (e.g., fingerprint or retina scan) for each use of the private key.

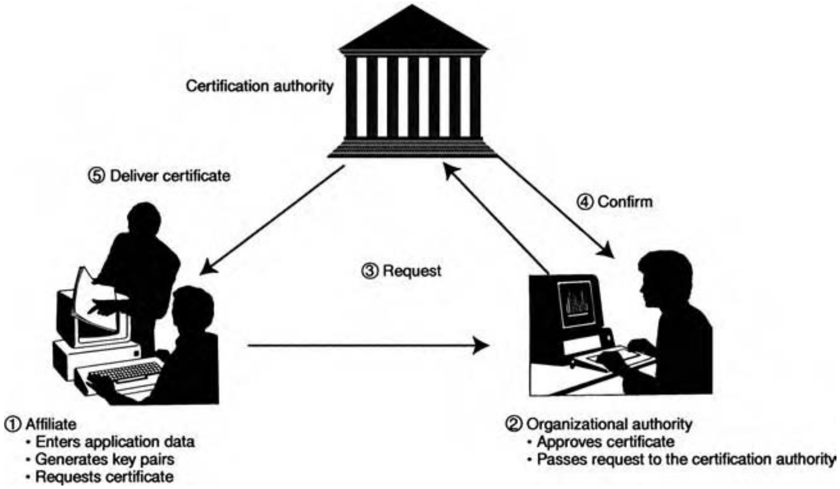


Exhibit 1. The Registration Process

Exhibit 1 shows a possible scenario for obtaining a certificate. The registration process might work as follows:

- The affiliate (i.e., candidate for certificate) fills out the application, generates private-public key pairs, and sends for the certificate, enclosing his or her public key.
- The organizational authority approves the application.
- The organizational authority passes the certificate application to the certification authority.
- The certification authority sends back a message confirming receipt of the application.
- After proper proofing, the certification authority sends the certificate to the applicant-affiliate.
- The applicant-affiliate then loads the certificate to his or her workstation, verifies the certificate authority's digital signature, and saves a copy of the certificate.

Digital Signatures. Exhibit 2 illustrates how a digital signature ensures the identity of the message originator. It shows how a message recipient uses an originator's digital signature to authenticate that originator.

On the Web, authentication could work as follows:

- The originator creates a message and the software performs a hash on the document.
- The originator's software then signs the message by encrypting it with the originator's private key.

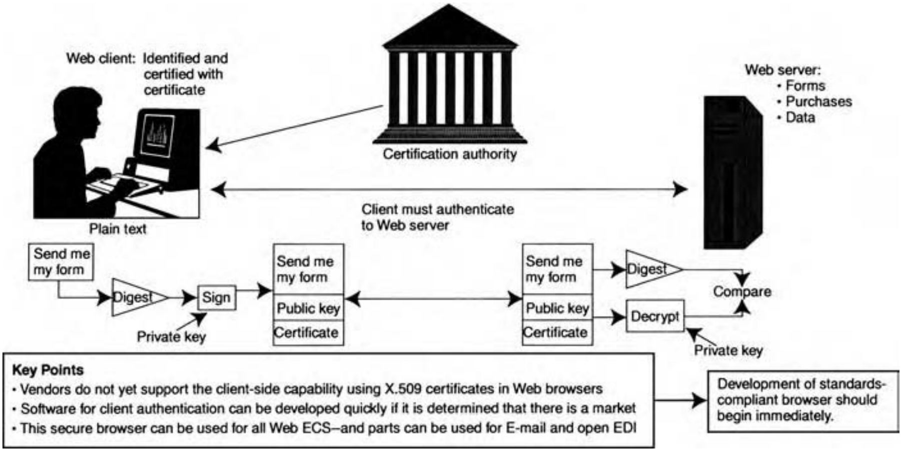

Web client: Identified and certified with certificate
Certification authority
Web server:
• Forms
• Purchases
• Data
Client must authenticate to Web server
Plain text
Send me my form
Digest
Sign
Private key
Send me my form
Public key
Certificate
Send me my form
Public key
Certificate
Digest
Decrypt
Compare
Private key
Key Points
• Vendors do not yet support the client-side capability using X.509 certificates in Web browsers
• Software for client authentication can be developed quickly if it is determined that there is a market
• This secure browser can be used for all Web ECS–and parts can be used for E-mail and open EDI
Development of standards-compliant browser should begin immediately.

Exhibit 2. Client Authentication

- The originator sends the message to the server attaching his or her public key and certificate to the message if necessary.
- The server either requests the originator's public key from a certificate/key repository or extracts the certification from the originator's message.

With this service, the authentication authority could either attach an authentication message verifying the digital signature's authenticity to the originator's message or provide that authentication to the recipient via a publicly accessible database. Upon receipt, the recipient would either acknowledge the originator's authenticity via the attached authentication message or access the public key and certificate from the publicly accessible database to read the signature.

To provide such levels of assurance, the certification authority must establish proofing stations where individuals and organizations can present themselves with appropriate identification and apply for certificates. The authority must also maintain or be part of a legal framework of protection and be in a position to mount an enforcement process to protect customers against fraud.

Certificate Repository

The certificate authority also provides the vehicle for the distribution of public keys. Thus the certificate authority would have to maintain the public key certificates in a directory server that can be accessed by authorized persons and computers.

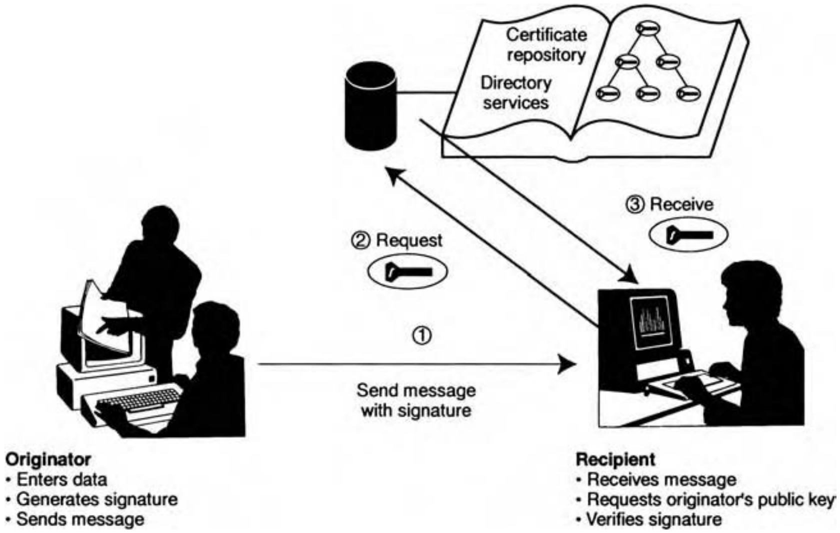


Exhibit 3. Certificate Repository

Exhibit 3 shows how subscribers might use such a repository. Certificates could be retrieved on demand along with their current status. Additional information, such as e-mail addresses or fax numbers, could also be available on demand.

The repository would work as follows:

- The message originator creates a message, generates a digital signature, and sends the message.
- The recipient sends a signed message requesting the originator's public key from the certificate repository.
- The certificate repository verifies the requester's signature and returns the public key to the recipient.

The certificate authority could also use the certificate repository to maintain a certificate revocation list, which provides notification of certificates that are revoked pursuant to a suspected compromise of the private key. This service could also require that the authority report such compromises via a compromised key list to special customers — possibly those enrolled in a subscribed service — and that such notifications be made available to all customers.

Finally, transactions involving certificates issued by other certificate authorities require that a cross-certification record be maintained and made publicly available in the certificate repository.

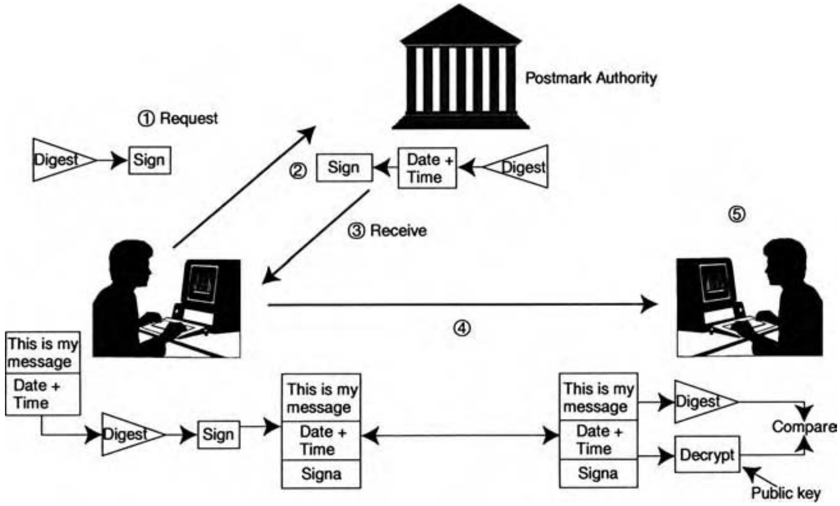


Exhibit 4. Electronic Postmark

Electronic Postmark

A service providing an electronic date and time postmark establishes the existence of a message at a specific point in time. By digitally signing the postmark, the postmarking authority assures the communicating parties that the message was sent, was in transit, or received at the indicated time.

This service is most useful when the recipient requires the originator to send a message by a specified deadline. The originator would request the postmark authority to postmark the message. The authority would receive a digest of the message, add a date and time token to it, digitally sign the package, and send it back to the originator, who would forward the complete package (i.e., signed digest, time stamp, and original message) to the recipient as shown in Exhibit 4.

Electronic postmarking functions as follows:

1. The originator sends a request to the postmark authority to postmark a message or document (i.e., a digital digest of the message or document).
2. The postmark authority adds date and time to the message received and affixes its digital signature to the entire package.
3. The postmark authority sends the package back to the originator.
4. The originator sends the original message or document plus the postmarked package to the recipient.
5. The recipient verifies the postmark authority signature with the authority's public key and reads the message or document.

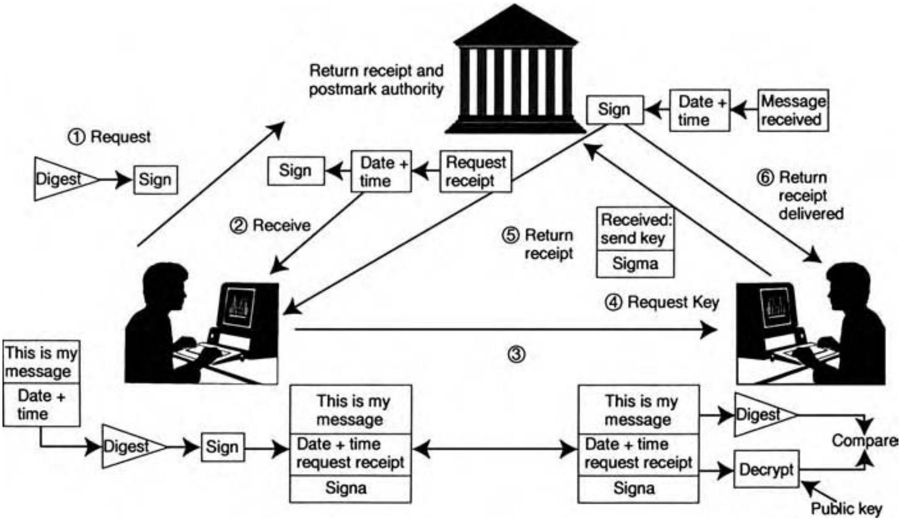


Exhibit 5. Return Receipt

Return Receipts

This service reports one of three events: that a message has transited the network, that it has been received at the recipient's mailbox, or that the recipient has actually decoded and opened the message at a specific date and time. In the latter instance, the transaction delivered to the recipient that has been encrypted might be set up only to be decrypted with a special one-time key, as shown in Exhibit 5. This one-time key could be provided by the postmark authority upon receipt of an acknowledgment from the recipient accompanied by the recipient's digital signature.

Here is how return receipt might work:

1. The originator sends a message digest to the return receipt and postmark authority (the authority) with a request for a postmark and return receipt.
2. The authority receives the message digest, adds date and time, encrypts the result, attaches a message to the recipient to request the decryption key from the authority upon receipt of the message, and affixes its digital signature to the package.
3. The authority returns the postmarked, receipted package to the originator, who sends it to the recipient.
4. The recipient receives the message package and makes a signed request for the decryption key from the authority.

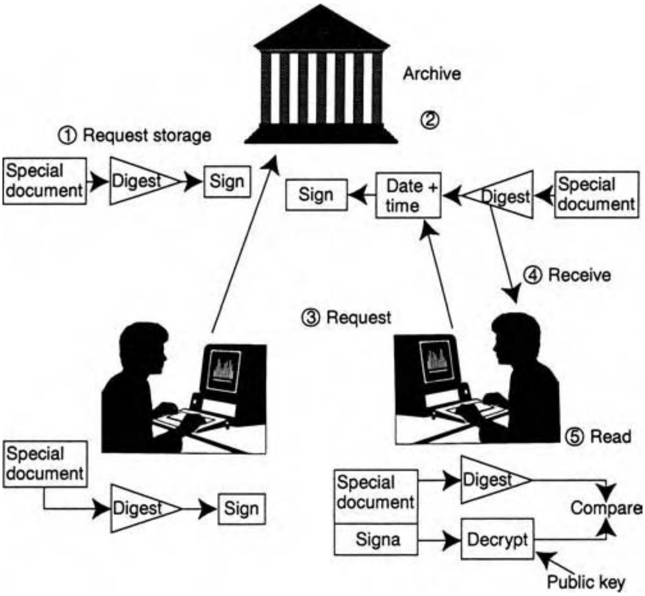


Exhibit 6. Storage and Retrieval

5. The authority receives the recipient's request, verifies the recipient's digital signature, and sends the decryption key to the recipient, who then decrypts and reads the message.
6. The authority simultaneously forwards the return receipt to the originator.

Storage and Retrieval Services

These services include transaction archiving where copies of transactions are held for specified periods of time, as illustrated in Exhibit 6. The service might also include information (i.e., documents, videos, or business transactions) that can be sealed, postmarked, and held in public storage to be retrieved via any authorized access. Likewise, encrypted information (i.e., documents, videos, or business transactions) can be sealed, postmarked, and further encrypted and held in sealed storage for indefinite periods of time. Each of these storage and retrieval capabilities must carry legal standing and the stamp of authenticity required for electronic correspondents.

Storage and retrieval works as follows:

1. The originator sends a request to the archive to archive a document or message for a specified period of time and designates this information as publicly retrievable.

2. The archive adds date and time to the message, verifies the identity of the originator, affixes a digital signature to the package, and archives the package.
3. A customer requests the document from the archive.
4. The archive retrieves the document, adds a date and time stamp to the package, affixes another digital signature to the new package, and sends it to the recipient.
5. The recipient verifies the first and second archive signatures and reads the message.

USE OF COMMERCIAL EXCHANGE SERVICES

E-commerce services may be used in one of three ways:

1. The originator sends a message to the authority with a request for service, the authority provides the service and returns the message to the originator, and the originator then forwards the message to the recipient.
2. The originator sends a message to a value-added network, which then forwards the message to the authority with a request for services. The authority provides the service and returns the message to the value added network, which then forwards the message to the recipient.
3. The originator sends a message to the authority with a request for service and the address of the recipient. The authority then forwards the message directly to the recipient.

All these services could be provided by a single authority, by a hierarchy of authorities, or by a network of authorities, each specializing in one or more of these services.

AVAILABLE TECHNOLOGIES FOR ELECTRONIC COMMERCE

Currently, three major technologies are capable of providing electronic commerce services — e-mail, the World Wide Web, and open EDI. As is typical of advanced technologies, security elements are the last to be developed and yet are essential if these technologies are to be deemed trustworthy for electronic commerce.

The issues of confidentiality, confirmation of identity, time certainty, and legal protection apply to all these technologies. The solutions — certification, key repositories, postmarking, return receipts, and storage and retrieval — are equally applicable to each of these technologies. Although the state of universality and interoperability varies among these technologies, they are all in a relative state of immaturity.

Secure E-Mail

Electronic messaging's most classic manifestation is e-mail. Because of its capacity for handling attachments, e-mail can be used to transfer official business, financial, technical, and a variety of multimedia forms.

DMS and PEM. Both the Department of Defense standard for e-mail, which is based on the ITU's X.400 standard for e-mail (called the Defense Message System or DMS), and the Internet e-mail standard, the SMTP, have made provisions for security. The DMS uses encapsulation techniques at several security levels to encrypt and sign e-mail messages. The security standard for the Internet is called Privacy Enhanced Mail (PEM). Both methods rely on a certificate hierarchy and known and trusted infrastructure. Neither method is fully developed.

Secure World Wide Web

The phenomenal growth of the Web makes it a prime candidate for the dissemination of forms and documents. Organizations view the Web as a prime tool for services such as delivery of applications and requests for information. However, Web technology has two competing types of security: one at the application layer that secures HTTP-formatted data (known as SHTTP), and one at the socket layer that encrypts data in the format in which it is transported across the network.

In addition, vendors do not yet support either client-side authentication or the use of X.509 certificates. Although software for such activities as client authentication can be developed relatively quickly, vendors have to be convinced that there is a real market for such products. This technology is about to emerge and, although it will emerge first to support Web applications, it will also speed the development of e-mail and EDI security services.

Secure Open EDI

Until now, EDI has been used in closed, value-added networks where security and integrity can be closely controlled. Signing and encryption have been proprietary to the EDI product in use or to the value-added EDI network provider.

By contrast, open EDI, running across open networks, requires adherence to the standards that are still being developed and a yet-to-be developed infrastructure that can ensure trusted keys. To date, the various schemes to accelerate the use of open systems for EDI have not captured the imagination of EDI users and providers.

THE OVERRIDING ISSUE: A PUBLIC KEY CERTIFICATE INFRASTRUCTURE

The suite of services and technologies described in this chapter depend on trusted public keys and their bindings to users. Users could be completely assured of the integrity of keys and their bindings if they were exchanged manually. Because business is conducted on a national and international scale, users have to be assured of the integrity of the registration authority and the key repository in an inevitably complex, electronic way.

One as-yet-unresolved issue is whether such an authority or authorities should be centralized and hierarchical or distributed. The centralized, hierarchical scheme would mean that certification authorities (and purveyors of the accompanying services) would be certified by a higher authority that, in turn, might be certified by yet a higher authority — and so on to the root authority. This kind certification would create a known chain of trust from the highest to the closest certification authority. This scheme is often referred to as the Public Key Infrastructure (PKI).

The alternative assumes that the market will foster the creation of a variety of specialized certification authorities to serve communities of interest. A complicated method of cross-referencing and maintaining those cross-references in the certificate repository for each community of interest would then develop.

The outcome of this debate is likely to result in a combination of both methods, such as several hierarchies with some kind of managed cross-referencing to enable public key exchanges between disparate communities of interest when required. Some of the issues yet to be resolved include:

- Agreement on the exact contents of certificates
- Definition of the size of prime numbers used in key generation
- Establishment of the qualifications required for obtaining a certificate
- Definition of the identification and authentication requirements for certificate registration
- Ruling on the frequency with which certificates are renewed
- Agreement on the legal standing and precedence for such technology

CONCLUSION

Groups such as the Internet Engineering Task Force, the Federal Government Public Key Infrastructure users group, and even the American Bar Association are tackling the knotty issues discussed in this chapter.

Toolkits are now available that allow the user to become his or her own certificate authority, thus enabling everyone to get into the act. Private companies such as VeriSign are establishing themselves as certification authorities so that users can give their public keys and certificates credence. The National Security Agency wants to become the certificate authority for the U.S. federal government. The U.S. Postal Service is intent on offering electronic commerce services to businesses and residences by acting as the certificate authority and provider.

An infrastructure will emerge, and it will probably work for users in a way very similar to the one that was described in this chapter.

ABOUT THE AUTHOR

David Litwack is president of dml Associates in Fairfax, VA.

Chapter 33
Evaluating and Selecting E-Commerce Software Solutions

Duane E. Sharp

Although businesses have used EDI (electronic data interchange) for years, the growth of Internet-based E-commerce presents new challenges. This chapter outlines the primary challenges and introduces several vendors' solutions.

The value and extent of commerce on the World Wide Web — E-commerce — are increasing dramatically, and the systems and software that make business transactions possible are proliferating. The Internet is proving to be an important element in many business operations, and in future years will become a major business resource for both buyers and sellers across a broad spectrum of business sectors. E-commerce covers an entire spectrum of business activities, ranging from merchandising and marketing of products and services, to electronic data interchange (EDI), involving electronic payment systems and order management.

According to a report published by the eMarketer — *The 1998 eCommerce Report* — consumers will buy 14 times more goods in 2002 than they did in 1997, for a total estimated value of $26 billion. The report points out that the world of E-commerce is growing into a rapidly evolving market that will "turn business as we know it upside down," particularly since the business-to-business segment of E-commerce is outpacing the consumer sector of E-commerce worldwide by a significant factor.

In fact, business-to-business E-commerce will account for the majority of Web-based revenues through 2002, which is estimated by one industry

0-8493-1161-6/02/$0.00+$1.50

analyst to reach $268 billion by 2002. Fortune 500 companies continue to dominate the online world and the top 10 percent of E-commerce businesses among these companies will account for the majority of Internet business volumes.

The eMarketer report also notes that companies can gain an economic advantage in conducting business online, because the cost of reaching additional customers is dramatically reduced once companies are linked to the Internet. The Web will provide opportunities for new value chains, distribution systems, and pricing structures with the capability for sales and customer support, maintaining contact with suppliers and partners via extranets, as well as communication and coordination resources.

With these statistics as background, it is important for information systems professionals to be able to evaluate and select the systems that will maximize the benefits to be achieved through E-commerce. The focus of this analysis is business-to-business E-commerce.

One of the trading concepts that is a catalyst for the rapid growth of business-to-business E-commerce is the “trading hub” concept. Trading hubs are venues that bring together thousands of buyers and sellers on a global basis, to trade freely with virtually perfect information flow on price, product, distribution, and delivery terms.

While many business sectors will undoubtedly adopt E-commerce sooner or later, those sectors with a strong stake in E-commerce, and which may already have experience with certain elements — such as electronic funds transfer — are moving quickly to E-commerce. One survey of chief executives indicated that the financial sector will see the most change, with over 70 percent of companies in this sector estimated to have adopted some form of E-commerce in the past two years. Banks are at the center of E-commerce and have a vested interest in it; and while some financial institutions still do not trust the Internet, many of them are satisfied with the security systems now available and are actively promoting E-commerce transactions.

CHALLENGES TO E-BUSINESS

In the primary area of E-commerce under consideration in this chapter — business-to-business — online transactions have been used by many businesses for several years. This has been the result of prevailing market forces and the increasing use of technology in business transactions. Several years ago, major retailers began requiring suppliers to adopt EDI, which allowed the suppliers to access online inventories through private networks, check current status of items, automatically replenish inventories, and receive payment directly to their bank accounts.

Probably the three biggest challenges for organizations that want to implement Internet-based e-business are security, privacy, and universal access. Every organization needs to properly present and position its company and its products and services. Also of significant concern to companies are the security of transactions, protection from viruses, and the protection of business systems from hackers. Access to services offered needs to be provided on as wide a basis as possible.

Several core systems need to be implemented within a network architecture to conduct business on the Internet, and to satisfy an organization's concerns about the primary aspects of E-commerce. These include the following major application categories, which will be reviewed in this chapter:

- Designing and maintaining online storefront Web sites
- Providing transaction security

DESIGNING AND MAINTAINING ONLINE STOREFRONT WEB SITES

The first category — designing and maintaining online storefronts — involves developing graphic concepts for the storefront Web site and laying out the site in the way a storefront designer would do for a traditional retail environment, using the wide range of graphics available with computer displays. The following products are a sampling of storefront design software available from several vendors.

IBM

Net.Commerce from IBM, first released in early 1997, offers tools for building and maintaining online storefronts. The second version (2.0) of this software provided the capability to host multiple storefronts, each with its own URL (universal remote location), on the same server. Along with improved scalability, this version is built on open standards and offers considerable flexibility in operating platforms, allowing users to move from single to multiple Windows NT, AIX, Sun, Solaris, AS/400, and System/390.

Net.Commerce lets end users browse, save, query, and order items in an interactive catalog, and the Net.Commerce Administrator tool lets Web storefront administrators create and manage online product templates.

Oracle

Oracle Internet Commerce Server (ICS) is designed for companies setting up their own Web storefronts, and runs only on Oracle databases. The original version of this software was priced at $20,000 per processor and included the enterprise edition of Oracle 7 database software and the advanced edition of Web Application Server 3.0. Later versions of Internet Commerce Server offered a $5000 product to existing Oracle customers. Like other vendors, Oracle will bring together its business-to-business product and its consumer product in a single commerce platform, written in Java. Among the features of ICS are customizable templates for creating catalog pages, order processing, and open interfaces for handing transactions off to back-end systems. Various versions are available, running on Windows NT and UNIX.

Several plug-in cartridges are available for Oracle's ICS, including Verifone and Cybercash for payments, Taxware for tax calculation, TanData for shipping and handling, and Portland Software's secure packaging for delivering software online.

SpaceWorks

OrderManager provides VARS with a flexible self-service ordering utility for the Internet, a company intranet, an extranet, or a combination of all three. This product has the potential to reduce training cycles and clogged telephone lines. It is targeted at the business-to-business marketplace, primarily to manufacturers or wholesale distributors that sell to other businesses. SpaceWorks' in-house staff creates sample E-commerce applications that link customers' legacy systems to catalogs and process on the Web sites hosted by SpaceWorks' service bureau.

TRANSACTION SECURITY

This aspect of E-commerce is one of the most contentious of technical issues concerning business on the Internet. The guaranteed security of business transactions is a vital component in the acceptance and success of conducting electronic business, whether it is an individual consumer buying a product or a company conducting a range of confidential business transactions.

Information that is transmitted via the Internet must be kept confidential, whether it is credit card information, bank account numbers, or other sensitive data that an organization wishes to protect in the conduct of its business operations.

To accomplish the required level of security, software developers use several encryption techniques, each of which has advantages and disadvantages. Encryption scrambles the original message to make it incomprehensible to those who do not have the "key" to decrypt or decipher it.

There are two basic types of encryption systems: public key and private key. Private key uses the same key to encrypt and decrypt information, and is not secure across networks. Public key uses two different but related keys for encryption and decryption — a public key and a private key — enabling much higher security across networks.

Public key encryption takes much longer than private key encryption — ten times longer to provide similar security. Because of this public key characteristic, major Internet browsers, such as Netscape Navigator and Microsoft Internet Explorer, use a hybrid of public key encryption. Navigator uses secure sockets layer (SSL) cryptography, while Explorer uses both SSL and private communications technology (PCT) cryptography.

The issue of security on the Internet is one of the domains of the Internet Engineering Task Force (IETF), a body responsible for Internet standards. The IETF is supporting a new protocol similar to SSL, called transport security layer (TLS), which provides host-to-host security across the Internet.

Other security initiatives are being developed by financial institutions, including major credit card organizations such as Visa and Mastercard. The most popular is a standard called secure electronic transaction (SET), a technique that provides higher levels of confidentiality and authentication than either public or private key, and is being adopted by these organizations and financial institutions for consumer purchases. While credit card companies have been quick to realize and adopt this technology, it is still being tested throughout North America.

SET adds another level of security to a heavy-duty encryption system for transaction messages. All parties in a SET transaction have a digital certificate — a numerical ID. The purchaser's certificate is stored on a hard drive, and when a purchase is made by credit card, the ID is transmitted along with the transaction. While SET has been slow to catch on with vendors and ISPs using other security technologies, it offers some enhanced security features that will benefit those organizations adopting it.

Software for Transaction Security

There are several vendor organizations with security software for the Internet, based on accepted or projected standards, such as SET. The following section describes the products of some of these market leaders.

Entrust Technologies. A subsidiary of Nortel, this company has developed software security solutions using public key technology for corporate networks, intranets, and the Internet. Security products developed and marketed by the company include Entrust, Entrust/Lite and

Entrust/WebCA, as well as the Entrust/Toolkit line of application programming interfaces. The company also has released an SET product, Entrust/CommerceCA, that provides a secure method for transmitting financial information over unsecured networks.

Verifone Inc. This company, a division of Hewlett-Packard, develops and markets merchant software (vPOS) using the SET security protocol. This product handles highly encrypted, SET-based credit card messages, and the digital certificate that provides authentication of the consumer, as well as the merchant, and then transmits the message to the financial institution.

Certicom Corp. This company uses an encryption technology called elliptic curve, a public key system that uses less bit space than conventional cryptography, with advantages of speed, bandwidth efficiency, and increased storage. This technology is used in Motorola's CipherNet security software, enabling rapid integration of security features into software applications.

Security software is a rapidly evolving area, as software developers mix and match various encryption technologies and other security techniques, and strategic partnerships continue to be forged among developers of security software and vendor organizations with E-commerce systems and products.

RATING E-COMMERCE VENDORS

One of the innovative evaluation services offered to companies that want to move into E-commerce is WebTrust. Developed jointly by the Canadian Institute of Chartered Accountants (CICA) and the American Institute of Certified Public Accountants (AICPA), WebTrust is designed to make it easier for both businesses and consumers to evaluate Web sites and to use E-commerce.

Chartered accountants who have received WebTrust training audit Web sites to assess whether they meet WebTrust criteria for good electronic commerce practices and security. If a business entity meets the criteria, it can display the WebTrust logo — a sign that will reduce the consumer's concerns about entering into a transaction with the site — on the E-commerce home page.

WebTrust addresses three major E-commerce areas:

1. Business practices disclosure: confirming that the entity discloses its business practices for E-commerce transactions and executes them in accordance with these practices, which include:
   a. Descriptive information about the nature of the goods that will be shipped or the services that will be provided where customers

can obtain warranty, service, and support related to the goods and services purchased on its Web site

b. Information to enable customers to file claims, ask questions, and register complaints

2. Transaction integrity: maintaining effective controls and practices to ensure that E-commerce orders are completed and billed as agreed, including:

a. Controls to ensure that each order is checked for accuracy and completeness, and that acknowledgment is received from the customer before the order is processed

b. Ensuring correct goods are shipped in the correct quantities in the timeframe agreed, that back orders and other exceptions are communicated to the customer, and prices and other costs are displayed before requesting acknowledgment of the order

c. Orders are billed and electronically settled as agreed, and errors are promptly corrected

To meet the WebTrust criteria summarized above, the entity would likely have a combination of automated and manual control procedures in place. The impact of wizards and built-in tools also needs to be addressed.

3. Information protection: implementing and maintaining effective controls and practices to ensure that private customer information is protected from uses that are not related to the E-commerce transaction, which include:

a. The protection of private customer information, such as credit card number and other personal or corporate confidential information, during transmission over the Internet and while it is stored in its E-commerce system

b. The business entity's access to the customer's computer

c. Protection of the customer's computer files

To meet these criteria, the business entity being evaluated would utilize an acceptable encryption protocol and have a firewall in place. In addition, employees or contractors with access to the system would be governed by security policies and tools.

The initial WebTrust audit would generally address a two- to three-month period. In order to maintain the logo, an update would be conducted at least every three months.

IT professionals can help their organizations succeed in E-commerce by promoting the adoption of certification programs such as WebTrust and by facilitating the implementation of the technical infrastructure and controls that will help ensure that the businesses conducting electronic commerce meet the criteria.

SUMMARY

The Internet has opened up a whole new era of business interconnection, allowing many more companies to take advantage of the efficiencies of network commerce. By evaluating and selecting the software solutions that are best configured for the organization's business requirements — both for electronic presentation of products and/or services and for security of transaction — the IS professional can ensure a viable, secure E-commerce environment.

A GLOSSARY OF TERMS FOR E-COMMERCE SOFTWARE

cookies: Snippets of information delivered from a Web site to the user's (client's) browser, and then stored on the hard drive. The information can be something like the time of one's last visit or the pages one downloaded. Cookies can be read by that Web site the next time one visits.

digital certificates: Digital IDs used to present credentials online. Digital certificates are issued by companies that act as "trusted third parties." In the SET (secure electronic transaction) protocol, the buyer, the merchant, and banks for these parties all have digital certificates.

EDI: An acronym for electronic data interchange. EDI provides electronic formats that allow for an exchange of business data between companies over networks.

digital wallet: Software that stays resident on the hard drive of an online shopper. When the shopper is ready to make a purchase, the wallet pops open to reveal payment options. Some wallets hold credit cards with encrypted information; other wallets hold digital coins.

firewall: A network firewall is a security system that controls access to a protected network. Firewalls are often used by organizations that want to connect to the Internet without compromising the security of proprietary systems and data.

PCT encryption: Private communications technology, an encryption method developed by Microsoft and available on its Internet Explorer 3.0 and 4.0 browsers. Similar to SSL, with a combination of public and private key encryption, it appears to have some streamlining of features that may make it more efficient than SSL, but it is not widely accepted.

RSA encryption: Based on a public key system, which means that every user has two digital keys — a public key to encrypt information and a private key to decrypt. Authentication of both sender and recipient is provided.

SET: Secure electronic transaction protocol is a means for authenticating credit card purchases on the Net. Digital signatures are used by all parties. Transaction information is encrypted using 1024-bit RSA encryption.

SSL encryption: Secure sockets layer, developed by Netscape to provide data encryption and authentication of servers or clients. It can be used for any function on the Internet.

shopping cart: A piece of software that operates on an online storefront. The shopping cart keeps track of all the items that a buyer wants to purchase, allowing the shopper to pay for the entire order at once.

TLS encryption: Transport layers security is a protocol for Internet host-to-host security proposed by the Internet Engineering Task Force.

ABOUT THE AUTHOR

Duane E. Sharp is president of SharpTech Associates, a Canadian company specializing in the communication of technology. An electronic engineer with more than 25 years of experience in the IT field, he has authored numerous articles on technology and a textbook on interactive computer terminals, and chaired sessions at Comdex Canada. He can be reached at desharp@netcom.ca.

Section VI
The E-Enabled Toolkit

The chapters in this section focus on the tools that should be part of a toolkit for building business solutions for the Internet. This includes such standards such as the Extensible Markup Language (XML), C^{++}, and Java.

XML, in particular, allows developers to both describe and build structured data in a format that facilitates seamless data exchange between Internet-based applications. This tool is designed specifically for the Internet and for Web-to-Web communication.

XML was developed by the World Wide Web Consortium (W3C) body, starting in 1996. Much of the work was done by the XML Working Group (XWG), which was established by W3C. XML is a subset of the Standard Generalized Markup Language (SGML) that was defined in the ISO standard 8879:1986.

"Selecting Hardware and Operating System Software for E-Commerce" (Chapter 34) describes some of the tools that are available for building E-commerce solutions.

"XML — Rosetta Stone for Data" (Chapter 35) explains the substantial impact XML is expected to have on how information is communicated between disparate applications both within and outside the organization. The Rosetta Stone, discovered in Egypt in the late eighteenth century, became a key to understanding Egyptian writing. The author expects XML, a metalanguage, to play a similar role in the new IT environment.

"XML: A Common Data Exchange Language for Businesses" (Chapter 36) discusses building business solutions and exchanging data between applications with this development standard.

"Reengineering Information Systems with XML" (Chapter 37) examines how valuable legacy information can be extracted and enriched using XML and making it more accessible through hypertext.

"Linux and the Web" (Chapter 38) presents an overview of Linux from an architectural and design perspective. An in-depth look at the operating system is followed by a review of cost-free applications that are available to enable the setup of an E-commerce infrastructure.

"Java and Its Supporting Technologies" (Chapter 39) explores Java and the host of supporting technologies that complement this programming language.

"Java and C^{++}: Similarities, Differences, and Performance" (Chapter 40) explains the important features of Java, compares execution performance to study the overhead of interpretation of the virtual machine instructions, and discusses the Java Programming Language in the context of the C^{++}.

"JavaBeans and Java Enterprise Server Platform" (Chapter 41) discusses both JavaBeans and the Java Enterprise Server platform as E-commerce development environments. This chapter explains how to leverage these tools in a "write once, run anywhere" capacity.

Chapter 34
Selecting Hardware and Operating System Software for E-Commerce

Duane E. Sharp

The growth of the Internet as a viable business vehicle is one of the phenomena of modern technology and has already had a significant impact on the business community, providing new methods of conducting business on a global basis.

The Internet has been described as a "powerful, but elusive new sales channel" for global business because it has the capability to reduce transaction costs, expand markets, improve customer service, and enable unique one-to-one marketing opportunities. It is elusive because start-up costs remain high and product offerings are often not integrated with existing corporate business systems. However, this situation is changing as vendors of hardware and software products are caught up in the business opportunities represented by Internet E-commerce.

The infrastructure required to conduct E-commerce on the Web already exists in many businesses, where applications such as EDI have been using business-to-business networks for some years. These organizations recognize the significant cost savings and convenience of online business-to-business transactions. For suppliers, online commerce can shrink the cost of doing business, help target customers more effectively, and help retain them.

0-8493-1161-6/02/$0.00+$1.50

EDI has gradually shifted to extranets — networks that open up sections of an enterprise's intranet to suppliers and vendors. Organizations in a variety of business sectors — automotive, aerospace, etc. — that use this technology are able to handle spare parts orders efficiently, with significant savings in person-hours, as well as reduced communication and paper-handling costs, all of which can lead to enhanced profitability.

The Web has become particularly useful for commodity products — computers, software, and electronic products, to name a few. For example, Cisco Systems, a major player in networking products, reports that its revenue from Internet sales increased dramatically, from just 5 percent to 33 percent of its annual sales, since it began selling computer networking products from its Web site in 1996.

As Deloitte & Touche pointed out in its "1998 Annual Report on The Software Industry," the impact of the changes that are now taking place in electronic commerce will be dramatic. New and streamlined supply chains will develop, especially in industries such as manufacturing, where supply chain management is particularly crucial. Labor-intensive tasks like issuing and following up on purchase orders will require fewer people and less paper, as business moves ever closer to the long-awaited paperless society.

It is important for IT managers to recognize the trends and shifts in technology that will occur over the next few years, to enable them to effectively implement E-commerce systems. This chapter provides a basis for evaluating some of the current hardware products and operating system architectures offered by vendors for E-commerce applications. Trends in hardware developments at the microchip level are discussed, as well as major vendor hardware products that have been optimized to make them efficient in E-commerce environments.

HARDWARE FOR E-COMMERCE

Changing Chips

Developments in microchip technology are having a significant impact on the architectures that vendors are providing to implement E-commerce applications, as well as placing new demands on software. High-performance RISC (Reduced Instruction Set Computing) and CISC (Complex Instruction Set Computing) chips have reached their limits after supplanting Pentium-based chips in large enterprisewide networks. A new chip, the Merced, being jointly developed by Intel and Hewlett-Packard, may emerge as the chip of the future. Because of these enhancements in chip technology, some industry analysts claim that the typical lead that hardware has on software may double in the next year.

Merced is a 64-bit chip, a hybrid of RISC and CISC technology, named IA-64 (Intel Architecture 64-bit), and will deliver the best of both RISC and CISC features. This chip will combine RISC-chip performance with X86 compatibility, allowing it to run programs designed for X86 and Pentium computers. Because Merced is still under development, its characteristics will need to be verified under real-world application environments. However, Intel's market leadership in chip design and its manufacturing reputation have already caught the interest and commitment of major hardware vendors, such as Sun, which plans to port its Polaris operating system to support the Merced chip.

Mainframes

Coincidental with new developments in chip technology, mainframes are reappearing as a computing force in a new generation of products that could become important elements in the network server world and in E-commerce applications. Some business environments that have retained and upgraded their mainframe platforms are already reaping the benefits these computers can bring to network environments.

With their fast response times and capability to process increasingly complex database transactions — a significant requirement in E-commerce environments — these reborn relics from the past are natural platforms for high-volume transaction processing. Software products are now available that can link front-end Windows NT and UNIX servers to legacy systems and databases residing on mainframes. Security improvements are also being developed to meet the requirements of the Internet and extranets, and software products will increase the acceptance of mainframes as enterprise servers suitable for Internet applications.

As well as being suitable platforms for intranets and for Web servers, where high volumes of transactions are involved, they are also scalable and provide fast response time for dynamic Web sites and complex database queries. Although security of transactions on the mainframe was an early concern, mainframe vendors are embedding encryption technology into mainframe chips, and some vendors even include a secure Internet firewall in their mainframe operating systems.

Vendor Server Products

For E-commerce applications, the hardware on which these applications run is an important part of the solution mix, and a few major vendors have adapted their standard hardware offerings to meet the specialized requirements of servers for E-commerce applications. In addition, these vendors have developed core applications and adapted their operating systems to enable E-commerce applications to be run efficiently. Some of the organizations that have already entered the market with hardware products adapted for E-commerce applications are IBM, Hewlett-Packard (HP), and Sun Microsystems. Other hardware vendors, such as Compaq — with its newly acquired Digital Equipment Corp. and Silicon Graphics — are not far behind.

The focus of this chapter is on those vendors that have launched E-commerce products and have a customer base for these products. Several of these vendors, specifically IBM, HP, and Sun Microsystems, have configured their standard hardware offerings for E-commerce, with bundled software and networking interface capability, as well as other features, such as security, designed to make E-commerce applications run efficiently. The following descriptions of the features of these products provide some points of evaluation for organizations entering the E-commerce arena.

IBM. IBM has a range of E-commerce servers, from mainframes to client/server products, configured to provide optimum platforms for E-commerce applications. These products are scalable to meet the requirements of small to large businesses, and run a variety of IBM E-commerce software products — Lotus Domino, Java, and Net.Commerce.

In the mainframe category, IBM offers the S/390, with its operating system, OS/390. The capability to handle thousands of users running E-commerce applications, and to integrate existing databases with the new breed of E-commerce applications, makes this product ideal for large organizations that have significant investments in mainframe computers and legacy systems.

The latest generation of mainframes provides improved technology in small cabinet physical configurations, using CMOS (complementary metal oxide semiconductor) chips. Lower production costs for these processors resulted in a dramatic drop in the average cost per MIPS, from $93,000 in 1990 to $7500 in 1997, and the per-MIPS cost is forecast to move even lower as production economies increase.

The drawbacks for mainframes in today's network server environment are the same as they were in the heyday of the mainframe: software costs and upgrades are expensive, largely because software vendors persist in charging mainframe license fees on the basis of processor size rather than on usage, a more logical cost model for this hardware category.

Cost of ownership for the new mainframes is about the same or less than for UNIX or Windows NT-based distributed systems, partly due to economies of scale available with mainframes, but also as a result of the greater administrative costs of client/server systems.

Moving down the hardware scale to smaller servers, IBM has three primary offerings, in order of size and capability: the RS/6000, the AS/400e, and Netfinity, each one configured for E-commerce applications such as Lotus Domino and Net.Commerce.

In addition to its server products, IBM has "circled" the E-commerce product market with other products — software, communications servers, networking products, storage systems, and printers — all of which add to its product mix and capability to provide complete E-commerce solutions.

HP. Hewlett-Packard's primary E-commerce offering is Domain Commerce, a bundled, scalable software solution for E-commerce applications, running on HP NetServer products, and designed to provide a consistent level of service, handling transaction traffic and user priorities.

Included in the Domain Commerce platform, which is software centric, is a range of software to determine user and service classes; manage peak-stage windows; enable customers to centrally manage systems, network, and E-commerce applications, commerce gateway and POS; and provide network security and advanced graphics capability.

Domain Commerce includes the following functional modules:

- **HP ServiceControl:** providing server overload protection and customer and transaction prioritization
- **Domain Management:** enabling management of the system, network, and E-commerce applications from a standard browser
- **VeriFone vPOS:** advanced, secure, point-of-sale for Internet payment transactions
- **OpenPix:** imaging software for communications and businesses that want to increase Web-based transactions through image-rich Web sites
- **Netscape Enterprise Server:** an enterprise Web server for business applications, providing Web publishing and document management capabilities

Options available with Domain Commerce include storefront software, robust transaction engines, and global business systems, as well as a range of other E-commerce products — high-availability servers, encryption accelerators, and enterprise firewalls — available from HP partners.

Domain Commerce will initially be available for the HP UNIX operating system (HP-UX), but will be ported to Windows NT.

Sun Microsystems. Sun has focused on the development of applications in the Java language, which it introduced to the industry a few years ago, and has several E-commerce applications running under Solaris, its 32-bit, UNIX-based operating system. Sun has placed considerable importance on the E-commerce market, and has over 300 electronic commerce solutions providers.

Sun's server product lines — the Netra and Ultra families — are scalable and are designed to support transactions from single users to thousands of users. Security products for these platforms are provided by Sun software partners, configured to meet specific enterprise requirements for secure E-commerce.

Sun's solution providers offer a wide range of functionality to address all requirements for commerce-enabled enterprise, including Internet commerce merchant software, billing and payment systems, security, Internet EDI, search and navigation, profiling and usage analysis, information push, content/Web site creation and management, and document management and workflow. Sun's E-commerce solutions integrate elements from four major areas:

1. Scalable, high-performance platforms
2. Security
3. Java and Java Commerce
4. Partner programs and professional services

Java's 'Write Once, Run Anywhere' streamlines software development, and Java Commerce provides a complete Internet-based infrastructure for electronic commerce. It is an open platform that can support all standards and payment protocols running concurrently in the same environment.

OPERATING SYSTEM SOFTWARE FOR E-COMMERCE

The major hardware vendors referred to above offer their own operating systems, usually variants of UNIX, and application suites for E-commerce environments, either developed by the vendor itself or by software partners. However, major operating system vendors such as Microsoft and Novell, both of which have significant presence in networking environments, will also be active players in the E-commerce marketplace.

In the following paragraphs, the E-commerce strategies and operating system environments provided by these two industry leaders in networking and operating systems, are reviewed and compared.

Microsoft Windows NT

Microsoft owns a large percentage of the network market, and its Windows NT product is growing rapidly against competitors NetWare and UNIX. NT

Server had the fastest growth of worldwide software license shipments in 1997, with a 73 percent increase. It was expected to surpass UNIX shipments in 1999 and NetWare in the year 2000.

The importance of E-commerce to Microsoft is reflected in the company's establishment of *The Microsoft Internet Commerce Strategy* — a comprehensive mix of servers and tools for the E-commerce environment. This strategy has been implemented by integrating commerce functionality into the Microsoft® BackOffice™ family, to provide a commerce-enabled server back end, running on the Windows NT operating system, and based on three core software products:

1. *Microsoft Site Server, Enterprise Edition:* for the deployment and management of commerce-enabled Web sites (includes Commerce Server)
2. *Microsoft Internet Explorer:* a Web browser, for a commerce-enabled desktop
3. *Microsoft Wallet:* for secure, convenient purchasing, as part of the commerce-enabled desktop

Microsoft's E-commerce Internet strategy includes the integration of its own operating systems and commerce tools offered by its developer partners.

Novell

Unlike Microsoft, its primary competitor in networking operating systems, Novell has chosen to partner with developers, rather than develop its own strategy and core tools for E-commerce. In this context, Novell's network operating system (NetWare) can be viewed as an "umbrella" under which E-commerce applications developed by its partners will run. With its significant leadership position in the networking market and its large, global customer base, there will undoubtedly be many E-commerce environments operating under the NetWare operating system.

However, the position taken by Novell is quite different from the IBM, HP, Microsoft, and Sun Microsystems strategies, which provide those vendors with a significant degree of control over their approaches to E-commerce and the development and management of core products for this market.

As well, Novell has already had some rearrangement in its partnership relationships in core areas of E-commerce software, which tends to weaken its overall position in this market as it unfolds, reflecting a strategy that has not yet achieved stability.

CONCLUSION

E-commerce, operating under industry-standard operating systems and integrated with a range of new application software to handle business transactions on the Internet, is a rapidly growing and, as market analyses indicate, a significant thrust in global business activities. For the IT manager, there are a reasonable number of choices available to implement E-commerce, ranging from major vendor hardware and integrated operating system with vendor E-commerce software, and software vendor E-commerce suites and partner software with partner products in special function areas, to the integration of a customer-selected range of E-commerce applications running under an industry-standard network operating system.

ABOUT THE AUTHOR

Duane E. Sharp is president of SharpTech Associates, a Canadian company specializing in the communication of technology. An electronic engineer with more than 25 years of experience in the IT field, he has authored numerous articles on technology and a textbook on interactive computer terminals, and has chaired sessions at Comdex Canada. He can be reached at desharp@netcom.ca.

Chapter 35
XML — Rosetta Stone for Data

John van den Hoven

The eXtensible Markup Language (XML) is the universal format for structured documents and data. It is a method for putting structured data in a text file in order to share both the format and the data (making the data self-describing) on the World Wide Web, intranets, and elsewhere. The XML specification is defined by the World Wide Web Consortium (W3C; http://-www.w3.org).

Examining the component parts of the term "eXtensible Markup Language" can further enhance the definition of XML. A markup language is a system of symbols and rules to identify structures in a document. XML is extensible because the markup symbols are unlimited and self-defining. Thus, XML is a system of unlimited and self-defining symbols and rules, which is used to identify structures in a document.

DESCRIPTION

XML is a metalanguage (in the form of a set of rules) used to define other domain- or industry-specific languages that describe data. To construct its own XML language (or vocabulary), an enterprise creates a Document Type Definition (DTD), which is a set of rules that define what tags appear in a specific document type or, in database terms, the schema. Just as the data definition language (DDL) is used to declare data elements and their relationships in a database, so too does the DTD in XML.

A DTD in XML consists of tags (sometimes referred to as elements) and attributes. Tags (words bracketed by '<' and '>') and' attributes (in the form `name="value"`) are used to delimit pieces of data but the interpretation of the data is done by the application that reads it. It can be used to describe the contents of a very wide range of file types — including Web pages, spreadsheets, database files, address books, and graphics — to a very detailed level.

0-8493-1161-6/02/$0.00+$1.50

HISTORY AND CONTEXT

Hypertext Markup Language (HTML) was created in 1990 and is now widely used on the World Wide Web as a fixed language that tells a browser how to display data. XML is a simple, very flexible text format that was created in 1996 (and became a W3C standard in 1998) to solve HTML's shortcomings in handling very large documents. XML is more complex than HTML because it is a metalanguage used to create markup languages, whereas HTML is one of the languages that can be expressed using XML.

While HTML can be expressed using XML, XML itself is a streamlined, Web-compatible version of Standard Generalized Markup Language (SGML), which was developed in the early 1980s as the international standard metalanguage for markup. SGML became an International Organization for Standardization (ISO) standard in 1986. It can be said that XML provides 80 percent of the benefit of SGML with 20 percent of the effort.

ROSETTA STONE

The impact of XML can be compared to that of the Rosetta stone. The Rosetta stone was discovered in Egypt in the late eighteenth century; it was inscribed with ancient Egyptian hieroglyphics and a translation of them in Greek. The stone proved to be the key to understanding Egyptian writing. It represents the "translation" of "silent" symbols into a living language, which is necessary to make these symbols meaningful.

The interfaces used in today's enterprises have become the modern form of hieroglyphics. XML promises to play a similar role to that of the Rosetta stone by enabling a better understanding of these modern hieroglyphics and in making the content of the data in these interfaces understandable to many more systems.

VALUE OF XML

XML is a simple yet powerful format for representing data in a neutral format that is independent of the database, the application, and the underlying system. XML has many potential uses, including electronic publishing, improving Web searches, and as a data-exchange mechanism.

Electronic Publishing

XML provides a standardized format that separates information content from presentation, allowing publishers of information to "write once and publish everywhere." XML data defines the structure and content, and then a stylesheet is applied to it to define the presentation. These stylesheets are defined using the eXtensible Stylesheet Language (XSL) associated with XML to format the content automatically for various users and devices. Through the use of XML and XSL, documents can be explicitly tailored to the requirements of information users. This can be done by providing customized views into information to make the information content richer, easier to use, and more useful.

XML and XSL can also be used by information publishers to publish information content in multiple formats and languages to multiple devices. XML is a generic source format that allows different stylesheets to be applied to put the content into an endless range of printed and electronic forms. These forms can be delivered through a wide variety of devices such as Web browsers, televisions, receivers, personal digital assistants (e.g., the Palm Pilot), digital cell phones, and pagers.

Web Searching

XML makes it easier to search and combine information from within the enterprise and from the World Wide Web. XML does this because documents that include metadata — data about data — are more easily searched because the metadata can be used to pinpoint the information required. As a result, XML makes retrieval of documents and data much faster and more accurate than it is now. The need for better searching capabilities is especially evident to those searching for information on the Internet today.

Data Exchange

Any individual, group of individuals, or enterprise that wants to share information in a consistent way can use XML. The information can be in the form of documents or data. XML enables automated data exchange without requiring substantial custom programming. It is far more efficient than e-mail, fax, phone, and customized interface methods that most enterprises use today to work with their customers, suppliers, and partners. XML simplifies data exchange within and between enterprises by eliminating these cumbersome and error-prone methods.

Within an enterprise, XML can be used to exchange data and documents between individuals, departments, and the applications supporting these individuals and departments. It can also be used to derive greater value from legacy applications and data sources by making the data in these applications easier to access, share, and exchange. This is especially important to facilitate (1) data warehousing to allow access to the large volumes of legacy data in enterprises today; (2) electronic commerce applications which must work with existing applications and their data formats; and (3) Web access to legacy data.

One of the most important forms of potential benefit XML offers is to enable enterprises with different information systems to communicate with one another. XML can facilitate the exchange of data across enterprise boundaries to support business-to-business (B2B) communications. XML-based Electronic Data Interchange (EDI) standards (XML is complementary to EDI because EDI data can travel inside XML) are extending the use of EDI to smaller enterprises because of XML's greater flexibility and ease of implementation. XML is transforming data exchange within industries XML is used to define platform-independent protocols for the exchange of data.

FUTURE DEVELOPMENTS IN XML

XML is undergoing rapid innovation and experiencing wide adoption. Greater utilization of XML, the addition of XML capabilities to database management systems, and greater standardization are some of the key future areas of development in XML.

Use of XML

XML is likely to become the de facto standard for integrating data and content during the next few years, because XML offers many advantages as a mechanism for flexible data storage and data exchange and as a means of bridging different technologies such as object models, communications protocols, and programming languages. Since XML allows internationalized media-independent electronic publishing, it will eventually replace HTML as the language of choice for designing Web sites.

XML will also be increasingly used to target this media-independent content to new delivery channels. The Wireless Markup Language (WML) is an example of an XML-based language that delivers content to devices such as smart cell phones and personal digital assistants. XML will do for data and documents what Java has done for programs; it will make the data both platform-independent and vendor-independent. Therefore, XML will be used on an ever-increasing range of devices in the future.

XML in the DBMS

Most of the critical business data in enterprises today is managed by database management systems (DBMS). In order to realize the promise of XML as an enabler for exchanging business data, the DBMS must provide full support for XML. One of the ways this will happen is that XML will be used to extend existing data access standards such as Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), and ActiveX Data Objects/Object Linking and Embedding Database (ADO/OLE DB) by Web-enabling these standards and providing a common data format for persistence.

XML will also be supported directly by the DBMS. With the growing importance and awareness of the benefits of XML, key DBMS vendors such as IBM, Informix, Microsoft, Oracle, and Sybase are adding support for XML to their database products. This support will include the ability to publish database information (including catalog information or data from the database) as XML documents, to write XML documents to database tables, to query XML data using SQL syntax, and to store XML documents (such as Web pages) in the database.

The trend is to bring together and manage unstructured data (such as documents, text, and multimedia) in much the same way that structured data is managed by the DBMS today. XML will play a key role in allowing this to happen.

XML Standards

XML standards are in the form of technology standards and vocabulary standards. The XML 1.0 specification provides the technology foundation by providing a syntax for adding metadata to text data. Other technology standards will be developed and deployed to extend the value of XML. These include XML Name-spaces, which specifies the context in which XML tag names should be interpreted; XML Schema, which allows XML structures to be defined within XML itself; eXtensible Style Sheets (XSL), which includes XML formatting objects to describe how a document should be displayed or printed along with XSL Transformation for converting XML data from one XML structure to another or for converting XML to HTML; and Document Object Model (DOM), which is a standard set of function calls for manipulating XML and HTML files from a programming language.

The other area of XML standards is vocabulary standards. XML was designed to let enterprises develop their own markup languages for specific purposes, and that is now happening in abundance. As a result, there is much work still to be done in terms of standardizing the vocabularies within and between industries. These vocabulary standards for XML are being created at a rapid pace as vendors and users try to establish standards for their own enterprises, for industries, or as general-purpose standards.

The finance, technology, and health care industries are areas in which much XML vocabulary development is taking place. In finance, the Financial Products Markup Language (FpML) and Financial Information Exchange Markup Language (FIXML) are being established as standards for exchanging contracts and information about transactions. In technology, the RosettaNet project is creating a standardized vocabulary and defining an industry framework to define how XML documents and data are assembled and exchanged. In healthcare, the Health Level 7 (HL7)

Committee is creating a standard document format, based on XML, for exchanging patient information between healthcare organizations such as hospitals, labs, and healthcare practitioners.

Common XML vocabularies for conducting business-to-business commerce are also being established through Microsoft's BizTalk, CommerceNet's eCo Framework, OASIS's ebXML, and through other initiatives. Electronic Data Interchange standards bodies, vertical industry consortia, and working groups such as the Open Applications Group will also contribute to the development of common XML vocabularies.

CONCLUSION

XML is rapidly becoming a key technology for deciphering the modern hieroglyphics of nonstandard, customized interfaces for exchanging data. Weinberg's Law states that "if builders built buildings the way computer programmers write programs, the first woodpecker that came along would have destroyed all civilization." Weinberg's Law seems applicable to most of the fragile approaches for sharing and exchanging data in today's enterprises.

XML is the Rosetta stone for data because it is a universal and open standard that enables different enterprises using different applications, terminologies, and technologies to share and exchange data and documents. XML is a key integrating mechanism between the enterprise and its customers, suppliers, and partners. It provides a standard foundation for exchanging data that promises to reduce the cost, complexity, and effort required in integrating data within enterprises and between enterprises. This is becoming more and more essential in a global electronic economy where an enterprise must be linked to an ever-expanding chain for business-to-business exchange.

Chapter 36
XML: A Common Data Exchange Language for Businesses

Charles Dow

The Extensible Markup Language (XML), first published on February 10, 1998, is a standard for structured documents and data on the Web. It is a product of the W3C (World Wide Web Consortium), an international industry group made up of about 470 organizations and run by the MIT Laboratory for Computer Science in the United States. The standard has its roots in work done decades ago and has come about due to contributions from many sources. The latest recommendation is XML 1.0 Second Edition. It is a vendor-neutral, license-free, platform-independent standard.

WHY XML?

The question should really be "Why now?" The late 1990s saw a whirl of IT activity with the Internet being the driving force for XML and many other IT-related technologies.

HTML swiftly became the common language of Web pages — information that can be viewed and printed. But the need to go beyond those simple goals was pressing hard, including the need for easy exchange of data over the Internet. XML was born. It seems determined to become the business language of the Internet with many dialects to serve different business needs.

The Fundamental Problem

Imagine going to a foreign country or to a meeting where no one shared a common language. How do you communicate? Usually with great difficulty,

0-8493-1161-6/02/$0.00+$1.50

using a version of sign language. Unfortunately, computers do not have that option.

Any application or program that needs to communicate needs to receive and/or send data or information — information that the receiver can massage, manipulate, or otherwise process. The first information that comes to mind is financial information, but it could also be health-related information, library information, or EDI messages.

For applications to communicate there is the need to establish a common format that everyone agrees to live by. This seems simple enough in theory, but not in practice. For instance, recently my QA group discovered an interest rate of 20.01 percent. Odd? Yes, but only until you realize that the year is 2001. One of the applications that we outsourced had changed a format without informing us.

What if you need to coordinate your development with hundreds of other companies? What about when you do not know the companies? The answer is applications need to be loosely coupled so development can proceed without risking the "breaking" of them. By establishing an interface that can be interrogated and parsed, we solve most of those problems.

Even today most Web sites provide information that is only intended to be read or printed. This impacts search engines — as anyone soon learns who tries to find something specific, e.g., a source for parts for a York Concorde rowing machine — because there is no way to search for product names or something similar. A search produces an enormous amount of information that you have to discard on your way to finding what you are looking for. Search engines try to reduce that burden but they are only partly successful.

For those who are trying to aggregate information, e.g., price information from different sources using the HTML content that is sent to a browser, it is a near-impossible challenge. The information sent has some structure that can be used, but your interpretation of it is not guaranteed to be error free. Entire businesses have arisen to provide software to "scrape" information from the HTML pages.

A Business Driver

EDI was the early way of doing eBusiness. But it was and still is relatively expensive. The Internet coupled with XML has opened a new and cheaper way. "However, XML's current dark secret is that it's slower than EDI. The messages must be larger — as much as ten times larger — requiring greater bandwidth and more cycles to move and process."[1] Current thinking is that translation from one to the other will allow XML and EDI to integrate in the short run.

Accuracy

Like a secret that is told from one person to another, after four or five people have repeated it, the wording of the secret is often no longer identical to what started the chain of events. With XML everyone can work with the original data without constantly reinterpreting it as it is passed from one program to another. There are none of the problems of data-type mismatching where a program is sending 24 digits and the receiver uses 12, etc.

A SOLUTION

XML seems determined to be the solution. In my opinion, the XML standard is following the "make it work, then make it fast" philosophy. Now that it has been shown to work, compression techniques (and bandwidth) will tackle the speed issue.

Make It Work

At the birth of the Web, there was the initial problem of the need to share and link documents over the Web. Standard Generalized Markup Language (SGML), an ISO standard (ISO 8879) based on work that started in the 1960s, had been devised and standardized by 1986 to essentially solve the problem of technical documentation.

Using a simplified version named HTML the problem of sending over the Web was solved. The solution was so straightforward and well suited to the need that it was an immediate success. However, most of the tags were designed to format the information to look good, e.g., bold, font size, and underlined. It was not intended to allow you to process the data beyond displaying it.

Make It Better

XML can now be viewed as the "make it better" step. Soon after the popularity of the Internet had been established, it was viewed as the plumbing for the interconnection of all the world applications. Firms like Nortel talked about Web tone.

XML provides a way to add additional structure and rules to the content. Once you have structure, you can manipulate it and process it — distributed computing at its finest.

XML used the best of SGML and by design focused on data interchange, not just the publishing of documents. Of course, the former also allows the latter to function very elegantly. It was an elegant compromise that built on a lesson that was learned from the HTML success, "that simplicity and flexibility beat optimization and power in a world where connectivity is key."[2]

XML allows you to select only what you need and, more importantly, validate that the source has been received intact.

XML[3] has additional features that HTML lacks:

- *Extensibility.* XML, unlike HTML, allows users to specify their own tags and attributes.
- *Structure.* XML supports the specification of deep, nested structures required to represent database schemas and other hierarchies.
- *Validation.* The XML specification allows rigor, allowing applications to validate the data they read.
- *Unicode.* XML also embraced Unicode, which allows any or all languages to be published in the same document, making the first W in WWW truly meaningful.

XML now frees us from the page metaphor of HTML. Now a browser or other application can receive quite a bit of information allowing local processing. Then the user can filter, sort, or reorganize and display selected data without needing to request additional information from the source. Did you ever try to select a flight or a series of flights?

THE BENEFITS

The benefits of structuring the content sent over the Internet are obvious; they include the following:

- Search engines could focus on desired attributes of information only, filtering out the unneeded stuff.
- Web sites could present views that are appropriate to the devices used to access them.
- Information could be manipulated intelligently.
- The Web becomes a huge distributed database of goods, services, and information — no longer just a huge library of articles.
- Applications can be loosely coupled, able to interoperate.

XML EXPOSED

If you have ever used HTML or any derivative of SGML you would be familiar with tags and attributes. Like HTML, XML makes use of *tags* (words bracketed by "<" and ">") and *attributes* (of the form name = "value"). HTML specifies what each tag and attribute means, usually including how to display the text in a browser. XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data to the application that processes it; in this case the browser is viewed as only one application.

Tags

Tags are words (labels) bracketed by "<" and ">."

Exhibit 1. Sample XML Code

```
<ARTICLE>
<HEADLINE>Fredrick the Great meets Bach</HEADLINE>
<AUTHOR>Johann Nikolaus Forkel</AUTHOR>
<PARA>
One evening, just as he was getting his
<INSTRUMENT>flute</INSTRUMENT> ready and his
musicians were assembled, an officer brought him a list of
the strangers who had arrived.
</PARA>
</ARTICLE>
```

Attributes

Attributes take the form name = “value,” i.e., name, value pairs.

A Sample of XML

Exhibit 1 shows sample XML code.[4]

Observations?

Some observations about XML include the following:

- A text file, no binary[5]
- Readable by a human
- Verbose
- Platform-independent by design

These are by design. The verbose quality both of the tags and the text format was for simplicity.

THE PROPER NAME

An XML document is the proper name when the XML has been constructed correctly. The data, especially when nested, looks like a collection of trees. The use of the word “document” really indicates that the data contained in the XML can be represented as a document.

The XML standard is a set of rules, conventions, and guidelines.

WHERE THE ACTION IS

Vocabularies level the playing field. HTML works well because there is a small set of tags where the meaning is known. For now, let us ignore the DHTML wars.

XML vocabularies try to solve that problem for other areas. Vocabularies are designed to allow multiple users to easily share data and interpret data about a subject. Normally an application, or a class of device, e.g., a PDA, would use a vocabulary.

For just about every need, there seems to be a group that is working on a vocabulary. Do you know what is occurring in your industry? Vocabularies are documented in a Document-Type Definition (DTD). See the section on DTDs later in this chapter. Some examples of vocabularies follow.

HTML

Although HTML preceded XML, it has a published DTD. If you switch to seeing the source of an HTML page in your browser, you will see something such as:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
```

WML

Wireless Markup Language (WML), part of the WAP specification, is a language for describing wireless-specific content.

ADEX

Ad Exchange (ADEX), the Newspaper Association of America (NAA) Classified Advertising Standards Task Force, has created an XML vocabulary for classified ads.

SMIL

Synchronized Multimedia Integration Language (SMIL) is used for multimedia with precise timing, e.g., sound and video.

ebXML

Electronic Business XML (ebXML) includes specifications for messaging services, collaborative partner agreements, and registry and repository. It was designed to be a global standard for creating E-commerce applications.

XML BASICS

This section of the document will review some of the main XML terms that a reader will come across. There are other more esoteric terms that an interested reader may want to research and get familiar with.

The Parser

No understanding of XML would be possible without recognizing the key role of the parser, a.k.a. the XML processor program. The parser understands XML's strict rules, conventions, and guidelines. The parser is used to read XML documents and provide access to their content and structure.

Exhibit 2. XML for a Simple Memo

```
<!DOCTYPE memo [
<!ELEMENT memo(to, from, date, subject?, para+) >
<!ELEMENT para(#PCDATA) >
<!ELEMENT to(#PCDATA) >
<!ELEMENT from(#PCDATA) >
<!ELEMENT date(#PCDATA) >
<!ELEMENT subject(#PCDATA) >
]>
```

Document-Type Declaration

Think of the document-type declaration as the container for the rules. It could either have all the rules contained within [and] brackets, similar to the example that follows, or reference an external file, similar to the HTML example in an earlier paragraph. The rules are found in the DTD.

Document-Type Definition

The DTD keeps the specific rules for that vocabulary.

Example. For a simple memo the XML DOCTYPE and DTD might take the form shown in Exhibit 2.[6]

Without going into too much detail, there are a few pieces of information that you need to make sense of the above. An element can be described as a logical grouping. The question mark after "subject?" means that element is optional. The plus sign after the "para+" means that at least one paragraph is required. And if you really want to know, "#PCDATA" means parsed character data, i.e., text with no tags that are unknown.

Schemas

Think of them as DTDs on steroids. DTDs tell you what is valid syntax. But it does not specify what the content is allowed to be. DTDs expect everything to be text. Schemas, on the other hand, specify what the content is allowed to be, so you can perform data validation, e.g., timeDuration, decimal, and binary.

Well-Formed

This essentially means that the document contains one or more elements, delimited by start- and end-tags that nest properly within each other.

Valid

A document is valid if it conforms to a document-type declaration. The benefit of DTDs and Schemas is efficiency and clarity. The user of the XML knows exactly what is allowed, and with Schemas we can validate the content.

The DOM

We are at the point where XML is beginning to serve two masters:

1. The developers of Web documents, which allow Web authors to have a consistent approach to dynamic content
2. Programmers who want their applications to interoperate

The Document Object Model (DOM) is a platform- and language-neutral interface. It specifies a standard set of interfaces that allow dynamic access and updating of the content, structure, and style of XML and HTML documents. The specification has different levels and optional modules (and a method for interrogating what is supported).

The DOM provides developers with a common programming model across a variety of languages and applications, allowing consistent use of different vendors' tools and browsers using the standard DOM interfaces.

It provides a relatively complete object-oriented (OO) programming interface. And like any good OO interface, it allows you to hide the internal details, so the programmer is free to optimize, tune, and tweak without consulting the users of his interfaces.

With the DOM,[7] programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the DOM, with a few exceptions.

The DOM does not replace COM or CORBA. Both DOM implementations and applications may be implemented using systems based on COM or CORBA.

DOM Implementation. This is a piece of software that wraps the XML or HTML document and provides the DOM interface. Browsers typically perform this service.

DOM Application. This is any application that uses the DOM implementation. Scripts, e.g., JavaScript or VBScript, and Java programs are some examples of DOM applications.

MORE XML

XLink

XML Linking Language (XLink) describes a standard way to add XML elements that perform a function similar to hyperlinks in a HTML file.

XPath

XML Path Language (XPath) is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer. It got its name from its use of a path notation similar to URLs for navigating through the hierarchical structure of an XML document.

XPointer

XPointer is a standard way to describe an URL, i.e., a reference to a piece of data inside an XML file. It is designed to allow you to get to a specific point in the XML.

XSLT

XSL Transformations (XSLT) is a language for transforming XML documents into other XML documents. The transformation is achieved by associating patterns with templates.

XSLT was expected to be used with XSL but was designed and is capable of being used independently of XSL. Outside of XSL, it is useful for rearranging, adding, or deleting tags and attributes. Transformations expressed in XSLT are called stylesheets.

XSLT makes use of the expression language defined by XPath for selecting elements for processing, for conditional processing, and for generating text.

XSL

XSL, which is a stylesheet language for XML, specifies an XML vocabulary for specifying formatting. The W3C[8] specifically points out that CSS (cascading style sheets), the style sheet language, which is as applicable to XML as it is to HTML, is simpler; thus, use it when you can — for everything else use XSL.

Note: In practice, you would transform the XML document using XSLT and then use CSS to display the document.

XML Namespaces

XML Namespaces is a specification that describes how you can associate a URL with every single tag and attribute in an XML document. What that URL is used for is up to the application that reads it.

ASSOCIATED STANDARDS

Associated standards are:

- Unicode and ISO/IEC 10646 for characters
- Internet RFC 1766 for language identification tags

- ISO 639 for language name codes
- ISO 3166 for country name codes

CONCLUSION

This chapter examined XML as a suite of language tools that support standardized communication between business applications. XML is a natural extension of earlier Internet languages (like HTML), and one that allows transaction-based processing over the Web. It is also the start of a new class of tools that will continue this process.

Notes

1. Dejesus, Edmund X., EDI? XML? Or Both? January 08, 2001. http://www.computerworld.com/cwi/story/0,1199,NAV47_STO55904,00.html
2. Microsoft's Vision for XML. http://xml.coverpages.org/bosworthXML98.html
3. Bosak, Jon, XML, Java, and the Future of the Web. Sun Microsystems. Last revised 1997.03.10, http://www.ibiblio.org/pub/sun-info/standards/xml/why/xmlapps.htm
4. Source: http://www.w3.org/TR/REC-CSS2/intro.html#q2
5. Binary data can be included if you really desire.
6. Bryan, Martin, An Introduction to the Extensible Markup Language (XML), The SGML Centre. http://www.personal.u-net.com/~sgml/xmlintro.htm
7. Robie, Jonathan, Ed., What Is the Document Object Model? Texcel Research. http://www.w3.org/TR/REC-DOM-Level-1/introduction.html
8. http://www.w3.org/Style/CSS-vs-XSL

References

1. Extensible Markup Language (XML). http://www.w3.org/XML/
2. Kayl, Kammie, Ahead of Its Time: ebXML on Track for an Early Delivery. http://www.javasoft.com/features/2000/12/ebxml.html?frontpage-headlinesfeatures
3. Electronic Business XML. http://www.ebxml.org/
4. Bosak, Jon and Bray, Tim, Scientific American: XML and the Second-Generation Web. http://www.sciam.com/1999/0599issue/0599bosak.html
5. Standard Generalized Markup Language (SGML): Overview and New Developments. http://www.nlc-bnc.ca/publications/netnotes/notes3.htm

ABOUT THE AUTHOR

Charles Dow is vice president of Product Engineering–Banking Systems, SLMsoft.com Inc.

Chapter 37
Reengineering Information Systems with XML

Chao-Min Chiu

Hypertext functionality through the World Wide Web (WWW) can be provided for hypertext-unaware information systems (IS) with minimal or no changes to the IS. IS includes financial information systems, accounting information systems, expert systems, decision-support systems, etc. IS dynamically generate their contents and thus require some *mapping mechanism* to automatically map the generated contents to hypertext constructs (nodes, links, and link markers) instead of hypertext links being hard-coded over static contents.

No systematic approach exists, however, for building mapping rules to supplement an IS with hypertext support through the WWW. Mapping rules infer useful links that give users more direct access to the primary functionality of IS, give access to metainformation about IS objects, and enable annotation and ad hoc links. This chapter offers a set of guidelines for analyzing IS and building mapping rules.

HTML is for displaying instead of describing data and only supports a fixed and limited tagset.[1] Extensible Markup Language (XML) addresses the limitations of HTML. XML allows the development of custom and domain-specific elements and attributes. When reengineering applications or IS for the Web, XML seems to be a more potential approach than HTML because it offers extensible, human-readable, machine-readable, semantic, structural, and custom markup. This chapter gives one XML example to demonstrate the feasibility of using XML for representing mapped information generated by information systems.

0-8493-1161-6/02/$0.00+$1.50

BENEFITS OF HYPERTEXT SUPPORT AND USING XML

What benefit do users gain from providing information systems with hypertext support? Users may find it difficult to understand and take advantage of the myriad interrelationships in an IS knowledge base (data, processes, calculated results, and reports). Hypertext helps by streamlining access to, and providing rich navigational features around, related information, thereby increasing user comprehension of information and its context.[2] Augmenting an IS with hypertext support results in new ways to view and manage the information system's knowledge by navigating among items of interest and annotating with comments and relationships (links).[3]

Many resources of information systems have been made available to users through the WWW. However, those implementations do not meet our definition of integration with IS since they do not infer useful links that give users more direct access to the primary functionality of IS, give access to metainformation about IS objects, and enable annotation and ad hoc links.

WWW is mainly used to browse predefined HTML documents. Dynamically generated information from databases and legacy systems can also be converted to HTML format and viewed in the Web browser. However, HTML has some limitations. It is for displaying instead of describing data.[4] HTML documents are machine-readable but not human-readable. HTML only supports a fixed and limited tagset.[1] Extensible Markup Language (XML) addresses the limitations of HTML. XML[5] is a subset of Standard Generalized Markup Language (SGML). XML aims to develop machine-readable, human-readable, structured, and semantic documents that can be delivered over the Internet.[6,7] An important feature of XML is that it is extensible.[8] XML allows the development of custom and domain-specific elements and attributes. XML also supports descriptive information about document contents, called metadata. The extensibility and flexibility of XML enable it to integrate information from disparate sources and heterogeneous systems.[4] When reengineering applications or IS for Web, XML seems to be a more potential approach than HTML because it offers extensible, human-readable, machine-readable, semantic, structural, and custom markup.

SYSTEM ARCHITECTURE

In this section, a framework with seven logical components is proposed. This architecture emphasizes the integration of WWW with IS, providing hypertext functionality to each IS. Note that actual architectures may implement the functionality in different modules than those shown here.

Because the WWW browser and server are normal WWW components, only the functionality of other components will be described.

An IS is an application system with which users interact to perform a certain task that dynamically produces output content for display. IS instances are things written within an application package, such as an individual worksheet or database.

An IS wrapper translates and routes messages between its IS and the WWW server. An IS wrapper also provides its IS mapping rules to infer links and nodes from outputs of its IS. An IS wrapper must map commands sent by the WWW browser to actual system commands and invoke its IS to execute actual IS commands. A comprehensive IS wrapper allows one to integrate an existing IS with few or no changes.

The master wrapper coordinates schema mapping and message passing among different IS domains, thus aiding IS-to-IS integration. It also provides the following functionality: (1) decodes attributes (e.g., object type, object ID, and command) that underlie a link anchor; (2) searches the knowledge base for commands (e.g., list fields) that access various relationships on the selected IS object based on the object type; (3) maps commands to link anchors; and (4) forms an XML document that includes mapped link anchors and sends the document to the WWW server. Note that the master wrapper can be implemented with CGI scripts, server-side Java, server APIs, ASP, etc.

A knowledge base stores commands for accessing various relationships on IS objects and information that cannot be accessed directly from ISs (e.g., relationships in E-R diagrams). A linkbase stores user-created annotations and ad hoc links.

Note that the WWW server will serve any IS application that has an appropriate wrapper. To integrate a new IS with the WWW, one has to build a wrapper and store information (e.g., commands) in the knowledge base (see Exhibit 1). Thus, to provide an IS application with hypertext support, the developer must declare mapping rules in the master wrapper and its IS wrapper. The basic concept underlying the proposal here is the use of *mapping rules* to automatically provide hypertext functionality to IS when integrating them with the WWW. Note that one set of mapping rules can serve all instances of an IS.

BUILDING MAPPING RULES

This section discusses the process of analyzing ISs and building mapping rules.[9] These rules would reside in and become invoked by the master wrapper or IS wrapper. We explain each by using a financial information system built using MS-Excel as the target IS. When providing large information

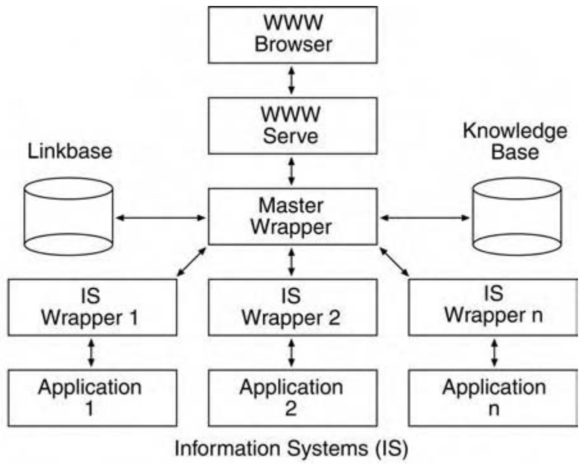


Exhibit 1. A Framework for Integrating IS into the WWW

systems with hypertext support, a facility that automatically infers useful links from dynamically generated information will be helpful. Mapping rules are such routines that convert dynamically generated information to hypertext constructs (nodes, links, and anchors). Mapping rules make extensive use of three features that Halasz[10] identified as outstanding issues in hypermedia research more than ten years ago, and that still have not been addressed in many hypermedia or WWW applications: (1) creating and manipulating virtual structures of hypermedia components; (2) computing over the knowledge base during link traversal; and (3) tailoring the hypermedia network.[11] Information systems in which component type or classes are easily recognized can benefit remarkably from the automatic link generation approach.[12] The process of building mapping rules is divided into four steps.

Step 1: Identify IS Objects

This step is to identify the data objects of interest. In a financial information system built using MS-Excel, objects include workbooks, worksheets, and cells.

Step 2: Identify Relationships among IS Objects

Bieber and Vitali[3] identify several types of relationships for system objects, including those to be discussed later. Identifying explicit and implicit relationships forces developers to consider which information users are interested in and then build mapping rules to access this information.

Each of the following relationships gives the user easy access to some aspect of an object.

- *Schema relationships:* access to the kind of domain-specific relationships in a schema or application design
- *Operational relationships:* direct access to information about IS objects using operational commands supported by the IS; in a financial information system built using MS-Excel, this includes Excel commands over specific objects
- *Structural relationships:* access to related objects based on the application's internal structure; in a financial information system built using MS-Excel, these include "contains" links among workbooks, worksheets, and cells
- *Metainformation relationships:* access to attributes of and descriptive information about IS objects; in a financial information system built using MS-Excel, these include its formula, explanation, cell type, etc.
- *Annotative relationships:* relationships declared by users instead of being inferred from the system structure; all users should be able to annotate objects even when without direct write access; a linkbase will store user-created comments

Sometimes, meaningful relationships cannot be accessed directly from IS, so developers must declare those relationships and store them in the knowledge base.

Step 3: Identify Commands for Accessing Relationships on IS Objects

Commands underlying the <Command> tags (see Exhibit 2) give users direct access to various relationships on IS objects. After identifying its implementation commands, one can then build a relationship mapping rules. Exhibit 2 lists some of these commands for worksheet and cell relationships. The displayed commands may be different from actual system commands. The IS wrapper should pass the actual system commands to its IS.

Step 4: Build Mapping Rules

The main purpose of mapping rules is to infer useful links from the output dynamically generated by an IS and commands for operating on IS objects. Note that one set of mapping rules can serve all instances of an IS. Two types of mapping rules can be identified: Command_Rule and Object_Rule. Bieber[11] and Wan[13] implemented mapping rules (or bridge laws) using Prolog. However, for clarity mapping rules are discussed here in terms of functional procedural calls.

Command_Rule (System, Type). This rule infers commands for operating upon the selected object. This rule should be included in the master

Exhibit 2. Commands for Operating upon Worksheets and Cells

Object	Relationships	Commands	Functionality
Worksheet	Operational	viewChart	Draw a chart based on values of the selected sheet
Worksheet	Structural	showData	Show contents of a sheet
Worksheet	Annotative	add	Create comments on a sheet
Worksheet	Annotative	view	View comments on a sheet
Worksheet	Operational	newSheet	Create a new Excel sheet
Cell	Operational	update	Change the value of a cell and do a what-if analysis
Cell	Metainformation	explain	Get the explanation or formula of a cell
Cell	Annotative	add	Create comments on the selected cell
Cell	Annotative	view	View comments on the selected cell
Cell	Metainformation	display	Display the formula (in Excel format) of a cell

wrapper (see Exhibit 2). The "system" parameter is used to discriminate among different information systems. This rule should provide the following functions:

- Search the knowledge database for commands for accessing various relationships on the selected IS object
- Map commands to command links
- Form an XML document that includes mapped links and send the document to the Web server

For example, Command_Rule ("FIS," "SHEET") will execute the aforementioned functions and create an XML document. To conserve space, just one command (showData) is listed here. In reality, the system would present all commands (or a filtered subset). Some available commands are listed in Exhibit 3.

The ID, "FIS,FL,EBIT-EPS," indicates the EBIT-EPS worksheet of the FL (Financial Leverage) workbook. The IS wrapper is an ASP application called "MasterWrapper." Note that the <Command> tag has a "Command" parameter whose value is "showData." This type of link anchor is called a command link anchor.

Object_Rule (Command, ID, Type). This rule sends the system command to the IS and infers useful information from outputs dynamically generated by the IS. This rule should be included in the IS wrapper (see Exhibit 1). This rule has four parameters. The object identifier (ID) is the key to determine which object of the given system the command should operate on. For example, the ID, "FIS,FL,EBIT-EPS,F16," means the "F16" cell of the "EBIT-EPS" worksheet; FL means the financial leverage workbook.

Exhibit 3. Commands for an EBIT Worksheet (Commands underlying the <Command> tags give users direct access to various relationships on IS objects)

```
<? xml version = “1.0” ?>
<?xml-stylesheet type="text/xsl" href="Commands.xsl"?>
<! DOCTYPE Commands SYSTEM “Commands.dtd”>
<Commands>
<Command href=”MasterWrapper.asp? ID=FIS,FL,EBIT-EPS&Type=Sheet&
Command=showData” Target="objFrame">showData</Command>
. . . .
</Command>
```

Exhibit 4. Tags That Represent Objects of an EBIT Worksheet

```
<? XML VERSION = “1.0” ?>
<?xml-stylesheet type="text/xsl" href=" FIS-EBIT-EPS.xsl"?>
<! DOCTYPE FIS-EBIT-EPS SYSTEM “FIS-EBIT-EPS.DTD”>
<FIS-EBIT-EPS>
<EBIT href="MasterWrapper.asp?ID=FIS,FL,EBIT-
  EPS,F16&Format=Formula&
Type=Cell&Command=No" Target="cmdFrame">213000</EBIT>
. . . .
</ FIS-EBIT-EPS >
```

This rule should provide the following functions:

- Map commands to actual IS commands
- Send actual commands and other parameters to the IS
- Receive display output from the IS
- Infer object links from the output generated by the IS
- Create the XML document with inferred links and send the document to the Web server

Exhibit 4 shows an example in which the command accesses a structural relationship (from step 2). Object_Rule (“showData”, “FIS,FL,EBIT-EPS,F16”, “Cell”) will execute the five aforementioned functions and send the following XML document to the Web server. To conserve space, only the cell called “EBIT” is listed.

When a user clicks on the tag, the “MasterWrapper.asp” application will be invoked and then the command rule will be called to infer available commands for this object. Note that the <EBIT> tag has a “Command” parameter whose value is “No.” This type of link anchor is called an object link anchor.

IMPLEMENTATION

A proof-of-concept prototype has been created using a financial information system as the target IS. The financial information system was built using Microsoft Excel. The masterwrapper is an ASP program (i.e., MasterWrapper.asp). The IS wrapper has two parts: an ASP program (i.e., FISWrapper.asp) and a DLL (i.e., FISWrapper.DLL) built using Visual Basic. The following discussion shows a document type definition (DTD) that defines tags for marking up information (i.e., EBIT/EPS analysis) generated by the financial information system; explains how to use the XSL technology of IE 5.0 to display the EBIT/EPS XML document; explains how to program Command_Rule and Object_Rule using flowcharts (see Exhibits 9 and 10); and presents some outputs from the prototype.

Document Type Definition

A DTD describes the structure and grammar of a class of documents. Exhibit 5 shows a DTD that contains a set of elements and attributes for representing an EBIT-EPS analysis (see Exhibit 6).

Exhibit 7 lists a XML document that is created based on the EBIT/EPS DTD (see Exhibit 5). This document is generated by the Object_Rule of IS wrapper (i.e., FISWrapper.DLL). The following briefly describes two XML elements in Exhibit 7.

- <?xml version="1.0"?>
  A XML document must begin with this statement.
- <?xml-stylesheet type="text/xsl" href="FIS-EBIT-EPS.xsl"?>
  This statement specifies the URL of the XSL file that is responsible for displaying the FIS-EBIT-EPS document.

Extensible Style Language (XSL)

The World Wide Web Consortium (W3C) first published the XSL Working Draft in August 1998 and the proposed Recommendation for XSL 1.0 in May 1999. Internet Explorer 5.0 supports a subset of the W3C Extensible Stylesheet Language (second XSL Working Draft).[14] The prototype uses the XSL technology of IE 5.0 to display XML documents. Exhibit 6 lists the XSL file for displaying the EBIT/EPS document (see Exhibit 5). Some XSL elements are briefly described in Exhibit 6.

- <?xml version="1.0"?>
  An XSL file is also a XML document, so it must begin with this statement.
- <xsl:stylesheet>
  This tag means that the document is an XSL file.

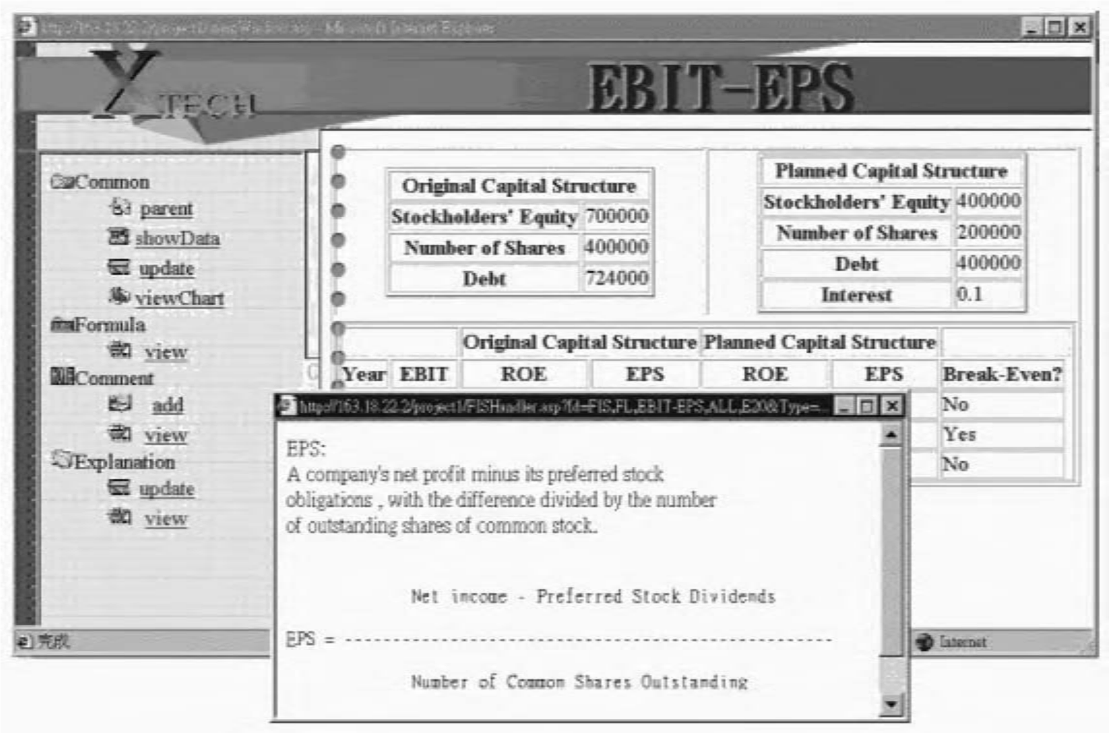
Exhibit 5. The DTD for Representing a Class of EBIT/EPS Document

```
<!- - FIS-EBIT-EPS : The top-level element - ->
<! ELEMENT FIS-EBIT-EPS        (Original-Capital-Structure, Planned-Capital-Structure,
                                 EBIT-EPS+) >
<! ELEMENT Original-Capital-Structure   (Stockholders-Equity, Shares, Debt) | (ROE, EPS)>
<! ELEMENT Stockholders-Equity   (#PCDATA)>
<!- - Shares element: stands for number of shares. - ->
<! ELEMENT Shares                  (#PCDATA)>
<! ELEMENT Debt                (#PCDATA)>
<! ELEMENT Planned-Capital-Structure   (Stockholders-Equity, Shares, Debt, Interest) |
                               (ROE, EPS)>
<! ELEMENT Interest            (#PCDATA)
<!- - EBIT-EPS element: stands for earnings before interest and tax and earnings per share. - ->
<! ELEMENT EBIT-EPS                (Year, EBIT, Original-Capital-Structure,

                               Planned-Capital-Structure, Break-Even)>

<! ELEMENT Year                (#PCDATA)>

<!- - EBIT element: stands for earnings before interest and tax. - ->

<! ELEMENT EBIT                (#PCDATA)>

<!- - ROE element: stands for return on equity. - ->

<! ELEMENT ROE                 (#PCDATA)>

<!- - EPS element: stands for earnings per share. - ->

<! ELEMENT EPS                 (#PCDATA)>

<! ELEMENT Break-Even          (#PCDATA)>

<! ATTLIST Stockholders-Equity

href           CDATA        # REQUIRED

Target         CDATA        # REQUIRED >

<! ATTLIST Shares

href           CDATA        # REQUIRED

Target         CDATA        # REQUIRED >

<! ATTLIST Debt

href           CDATA        # REQUIRED

Target         CDATA        # REQUIRED >
```

Exhibit 5. The DTD for Representing a Class of EBIT/EPS Document (Continued)

```
<! ATTLIST Interest

href          CDATA        # REQUIRED

target        CDATA        # REQUIRED >

<! ATTLIST Year

href          CDATA        # REQUIRED

Target        CDATA        # REQUIRED >

<! ATTLIST EBIT

href          CDATA        # REQUIRED

Target        CDATA        # REQUIRED >

<! ATTLIST ROE

href          CDATA        # REQUIRED

Target        CDATA        # REQUIRED >

<! ATTLIST EPS

href          CDATA        # REQUIRED

Target        CDATA        # REQUIRED >

<! ATTLIST Bbeak-Even

href          CDATA        # REQUIRED

Target        CDATA        # REQUIRED >
```

- <xsl:template match="/">
  This element indicates that this template corresponds to the root (/) of the XML document. A template defines how XML elements should be displayed.
- <xsl:apply-templates />
  This element selects all the children of current element and asks XSL processor to find and apply an appropriate template.



Original Capital Structure	
Stockholders' Equity	700000
Number of Shares	400000
Debt	724000

Planned Capital Structure	
Stockholders' Equity	400000
Number of Shares	200000
Debt	400000
Interest	0.1

		Original Capital Structure		Planned Capital Structure		
Year	EBIT	ROE	EPS	ROE	EPS	Break-Even?
						No
						Yes
						No

http://163.18.22.2/project1/FISHandler.asp?Id=FIS,FL,EBIT-EPS,ALL,E20&Type=...

EPS:
A company's net profit minus its preferred stock obligations , with the difference divided by the number of outstanding shares of common stock.

```
            Net income - Preferred Stock Dividends
EPS = ---------------------------------------------------
            Number of Common Shares Outstanding
```

Exhibit 6. The Interface of the Prototype

Exhibit 7. An XMl Document Created Based on the EBIT/EPS DTD

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="FIS-EBIT-EPS.xsl"?>
<!DOCTYPE FIS-EBIT-EPS SYSTEM "FIE-EBIT-EPS.dtd">
<FIS-EBIT-EPS>
<Original-Capital-Structure>
<Stockholders-Equity href="MasterWrapper.asp?ID=FIS,FL,EBIT-
  EPS,C16&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">781000</Stockholders-Equity>
<Shares href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,C17&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">400000</Shares>
<Debt href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,C18&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">724000</Debt>
</Original-Capital-Structure>
<Planned-Capital-Structure>
<Stockholders-Equity href="MasterWrapper.asp? ID=FIS,FL,EBIT-
  EPS,C21&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">400000</Stockholders-Equity>
<Shares href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,C22&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">200000</Shares>
<Debt href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,C23&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">400000</Debt>
<Interest href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,C24&Format=Data&
    Type=Cell&Command=No" Target="cmdFrame">0.1</Interest>
</Planned-Capital-Structure>
<EBIT-EPS>
<Year href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F15&Format=Formula&
    Type=Cell&Command=No" Target="cmdFrame">1990</Year>
<EBIT href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F16&Format=Formula&
    Type=Cell&Command=No" Target="cmdFrame">213000</EBIT>
        <Original-Capital-Structure>
            <ROE href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F19&Format=Formula&
                Type=Cell&Command=No" Target="cmdFrame">0.27</ROE>
            <EPS href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F20&Format=Formula&
                Type=Cell&Command=No" Target="cmdFrame">0.53</EPS>
        </Original-Capital-Structure>
        <Planned-Capital-Structure>
            <ROE href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F23&Format=Formula&
                Type=Cell&Command=No" Target="cmdFrame">0.43</ROE>
            <EPS href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F24&Format=Formula&
                Type=Cell&Command=No" Target="cmdFrame">0.87</EPS>
    </Planned-Capital-Structure>
<Break-Even href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,F26&Format=Formula&
    Type=Cell&Command=No" Target="cmdFrame">NO</Break-Even>
</EBIT-EPS>
<EBIT-EPS>
```

Exhibit 7. An XMl Document Created Based on the EBIT/EPS DTD (Continued)

```
<Year href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G15&Format=Formula&
   Type=Cell&Command=No" Target="cmdFrame">1991</Year>
<EBIT href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G16&Format=Formula&
   Type=Cell&Command=No" Target="cmdFrame">180000</EBIT>
<Original-Capital-Structure>
<ROE href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G19&Format=Formula&
   Type=Cell&Command=No" Target="cmdFrame">0.23</ROE>
<EPS href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G20&Format=Formula&
   Type=Cell&Command=No" Target="cmdFrame">0.45</EPS>
   </Original-Capital-Structure>
   <Planned-Capital-Structure>
      <ROE href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G23&Format=Formula&
         Type=Cell&Command=No" Target="cmdFrame">0.35</ROE>
      <EPS href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G24&Format=Formula&
         Type=Cell&Command=No" Target="cmdFrame">0.7</EPS>
   </Planned-Capital-Structure>
<Break-Even href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,G26&Format=Formula&
   Type=Cell&Command=No" Target="cmdFrame">NO</Break-Even>
</EBIT-EPS>
</FIS-EBIT-EPS>
```

- <xsl:apply-templates select="FIS-EBIT-EPS/Original-Capital-Structure" />
  This element selects all "Original-Capital-Structure" elements and asks XSL processor to find and apply an appropriate template.
- <xsl:template match="Shares">
  This statement indicates that this template corresponds to the "Shares" element.
- <xsl:template match="text()">
  This template is for handling text element (i.e., an element that has character data between start and end tags).
- <xsl:value-of />
  This statement returns the value of current element as text.
- <xsl:value-of select="@href"/>
  XSL patterns denote attributes with the @ symbol. This statement selects elements that have an href attribute and returns the value of the element's href attribute as text.
- <xsl:attribute name="href">
  This statement creates an attribute and attaches it to the target element.

Note that "&#38;" is replaced with "&" to make the document in Exhibit 8 readable.

Exhibit 8. The XSL File for Displaying the EBIT/EPS Document

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML>
  <STYLE>
    A { text-decoration:none }
    A:link { color:blue }
    A:visited { color:blue }
    A:active { color:red }
    TH.org {background-color:lightgreen}
    TD.org {background-color:cyan}
    .over {background-color:lightcyan}
    .click {background-color:lightcoral}
  </STYLE>
  <SCRIPT language="JavaScript">
    var clicked = "";
    function MouseOver(n) {
      if(n != clicked) {
        document.all[n].className = "over";
      }
      window.event.cancelBubble = true;
    }
    function MouseOut(n) {
      if(n != clicked) {
        document.all[n].className = "org";
    }
      window.event.cancelBubble = true;
    }
  </SCRIPT>
<BODY>
  <TABLE id="sheet" border="1">
    <TR>
       <TD> <CENTER>
         <TABLE border="2">
          <TR> <TH colspan="2">
            <A href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,B15&Format=Text&
              Type=Cell&Command=No" Target="cmdFrame">Original-Capital-
              Structure</A>
          </TH></TR>
          <TR> <TH>
            <A href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,B16&Format=Text&
              Type=Cell&Command=No" Target="cmdFrame">Stockholders-
              Equity</A>
            </TH>
            <xsl:apply-templates select="FIS-EBIT-EPS/Original-Capital-Structure/
            Stockholders-Equity" />
```

Exhibit 8. The XSL File for Displaying the EBIT/EPS Document (Continued)

```
         </TR>
         <TR> <TH>
           <A href="MasterWrapper.asp? ID=FIS,FL,EBIT-EPS,B17&Format=Text&
           Type=Cell&Command=No" Target="cmdFrame">Shares</A>
         </TH>
         <xsl:apply-templates select="FIS-EBIT-EPS/Original-Capital-Structure/
         Shares" />
       </TR>
       <TR> <TH>
           <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,B18&Format=Text&
             Type=Cell&Command=No" Target="cmdFrame">Debt</A>
         </TH>
         <xsl:apply-templates select="FIS-EBIT-EPS/Original-Capital-Structure/
         Debt" />
       </TR>
       </TABLE> </CENTER>
   </TD>
    <TD>
     <CENTER> <TABLE border="2">
       <TR>
         <TH colspan="2">
           <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,B20&Format=Text&
             Type=Cell&Command=No" Target="cmdFrame">Planned-Capital-
             Structure</A>
         </TH> </TR>
       <TR> <TH>
           <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,B21&Format=Text&
             Type=Cell&Command=No" Target="cmdFrame">Stockholders-
             Equity</A>
         </TH>
         <xsl:apply-templates select="FIS-EBIT-EPS/Planned-Capital-Structure/
         Stockholders-Equity" />
       </TR>
       <TR><TH>
         <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,B22&Format=Text&
           Type=Cell&Command=No" Target="cmdFrame">Shares</A>
         </TH>
         <xsl:apply-templates select="FIS-EBIT-EPS/Planned-Capital-Structure/
         Shares" />
       </TR>
       <TR><TH>
           <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,B23&Format=Text&
             Type=Cell&Command=No" Target="cmdFrame">Debt</A>
         </TH>
         <xsl:apply-templates select="FIS-EBIT-EPS/Planned-Capital-Structure/
         Debt" />
```

Exhibit 8. The XSL File for Displaying the EBIT/EPS Document (Continued)

```
      </TR>
      <TR><TH>
          <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,B24&Format=Text&
            Type=Cell&Command=No" Target="cmdFrame">Interest</A>
        </TH>
        <xsl:apply-templates select="FIS-EBIT-EPS/Planned-Capital-Structure/
        Interest" />
      </TR>
    </TABLE> </CENTER>
  </TD>
</TR>
  <TR> <TD colspan="2"> </TD> </TR>
  <TR><TD colspan="2">
    <TABLE border="1">
    <TR>
      <TD colspan="2"> </TD>
      <TH colspan="2">
        <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E18&Format=Text&
          Type=Cell&Command=No" Target="cmdFrame">Original-Capital-
          Structure</A>
    </TH>
    <TH colspan="2">
        <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E22&Format=Text&
          Type=Cell&Command=No" Target="cmdFrame">Planned-Capital-
          Structure</A>
    </TH>
    <TD colspan="2"> </TD>
  </TR>
    <TR> <TH>
        <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E15&Format=Text&
          Type=Cell&Command=No" Target="cmdFrame">Year</A>
      </TH>
      <TH>
        <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E16&Format=Text&
          Type=Cell&Command=No" Target="cmdFrame">EBIT</A>
      </TH>
      <TH>
        <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E19&Format=Text&
          Type=Cell&Command=No" Target="cmdFrame">ROE</A>
      </TH>
      <TH>
        <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E20&Format=Text&
          Type=Cell&Command=No" Target="cmdFrame">EPS</A>
      </TH>
```

Exhibit 8. The XSL File for Displaying the EBIT/EPS Document (Continued)

```
        <TH>
          <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E23&Format=Text&
            Type=Cell&Command=No" Target="cmdFrame">ROE</A>
        </TH>
        <TH>
          <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E24&Format=Text&
            Type=Cell&Command=No" Target="cmdFrame">EPS</A>
        </TH>
        <TH>
          <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E26&Format=Text&
            Type=Cell&Command=No" Target="cmdFrame">Break-Even?</A>
        </TH>
        <TH>
          <A href="MasterWrapper.asp?ID=FIS,FL,EBIT-EPS,E28&Format=Data&
            Type=Cell&Command=No" Target="cmdFrame"></A>
        </TH>
        </TR>
          <xsl:apply-templates select="FIS-EBIT-EPS/EBIT-EPS" />
        </TABLE>
      </TD>
    </TR>
  </TABLE>
</BODY>
</HTML>
</xsl:template>

<xsl:template match="text()"> <xsl:value-of /></xsl:template>

<xsl:template match="EBIT-EPS">
  <TR><xsl:apply-templates /> </TR>
</xsl:template>

<xsl:template match="Year">
  <TD> <A>
       <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
      <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
      <xsl:apply-templates />
  </A></TD>
</xsl:template>

<xsl:template match="ROE">
  <TD> <A>
    <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
```

Exhibit 8. The XSL File for Displaying the EBIT/EPS Document (Continued)

```
    <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
    <xsl:apply-templates />
  </A></TD>
</xsl:template>

<xsl:template match="EPS">
  <TD> <A>
    <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
    <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
    <xsl:apply-templates />
  </A></TD>
</xsl:template>

<xsl:template match="Break-Even">
  <TD> <A>
    <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
    <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
    <xsl:apply-templates />
  </A> </TD>
</xsl:template>

<xsl:template match="Planned-Capital-Structure">
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="Original-Capital-Structure">
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="Stockholders-Equity">
  <TD> <A>
      <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
      <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
      <xsl:apply-templates />
  </A> </TD>
</xsl:template>

<xsl:template match="Shares">
  <TD> <A>
      <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
      <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
      <xsl:apply-templates />
  </A> </TD>
</xsl:template>
```

Exhibit 8. The XSL File for Displaying the EBIT/EPS Document (Continued)

```
<xsl:template match="Debt">
  <TD> <A>
      <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
      <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
      <xsl:apply-templates />
  </A> </TD>
</xsl:template>

<xsl:template match="Interest">
  <TD> <A>
      <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
      <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
      <xsl:apply-templates />
  </A> </TD>
</xsl:template>

<xsl:template match="EBIT">
  <TD> <A>
      <xsl:attribute name="href"> <xsl:value-of select="@href"/> </xsl:attribute>
      <xsl:attribute name="Target"> <xsl:value-of select="@Target"/> </xsl:attribute>
      <xsl:apply-templates />
  </A> </TD>
</xsl:template>
</xsl:stylesheet>
```

Flowcharts for Mapping Rules

Two flowcharts (Exhibits 9 and 10) are used to describe the structure of Command_Rule and Object_Rule. In Exhibit 9, a simple example (i.e., click on the “EBIT” object) is used to explain how to program this rule.

In Exhibit 10, a simple example (i.e., click on the “explain” command link) is used to explain how to program Object_Rule. The Object_Rule is divided into subrules (i.e., procedures).

Outputs

Exhibit 6 shows the interface of the prototype. The left frame of our prototype’s interface contains three panels: function, command, and system administration. Exhibits 11 and 12 show outputs of executing some system commands. Exhibit 13 shows the function panel, which contains all financial analysis functions of the prototype.

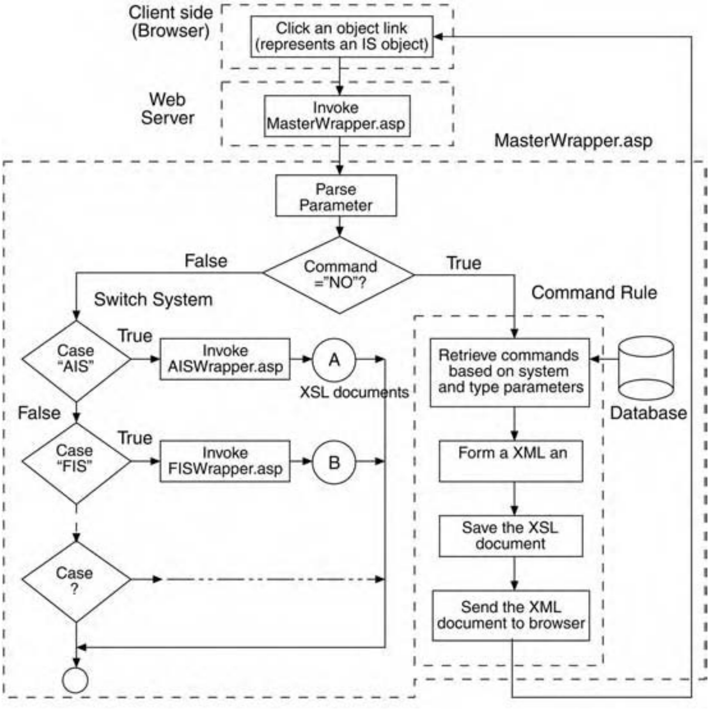


Exhibit 9. The Program Flow of MasterWrapper.asp

FUTURE WORK

Bieber[15] takes a two-stage approach to engineering applications for the World Wide Web. First, the software engineer performs a relationship-navigation analysis (RNA), analyzing an existing or new application specifically in terms of its intra- and interrelationships. We plan to use the RNA technique to supplement our mapping rule approach and form a relationship-navigation-rule analysis (RNRA) technique.

Second, a dynamic hypermedia engine (DHymE), automatically generates links for each of these relationships and metaknowledge items at runtime, as well as sophisticated navigation techniques not often found on the Web (e.g., guided tours, overviews, and structural query) on top of these links. Our mapping rules infer useful links that give users more direct access to the primary functionality of the IS, give access to various relationships about IS objects, and enable annotation and ad hoc links. It is also planned to extend the mapping rules and XML DTD to support sophisticated navigation techniques (e.g., guided tours, overviews, and structural query).

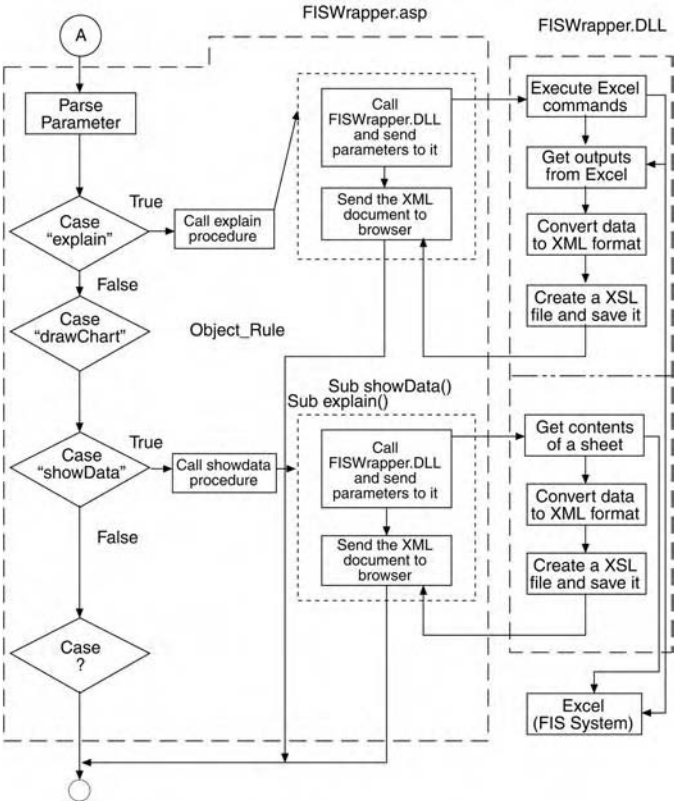
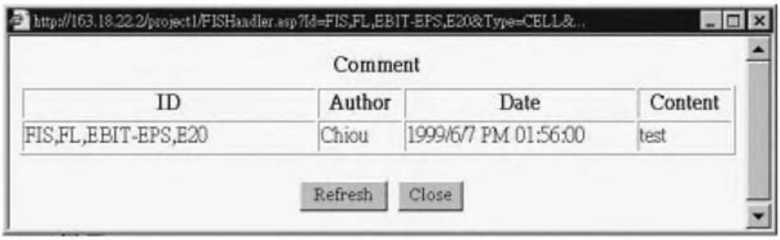


Exhibit 10. The Program Flow of FISWrapper (i.e., FISWrapper.asp and FISWrapper.dll)


http://163.18.22.2/project1/FISHandler.asp?Id=FIS,FL,EBIT-EPS,E20&Type=CELL&...

Comment

ID	Author	Date	Content
FIS,FL,EBIT-EPS,E20	Chiou	1999/6/7 PM 01:56:00	test

Exhibit 11. The Pop-Up Dialog Box Displays the Output of Executing the explain Command

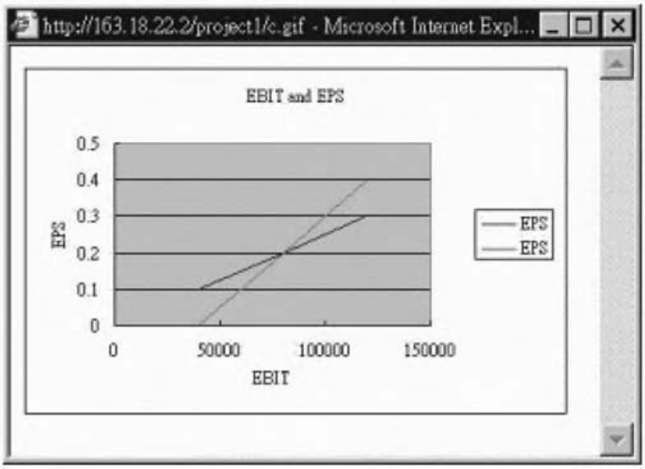
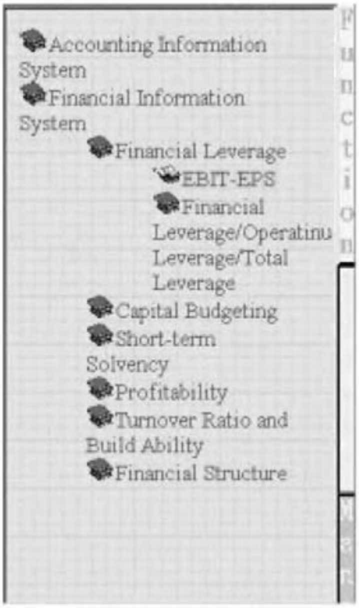


Exhibit 12. The Output of Executing the viewComment Command



Accounting Information System

Financial Information System

- Financial Leverage
  - EBIT-EPS
  - Financial Leverage/Operatinu Leverage/Total Leverage
- Capital Budgeting
- Short-term Solvency
- Profitability
- Turnover Ratio and Build Ability
- Financial Structure

Exhibit 13. The Function Panel Containing All Financial Analysis Functions of the Prototype

CONCLUSION

This chapter makes two contributions. First, a set of guidelines is proposed for analyzing IS and building mapping rules. Second, it provides one XML DTD, XML document, and XSL file to demonstrate the feasibility of using XML for representing mapped information in a human-readable, machine-readable, structured, and semantic way.

The WWW provides a way to integrate hypermedia into information systems. It is believed that integrating information systems in the business world with WWW should constitute a major thrust for the WWW research. It will go a long way toward making applications more understandable. When reengineering applications for the WWW, dynamic relationship mapping could be an effective way to add additional hypermedia links. This will assure that new WWW applications (especially IS) will have hypertext in them.[6] It is hoped that this chapter will call people's attention to mapping rules for better integrating the WWW and information systems.

XML aims to develop machine-readable, human-readable, structured, and semantic documents that can be delivered over the Internet. An important feature of XML is that it is extensible. XML allows the development of custom and domain-specific elements and attributes. When mapping outputs from information systems to hypertext constructs, XML seems to be a more potential approach than HTML.

References

1. N. Walsh, A Technical Introduction to XML, URL: http://nwalsh.com/articles/xml/, 1998.
2. M. Bieber and C. Kacmar, Designing Hypertext Support for Computational Applications, *Communication of the ACM,* 38(8), 99–107, 1995.
3. M. Bieber and F. Vitali, Toward Support for Hypermedia on the World Wide WWW, *IEEE Computer,* 30(1), 62–70, 1997.
4. Microsoft Corporation, XML: Enabling Next-Generation Web Applications, URL: http://www.microsoft.com/xml/articles/xmlwp2.htm, 1998.
5. T. Bray, J. Paoli, and C.M. Sperberg-McQueen, Extensible Markup Language (XML). 1.0, URL: http://www.w3.org/TR/1998/REC-xml-19980210, 1998.
6. R. Khare and A. Rifkin, XML: A Door to the Automated Web Applications, *IEEE Internet Computing,* 1(4), 78–87, 1997.
7. D. Connolly, R. Khare, and A. Rifkin, The Evolution of the Web Documents: The Ascent of XML, XML Special Issue of the *World Wide Web Journal,* 2(2), 119–128, 1997.
8. J. Bosak, XML, Java, and the Future of the Web, URL: http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm, 1997.
9. C.M. Chiu, Fusing the World Wide WWW and the Open Hypermedia System Technologies, Ph.D. Dissertation, Rutgers University, New Brunswick, NJ, 1997.
10. F. Halasz, Reflection on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems, *Communications of the ACM,* 31(7), 836–855, 1988.
11. M. Bieber, Automating Hypermedia for Decision Support, *Hypermedia,* 4(2), 83–110, 1992.
12. A. Garrido and G. Rossi, A Framework for Extending Object-Oriented Applications with Hypermedia Functionality, *The New Review of Hypermedia and Multimedia,* 2, 25–41, 1996.
13. J. Wan, Integrating Hypertext into Information Systems through Dynamic Linking, Ph.D. dissertation, New Jersey Institute of Technology, Newark, 1996.

14. Microsoft Corporation, XSL Developer's Guide, URL: http://msdn.microsoft.com/xml/xslguide/default.asp, 1999.
15. M. Bieber, Web Engineering, URL: http://megahertz.njit.edu/~bieber/web-engineering.html, 1998.

ABOUT THE AUTHOR

Chao-Min Chiu is an assistant professor in the Department of Information Management at National Kaohsiung First University of Science and Technology. His current interests include electronic commerce, integrating information systems into the WWW, and knowledge management. He can be contacted at cmchiu@ccms.nkfu.edu.tw.

Chapter 38
Linux and the Web

Srinivas Padmanabharao

A real threat to Microsoft's dominance of the desktop operating system workspace could come from the creation by a Finnish student, Linus Torvalds, in the form of a freely available operating system — Linux. Linux was originally developed by Linus and then enhanced, debugged, and completely rewritten by thousands of programmers around the world, and today it poses a credible threat to the domination of Microsoft. While Linux is still a few years away from appearing on every desktop around the world, its use as a server operating system has shown a remarkable increase over the past year. This chapter presents a look at the world of Linux.

LINUX: THE OPERATING SYSTEM

Linux is a UNIX-clone operating system. It is a freely available implementation of the published POSIX standards and does not use any of the UNIX source code. Linux traces its roots to another free operating system for the x86 architecture called MINIX, developed by Andy Tanenbaum. Linus Torvalds, inspired by the desire to create a better MINIX, took this codebase, modified the kernel, added a driver for keyboard and screens, and released it as Linux under the General Public License (GPL) in 1991. Since then, thousands of users have contributed to and enhanced the basic Linux kernel and code.

In essence, Linux is just a kernel and needs to be packaged with lots of applications, drivers, and tools to make it into a complete usable operating system. However, Linux is commonly used to refer to the kernel along with all other pieces of software needed to make the kernel useful.

Obtaining Linux

The two most common ways of obtaining Linux are:

- *By ftp over the Internet.* Linux is usable under the General Public License (GPL). This means that Linux is available, free of charge, in both binary and source code forms and can be used without the

0-8493-1161-6/02/$0.00+$1.50

need for any agreements, etc. by as many users as a system will support. There are many mirror sites over the Internet from where one can download Linux,[1] and choosing the nearest site is usually better. First-time users are better off downloading a stable version of the system; a version x.y.z, where y is an even number (including zero), is a stable version.

- *Through a commercial distribution.* Although Linux is itself free, one is allowed to charge a fee for packaging it into an easy-to-install product usually referred to as a distribution. Such distributions come with all the other associated software needed to make it useful, that is, desktop interfaces, network management tools, etc., and documentation. They are made available via CD-ROM. Examples of such companies are RedHat (Version 5.1 costs U.S.$49.95), Caldera, and Slackware. It must be remembered, however, that all software found on the CD-ROM can also be obtained free over the Internet, and it might be a worthwhile exercise to go through, for the adventurous types.

Hardware Requirements

If one has purchased a PC anytime during the last five years, one probably already has a machine that is ready for Linux. (Of course, if one has not, then it is probably time to call the nearest museum and claim possession of an antique relic.) While Linux was initially written for the PC, versions are available for other bigger machines. IBM is porting Linux to its RS6000 machines — if one needs proof of its growing popularity.

Linux supports Intel processors 386 and higher along with their clones. It also supports many DEC Alphas, Sun SPARCs, and Power PCs to varying extents. On the memory side, Linux will function reasonably on anything greater than 8MB of RAM. However, if one is using the X-windowing system and planning to use a lot of programs, then a single user system will perform satisfactorily with 32MB of RAM. One might want to consider adding more memory, around 8 to 16MB per additional user of the system. Linux can support up to 1GB of RAM. The falling price of storage makes it affordable and sensible to have at least about 4GB of hard disk space, although the base Linux kernel can be installed in less than 15MB.

Drivers for most types of peripherals (e.g., a mouse and sound card, video card, network card, and modems) can be found. Most distribution vendors maintain a more complete and updated list of supported hardware. If a driver cannot be found for a very special piece of hardware, one is welcome to write one and contribute to the Linux effort.

Installing Linux

Linux has come a long way from the days when it was a hacker's operating system. If installing Linux off a commercial distribution, then one will also usually get installation support for the price one has paid. Documentation is also available, along with the distribution. If installing Linux off the downloaded code from the net, then one can also get a guide on installation from the same site. Linux HOWTO documentation is a good source.[2] Instead of going into the gory details of a screen-by-screen description (i.e., a command-by-command description), emphasis is placed on the need for planning for the installation. Answer at least the following questions satisfactorily before jumping ahead.

1. What are hardware specifications on the machine? Get all the details of the hardware on the system. This will allow for installation of the needed drivers and prevent many a headache later.
2. What is this system going to be used for? If installing Linux on a machine at home with the intention of using it for personal pursuits, then one might want to make sure to also install the Xfree86, which provides a graphical user interface. If one intends to use it as a Web server, ensure that there is enough space to hold the Web server software (e.g., Apache), enough space for all the Web pages one will host, and enough space for any HTML editors one wishes to use.
3. Does one want to use two operating systems? It is not unlikely, especially if the system is at home, that one will want to retain the other operating system that is already present (like MS-DOS or some flavor of the Windows family). In that case, one might want to provide for the option of a prompt, i.e., Linux will ask you which system you wish to use at bootup time. Of course, one can access other files from the Linux environment by mounting it appropriately.

Some Features of Linux

Having gone through the installation ordeal successfully, rest assured that you have become the owner of a system that runs one of the best operating systems available. Here are a few of the features of the operating system.

- Multitasking and multi-user. Like any UNIX system, Linux supports many users, each running many programs simultaneously. Linux also supports virtual consoles; this feature allows one to have multiple sessions and log-in as two users simultaneously (use ALT-F1 to switch between sessions). Use the feature judiciously — determining one's current status can get pretty confusing, which is why one has commands like *whoami*.

- Multithreading. Linux has native kernel support for multiple independent threads of control within a single process memory space. It runs in protected mode on the 386, implementing memory protection between processes, so that one program cannot bring the entire system down.
- Memory management. Linux demand loads executables; that is, it reads only those parts of a program that are actually used. Linux increases speed by using shared copy-on-write pages among executables. This means that multiple processes can use the same memory for execution. When one tries to write to that memory, that page is copied somewhere else. Linux uses a unified memory pool for user programs and disk cache, so that all free memory can be used for caching, and the cache can be reduced when running large programs.
- Multiple file systems. Linux supports several common file systems, including Minix, Xenix, and all the common system V file systems, and has an advanced file system of its own, which offers file systems of up to 4TB, and names up to 255 characters in length. It provides transparent access to MS-DOS partitions (or OS/2 FAT partitions) via a special file system. VFAT (Windows NT, Windows 95) support and FAT-32 is available in Linux 2.0. A special file system called UMSDOS is also available; it allows Linux to be installed on a DOS file system. Linux supports a CD-ROM file system that reads all standard formats of CD-ROMs.
- Compatibility. Linux is compatible with UNIX, and most applications that have been written for UNIX can be recompiled to run on Linux with little or no modification. Linux is highly interoperable and can co-exist in a diverse environment with Netware and Windows NT.
- Networking support. Linux supports TCP/IP networking, including ftp, telnet, etc., which makes it ideal for use on a Web server.

A comparison of Linux with most of today's popular operating systems can be found on the Web;[3] it can be used to compare the performance of Linux with other operating systems.

LINUX APPLICATIONS

An operating system by itself provides very limited functionality for end use. It is the applications like databases, word processors, and development tools that determine the ultimate success of any operating system. It is in this respect that Linux has had an amazing success story. Because the Linux source code is freely available, it has provided developers around the world with the freedom to develop applications for this operating system without the need for acquiring expensive licenses from any vendor. This freedom has led to the development and free availability of a wide variety of software applications, such as:

- Development tools, including compilers, assemblers, and debuggers
- Text editors and word processing software
- A whole host of Internet applications, such as usenet news readers and e-mail agents
- World Wide Web development tools, Web servers, and browsers
- Graphics creation and manipulation tools
- Databases

A complete list of all applications available is maintained under the Linux Software Map (LSM).[4] The Linux Documentation project[5] is an excellent source of documentation for all Linux-related material, including applications for Linux.

ENABLING E-COMMERCE

In the Internet-crazy times of today, one of the key objectives of any company is to prepare itself for E-commerce. Today, companies are moving beyond the realm of providing static information on their Web sites and actively enhancing their Web sites to provide opportunities for interaction with the customer. In this context, the key enabling technologies that a company must consider include:

- A database to store information
- A Web server software to present a front end to the world
- A tool to connect between the two and provide a mechanism for actively updating the database with customer details or provide customer-requested information

Keeping with the spirit of *free*dom in Linux, this chapter will review the Apache Web server, which runs on more than half the servers on the Internet; a database called PostgreSQL; and an interconnection tool called PHP.

PostgreSQL

PostgreSQL originates from a research project in Professor Michael Stonebraker's group at Berkeley. It is a high-performance, robust object relational DBMS. It provides a full-featured API for development of client/server or n-tier applications. Also, via PHP/FI, it can be easily integrated with any Web site as a high-performance back end. PostgreSQL is freely available under the GNU public license, and a copy of the PostgreSQL database can be obtained from its Web site (http://www.postgresql.com).

Key Features. Key features of the PostgreSQL database include:

- Web/Apache interface
- Graphical interface
- APIs: C, C++, TLC, Perl, Python, and Java
- ODBC
- JDBC
- Online backup
- Regression testing package included to ensure reliability

PHP

PHP is a server-side, cross-platform, HTML-embedded scripting language. Rasmus Lerdorf conceived PHP sometime in the fall of 1994. It was initially known as Personal Home Page Tools. PHP Version 3.0 is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java, and Perl — with a couple of unique PHP-specific features thrown in. The goal of the language is to allow Web developers to write dynamically generated pages quickly. PHP can be obtained free of charge from its Web site (http://www.php.net).

Key Features. Key features of the PHP interconnection tool include:

- HTTP authentication; the HTTP authentication is available only when PHP is running as a module in the Apache server
- GIF creation
- File upload support
- HTTP cookie support
- Database support; this is probably the most powerful feature of PHP, with its ability to connect to both commercial databases (e.g., Oracle and Informix) and free databases (e.g., PostgreSQL)
- Regular expressions for complex string manipulation
- Error handling
- Connection handling
- PHP source viewer

Apache Web Server

One of the key pieces of software needed on a Web server is the HyperText Transfer Protocol (HTTP) server. When an end user enters a URL in order to view the Web site at the server end, it is this HTTP server that processes the request and sends the required information back for formatting and displaying by the browser. One can download Apache free of charge from its Web site (http://www.apache.org).

Key Features. Key features of the Apache Web server include:

- A powerful, flexible, HTTP/1.1-compliant Web server
- Implements the latest protocols, including HTTP/1.1 (RFC2068)
- Is highly configurable and extensible with third-party modules
- Can be customized by writing modules using the Apache module API
- Provides full source code and comes with an unrestrictive license
- Runs on most versions of UNIX without modification
- DBM databases for authentication; it allows one to easily set up password-protected pages with enormous numbers of authorized users, without bogging down the server
- Customized responses to errors and problems
- Allows multiple DirectoryIndex directives
- Unlimited numbers of alias and redirect directives
- Content negotiation
- Multi-homed servers facility, which allows the server to distinguish between requests made to different IP addresses (mapped to the same machine)

CONCLUSION

The world of Linux and its applications offer great promise to developers and IT managers alike. Following the open source model of development, a number of diverse and reliable applications have been written and are available free of charge. Picking and choosing between them can enable today's IT manager to provide his company's presence on the Internet in an easy, quick, and cost-efficient manner. These applications offer even greater promise to those small businesses that are cash strapped.

ACKNOWLEDGMENTS

The author is thankful to all supporters of the Open Source concept and those who are involved in developing such wonderful applications. My friend Sanjay Verma has been very helpful in this endeavor.

Notes

1. ftp://sunsite.unc.edu/pub/linux.
2. http://metalab.unc.edu/LDP/HOWTO/Installation-HOWTO.html.
3. http://www.falconweb.com/~linuxrx/WS_Linux/OS_comparison.html.
4. http://www.execpc.com/lsm/.
5. http://metalab.unc.edu/LDP/.

ABOUT THE AUTHOR

Srinivas Padmanabharao is a consultant with Deloitte Consulting, Toronto, Ontario, Canada.

Chapter 39 Java and Its Supporting Technologies

Ramesh Venkataraman

Java started out as a part of a software development effort intended to create consumer electronics devices with software embedded in them (Gosling and McGilton, 1996). Numerous problems with writing embedded software using languages available at that time led the designers of the project to develop a new language. The result of this effort was the Java language, which had the following objectives: "A simple, object-oriented, network-savvy, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, dynamic language" (Gosling and McGilton, 1996).

The two best-known features from the above list are the architecture neutrality of Java and the object-oriented nature of the language. The first feature allows the same Java code to be run on different operating system platforms, such as Windows, UNIX, or Macintosh. The object-oriented nature of the language leads to software systems that are easier to maintain and promotes code reuse. The majority of this section is devoted to a discussion of these two features.

Architecture Neutrality

The Java language is designed so that one can follow the Write Once, Run Anywhere model. This means that irrespective of the operating system on which one has designed, developed, and tested a software system, one can deploy the system on another platform without having to rewrite any pieces of the code. This is an immense benefit for corporations that support a diverse set of platforms. Traditionally, diversity in platforms has meant that software written for a specific platform (say, Windows) had to be ported to every other platform that it needed to run on. Porting

0-8493-1161-6/02/$0.00+$1.50

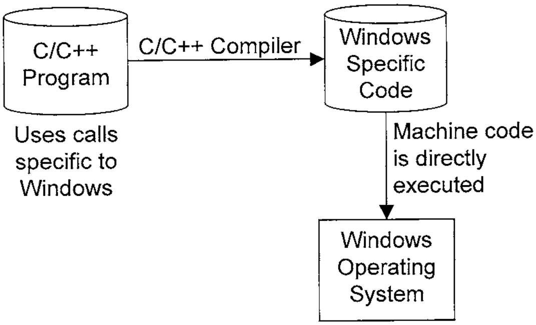


Exhibit 1. Typical Compilation and Execution of a Program

required a significant number of code rewrites, which in turn required a significant commitment of additional resources. Worse still, every time a new release of the software was developed, resources had to be allocated to incorporating the new changes into every platform supported by the software. Developing code in Java eliminates the need for porting code from one platform to another.

To understand how Java provides architecture neutrality, it is important to take a closer look at what happens when one develops systems using any other language. Assume the development of a new custom order-processing application program that is intended to run on the Windows platform. Further assume that the program is to be written in C or C++. The system itself is probably going to consist of many program files. In order to create the graphical user interface, the code is likely to make several (probably thousands) calls to library functions specific to the Windows platform. For simplicity, assume that this is the only specific functionality that the program uses. Running this program through a compiler will generate an executable file that is specific to the Windows platform. If one now decides to support the same application program on a UNIX or Mac platform, at a minimum all the code that relates to calls made to the library functions will need to be modified to correspond to the libraries supported by UNIX or Mac. Given the sheer number of such calls, this task will require a significant amount of development work. It will also require hiring people with a different skill set than the original developers. Exhibit 1 illustrates this scenario.

What happens if one uses Java to write this same program? In order to create the graphical user interface, one will use the Java Abstract Windowing Toolkit (AWT). The AWT is part of the Java language and is thus the same for whichever platform the program is going to eventually run on. Thus, the need to rewrite code for different platforms is eliminated.

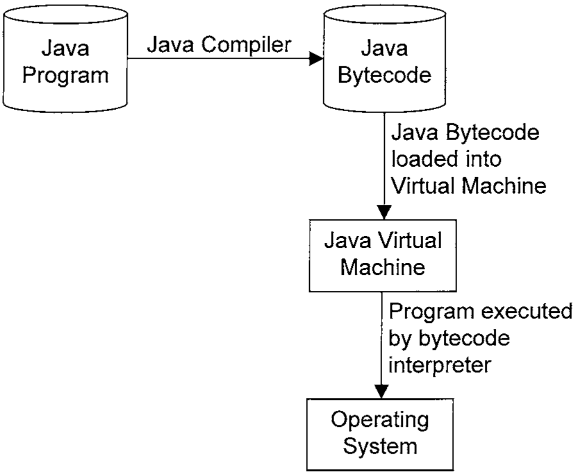

Java Program
Java Compiler
Java Bytecode
Java Bytecode loaded into Virtual Machine
Java Virtual Machine
Program executed by bytecode interpreter
Operating System

Exhibit 2. Compilation and Execution of a Java Program

This is because compiling a set of Java files generates object code (called bytecodes) that is targeted to run on the Java Virtual Machine (JVM). The JVM is a piece of software that runs on top of the operating system (Windows, UNIX, etc.). Thus, as long as the operating system supports the JVM, the code generated by compiling the Java program(s) will be able to run on any platform without requiring any changes to the code. Exhibit 2 illustrates the process of how a Java program runs on a particular machine. This paradigm is referred to as the Write Once, Run Anywhere model.™

To ensure the Write Once, Run Anywhere model,[1] it is important that programmers use 100% Pure Java in their programs. This means that they should not use any extensions to Java that tie the code to a specific operating system platform. For example, Microsoft Visual J++ provides users with some classes that are specific to the Windows platform. Using these classes in an application would make these Java programs incapable of running on a UNIX or Mac platform. This, of course, is one of the issues that was part of the Sun versus Microsoft lawsuit. Thus, if the ability to run on different platforms is important, it is critical to follow the guidelines for developing 100% Pure Java code specified by Sun (http://java.sun.com/100percent/).

Object Orientation

Object-oriented programming has, for the past decade and a half, been touted as the new paradigm for software development. Numerous articles and books have been written about the benefits of object-oriented systems development. Some example benefits include code reuse, robustness, adaptability to changing requirements, and easier maintenance of code. However, one of the key problems that plagued the object-oriented world in the past was that object-oriented languages were too difficult to use. In this author's opinion, Java is an example of a simple yet powerful object-oriented language. The biggest advantage of Java is that features inherent in the language make it easy for programmers to create code that has object-oriented characteristics. For example, the simple act of defining variables as private in a class goes a long way in enforcing encapsulation of objects. Using inheritance and polymorphism are also relatively easier in Java than in other languages. The end result is that programmers are less likely to violate object-oriented principles when writing their code.

Java is a language that allows novice programmers to write reasonably good code (somewhat unknowingly) and allows good programmers to write excellent code without wasting time on frustrating issues. For example, Java supports the notion of references (they are somewhat like pointers in other languages, without the associated headaches) and automatic garbage collection. These features relieve the programmer from hassles of memory management and pointer memory overruns — issues that have plagued C/C++ programmers for years. Of course, the current generation of programmers using Java will never experience the joys of tackling these problems.

Another example of how some of the built-in features of Java help novice programmers can be seen in Java's error/exception handling features. One of the most common mistakes made by novice programmers when writing programs that call system-level functions such as reading or writing to files, or network connections, is that they tend not to check to see if the calls actually succeeded. In Java, any pieces of code that perform functions that can lead to such errors are required to be put in a try/catch block. The catch part is mandatory and enables the programmer to deal with any errors that might occur when the program is run. By not allowing programmers to write code without dealing with the exceptions that may occur, Java forces programmers to write more robust code (mostly without much conscious effort from the programmer).

Java supports the basic tenets of object orientation, including objects and classes, encapsulation, inheritance, and polymorphism. Java only supports "true" single inheritance; that is, it is possible to inherit properties *and* behavior only from a single superclass. However, it does allow one to emulate multiple inheritance through the use of interface classes; that is, it is possible to inherit the behavior *alone* from multiple superclasses.

The Java development kit (JDK) comes bundled with a number of classes (grouped together into packages) that extend the power of the language. The current version of the JDK (1.2) has built-in support for creating graphical components (java.awt), using networking functions (java.net), accessing databases (java.sql), as well as common data structures such as hash tables, stacks, and a dynamic memory structure called vectors (java.util). In addition, it has built-in support for creating multithreaded applications through its thread mechanisms (java.lang.Thread) as well as distributed applications through Remote Method Invocation (RMI). The functionality provided by some of these packages is discussed below.

Remote Method Invocation (RMI)

RMI provides the foundation for writing distributed computing applications using Java. RMI allows a Java object running on one machine to use the services provided by another object by calling the public methods in objects on remote machines. This communication is achieved through the use of an interface that hides the fact that the objects are actually located on a remote machine.

RMI-based applications (like other client/server applications) consist of a client object (running on a local Java Virtual Machine) and a server object (running on a remote Java Virtual Machine). Objects that want to provide services to other objects register themselves with an RMI registry. Clients that need to use them simply access the RMI registry and look for objects that provide the services they need. RMI also provides a mechanism for distributed garbage collection, that is, managing references to objects in the RMI server, thus once again freeing the programmer from memory management concerns.

JDBC

JDBC[2] is a set of classes that allows Java applications and applets to issue SQL requests to databases and process the results of queries. The JDBC API provides support for establishing a connection, issuing a SQL statement, and processing the results. The JDBC API and SQL provide a standard database-independent mechanism for Java applications to communicate with a variety of databases.

To be able to connect to a database, one will need access to a JDBC driver. A JDBC driver provides the actual connection between one's Java application (applet) and a database. Currently, the JDBC-ODBC bridge driver is used extensively because it allows access to any database that has ODBC drivers. However, pure Java drivers are also available for many of the popular databases.

Java Foundation Classes and Swing

Java 1.0 provided basic support for creating windowed applications through the Abstract Windowing Toolkit (AWT). However, the model for event handling was revised completely when Java 1.1 was released. This allowed programmers to create complex graphical applications (and applets) using Java. However, these applications had their own unique look and feel. One could look or interact with a Java application and clearly identify it as being a Java program.

The new Swing classes are part of the Java Foundation classes that are bundled with Java, JDK 1.2. The key characteristic of Swing classes is that they allow one to control the look and feel of the application being developed. Currently, one can choose between the Java look and feel, the Win32 look and feel, and the Motif look and feel. One can also design one's own look and feel and then incorporate that in applications. It should be noted that selecting a look and feel does not constrain the portability of the program. For example, if one specifies the Win32 look and feel and tries to run a program on a UNIX machine, the Swing mechanism will determine that the look and feel does not exist on that machine and will simply default to the cross-platform Java look and feel.

SUPPORTING TECHNOLOGIES

Java, the core language (as described above), has found its way into a number of application areas and has resulted in the development of a number of supporting technologies. These technologies have helped make Java a popular alternative to other development languages. This section takes a brief look at a number of these applications and technologies.

Java and the Web

Applets. A key reason for Java's rise in popularity has been the ability to embed small Java programs in Web pages. These programs, called applets, were initially used to "liven up" Web pages by adding graphical characteristics to an otherwise "drab" text-based Web page. Game and banner applets probably led this category of Java programs. Applets are special Java programs that have some restrictions on what they can and cannot do. In particular, they cannot (1) access files on the local machine, (2) spawn off any external programs, (3) communicate with any other host (Web server) other than the one they originated from, and (4) find out information about the machine they are running on. These restrictions were put into place to make it safe for users to download Java applets from Web servers onto their machines.

While the majority of applets in use are currently still used to simply "beautify" Web pages, there is a growing trend to use applets as the client tier in multi-tiered applications. By using applets and server-side Java technology such as RMI or JDBC, one can create powerful multi-tiered applications (including Web applications). Applets provide a very simple alternative to complicated client-side software installation. To activate the client side of an application on a machine, users simply have to download the Web page containing the applet. Thus, it is not surprising that this model is increasingly preferred to the complicated software installations that are typical of current client/server applications, especially E-commerce applications. Applets are also being used extensively as client front ends to legacy systems.

However, applets still have the drawback that they need to be downloaded across a network. Thus, the size of the applet (usually they are not very large) and the speed of the network are key factors in determining whether an applet-based solution is appropriate. This is perhaps why client/server applications using Java are more popular in high-speed corporate intranets than across the (slower) Internet.

Servlets. Java has not only made an impact on the client-side of typical Web applications (through applets), but it has also made inroads into the server-side processing on the Web. To support server-side processing using Java, Sun introduced the servlet API. Using the servlet API, one can write Java programs that can process data from client applications that reside on the Web browser. All input and output are achieved through HTTP-style interaction between the browser and the programs. The servlet API provides an alternative to traditional CGI processing or other proprietary Web applications development protocols. The advantages of the using the servlet API over CGI is that one can take advantage of the multi-threaded nature of Java and create multiple threads to take care of multiple requests instead of creating a process for every request (as in CGI). Developers are also automatically able to take advantage of all the advantages of using Java such as its object-oriented features, advanced memory management, and exception handling capabilities, as well as have access to all the Java packages and classes, including database connectivity through JDBC, discussed previously. Given the lack of well-defined methodologies for developing Web applications and the subsequent nightmare of code maintenance that seems to be a staple of Web applications, using object-oriented design techniques on the server side certainly makes a lot of sense. As always, by using Java, one can ensure that code remains portable across platforms as well as Web servers, as long as the Web server supports the servlet API. This list includes Sun's Java Web server, Apache Server, Web Logic Application Server, IBM Internet Connection Server, and Lotus Domino Go Webserver. Third-party add-ons are also available to enable servlet support on Web servers that only support the CGI protocol. Given the advantages of Java servlets, it is likely that they will play a key role in the development of E-commerce applications.

Java in Consumer and Embedded Devices

Discussion thus far has revolved around the use of Java on desktop or server machines that support various operating systems such as the Windows suite, UNIX, or Mac. From the discussion in the previous section, it can be seen that in order to run Java on one of these platforms, one needs the Java Virtual Machine, combined with the core Java language and its supporting classes. These are commonly referred to as the Java application environment (AE) or the Java platform.

In keeping with the original intent of developing Java (i.e., to develop software that could be embedded in devices), there are specifications available for developing and running Java programs on embedded and consumer devices. Examples of such devices include pagers, televisions, telephones, personal digital assistants (PDAs), and routers. In order to serve the needs of these smaller devices, which typically have lower processor speeds and considerably less memory, there are three other Java application environments or platforms available for use with different types of devices. They are the Personal Java, Embedded Java, and Java Card platforms. Each of them is tailored for a particular category of devices and applications. Essentially, each of these platforms uses a subset of the functionality of the Java API and has a corresponding virtual machine tailored specifically for it. It should be noted that this does not violate the Write Once, Run Anywhere model. Programs written for a device that supports one of the above platforms (e.g., a cell phone) is completely portable to other devices that support the same platform (e.g., a desktop phone), if desired. Note that upgrades or changes in the hardware of these devices will not affect the software as long as the application environment can be supported on the new hardware. Also, development of code for all the platforms described can be done using the JDK or other development environments (see the next section).

The Personal Java platform is intended for use by devices that will be used by the business consumer. Such a consumer will possibly need to run a variety of applications such as a mini Web browser, e-mail, calendar software, etc. on the device. Examples of devices that the Personal Java platform is intended for include PDAs, Internet-enabled televisions, advanced (multi-purpose) mobile phones, etc. The Embedded Java platform is intended for use with devices such as cell phones, pagers, printers, and networking devices such as hubs, routers, and switches. These devices have very specific applications needs and functionality. The Java smart card platform is intended for use in devices with severely limited memory and processor speeds. Examples include the Java Smart Card and Java ring. These devices are the electronic equivalent of the modern-day plastic cards.

Jini

The ability to embed Java into different types of devices (using the various Java platforms described above), combined with the ubiquitous presence of networking technologies, presents a great opportunity for creating an entirely new set of uses for these electronic devices. Sun's Jini technology represents the first step in facilitating the development of such new applications. Sun introduced the Jini connection technology in January 1999. The emergence of this technology is a measure of the strides that Java has made, both in its popularity as well as its maturity.

Jini technology allows one to plug in different types of devices dynamically into a network. Devices that connect into a network publish the kinds of services they can provide to a lookup server. The process of looking up a server and registering services is done automatically. Jini defines the protocols that are used by the lookup server as well as the discovery and join protocols used by the individual devices. Jini technology is in turn based on Java and RMI (see the previous section). Thus, devices can be connected to the network with virtually no setup work. Devices can also join and leave the network at will without affecting the rest of the network. All that is needed is a connection to a network, a lookup server, and devices that implement Jini technology and are networkable.

Assume that one wants to connect a laptop to a digital video camera and control the video camera from a laptop. Further assume that both devices are network compatible; that a lookup server is in place on a network; and that the digital camera provides three types of services: move (up, down, left, right), take a snapshot, or take a series of pictures (at specified intervals). One way of setting up the two devices is to start by connecting the digital camera to the network. When the Jini-enabled video camera is connected to the network, it would find the lookup server and then register its services (along with their interfaces) with the lookup server. Because each device can be treated as an object, the details of how a task is performed are known only to the individual device. Other devices, for example the laptop, need only be concerned with the services and interfaces provided by the device. If they want to use a particular service, they would simply need to check with the lookup server on their network to see if any devices providing the services they need have registered with the server. Thus, in the example, all one needs to do is connect the laptop to the network. The laptop would find the lookup server and register itself with it. Because the laptop is a consumer of services (provided by other devices), the lookup server would provide a list of devices offering services (e.g., the digital video camera) to the laptop. The user of the laptop can then select the digital video camera from this list. This would cause the services and interfaces (Java objects) registered by the digital video camera to be downloaded from the lookup server, which then enables the laptop to communicate and use the camera through the network.

The key points in the example above are: (1) it was not necessary to set up the laptop with the drivers for the video camera. The laptop only needed to be concerned with the services provided (which it was able to download from the lookup server). The actual implementation of the service (the driver) is on the video camera. (2) The devices were able to "self-connect" themselves to the network thus providing true plug-and-play technology. (3) One did not have to be concerned with any network operating systems issues. As long as the devices (including the lookup server) were able to run Java and Jini, it was possible to hook them up to the network and use them.

Will Jini eliminate the need for complex operating systems? Will it eliminate the need to store separate device drivers for each device that one wants to use and eliminate the need to update device drivers on a regular basis? Jini technology promises to do all of the above. However, it requires that vendors that produce different types of digital devices embrace the technology and make their devices Jini enabled.

JAVA DEVELOPMENT ENVIRONMENTS

What choices are there if one wants to develop Java applications? What does it take to run Java on a machine? What does it take to run Java on a browser? This section provides answers to these questions and more.

The easiest and cheapest way to develop Java applications on a machine is to download the JDK from Sun. The JDK is completely free and it comes with everything needed to create Java applications. The drawback is that the JDK is entirely command line oriented.

Serious application development in Java will, however, require the use of a graphical integrated development environment (IDE). Many vendors such as IBM, Symantec, Borland, Sybase, and Microsoft provide IDEs that provide a graphically oriented environment for developing, debugging, and testing Java applications. IBM's product is called Visual Age for Java, Symantec's Java toolkit is Visual Café, Borland's is Jbuilder, Sybase's is called PowerJ, and Microsoft produces Visual J++. Each of these products has several editions, such as professional edition, database edition, standard edition, etc., that are targeted toward various types of applications developers. The version number of the product does not always reflect the version of Java supported by the IDE. Thus, it is important to check if the IDE supports 1.1 or 1.2 (or whatever version of Java one is trying to produce applications for).

If one is not interested in developing Java applications but is simply interested in running Java software on machines, all that is needed is the Java Runtime Environment (JRE) for the platform. The JRE version numbers coincide with releases of Java. For example, JRE 1.2.1 supports Java 1.2.

In order to run Java-enabled Web pages in a browser, the browser software needs to support the right version of the Java Virtual Machine. Since until recently the timing of browser releases was usually not in sync with Java's releases, ensuring compatibility between the browser and applets has been a difficult proposition. For example, Netscape version 4.0 only had support for Java 1.0 (Netscape 4.5 supports version 1.1). This means that applets developed using JDK 1.1 could not be run on the browser. However, Sun has now released the Java plug-in that enables one to run Java using the plug-in and the corresponding Java runtime environment instead of the version of Java supported by the browser. Using the plug-in will enable users to eliminate any compatibility problems due to conflicts between the version of Java used to create applets and the one supported in the browser.

CONCLUSION

Java and its supporting technologies had a significant impact on computing in the 1990s. For the first time, it was possible to develop applications for devices ranging from powerful workstations to digital cards, storage devices, high-end Web-enabled phones, and pagers using a common language — Java. The use of this easy-to-learn, object-oriented language results in code that is robust, portable, and easy to reuse and maintain. Writing code in Java enables developers to take advantage of the Write Once, Run Anywhere paradigm. For an IS project manager, the features mentioned above mean that the overall cost of systems development — especially the cost of implementation, maintenance, and porting — will be lower. The portability of Java also means that IS organizations can develop software applications without making a long-term commitment to a particular Web server, database system, operating system, or hardware platform.

Early fears about Java that revolved mainly around the security and speed of Java applications have been put to rest. Applications written in Java are (for most practical purposes) as fast as applications written using popular languages such as Visual Basic, PowerBuilder, C/C++, etc. Using Jini technology and current networking infrastructure, it has also become possible to plug in various kinds of devices as needed into a network and facilitate cooperation among these devices. Educational institutions and businesses are beginning to produce sufficient numbers of educated developers who can harness the power of the language. As long as Sun and its partners focus on the strengths of Java and continue to strive to bring Java into the mainstream of both systems and business software applications development, the future of Java and Java-enabled technologies looks very bright.

Notes

1. Write Once, Run Anywhere and 100% Pure Java are trademarks of Sun Microsystems, Inc.
2. It is interesting to note that JDBC is not an acronym for Java database connectivity, although that is what it essentially does.

References

1. A-Z Index of java.sun.com: http://java.sun.com/a-z/.
2. Java Platform Documentation: http://java.sun.com/docs/.
3. James Gosling and Henry McGilton, *The Java Language Environment: A White Paper;* http://java.sun.com/docs/white/langenv/.

ABOUT THE AUTHOR

Ramesh Venkataraman is an assistant professor in the Department of Accounting and Information Systems, Kelley School of Business, Indiana University, Bloomington, Indiana. His e-mail address is venkat@indiana.edu.

Chapter 40
Java and C++: Similarities, Differences, and Performance

Adam Fadlalla
Paul J. Jalics
Victor Matos

Object-oriented programming (OOP) is an improvement in programming technology, and C++ is the most visible and commonly used OOP dissemination vehicle. However, C++ drags with it a lot of the "history" of C, which includes assumptions from the 1960s that a systems implementation language needs to be completely "open" and (consequently) "sloppy" so that programmers can implement, inside the confines of an operating system, all the artifices and "magic" that might be desired.

While this position for total openness could be arguably justified, it clearly brings with it an undesirable amount of language inconsistencies, ambiguities, exceptions, and clumsiness. On the opposite side, an experience from Xerox's Mesa language indicates that a strongly type-checked language can go a long way in avoiding or eliminating execution-time errors. C++ was intended to go in this direction but backward compatibility to the very important C language left a great deal of the "slop" that was in C remaining in C++.

To make matters worse, the architecture of the initial C++ implementations was geared to converting the C++ source programs to C code so that existing C compilers could be used for compilation. As a result, a number of unwanted side-effect anomalies can occur. For example, it is possible in C++ to cast any variable or pointer to anything else; functions without

0-8493-1161-6/02/$0.00+$1.50

a return type may still return an integer; pointers are most often used without checking whether they were initialized; subscripts for arrays are not checked; and # preprocessor commands and variables can generate a multitude of erroneous code.

In addition to all these problems, there is one more disturbing factor: C++ is a compiled language that interacts with "real" computers. For each real computer, there is an OS interface to C++, and unfortunately, it is different for each operating environment! Try to take a Borland C++ GUI application and convert it to Microsoft C++. It is akin to starting over, in one sense, especially if the original author did not design the application to be portable.

This is much more so than in newer programming environments, like the two mentioned previously, because the integrated debugging environments (IDEs) and their "wizards" often generate a good deal of C++ source code. The programmer then inserts his application code in the midst of all the automatically generated code.

In summary, C++ is an extraordinary programming language that is unfortunately too complex, too error prone, and too difficult to teach/learn because of its wealth of features. The authors believe the intent of the Java developers was to retain the important features of C++ while eliminating unnecessary complexity and error-prone language features. At the same time, Java introduces some new techniques that are superior to its C++ counterparts; for example, all pointers are eliminated, all objects are always passed by reference, and address arithmetic is forbidden.

DIFFERENCES BETWEEN C++ AND JAVA

Java is different from its predecessor programming languages (C, C++, Modula, Ada) in several ways:

- Java is a small language; however, it provides support for object-oriented programming.
- Arrays are the only predefined data structure type offered by Java, yet they provide access to classes that could be used to support any user-defined data structure.
- Java includes a rich Graphical User Interface (GUI) and multimedia facilities (sound, images, animation). Java is intrinsically GUI oriented. Its applications — called applets — are displayed by a Web browser rather than directly delivered to the computer's monitor.
- Java provides a self-contained environment for concurrency control, as well as a set of primitive programming facilities to support networking operations.

The Exhibit 1 table takes a piecemeal approach to comparing the two languages. Concepts from a C++ standpoint are mapped into their equivalent versions in Java. Even though the list is not exhaustive, it contains many of the building blocks needed for a good perspective of the differences between C/C++ and Java. Exhibit 2 summarizes the Exhibit 1 table.

Exhibit 1. A Comparison of C/C++ and Java

C and C++	Java
Program Organization:	
C and C++ programs normally consist of two parts: a header file (.h) containing the class definition and a (.cpp) file containing the implementation of the class. However, one can write very unobvious C++ programs using the preprocessor commands: *#define, #ifdef,* and *typedef* constructions.	Each complete logical unit of Java code is placed alone into one single piece of code (.Java file). This creates a single source for a class. The disadvantage here is that one may not be able to get a good overview of the class declaration as it often spans many pages. Projects are made by combining several independent Java files together. There is no preprocessor activity. There are no header files. *#ifdef, #include, #define,* macros, and other preprocessor commands are not available.
Constants:	

C/C++ constants have no explicit type; they could be defined in a global mode.
#define MILETOKMS 1602
#define PI 3.141592
class circle {
  double radius;
  circle(double r) {radius= r; };
  double area() { return (2*PI*radius); }
};
Note: (1) semicolon at the end of the class is needed.
(2) #define variables are global.
	The data type of Java constants must be explicitly defined. Constants must be made *public* and placed inside a class.
	class circle {
	  //constants
	  static final double MILESTOKMS = 1602;
	  public static final PI = 3.141592;
	  //variable
	  double radius;
	  //method
	  circle(double r) {radius= r; };
	  double area () {return (2*PI*radius);};
	}
	Note: (1) semicolon at the end of class is not needed.
	(2) public constant *circle.PI* is global.

Global Variables:	
Variables, constants, and data types could be made *global* by positioning them outside of a class definition or the implementation of a class's methods.	Java has no global variables, so all variables must be defined in some class. This implements more fully the object-oriented nature of the language. There are *static* variables just like in C++ and if they have a *public* access, they can be used like global variables with a class prefix. For example: myCompany.phoneNumber; myCompany is the class, and phoneNumber a public static variable.

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
Structures:	

C++ structures offer a rude method of providing *data encapsulation.*
struct EnglishDistance {
int feet;
double inches;
};
An instance is defined as:
struct EnglishDistance myboat;
	In Java, there are no structures. Therefore, C++ structures must be converted into Java classes:
	class englishDistance {
	int feet;
	double inches;
	}
	An instance is defined as:
	englishDistance myBoat = new myBoat();

Functions and Methods:	

C/C++ programs are collections of functions. There is a distinguished function called *main,* which is the first to be executed. C++ supports a heterogeneous mix of both methods inside classes and C-like functions.
double calculateCircleArea (double radius)
{
return (2 * PI * radius);
}
Used in the context:
Double r1=5.4;
Area1 = calculateCircleArea (r1);
	Java has no functions. All the activity provided by functions must be written as methods inside a class. Methods in a function need to receive a fixed number of parameters. Java functions that return no value must be declared as returning void.
	Class circle {
	double radius;
	circle (double r) {radius = r; };
	double calculateArea()
	{return (2*myConst.PI * radius);};
	}
	Used in the context
	circle circle1 = new circle(r1);
	Area1 = circle1.calculateArea( );
	here circle1 is defined as an instance of the circle class, with an original radius r1.
	All Java methods are *virtual* so that the actual function called is determined at execution time. Also, the final modifier declares that a method may not be overridden by subclasses.

Access Modifiers:	
Visibility (*Hiding*) of data and member functions inside a class is controlled using access specifiers: private: can be accessed only *within* the class public: visible outside of the class as class.part protected: similar to private but accessible to the methods of any derived class	Java uses the concept of *packages.* A package is a container for classes and other packages. A class is a container of data and code. Java uses combinations of the access modifiers: public, private, and protected, but it adds more control in visibility by ruling how classes and packages interact with one another. Exhibit 2 summarizes the scoping of variables and functions.

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
Automatic Typecasting:	

C++ allows coercion in which loss of precision could occur. For instance, assigning a double to an integer (losing decimals) is a valid C operation:
int age = 20;
double dogAge = age / 7.0;
// next is a valid assignment in C++
// however it may produce a warning msg
age = dogAge;
	Java forces the programmer to *explicitly* typecast assignments operations in which a loss of representation may occur:
	int age = 20;
	double dogAge = age / 7.0;
	//explicit coercion is needed below
	age = (int) dogAge;

Operator Overloading:	

C++ offers operator overloading, which in many cases offers a great degree of elegance in dealing with data objects. Consider the C++ example:
class eDistance { //English Distance
private:int feet;
double inches;
public:
eDistance(int f, double I)
{feet= f; inches= i;};
eDistanceoperator + (eDistance d2) {
inches+=d2.inches;
feet+= d2.feet;
if (inches >12) {
feet++;
inches-=12.0;
};
return eDistance(feet,
inches);
};
};
which could be used in the context:
d3 = d1 + d2;
where d1, d2, d3 are instances of eDistance.
	Java does not support operator overloading. Overloading is implemented with normal methods that are functionally equivalent to the action of the operator.
	Class eDistance { //English Distance
	intfeet;
	doubleinches;
	eDistance add (eDistance d2)
	{
	double sumInches = inches +
	d2.inches;
	double sumFeet = feet + d2.feet;
	if (sumInches >12) {
	sumFeet++;
	sumInches-=12.0;
	}
	return new eDistance(sumFeet,
	sumInches);
	};
	}
	Used in the context
	d3 = d1.add (d2);
	where d1, d2, d3 are instances of English Distance.

Strings:	

C/C++ does not include support for strings. Those must be either treated as a user-defined object or a zero-terminated array of characters. For example:
#include <string.h>
char Name[20];
strcpy(Name, "Don Quixote");
size = strlen(Name);
	Java offers strings as *primitive* objects. Strings are part of the Java language specification; therefore, their behavior is rigorously well defined across programs.
	String Name = new String( );
	Name = "Don Quixote";
	size = Name.length( );

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
Input-Output Streams:	
C++ offers three standard streams: in, out, error, as well as the overloaded operators << and >>:	Java applications also support the concept of *streams*; however, there are no << or >> operators.

int age;
cout << "Enter your age: ";
cin >> age;
cout << "you said " << age << " years";
	int age;
	System.out.println ("Enter your age: ");
	System.out.flush();
	age = System.in.read();
	System.out.println("You said " +age+" years");

Command-Line Arguments:	
C/C++ passes an integer argc, indicating the total number of arguments present in the command line, and argv[ ], which is an array of pointers to chars:	Java applications use a *string* collection to pass parameters. Path/Prog names are not displayed.

int main (int argc, char* argv[ ]) {
for (int i=0; i<argc; i++)
cout << argv[i];
};
	public class test2 {
	public static void main (String[] arg) {
	for (int j=0; j<arg.length;j++)
	System.out.println (arg[j]); } ; }

Friends and Packages:	
C++ classes use *friend* functions to act as a bridge between two unrelated classes, or to improve readability by providing a more obvious syntax. When another class declares your class as a friend, you have access to all data and methods in that class.	Java has no *friend* functions. This issue has been controversial in the design of C++. The Java designers decided to drop it. However, Java variables that are (1) in a class which is part of a package, and (2) introduced without an access modifier, acquire the *friendly* access type of the package. Those variables could be called without *get* and *set* functions from other classes in the same package.
Inheritance:	

C++ provides multiple inheritance. An object has access to all the data and methods of its ancestors.
class student : public person,
public athlete,
public intellectual
{
// derived class definition
...
};
	Java implements single inheritance. Each object has only one ancestor; however, *interfaces* could be used to support dynamic method resolution at runtime. Java *interfaces* are like classes, but lack instance variables and their methods contain only a signature but no body. An interface is similar to a C++ abstract class that specifies the behavior of an object without specifying an implementation. A class may include any number of interfaces. The body of referenced methods must be provided in the subclass.

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
Inheritance: (Continued)	

The student class is derived from the three ancestor classes: person, athlete, and intellectual. Any data and methods defined in the upper classes are reachable from the subclass.	class student implements person, athlete,
	intellectual
	{
	// sub-class definition
	...
	}
	Keyword *implements* is used to introduce the names of *interfaces* other than the primary ancestor.

Basic Data Types:	
C++ offers the following primitive data types (given with their field widths in bits): char:8, int: 16, long:32, float:4, double:64, long double:80, unsigned char:8, unsigned int:16, unsigned long:32 C++ also has bitfields, enumerated types, unions, structs, and typedef to enrich the possibilities.	The following built-in data types with field widths in bits are available: boolean:1, byte:8, char:16, short:16, int:32, long:64, float:32, double:64. Please note that char is 16 bits wide and supports the international Unicode standard for all foreign alphabets, including Japanese and Chinese. Java has no *enumerated type,* no *bitfields,* no variable number of arguments for functions, no *struct,* no *unions,* no *typedef.*
Strings:	

C++ has no direct support for strings. Programmers must use zero-terminated arrays of characters or include add-in libraries (MFC, OWL, STL) to provide additional facilities to handle strings:
#include <string.h>
...
char myStr [80];
strcpy(myStr, "Hello");
strcat(myStr, " World");
This example uses an array of null-terminated chars, and the <string.h> library to support strings.
	Java uses the built-in class *string* to replace most uses of null terminated character strings as used in C++.
	class Greetings {
	public static void main(String args[ ])
	{
	String myStr; //a reference to the string
	myStr = new String("Hello World");
	myStr = "Hello" + " World"; //assign concat
	}
	}

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
Arrays:	

An array definition defines a starting address and reserves storage for allocation
`int deptNumber[4];`
`int dNumb[] = {10, 20, 30, 40 };`
	Arrays are defined in two steps. First, a reference to the repeating group is made, then new is used to allocate storage and define the array's size
	`int deptNumber[ ];`
	`deptNumber = new int[4];`
	`int deptNumb[ ] = new int[4];`
	`int dNum[ ] = {10, 20,30,40};`
	Arrays that have no claimed space using the new command are set to *null.* Since array subscripts are always checked, the most common form of program failure in Java is trying to use a null reference type that is caught by the Java virtual machine.

Pointers:	

C and C++ make extensive use of pointers. The addressing and dereferencing operators are used to signal a pointer to data and the value of such address:
`int Number = 10;`
`int *ptrNumber = &Number;`
This example has no equivalent translation in Java.
	Java has no pointers and thus no pointer arithmetic. Referencing and dereferencing operators (*, &) do not exist. There are, however, *references*, which is a safe kind of pointer.
	`BTreeType mytree; //reference`
	`mytree = new BTreeType( );`
	`   //initialization`
	Both arrays and objects are passed by *reference.* When one creates an instance variable of a class (BTreeType mytree;), it is just a reference variable to an instance of that class. The instance variable *must be initialized* with a new operator (mytree = new BTreeType( );) to create an actual instance of that class. The default value of all objects and arrays (i.e., reference types) is *null* (which means an absence of a reference).

Operators:	
There is a large number of arithmetical and logical operators such as: ++, – –, +, –, *, /, %, &&, !, | |, etc.	Java supports almost all C++ operators, including bit-wise logical operators, and has minimal extensions.
Flow Control:	
Typical flow control structures are: if [else], for, while, switch, break, continue, goto.	Same as C/C++ with the exception of the goto statement, which is not implemented. Java also includes a synchronized statement to protect critical sections of multithreaded applications.

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
OS Utilities:	
Input/output operations are not intrinsically defined in the language, but rather they are acquired using the standard C libraries (stdio.h, stdlib.h, etc.). Another alternative is to use the more powerful set of classes or class libraries provided by the Microsoft MFC and the Borland OWL. However, those libraries are different and incompatible for every compiler.	Java has designed into it a set of predefined and standardized classes for *I/O, Graphical User Interface (GUI)* access at a high-level, *networking, multiprogramming* with threads, and other operating system services.
Linking:	
In C++, one typically links the application into an executable (myBtreeApp.exe), which is all loaded into memory when the program is started.	Uses a built-in *dynamic linker* and it is common to load new .class files in the middle of execution. Java has no .exe form, but is instead a *collection* of .class object files. Thus, Java .classes are often downloaded from the Internet and executed with subsequent parts (.class modules) downloaded as needed.
Executing:	
C++ is compiled into highly efficient machine language.	Java is typically *interpreted;* thus, Java has (for the present) compromised performance to get portability.
Security:	
C++ has an infinite number of ways in which its integrity and security can be defeated.	Java has a *Code Verifier* that checks each .class module before it is loaded to verify that it is well behaved and obeys the basic rules of the Java language. The *Class Loader* maintains a list of classes it has loaded, as well as the locations from which they came. When a class references another class, the request is served by its original class loader. This means that classes loaded from a specific source can be restricted to interact only with other classes retrieved from the same location. Finally, the *Security Manager* can restrict access to computer resources like the file system, network ports, external processes, memory regions outside of the virtual machine, etc.
Internet:	
C++ was not designed with the Internet in mind; therefore, there is nothing directly connecting them.	Java provides a GUI "interpretable" code-type called an applet. Applets are applications that can be *run in a standardized environment,* which happens to be provided by a browser such as Netscape or MS-Internet Explorer. Therefore, one can develop a GUI application that can run on the Internet browser of any computer.

Exhibit 1. A Comparison of C/C++ and Java (Continued)

C and C++	Java
Miscellaneous:	
• Debugging code can be included in the source directly and conditioned on constant variables; thus, the Java compiler will remove the code if these variables are not set. • One does not need a *destructor* member function very often in Java because returning new and other heap storage is done automatically by Java when it is no longer pointed to by any reference types. • Java exception handling is similar to C++ but not exactly the same. The try block is followed by potentially several catch functions, and a clean-up function finally, which is executed in any case. The throw statement generates an exception, and a throws clause in a function declaration indicates that a specific exception might occur in that function. This policy promotes error information to the same level of importance as argument and return typing. • C++ classes sometimes have a "fragile base class" problem that makes it difficult to modify a class that has many derived classes; changing the base class may require recompilation of the derived classes. Java avoids this problem by dynamically locating fields within classes.	

Exhibit 2. Summary of Differences between C/C++ and Java

	Access modifiers applied on a variable or member function				
Data hiding object	**Private**	**No modifier**	**Private protected**	**Protected**	**Public**
Same class	Yes	Yes	Yes	Yes	Yes
Same package subclass	No	Yes	Yes	Yes	Yes
Same package nonsubclass	No	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes	Yes
Different package nonsubclass	No	No	No	No	Yes

IMPACT OF THE MISSING C++ FEATURES IN JAVA

With about 10,000 lines of Java programming behind them and 7+ years of C++ programming, the authors of this chapter hazard a guess as to the impact on Java's potential of some of the features of C++ that are missing in Java:

- *Multiple inheritance.* While there are many examples of cases where multiple inheritance might be the "best" solution, the concept is not totally clear, and at least, it is bothersome. In their experiences, multiple inheritance has not been used very much in C++, so the impact of losing it should be minimal. Nonetheless, there is the possibility of implementing multiple inheritance in Java using interfaces.

- *The C++ preprocessor* is certainly the source of many possible errors, but it is also a powerful tool for managing different versions, such as the performance tests described in the following. For example, the timing code in the benchmarks was inserted into each benchmark using #include. In Java, these were simply copied manually into each benchmark (when a bug was discovered in the timing code, it was time to hand-edit 19 different versions of the same code). Also in the C++ timing code, an array of structures was used in which the test procedure address was stored along with the repetition count, etc. In Java, this could not be accomplished and so another procedure layer needed to be inserted that used a switch statement to call any of the tests in the current group with appropriate parameters.
- *Pointers.* While it is true that C/C++ have traditionally relied heavily on "pointer logic" to access data and manipulate parameters, it is also clear that a significant portion of the program development effort is related to cleaning up that type of logic. Dangling pointers and long chains of indirect addressing are by far the most perturbing and error-prone elements of C/C++ programming. Certain retraining might be needed by the C programmer to adjust to a pointer-free environment. Notice, however, that the concept of *reference* to an object is somehow similar to the pointer notion, just simpler.

There are not many other limitations that will imperil the C programmer from making a smooth transition to Java.

JAVA PERFORMANCE CHARACTERISTICS

Among the greatest concerns in Java is execution performance: Java is designed to be *interpreted* — rather than executed. Java .class "executable" programs are pieces of machine code for a nonphysically existent computer. This hypothetical computer is totally implemented in software, and is called the *Java Virtual Machine.* In order to execute .class programs, one can use a computer program (a Java-enabled browser such as Netscape, or an emulator such as Jview.exe) that interpretatively executes the machine instructions in the .class files.

The industry rule of thumb for the execution speed of interpreted code is that it is slower by an *order of magnitude.* While it is true that computer hardware speeds are becoming ever faster, the appetite for solving larger and more complex problems means that the execution speed is still very crucial.

A number of Java benchmarks were found when looking at the issue of execution evaluation with the Sun Microsystems White Paper. Most of those studies were meant to compare the following:

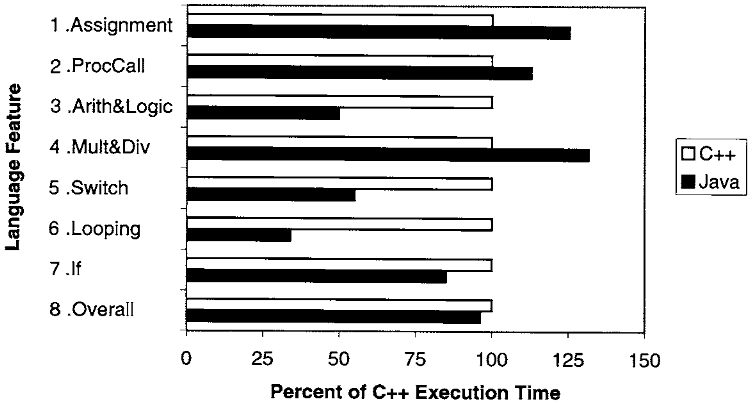


Exhibit 3. CPU Language Features: Java versus C++

- The speed of one Java implementation versus another.
- The relative merits of different Internet browsers when executing applets.
- The rating of different hardware platforms when executing the same code. However, just a few of those benchmarks made some comparison between C++ and Java.

An exhaustive analysis of the two languages on a feature-by-feature basis ensued. The focus was to obtain a gross evaluation of how far apart one tool is (C++) from the other (J++). A number of C++ benchmarks programs that were written earlier to study C++ performance were converted to Java. All benchmarks were in C++ first and then again in Java to study the relative performance of C++ versus Java.

The performance tests were executed on the same computer, a Pentium 133-MHz laptop with 32MB of memory, running Windows 95. The software systems were Visual C++ 4.0 compiler from Microsoft and Microsoft Visual J++ version 1.1.

A battery of tests, called the *X tests,* measure CPU execution speeds of different language features (see Exhibits 3 and 4). Both the Java and the C++ test programs were executed using a non-GUI interface. C++ code in the form of .exe files targeted a plain DOS platform, while the Java programs were produced as "applications" to be executed by the JVIEW emulator.

As summarized in Exhibit 3, a test of CPU language features, Java versus C++, revealed the following:

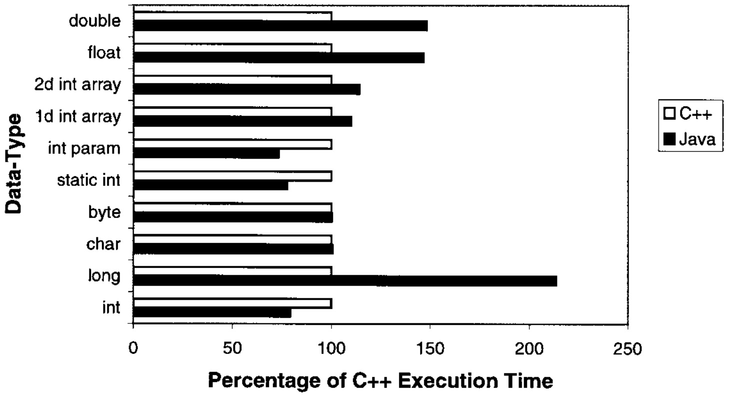


Exhibit 4. Java Data-Type Performance versus C++

- A simple assignment is 25 percent slower in Java than in C++.
- Procedure call overhead in actual time is 13 percent slower in Java when compared to C++. However, in C++, a procedure to add two integers executes 8.1 times as long as the "in-line" assignment c = a + b; in Java, this factor is 2.1; thus, the additional overhead for the procedure call in this case in Java is one fourth as much when compared to C++.
- Arithmetic, logical, and shift statements appear overall to be about 50 percent faster in Java.
- Multiply and Division statements (*, /, %) appear to be about 32 percent slower in Java as compared to C++.
- The switch statement performance in Java is 45 percent faster than in C++.
- Looping statements (for, do while, while) appear to be 3 times as fast in Java as in C++.
- If statement execution times in Java are faster by 15 percent.
- The overall execution time of Java is about 4 percent less than the corresponding C++ measurement.

The only surprise here is the low procedure-call overhead in Java, and the insight that the performance of the main language facilities in Java is comparable to the known-to-be-efficient C++.

As summarized in Exhibit 4, Java data-type performance testing found that:

- Double-precision floating point in Java is 52 percent slower than in C++.

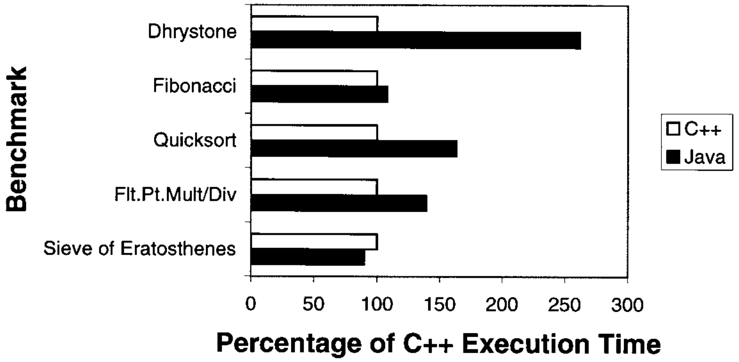


Exhibit 5. Routine-Level CPU Benchmarks

- Java single-precision floating point is 53 percent slower than in C++.
- Two-dimensional array access in Java takes 14 percent longer than in C++.
- Accessing elements of a one-dimensional array in Java takes 10 percent longer than in C++.
- Int parameters in Java take 27 percent less execution time than in C++.
- Statements involving Java static variables perform 22 percent faster than in C++.
- The Java byte data-type (8-bits) takes about the same execution time as the 8-bit char types do in C++.
- The Java char data-type (16-bits) take about the same time as the C++ char (but the C++ 16-bit short takes 6.5 percent more execution time than the Java char).
- Java statements involving long integers take 113 percent more execution time than do C++ long integers. Note that Java ints are 32 bits and Java longs are 64 bits, whereas C++ ints and longs are each 32 bits on the Pentium chip.
- Java ints perform 21 percent faster than integers in C++.

All in all, paying attention to the choice of data types can have a significant impact on performance. The single, striking example is the use of long integers, which has no performance penalty in C++ on the Pentium, but is over twice as slow in Java, where it is a 64-bit "super long."

As summarized in Exhibit 5, routine-level CPU benchmarks found that:

- The Dhrystone benchmark, which is designed to be typical of the CPU part of what programs do (no I/O), takes 2.61 times as much time as in C++.
- The highly recursive Fibonacci number calculation performs 8 percent slower in Java when compared to C++.

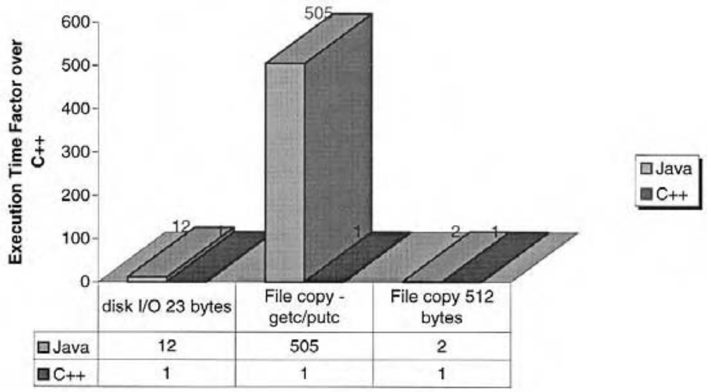


Exhibit 6. Routine-Level I/O Test

- The QuickSort sorting algorithm performs take 63 percent more time in Java when compared to C++.
- A floating point multiply and divide sequence takes 39 percent more time in Java when compared to C++.
- The Sieve of Eratosthenes code takes 10 percent less time in Java when compared to C++.

These measurements simply confirm that the measurements in the first two sections can be reaped in a routine that does some specific useful task.

What is also clear from the tests previously discussed is that unless one pays extraordinary attention to I/O performance, there is little hope of getting a reasonable whole-program execution time when compared with the possibilities in C++.

As summarized in Exhibit 6, routine level I/O tests discovered the following:

- Writing 23-character records to a disk file takes 12.2 times as long in Java as in C++.
- Copying a file using readByte/writeByte of a Random AccessFile takes 505 times longer in Java than the corresponding getc/putc commands in C++.
- Copying the same file by using 512-byte write statements in Java takes only 56 percent more time than in C++.

What is clear from the preceding tests is that unless a programmer pays extraordinary attention to I/O performance, there is little hope of getting a reasonable whole-program execution time when compared with what is possible in C++.

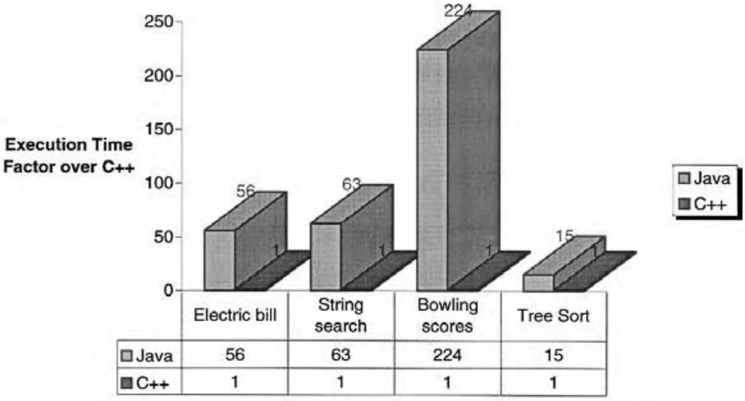


Exhibit 7. Simple Whole-Program Benchmarks

As summarized in Exhibit 7, a simple whole-program benchmark found that:

- An electric bill processing benchmark takes 56 times longer in Java than in C++.
- A string search of a file benchmark takes 63 times longer in Java than in the C++ version.
- A bowling scores calculation program takes 224 longer in Java as in the C++ version.
- A TreeSort benchmark takes 15 times longer in Java than in C++.

The actual performance factors here probably have a great deal to do with the I/O usage in the above benchmarks, for which improvement methods will be discussed in the following paragraphs.

The Ctax benchmark found the following:

- A Java procedure call of a static function takes only 60 percent of the time of a global function in C++.
- A normal Java member function takes 8 percent less time than in C++. Also, the normal member function in Java takes 45 percent more time than a static function.
- Since all Java member functions are virtual (i.e., exact function is decided at execution time), performance is identical to the above even if using polymorphism, overridden or final.
- The performance of passing parameters to a procedure is about 4 percent extra for each parameter used. (In C++, the factor is about 3.5 percent extra.)

- The performance of a new 4-byte class without member functions is identical to the corresponding C++ new.

The Screen Emulator Benchmark discovered the following:

- Summing up the contents of a CGA 4000 char screen buffer in Java takes 16 percent more time than in C++.
- Summing up a char[25][160] in Java takes 56 percent more time than above in Java (the increase in C++ is 27 percent).
- Summing up a Java class aspot {char a,b;} in an array of classes [25][80] takes 113 percent more time than item 1 (in C++ this takes 24 percent extra time).
- Summing up a Java class cga that has a member function caref (row,col), which returns a reference to an aspot class instance, takes 263 percent more time than item 1 above (in C++, this takes 216 percent extra).
- Accessing class variables of another class instance in Java is 33 percent slower than a local variable (in C++, it is 3 percent slower).

IMPROVING THE PERFORMANCE OF JAVA PROGRAMS

The purpose of the previous section was to give an idea of the overall performance of the Java language using C++ as a framework of reference. The results were mixed, however; they seem to indicate that Java can be as fast as C++, thanks to the intervention of Just-In-Time (JIT) compilation, which does a "mini-compile" into native code as the program is being executed. Looking at the results, the authors questioned their methods a great deal since *so many of the tests showed Java to be as fast or even faster than C++*. However, it was a relief to find that other experimenters have found similar results.

To compound this evaluation, the obvious issue that not all Java compilers/viewers are the same must be added. It can be speculated that most of the relative gains in performance appear to depend primarily on the combination of the Java compiler used to generate the bytecode, and the Java Virtual Machine interpreter utilized for execution. To prove this point, a combination of compilers and viewers were tried. Consider the following findings:

- One of the measurements yields an execution time of 0.155 seconds when compiled with the Symantec Java 1.0d compiler and executed with the Microsoft Visual J++ 1.1 Jview interpreter.
- If using the Symantec Java interpreter, the performance doubles to 0.327 seconds.
- If using the Microsoft Visual J++ compiler to compile and execute, the execution time doubles again to 0.675 seconds.

- If, on the other hand, Microsoft Visual J++ compiler is used to compile but run on the Symantec Java interpreter, the execution time is highest at 0.793 seconds.

A dramatic factor of the previous measurements is that while Java "looks good" in most of the measurements, it also provides a disconcerting wide range of variability. The execution factor Java/C++ was 2, 5, 10, 100, or even 500 in the small number of measurements. To shed some light on this issue, a few performance improvements on some of the benchmarks programs were attempted. For example:

- The electric bill calculation benchmark started with an execution time of 23.79 seconds. Suspecting that the I/O time was causing the problem, the input file of 8000 integers was replaced with a table in memory. This had little effect. The only thing left was to put a buffering into the print output file. This reduced the execution time by 53.5 percent, down to 11 seconds.
- The string search benchmark originally took 52 seconds (a factor of 63 times the C++ version), and was basically searching for a word in an input file and writing the matching line with some trivial formatting. Java's substring match facility was already in use, so there was not much that needed obvious improvement. Again, maximum file buffering was put into both the input data file and the output report. This had the dramatic effect of reducing execution time to 4.5 seconds (a factor of 11.9 times the C++ version).
- The bowling scores program took 47 seconds to execute (a factor 191 times the C++ version). For this program, there is no input file and the output report is only a dozen lines. Yet when maximum buffering for the report file was put in, the execution time decreased to 43.5 seconds (a factor of 177 times the C++ version). The rest of the program does some calculations on two-dimensional arrays.
- The Java program profiler was used to assess where the time is spent in individual applications. In a large medical application with many screens, the time to bring up a given screen for the first time was excessive and the profiler indicated that most of the time was spent in class initialization for the various classes involved. The reason for this was a great mystery and a workaround was attempted: create the classes involved right as the program starts, when even the identity of the patient is unknown. Sure enough, this substantially reduced the excessive screen switching times. It is speculated that the Just-In-Time compiler might be generating native code for member functions for the class as it is being constructed (under VisualCafe).

I/O PERFORMANCE SUMMARY

Just from these few experiences, one can see that I/O performance can be dramatically improved by simply using buffered streams for both file input, file output, and report output. Also, Java has a readFully method that reads an entire file in one I/O call. Now that virtual and actual memories are so large, this can be a dramatic way of achieving performance improvement — of course, at some cost! In one medical data application, bringing up a patient screen took 20 seconds — an eternity if one is staring at the screen. When the reading of the patient file was changed to readFully, the time was reduced to 0.4 seconds (an improvement factor of 50). Obviously, those applications deemed to be real-time critical will benefit tremendously from this "read-ahead" protocol; on the other hand, not all applications need such a solution.

CPU PERFORMANCE SUMMARY

The CPU-bound portions of a program can also be improved with the knowledge derived from the measurements of the different data types previously described. One should always be aware of the relative performance of different compilers, and Java interpreters, which are changing fast as the Java development platforms come of age. It is important to point out that *Java applications are nearly 100 percent portable between any platform, operating system, and development system.* The Sun development kit, Symantec VisualCafe, and Microsoft Visual J++ were used alternately, with no compatibility problems thus far.

CONCLUSION

Java is emerging as an important software development language for the present and the future. The combination of practical choices made in terms of language features for an object-oriented programming language, a standardized set of system interfaces for input/output, graphical user interfaces, networking, operating system facilities, and a platform-independent executable code will likely make it an important tool for some time to come.

While performance is a serious issue, our measurements show that Java bytecode behavior can approach executable C++ code in a very impressive way. However, cases in which much work is needed to make the gap between both languages shrink to a reasonable value were also observed. The authors have seen the development of some production level software that generally meets the performance requirements of an interactive Windows application. Also, the technology is still emerging, so there will likely be more optimization of the JVM code generated, compilers that generate native code, a steady improvement in the performance of Java Virtual Machine emulators and Just-In-Time optimization, and even the possibility of auxiliary hardware CPUs that implement the Java Virtual Machine.

Recommended Reading

Kernigham, Brian and Ritchie, Dennis. *The C Programming Language,* Prentice-Hall, 1978.
Fadlalla, A., Jalics, P., and Matos, Victor, "Portability of GUI Based Object-Oriented Applications," *Systems Development Management*, Auerbach Publications, accepted January 1997.
Gosling, James. "The Java Language: A White Paper," Sun Microsystems, 1993.
"The Java Language Environment: A White Paper," Sun Microsystems, October 1995.
"The CaffeineMark Java Benchmark" from Pendragon Software.
"Jmark 1.0 Java Benchmark," *PC Magazine*, January 7, 1997, Vol. 16, No. 1, p. 182.
Bell, Doug. "Make Java Fast: Optimize!" *JavaWorld*, http://www.javaworld.com/javaworld.
Hardwick, Jonathan. Java Optimization site: http://www.cs.cmu.edu/~jcl/java/optimization.html.
The Fhourstones 2.0 Benchmark (ANSI C,Java), tromp@cwi.nl.
"Jstones" by Sky Coyote (Java, C++), sky@inergalact.com.
Jalics, P. and Blake, B. "Performance of Object-Oriented Programs: C++," *Systems Development Management*, Auerbach Publications, 1992.
Bishop, Judith. *Java Gently*, Addison-Wesley, Reading, MA, 1997.
Naughton, Patrick. *The Java Handbook*, Osborne-McGraw-Hill, 1997.
Deitel and Deitel, *Java — How to Program*, Prentice-Hall, Upper Saddle River, NJ, 1997.

ABOUT THE AUTHORS

Adam Faldalla, Ph.D., is assistant professor of computer and information science at Cleveland State University.

Victor Matos, Ph.D., is associate professor of computer and information science at Cleveland State University.

Paul J. Jalics, Ph.D., is professor of computer and information science at Cleveland State University.

Chapter 41
JavaBeans and Java Enterprise Server Platform

David Wadsworth

A majority of the world's data resides on mainframe servers. This legacy poses many challenges to the information systems (IS) community as it struggles with the demands of business units for new and innovative solutions to business problems. Organizations need to adopt a flexible, secure, and cost-effective architecture that will enable them to remain competitive and enable breakaway business strategies. Adoption of Java™ computing realizes these benefits by providing key technology enablers.

JAVA TECHNOLOGY REVIEW

The Java programming language was introduced to the public in May 1995. Key features of the language, such as platform independence and ease of programming, made it an instant success in the software development community. Other features such as safe network delivery and baked-in security have made the language the de facto standard for the development and deployment of Web-based applications.

Applications written in the Java programming language are compiled to bytecode that can run wherever the Java platform is present. The Java platform is a software environment composed of the Java Virtual Machine and the Java Core Application Programming Interfaces (APIs). Portability of applications is achieved because there is only one virtual machine specification which provides a standard, uniform programming interface on any hardware architecture. Developers writing to this base set of functionality can be confident that their applications will run anywhere without the need for additional libraries. Core libraries include functional support for GUI development, I/O, database connectivity, networking, math, components (JavaBeans), multi-threading, and many others.

0-8493-1161-6/02/$0.00+$1.50

Sun's Java computing architecture is an implementation framework that uses standard, currently available network protocols and services to deliver the power of Java applications to the widest possible base of Java platform-enabled devices and users. With this architecture, transactions can be moved transparently to the most cost-effective, appropriate support channel within a network owing to the portable, Write Once, Run Anywhere™ nature of Java applications.

JAVA PLATFORM COMPONENT ARCHITECTURES

Designing and developing applications by means of components has been available for many years. The challenge has been to embrace and extend existing technology with new. Until recently such an approach has been proprietary and difficult to deploy. The Java computing environment with JavaBeans, a component technology, and server architecture solution, Java Enterprise Server, enables organizations to greatly simplify access to business systems. What follows is a description of the JavaBeans component model and an overview of the Java Enterprise Server platform.

JAVABEANS

A JavaBean is a reusable Java software component that can be visually manipulated and customized in a builder tool. These application building blocks are constructed so as to easily communicate with one another in a common environment. They also have the ability to store their state "on the shelf" to be revived at a later date. Since they are written in the Java programming language for deployment on any Java platform, JavaBeans are the platform-independent components for the network.

JavaBean components can range from simple GUI elements, such as buttons and sliders, to more sophisticated visual software components, such as database viewers. Some JavaBeans may have no GUI appearance of their own, but can still be manipulated in an application builder.

The JavaBean API has been designed to be accessible by builder tools as well as manually manipulated by human programmers. The key APIs such as property control, event handling, and persistence can be accessed by both handcrafted applications and builder tools. As well as event handling, property control, and persistence, introspection and customization are distinguishing features of all JavaBeans.

Property Control

Property control facilitates the customizing of the JavaBean at both design and runtime. Both the behavior and appearance of a JavaBean can be modified through the property features. For example, a GUI button might have a property named "ButtonLabel," which represents the text displayed in the button. This property can be accessed through its getter and setter methods. Once properties for a bean are configured, their state will be maintained through the persistence mechanism.

Persistence

The attributes and behavior of a bean are known as the state of the bean. The persistence mechanism within the JavaBean API supports storage of this state once the bean is customized. It is this state that is incorporated into the application and available at runtime. This externalization can be in a custom format or the default. A custom external format allows the bean to be stored as another object type such as an Excel document inside a Word document. The default is reserved for those instances where the bean's state needs to be saved without regard to the external format.

Event Handling

Event handling is a simple mechanism that allows components to be connected based on their production and/or interest in certain actions. A component or series of components can be sources of events that can be caught and processed by other components or scripting environments. Typical examples of events would include mouse movements, field updates, and keyboard actions. Notification of these events generated by a component would be delivered to any interested component.

The extensible event handling mechanism for JavaBeans allows for the easy implementation of the model in application builder tools. Event types and propagation models can be crafted to accommodate a variety of application types.

Customization

Changing the appearance and behavior of a JavaBean is accomplished through the customization features of the JavaBean's API. Each JavaBean contains a list of exported properties that an application builder can scan and use to create a GUI property editor sheet. The user can then customize the bean using this dynamically created sheet. This is the simplest form of customization.

Another layer of customization is possible by attaching to the bean a customizer class that acts as a properties wizard. This wizard will have a GUI which can be employed to tailor the properties for the related bean in a guided tour fashion. Such wizards are more likely to be found associated with complex beans such as calculator beans or database connection beans. Once customization is complete, the properties will be stored using the persistence mechanism.

Introspection

The properties, methods, and events a JavaBean supports are determined at runtime and in builder environments by means of introspection. Introspection is a prescribed method of querying the bean to discover its inherent characteristics. Introspection is implemented using the Java programming language rather than a separate specification language. Therefore, all of the behavior of the bean is specifiable in the Java programming language.

One introspection model supported by the JavaBeans API provides a default view of the methods, events, and properties. This simple mechanism does not require the programmer to do extra work to support introspection. For more sophisticated components, interfaces are available for the developer of the bean to provide specific and detailed control over which methods, events, and properties are exposed.

Default, low-level reflection of the bean is used to discover the methods supported by the bean. Design patterns are then applied to these methods to determine the properties, events, and public methods supported by the component. For example, if a pair of methods such as setColor and getColor are discovered during the reflection process, the property “color” is identified by the application of the “get/set” design pattern for property discovery.

More complex component analysis can be built into the bean by the use of a BeanInfo class. This class would be used by a builder tool to programmatically discover the bean’s behavior.

Security

JavaBeans are governed by the same security model as all other Java applets and applications. If a JavaBean is contained in an untrusted applet, it will then be subject to the same restrictions and will not be allowed to read or write files on the local file system or connect to arbitrary network hosts. As a component in a Java application or trusted applet, a JavaBean will be granted the same access to files and hosts as a normal Java application. Developers are encouraged to design their beans so they can be run as part of untrusted applets.

Runtime Versus Design Time JavaBeans

Each JavaBean must be capable of running in a number of different environments. The two most important are the design and runtime environments. In the design environment a JavaBean must be able to expose its properties and other design time information to allow for customization in a builder tool. In some cases wizards contained in the bean may be employed to simplify this process.

Once the application is generated the bean must be usable at runtime. There is really no need to have the customization or design information available in this environment.

The amount of code required to support the customization and design time information for a bean could be potentially quite large. For example, a wizard to assist in the modification of bean properties could be considerably larger than the runtime version of the bean. For this reason it is possible to segregate the design time and runtime aspects of a bean so it can be deployed without the overhead of the design time features.

JavaBeans Summary

JavaBeans are the component object model for the Java platform. These device-independent components can be customized and assembled quickly and easily to create sophisticated applications.

JAVA ENTERPRISE SERVER PLATFORM

As organizations adopt Internet technologies to enable new business strategies, they are faced with the task of integrating all of their legacy applications, databases, and transaction services with Web-based services. Traditional applications designed in the client/server model do not deploy well in an Internet/extranet environment. While not new, multi-tier architectures for application development and deployment are best suited for extending the reach of a company's infrastructure to partners, suppliers, customers, and remote employees. The Java Enterprise Server platform provides such an architecture in an open and standards-based environment that incorporates existing infrastructure while extending their reach to intranets, extranets, and even the Internet. An extensible architecture, the Java Enterprise Server platform contains the APIs, products, and tools necessary to construct new enterprisewide applications and integrate with existing systems.

Traditional mission-critical applications are written to the APIs of the underlying operating system, thereby tying the application to a single operating system. Porting of the application to a new operating system is both difficult and expensive. These same applications may rely on a service, such as a transaction monitor. Access to this service will be through the software vendor's proprietary APIs, creating another platform lock and presenting a barrier to moving to a different service provider.

The Java Enterprise Server platform is designed to address these platform-lock issues. It extends the notion of "Write Once, Run Anywhere" to include "and integrate with everything." Based on a layer and leverage model, the Java Enterprise Server platform can be built on top of existing legacy systems such as transaction monitors, database access, system

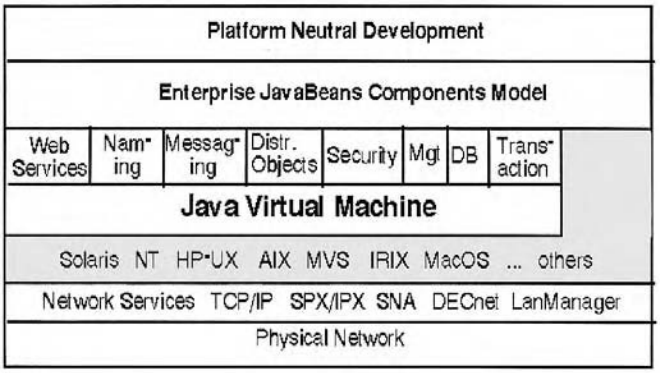

Platform Neutral Development
Enterprise JavaBeans Components Model
Web Services
Naming
Messaging
Distr. Objects
Security
Mgt
DB
Transaction
Java Virtual Machine
Solaris NT HP-UX AIX MVS IRIX MacOS ... others
Network Services TCP/IP SPX/IPX SNA DECnet LanManager
Physical Network

Exhibit 1. Java Enterprise Server Architecture

management, naming and directory services, and CORBA (see Exhibit 1). Interfaces to these services, as well as a component model that provides for application encapsulation and reuse, are integral to the Java Enterprise Server platform. The component model includes JavaBeans, components for the client, and Enterprise JavaBeans (EJBs) components for the server.

All of the benefits of rapid application development, scalability, robustness, and security of the JavaBeans component architecture are extended to the Java Enterprise Server platform. EJBs also have the ability to provide transactional services. Coupled with these benefits is an open architecture capable of providing ease of development, deployment, and management.

Enterprise JavaBeans, an extension of the JavaBeans architecture, provide a distributed component architecture for developing and deploying component-based, multi-tier applications. Business logic is encapsulated in the Enterprise JavaBeans promoting a high degree of reuse. Access to low-level services, such as session management and multi-threading, is simplified such that developers building applications do not need to deal directly with these functions.

Distributed applications developed with Enterprise JavaBeans can be deployed on any other platform without modifications. Support for transactions and messaging integrates with existing legacy systems and middleware.

The heart of the Enterprise JavaBean platform is the Enterprise JavaBean executive (see Exhibit 2). This runtime executive is used to execute the components that provide the services required by an application. Through its components, the executive manages load balancing, and handles multi-threading, transaction management, security, and connection

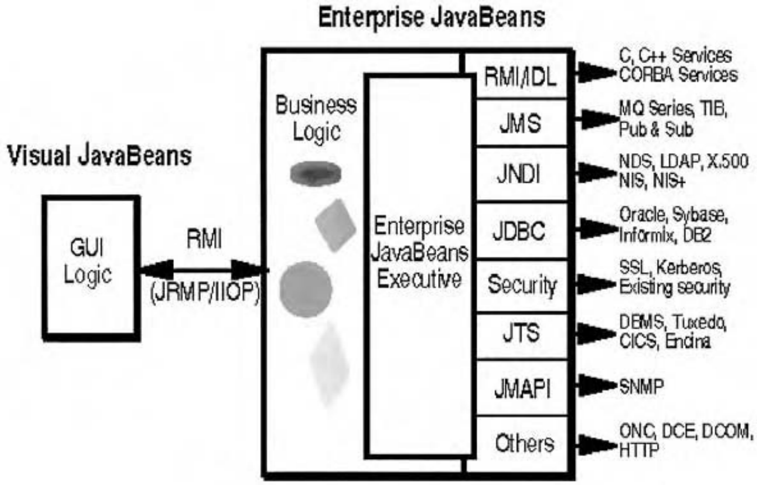

Enterprise JavaBeans
Visual JavaBeans
GUI Logic
RMI
(JRMP/IIOP)
Business Logic
Enterprise JavaBeans Executive
RMI/IDL
JMS
JNDI
JDBC
Security
JTS
JMAPI
Others
C, C++ Services CORBA Services
MQ Series, TIB, Pub & Sub
NDS, LDAP, X.500 NIS, NIS+
Oracle, Sybase, Informix, DB2
SSL, Kerberos, Existing security
DBMS, Tuxedo, CICS, Encina
SNMP
ONC, DCE, DCOM, HTTP

Exhibit 2. Enterprise JavaBeans Framework

management. This frees programmers so they can focus on developing the components that contain business logic.

Communication between the client and server in an application does not need to rely on any particular protocol. Both client and server sides of the application are coded using the Java programming language. At deployment time, the underlying communication stubs are generated automatically. The Java programming language introspection of the application class files is used to generate the communication stubs.

Unlike JavaBeans, which use the Java event model, Enterprise JavaBeans use the distributed CORBA event model. The event model supported by the Java programming language is well suited for local, tightly integrated applications, but does not perform as well in a networked environment where high latency and insecure networks are common. Enterprise JavaBean events are propagated across the network over CORBA's Internet InterORB Protocol (IIOP) to other components.

Enterprise JavaBeans can be configured automatically as CORBA objects and then accessed through IIOP by clients. These client applications do not have to be written in the Java programming language to access the components. EJBs can also function as COM/DCOM objects for Windows clients.

Access to several key services are offered as part of the Enterprise JavaBean specification (Exhibit 2). These services are offered through specific Java platform APIs, such as JavaIDL/RMI for accessing CORBA, DCE, or ONC services; Java Message Service (JMS) for access to messaging systems such as MQ Series; Java Naming and Directory Interface (JNDI) for accessing multiple naming and directory services such as LDAP and NDS; Java Database Connectivity (JDBC) for connecting to various relational and nonrelational databases; Java security APIs providing for encryption and authentication; Java Transaction services (JTS) providing a Java programming language binding to the object transaction services (OTS) of CORBA; Java Management API (JMAPI) providing for the management of networked resources such as workstations and routers; and Web services through the Java Server API. Each is detailed below.

JavaIDL

The Java Interface Definition Language (IDL) provides standards-based interoperability and connectivity with CORBA. Through these interfaces, Java applications are able to access existing infrastructure written in other languages. This is one of the key interfaces for legacy system integration. JavaIDL is part of the Java Platform Core API set and is therefore available across multiple platforms.

Java Message Service

Java Message Service (JMS) provides an interface to messaging systems that provide publish/subscribe and message queue services. This platform-independent interface will also support the emerging push/pull technologies.

Java Naming and Directory Interface

Many different kinds of naming and directory services exist in today's enterprises. Directory services such as LDAP, NIS, and NDS provide networkwide sharing of information about the users, systems, applications, and resources that exist on an intranet or the Internet. User information can include log-in ids, passwords, security access, and electronic mail addresses. System information can include network addresses and machine configurations. The Java Naming and Directory Interface (JNDI) is independent of any specific naming and directory service implementation. Application developers can easily access multiple namespaces through JNDI. A single interface simplifies the access to composite namespaces as well as enabling an application to be portable across different platforms.

Java Database Connectivity

One of the earliest and now core APIs is the Java Database Connectivity API (JDBC). This is an SQL-based, database-independent API which frees developers from writing database vendor-specific code in their applications.

JDBC supports the common database functionality such as remote procedure calls, SQL statements, database connection, and result sets. Since JDBC is implemented via a driver manager that itself can be implemented in the Java programming language, applets can be delivered to the client with the database connectivity built in. Implementation drivers for all the major RDBMSs are already available for JDBC and a JDBC to ODBC bridge is standard in the Java Developer's Kit Version 1.1. JDBC drivers for object-relational DBMSs, as well as the IBM IMS, are also currently available.

Java Security API

Security is an integral part of the Java platform and extends to the Java Enterprise Server architecture. There are four key areas that are supported by various security APIs: authentication, authorization, privacy, and integrity.

Authentication is the system's ability to verify or recognize a user. Typically performed at application access or system sign-on, authentication is the first line of defense present in a comprehensive security model. The JavaCard APIs allow smart cards to be employed as secure user authentication devices. These physical cards, combined with a secure personal identification number (PIN), enable users to be recognized by the target system. Digital signatures, another authentication method, are also supported through the Java Virtual Machine.

Authorization is the means of determining which data, systems, and services a user can access. The Java Security APIs and access control lists (ACLs) are available for managing who can access what. ACLs can be built for each Enterprise JavaBean and consulted whenever the bean is accessed. Based on the user's role, some form of access can be given or denied. Transaction servers installed in the application enforce the ACL at runtime. Since ACLs are not a static structure, they can be moved around the network with an EJB object. These embedded ACLs can then be accessed by the application developer.

Privacy concerns are raised in the context of transmission of sensitive data across public networks. To protect data, such as credit card numbers, encryption is typically employed. The Java language cryptography APIs provide application or session-level encryption. This interface can support any encryption implementation, including DES.

As data passes through a network, be it private or public, there is a chance for malicious or accidental modification. To prevent such actions, it is necessary to be able to guarantee the integrity of the transmission. The same mechanisms for ensuring privacy can be used for maintaining integrity of network communications, namely session and application encryption.

Java Transaction Services

Java Transaction Services (JTS) within the Enterprise JavaBean framework are a low-level API not meant as an application programmer interface. JTS programming is targeted to the resource managers and TP monitor programmers. Currently available implementations include BEA Systems Jolt product for Tuxedo access or the IBM JavaCICS for access to mainframe CICS applications.

Java Management API

The Java Management API (JMAPI) is a set of interfaces for the development of distributed network, system, and application management applications. JMAPI is designed to be incorporated into a variety of devices, across diverse network protocols and numerous operating systems. With support for the Simple Network Management Protocol (SNMP), JMAPI can communicate directly with a variety of existing devices. In the future, device manufacturers will incorporate the JMAPI directly into their products. System administrators using applications developed on this foundation are able to easily manage their network, applications, or other systems from any Java platform located anywhere on the network.

Java Server API

The Java Server API is an extensible framework that can be employed to quickly develop network-centric servers. These servers are capable of providing network-based services, such as Web services, file and print services, proxy services, and mail services. To extend the functionality of a Java server, a developer can create servlets using the Java Servlet API. Java servlets are programs that can be local to the server or downloaded across the network and then executed on the Java server. These servlets are perfect for processing form data from HTML pages, replacing the platform-dependent CGI-bin scripts in use by many organizations.

SUMMARY

The ability to integrate with legacy systems and extend enterprise services to the network with platform-independent technologies are key benefits of developing a Java Enterprise Server strategy. Enterprise JavaBeans, the component architecture for the Java Enterprise Server, provide a software- and hardware-independent method to access these systems and make them available to business components. These components can easily access services such as transaction monitors and message systems, DBMSs, and naming services with the assurance of the Java platform's "Write Once, Run Everywhere."

ABOUT THE AUTHOR

David Wadsworth is a Java Evangelist with Sun Microsystems of Canada. His primary role is to proselytize and promote the benefits of the platform independence of Java computing within the Canadian marketplace. David can be reached at david.wadsworth@canada.sun.com.

Section VII
Solution Testing

Testing is complex in any systems environment, but even more so in the Internet space. Peak times can draw an unprecedented number of Internet users to a specific Web site all at the same time. For example, stock-trading sites may experience very little traffic until a sudden news release instigates a lot of buying or selling of certain stocks. Not only must the corresponding Web site be able to scale to the requirements of this sudden traffic flow, but the performance must remain acceptable and reliable as well. The users are also on disparate computer systems, interacting with the Web site in virtually every manner possible. The combinations and permutations of things to test are enormous. There is a clear need for a thorough testing environment and strategy in very Internet-related development projects with test cases that are unique to this new space.

"Integrated Automated Software Testing: A Life Cycle Approach" (Chapter 42) describes how to integrate automated testing into the systems development life cycle. The key enabler is to build and maintain an automated test library that every member of a development team can access.

"Web-Based Testing and Capacity Planning" (Chapter 43) examines both of these primary challenges. This chapter shows development managers how to define the scope of Web testing and monitoring server behavior, while leveraging historical growth statisitics, to ensure Web site availability to all users all of the time.

"Evaluating and Selecting Automated Testing Tools" (Chapter 44) discusses the various classes of testing tools and describes a framework for selecting the appropriate product groups for a specific organization and set of testing requirements.

"Testing Object-Based Applications" (Chapter 45) examines how to use an incremental approach to design, build, and test individual objects or components to improve the overall quality of a business solution.

"Security Testing Is Not All the Same: A Reference Taxonomy" (Chapter 46) provides a list of terms to classify the different types of services that are available and explains how each technique evaluates security controls in applications.

"Introduction to Penetration Testing" (Chapter 47) shows how to establish a set of procedures that are designed to bypass the security controls of a system or organization only for the purpose of testing its ability to withstand an attack.

"Performing Penetration Testing" (Chapter 48) details methods of executing the procedures discussed in Chapter 47.

Chapter 42
Integrated Automated Software Testing: A Life Cycle Approach

Linda G. Hayes

The shift toward client/server applications, the proliferation of interactive, graphic user interface, and the growing use of technology as a competitive weapon pose increased dangers from software defects in critical systems. Automation alone, however, cannot solve the underlying problems of a manual test procedure. Automation cannot improve an ineffective manual process, and it may confuse underlying organizational problems with technical ones.

Automating the test process presents both the challenge and the opportunity to discard traditional methods that are no longer effective. To successfully automate test methods, the goals and shortcomings of current methods must be critically examined. Without such an examination, automated test procedures will either fail or aggravate previous problems.

CURRENT TESTING PRACTICES

In most organizations, testing is not an orderly procedure but a cataclysmic event. Sandwiched between the coding deadline and the release date, testing is often the victim of the scheduling squeeze endemic in many software development environments. Coding schedules are often extended but the release dates remain fixed. As a result, software testing is either shortened or eliminated. Testing is often abandoned because many information system (IS) departments do not clearly know what constitutes a successful test. Because there is no definition of a successful test, the benefit of further

0-8493-1161-6/02/$0.00+$1.50

testing cannot be measured objectively against the risk of curtailed testing and the consequences of a missed release date.

The definition of a test process is often ambiguous. To test an application, one needs to know why the application is being tested. A facile answer is that the application is being tested for its conformance to requirements and specifications. This answer assumes that application requirements and specifications have been documented. In many systems development departments, such documents rarely exist.

Even if these documents do exist, the definition of a successful test is still unclear. Performing even the most basic tests comprehensively on a typical commercial online application requires thousands of test cases, which cannot predict performance or system-related failure.

The use of CASE tools has significantly advanced the art of software design and development, in many cases providing automated application generation. For these applications, the design specification is more likely to be a living document that accurately reflects the state of the system. Even if the design is well defined and the code can be relied upon to be consistent with it, the requirements for a successful test must also encompass the business functions to be satisfied by the system and all of the related factors, such as performance, interapplication interfaces, and production environment compatibility.

Even when the design is documented, performing even the most basic tests comprehensively on a typical online commercial application requires thousands of test cases to be executed each and every time the application or its operating environment changes. Ironically, the better defined the application is, the greater the number of known test requirements, and the more intimidating the testing task becomes, often overwhelming limited testing budgets and schedules.

Therefore, testing is usually performed ad hoc. Programmers or experienced users rely on their own experiences, prejudices, and skills to determine which conditions should be tested and which responses are correct. Testing is performed only on areas of change; the need for full regression testing is recognized but rarely fulfilled. Test coverage is spotty and its benefits are unpredictable.

This lack of a clearly defined test process cripples the testing effort. Without an objective definition of testing, it is impossible to predict the resources needed, the time required, or the coverage achievable. Staffing and scheduling are determined either by guesswork or by the availability of time and personnel. Software quality is ultimately determined in the trenches.

Automating such an environment simply preserves the existing conditions. Capture and playback testing tools, which were among the first tools used to automate testing, simply record manual test procedures as a series of keystrokes or mouse events and screen or window images. They not only fail to improve test coverage, but also divert resources and attention from the real problem: the lack of a definition for the test procedure.

Capture and playback tools also fail in a more fundamental way to automate the test process. These tools merely record the keystrokes and screen images produced during a single test session and repeat them at a later time. They can capture a single test session, but they cannot capture the general test procedure. Thus, capture and playback tools fail to provide a model for testing subsequent releases or related applications.

DEFINING TESTING THROUGH AUTOMATION

Automating the software testing process must be approached as an applications project so that the process can be defined and made to provide useful, general purpose models. Testing automation requires building a system that can carry out the mechanical procedures of the overall process. Such a system would allow testers to direct their attention to defining the inputs, outputs, sequences, and conditions that constitute a successful and comprehensive test.

Once test automation is viewed as an applications development process, some conclusions are inevitable. First, automated software testing tools should be developed by those skilled in software design and programming. Therefore, a test automation system is just another application whose end users happen to be testers. The once popular belief that testing tools should be selected for ease of use by those responsible for manual testing is no longer tenable. The manual tester is the automation customer or end user, not the developer.

Another conclusion is that test scripts must be distinguished from the information they process. Well-designed application programs do not contain hard-coded data and neither should well-designed test scripts. A test script should serve simply as a means for automatically applying the test case to the software and comparing the actual output to the expected output. Test scripts should also report results in meaningful terms that are designed to reveal defects at their sources.

The process of creating an automated testing system according to traditional development disciplines results in a maintainable library of scripts and test cases. The maintainability of the test library extends the life of the test automation system by making it feasible to incorporate the inevitable modifications and enhancements to the application under test into the test automation process without making obsolete prior test cases. This longer life means that test cases can be cumulative, so that the volume of test scripts and cases can keep pace with the expanding features and functions of the application that must be tested.

However, comprehensive test automation cannot be achieved by the test organization alone, any more than quality applications can be produced in vacuum by programmers. The testing process must encompass the efforts of every area that affects systems development, implementation, and support. The roles that various groups must play in the automated testing process are discussed in the following sections of this chapter.

THE TEST ORGANIZATION'S ROLE

If the test organization does exist in a company, it is usually a conglomeration of programmers, systems operators, and temporary workers. Testing has yet to be recognized as a professional discipline, and the test organization staff is treated as entry level.

Ironically, testers usually have a wealth of accountability and a paucity of authority. The entire burden of a system's quality may be theirs, but testers may not have any control over those who design, code, install, use, and maintain the system. The role of most quality assurance organizations may be advisory at best and perfunctory at worst.

Automated testing can improve this situation. If the automated test procedure has been implemented as a general model and separated from the applications to be tested, the test organization can use its time more productively. Testers can identify the sources of defects and determine ways to prevent the defects. Simple tracking of defects to particular modules or functions may reveal incomplete design specifications, sloppy coding practices, quirks in the operating environment, or unskilled use of the system.

However, by gaining a better understanding of what caused a defect in the first place, testers will be able to address causes rather than effects of poor system quality.

The role of the testing organization, therefore, is to define, design, assemble, and manage an automated system capable of assimilating test library components. This system will also be able to execute test components in a consistent and predictable manner and report results in a meaningful way. To fulfill this role, the test organization must be given authority to charge other organizations with responsibility for testing and to guide these organizations through the testing process.

THE SYSTEMS ANALYST'S ROLE

A systems development project typically begins with a request from the user community for more systems functions. The systems analyst translates this request from the business user's description into terms understandable to the systems developer. For example, a new group health insurance policy dictates that employees are to be categorized as officers, managers, staff members, or part-time employees. These categories will be used to apply corresponding rates and calculate the insurance deduction. Working from this description of the new policy, the systems analyst prepares a detailed design of the necessary files, fields, and calculations. Despite this detailed plan, much of the system is still not defined.

Because requirements customarily state what a system should do, they can be ambiguous as to what a system should not do. Many details of the requirements are implied and not explicitly stated. For example, the existence of a new category field and corresponding table implies that any category entered into an employee record must be checked against the table to ensure that the category exists.

More questions and implied answers follow from undefined user requirements. Could the category field be left without a value? If the category must be validated against the table, then the table must be created before employee entry or update is permitted. Could this affect normal employee processing? Can a table value be later deleted or modified? If so, what should be done to employee records containing a formerly valid table value?

The problem is that user requests are usually stated in highly ambiguous terms, whereas systems design and development is an intensely specific process. Very little debate takes place about a program's performance once the program has been written. But how do analysts know in advance whether a program will perform as its users require? In other words, how are user requirements defined in testable terms?

An obvious solution to the requirements definition problem is to define requirements in terms of test cases. An example can be found in hardware specifications for sheet metal used in constructing an aircraft. These specifications are never written as wishful narratives filled with such descriptive terms as wide, thin, and temperature resistant. They are stated in objective terms that can be tested for conformance. Instead of specifying thin, the hardware designer specifies [frac18] in thickness; instead of specifying large, the hardware designer specifies 8 ft by 12 ft.

Without such specifications, the sheet metal plant would be continually manufacturing sample parts, which the airplane assembler would be continually rejecting because they were either too thick or too thin, too large or too small.

Although it seems absurd, end users often continually reject new systems because of similar, vaguely stated reasons. In a typical scenario, an end user of a new group health insurance policy system exclaims, “Of course an employee should not be allowed to have an unidentified category. The systems analyst and programmer should have known better.”

This type of reaction is typical in the iterative cycle that has come to frustrate on-time delivery of quality software. In this cycle, requirements are vaguely defined and systems are developed through trial and error. Fortunately, this situation can be corrected. Requirements for a software application can and should be expressed with the same degree of specificity and at the same point in the development cycle as the hardware requirements are determined.

If the analyst required the end user to think through each step when requesting a function, procedural issues such as the following would come to the fore:

- In what order must these processes take place?
- What edits must take place?
- How are changes made?

While defining these steps and their inputs and outputs, the user would also be reminded of side issues and could determine how they should be processed instead of having the systems analyst or systems developer infer the appropriate action. More important, by defining requirements in terms of processes, inputs, and outputs, the end user would also be defining requirements in terms of executable test cases.

Defining user requirements in terms of executable test cases not only clarifies any ambiguous requests, but also provides the nucleus of the application test library, which can be used throughout the development life cycle. Thus, systems analysts can set the stage for acceptance tests that determine if user requirements are being met.

The test organization’s responsibility to the systems analyst is to define the vocabulary and grammar required to state an executable test. This vocabulary includes the nouns, verbs, and adjectives legal for stating a testing task and the structural rules for combining them into test cases. Without this vocabulary and these rules of grammar, stating test cases is subject to the same ambiguities as stating requirements.

A noun, for example, includes any defined component of the application: a file, a menu, a screen, a message. The verbs describe the actions that are appropriate for a particular noun. A field can receive input or display output, a menu item can be selected, a screen image can be compared, or an error message verified. The adjectives are the values: an exact field value, a particular menu selection, a named screen, or a certain message string. The combination of these elements states a testing task, and the sequence of tasks describes a test case in terms that are executable by the test automation system.

THE SYSTEM DEVELOPER'S ROLE

At one time, systems developers were often responsible for a type of software testing known as unit testing. To perform a unit test, the developer who wrote a module's source code also tested that module. This testing was rarely automated because it was often performed ad hoc and defined by the responsible systems developer. A unit test usually determined whether the code operated as the programmer expected, not whether it performed as the end user required.

The automation of testing gives systems developers opportunities to expand their role. One opportunity can be found in the initial coding phase, where developers can easily adopt certain standards and practices that would make future testing easier. When writing source code, developers could incorporate into the code such features as:

- Unique screen identifiers
- Centralized data dictionaries
- Naming conventions
- Standardized navigation and function keys
- Consistent error handling and recovery procedures

These features simplify test automation by providing consistent and predictable application behavior and allow testing techniques to be mapped into the application architecture.

A direct correspondence exists between the application source code library and the application test library. As the application's source code is enhanced and modified, its test library must undergo corresponding changes. Because developers are the architects and maintainers of the source code library, a compelling argument can be made that programmers can and should contribute to the test process.

At a minimum, the development organization should contribute to white box, or glass box, testing, which is based on the source code and ensures that each line of code executes properly. With their intimate knowledge of the code, systems developers have the background to define test cases, which cover such arcane areas as array ranges, sector boundaries, and branch instructions.

Other types of test cases that the development organization can contribute are those that perform negative tests (e.g., the verification of editing and error handling). Because checking for valid data types and values and error processing is performed at the code level, the programmer is capable of identifying test cases that will exercise each possible negative condition.

Instead of relying on individual unit tests, which can be ad hoc, the development organization can benefit from the test automation process by standardizing test cases that are made up of acceptable unit tests.

In an automated test environment, the development organization should consider testing needs while creating source code. The organization should provide test cases for unit and white box testing. These unit tests should conform to the test organization's vocabulary and rules of grammar so that they can be seamlessly integrated into the automated test system. It should also inform the test library of any changes to an application's source code that could affect the application test library.

SYSTEMS MANAGEMENT'S ROLE

For an application to be successful, it is not sufficient that the application's source code complies and executes without error and that the application performs all the required business functions. The application must also be able to function on the intended hardware and operating system. For example, many IS organizations have experienced situations in which an upgrade to an operating system or the installation of a new database manager caused an otherwise unmodified application to abend. Table entries may be lost, configuration options changed, system functions modified, or other such mishaps can occur in the complex systems environment in which most commercial applications reside.

The advent of client/server applications adds another dimension of complexity. Distributed application databases and shared execution responsibilities mean that multiple platforms and operating environments are involved, each with its own configuration issues and considerations. Interfaces exist not only between applications but within them, spanning an ever-expanding network of hardware and software.

Testing for problems encountered in such situations requires experience with the hardware and operating system that make up the application's production environment. Stress and performance testing ensure that the application can meet, in the intended production environment, such operational requirements as response time, transaction throughput rates, file size, and disk volume, in addition to functional requirements.

The technical expertise needed to create conditions to test for operational requirements is probably not within the capabilities of the test organization and is certainly beyond the scope of the systems analysis and development organizations. Therefore, systems management should contribute to the application test library cases and procedures that test such critical areas as performance, stress, volume constraints, and compatibility with the operating environment. This testing material is indispensable for ensuring that the application will perform as needed when it is installed for production.

In addition to contributing test cases and conditions to exercise the application throughout its operating environment, systems management must clearly define the necessary system configurations. The horsepower of the hardware platforms, the version and release of the operating systems, the configuration of the network, the privileges and security levels of the workstation, the state of the database, and myriad related issues determine whether the application environment that is tested is consistent with the one in which it will ultimately operate.

The test organization's responsibility is to implement explicit requirements for describing the system configuration for the automated test environment and to report any related problems that are discovered to systems management for resolution.

THE END USER'S ROLE

No matter how comprehensive the automated test library is, it cannot cover all the situations end users can create. End users can make mistakes, become confused, work illogically, and, in general, behave randomly. They are the wild cards in software testing, and testing cannot be considered complete without end-user involvement.

By automating testing, the test library, with its scripts and cases, is the main tool for testing. The test library is important because it allows actions, inputs, and outputs to be defined before an application is developed. Because test automation and the test library define the test process, random or ad hoc testing cannot be fully automated. This does not mean, however, that ad hoc testing is neither important nor unable to benefit from automation.

After the system has passed the rigors of the automated test library, it should be subjected to the rigors of end use. Potential end users should be allowed to work with the system in order to catch any system failures caused by unexpected use. For this phase of testing, end users are instructed to crash the system as well to make sure it meets their requirements. Most users find this phase of testing enjoyable and challenging as well as an opportunity to show off as they push the system to its limits and thereby expose problems.

During this phase, capture and playback tools regain their utility. These tools can be used to record user test sessions, and only the recordings of system crashes are retained. The recordings can reconstruct the exact sequence of events that caused the system to fail. They can also be studied or replayed to test for an error condition.

Another utility of capture and playback recordings arises from analyzing user interaction with the application to measure usability. By timing the interval between keystrokes and mouse events, counting the incidents of pressing the backspace or undo key, and other types of indicators, the recorded test session can yield useful information about potential problem areas or inefficiencies that may plague end users. Although usability testing is arguably a discipline separate and apart from traditional functional testing, it is no less important to software quality as an overall objective.

SYSTEMS SUPPORT'S ROLE

Despite efforts to prevent them, system failures do occur. The support organization is usually responsible for investigating these failures and working with the development organization to eliminate bugs from the system.

Although testing for these bugs is often rigorous, they commonly reappear. The coding error that caused the bugs to appear the first time could have been a common mistake, which could easily be repeated after the system has been changed. The same unusual combination of keystrokes that one end user made to accidentally crash the system may be reentered by a subsequent user. Despite comprehensive testing after the first version of the system has crashed, altered versions of the system may not be subjected to the same tests and will contain similar bugs.

It has been argued that these bugs are often due to highly improbable events and it is not economical to test for them each time an application is developed. Automation has changed the economics of software testing, however, and has extended the limits of how practical extensive testing can be. One factor that allows for extensive testing to be practical is the automated test library, which allows tests implemented during system repair to be preserved for later use.

Although the causes of a bug may seem unlikely to recur, the fact that a bug occurred once increases the likelihood it will recur. Testers can learn from bugs repaired in the past how to devise tests for similar bugs.

The support organization can also contribute to the test library in another way. Just as system failures can be a source for the automated test library, successful system operation can also be a source. Successful production runs offer the opportunity to take actual samples and incorporate them into tests of critical operations. Inventing hypothetical test cases can be academic, but successful runs readily provide real-world cases. Although a test based on an exact replication of a complete production cycle may not be possible, it is worthwhile to constantly and systematically extract samples that exemplify business transactions and processes. The systems support organization can contribute to the automated test library sample cases taken from daily operation.

The systems support organization closes the loop of comprehensive software testing. It provides formal feedback that enhances the testing process with cases and conditions encountered during production.

CONCLUSION

Automating faulty manual test practices is an exercise that produces no benefits. For the automation of software testing to be worthwhile, it should be approached as a systems development project. Test automation should be designed to define the test process in objective terms and to allocate responsibility for quality to those organizations active in the systems development life cycle. To achieve this design, the role of the test organization must be fundamentally changed. The test organization must work with analysts, developers, systems managers, end users, and support personnel to assemble a test library. This library will be able to supply reliable, consistent, and comprehensive test coverage as systems and organizations within the IS department change over time.

ABOUT THE AUTHOR

Linda G. Hayes, B.B.A., M.S., C.P.A., J.D., is a founder, president, and CEO of AutoTester, Inc., which is based in Dallas and develops automated software testing solutions.

Chapter 43
Web-Based Testing and Capacity Planning

Trevor Clarke

Every day, more and more companies are entering the E-marketplace by offering their products and services through the Internet. This shift has led to fundamental changes in the product-development life cycle. The challenges facing CIOs and IT managers have increased accordingly, as they are expected to deliver complex applications and application environments in less time than traditional client/server applications to meet the more sophisticated demands of their customers and to remain competitive. Consequently, a much more rigorous testing process, completed in a shorter timeframe, is required.

Coupled with this new medium of transacting business is a much larger marketplace, which makes it increasingly difficult for IT managers to predict loads and appropriately provision infrastructure. Failure to sufficiently provision the infrastructure will result in performance degradations and, ultimately, the loss of customers. This article addresses two key challenges facing CIOs and IT managers: Web-based testing and capacity planning in a rapidly changing Internet environment.

THE ADDED COMPLEXITIES

Web-based systems introduce many additional and different complexities over traditional client/server systems and the earlier mainframe environments. As businesses go online, there are many unknowns that could adversely affect the success of their E-business venture. The following list identifies some of the major complexities and unknowns that a testing organization will have to consider to ensure a quality service.

0-8493-1161-6/02/$0.00+$1.50

1. *Speed.* The increased competition faced by companies doing business on the Internet has resulted in shorter development life cycles. To meet customer expectations, companies must respond quickly to market demands and continuously improve their site to keep existing customers and attract new customers. Testing must also be completed in much shorter time frames than experienced with client/server solutions.
2. *Scenario development.* A key challenge with Web-based systems is the development and testing of all possible scenarios of user interaction with the system. For transaction based-systems, rigorous testing needs to occur to ensure the integrity of transactions as users can willingly or unwillingly be disconnected from the system. Also, transaction integrity needs to be ensured during peak activity when performance degradations and system timeouts are more likely. Finally, the testing organization also needs to consider that users may freely navigate forward or backward within a Web site and may cause unwanted duplication of transactions.
3. *Performance testing.* Ensuring the performance of a Web-based system is another key challenge as some components are not under direct control of the enterprise. The system or the network could cause performance issues in a Web-based environment. Keynote Systems, the Internet performance authority, indicates that Internet performance problems are generally not server problems.[1] Keynote has demonstrated that most performance problems occur out in the Internet infrastructure between the users and Web servers at network access points (NAPs), routers, or in a domain name server (DNS). Assuring performance could equate to a company's ability to attract and keep customers loyal to its Web site.
4. *Capacity planning.* Effective planning of system and network capacity becomes difficult as business becomes global when online. Ineffective planning could lead to excessive performance issues that result in loss of customers.
5. *Security.* Additional security risks are associated with Web-based systems as they operate in a relatively "open" environment and could provide access to the company's confidential systems and data by unauthorized users. Simple bugs in the Web server could enable users to corrupt or steal data from the system or even render the systems unavailable.
6. *Multiple technologies.* A complete testing cycle would include all possible software configurations that users leverage to access a site (primarily Netscape or Microsoft's Explorer). Configurations may include various browser versions and service packs.

THE TESTING CYCLE

Utilizing typical client/server testing approaches will not address the many added complexities resulting from a Web-based system. Additionally, the more aggressive time schedules involved in Web-site development projects result in the need for an organization to develop a different and effective approach.

Defining Testing Scope

Determining the testing scope is critical to the success of a Web-based testing project. Due to the short time frame associated with Web-site testing, it can become difficult to test all components of the application and network. When possible, testing the complete Web-based environment is ideal. However, when time and budget constraints are incorporated, an organization may need to determine critical requirements and potential high-risk areas and focus testing efforts on these areas.

Critical requirements and high-risk areas can be determined by analyzing the requirements to determine the functionality that is most important to the success of the Web site, the areas within the Web site that will draw most customer focus (both positive and negative), and areas of the Web site that pose security threats.

Testing scope may include the complete system environment, including network performance testing. Alternatively, testing scope may be isolated to a particular module of the Web site or system environment (e.g., Web server, application server, database, etc.). Although not every component of the Web-based application or infrastructure may be tested before production, it is recommended that testing continue post-production for components not initially tested.

Test Planning

Based on the testing scope, the testing organization needs to plan the testing phase, including the types and timing of tests to be performed in both the pre- and post-release stages. The following testing types would be executed in a complete testing cycle:

1. *Unit testing.* Unit testing is the process of testing individual application objects or functions in an isolated environment before testing the integration with other tested units. Unit testing is the most efficient and effective phase in terms of defect detection.
2. *Integration testing.* The purpose of integration testing is to verify proper integrated functioning of the modules (objects, functions) that make up a subsystem. The focus of integration testing is on cross-functional tests rather than on unit tests within one module.

3. *End-to-end testing.* End-to-end testing is a comprehensive test of the integration of subsystems and interfaces that make up the Web site. Typically, end-to-end testing models all possible scenarios of user or business activity on the Web site. Included within this testing phase is the verification of all links to other Web sites, whether internal or external (referred to as link testing). Link testing is a key activity that should be completed on a recurring basis because Web sites tend to change URLs or are discontinued.
4. *Security testing.* Although implemented security measures are considered as part of the end-to-end solution, this testing type is kept separate due to its importance. Security testing involves two key processes. The first is the assurance that unauthorized users are blocked from accessing data and systems not intended for the user population. The second involves the testing of the data encryption techniques employed by the organization.
5. *Regression testing.* Regression testing ensures that code changes made during application testing or post-production have not introduced any additional defects into previously tested code.
6. *Usability testing.* Usability testing ensures that the presentation, flow, and general ergonomics of the Web site is accepted by the intended user community. This testing phase is critical because it enables an organization to measure the effectiveness of the content and design of the Web site, which ultimately leads to the ability to attract and keep customers.
7. *Stress testing.* Stress testing observes the capabilities of production hardware and software to continue to function properly under a predetermined set and volume of test scenarios. The purpose of stress testing is to ensure that the system can maintain throughput and efficient operation under different load conditions.

   Stress testing enables an organization to determine what conditions are likely to cause system (hardware or software) failures. This testing phase needs to consider the possible hardware platforms, operating systems, and browsers used by customers. Results from stress testing are also a key component used for capacity planning (capacity planning is discussed later).
8. *Performance testing.* Performance testing observes the response times of the systems (i.e., Web server, database, etc.) and capabilities of the network to efficiently transmit data under varied load conditions. Performance testing should enable an organization to determine and resolve bottlenecks within the application and infrastructure. Performance testing should also consider the possible hardware platforms, operating systems, and browsers used by customers.

If testing scope has been limited to a certain aspect of the system or network environment, only a limited set of tests will be completed in the pre-production phase. Based on the priorities set in the scoping phase, the test manager needs to determine the set of test types and resources required in the pre-production testing phase and those that will be completed in the post-production phase. The minimum testing that needs to occur for code changes is unit and integration testing for the modules affected by the code change.

The requirement for much quicker development and testing cycles has led to the creation of sophisticated software quality tools that automate many of the test types described above. Key competitors in this marketplace include Segue Software, Mercury Interactive, RadView Software, and RSW Software. The following paragraphs describe the solutions offered by each company.

Segue Software. Segue Software's Silk family of E-business testing products automates several threads of the testing process, including functional (unit) and regression testing (SilkTest), load and performance testing (SilkPerformer), and scenario testing (SilkRealizer). Seque also provides professional services to help install and configure the Silk products to test a company's products.

Additional value-add products in the Silk line include SilkMonitor (24 × 7 monitoring and reporting of Web, application, and database servers); SilkObserver (end-to-end transaction management and monitoring of CORBA applications); SilkMeter (access control and usage metering); and SilkRadar (automated defect tracking).

For more information, visit Seque Software's Web site at www.segue.com.

Mercury Interactive. Mercury Interactive provides the Astra suite of Web-based testing products. Specific modules include Astra LoadTest to test scalability and performance, and Astra QuickTest for functional and regression testing. Additional value-add tools include Astra SiteManager to manage the Web site and identify problems and user hotspots.

For more information, visit Mercury Interactive's Web site at www.mercuryinteractive.com.

Radview Software. Radview's WebLoad product line provides tools for verifying application scalability and integrity. Scalability and integrity refers to load and functional testing. Additional products include WebLoad Resource Manager to facilitate and coordinate testing and resources in the development life cycle.

For more information, visit Radview Software's Web site at www.radview.com.

RSW Software. RSW's e-Test suite of products provides solutions to test the functionality, scalability, and availability of Web-based applications. e-Load is used for load and scalability testing, while e-Tester is used for functional and regression testing. Additional value-add modules include e-Monitor, which provides 7 × 24 monitoring of deployed applications.

For more information, visit RSW Software's Web site at www.rswsoftware.com.

To significantly decrease the time required to perform testing, it is recommended to assess the organization's testing requirements and choose an automated software quality tool to expedite repetitive testing tasks. Additionally, these test tools will enable an organization to perform stress testing, which is key to ensuring sufficient network and server resource levels for the production environment.

Capacity Planning

Effective performance testing is difficult without an accurate depiction of future loads. Many companies simply over-engineer hardware and networks at high costs to minimize potential performance issues leading to service degradations or deal with performance issues on a reactive basis. Reacting to performance issues in today's highly competitive marketplace could ultimately lead to the loss of customers during system downtime or periods of poor performance. Planning capacity is a critical step required to ensure the future performance of a Web-based environment. The key components involved are network, server (e.g., memory, CPU, I/O) and storage capacity.

Establishing performance benchmarks and subsequently estimating future growth is critical to planning the capacity of the network and servers. Although benchmarks are published by the Standard Performance Evaluation Corporation for Web servers (www.specbench.org), their uses are limited and do not accurately represent a real-world integrated Web environment. Alternatively, benchmarks can be determined through stress testing and mapping of performance (e.g., response times) to specific network or hardware configurations under varying loads. Modeling tools and techniques can also be used to determine performance characteristics under varying loads.

Once initial benchmarks are established, future production loads can be estimated using historical growth statistics and growth predictions estimated by various Internet analyst groups (e.g., IDC, GartnerGroup, and Forrester Research). Subsequently, the growth forecasts can be put to test to determine the resource and scalability requirements of the network and hardware in the future. Note that peak loads of three to four times average loads should be tested during the stress test phase. Additional stress testing considerations include modeling higher-volume loads for cyclical periods. For example, online retailers may have much higher loads during the Christmas period than during the rest of the year. Ensuring performance, especially during these peak periods, will have an impact on Website success. For this reason, over-provisioning hardware or network components to a certain level is justified.

Although effective capacity planning should enable one's systems to handle future growth, monitoring of one's networks and server resources should continue to ensure that capacity is within acceptable limits.

CONCLUSION

Web-based applications have resulted in many challenges for the testing community. The ability of an organization to effectively prioritize the components requiring testing and to rapidly execute the tests is a requirement in a competitive E-marketplace. Leveraging the tools designed specifically for Web-based testing will enhance an organization's ability to get a quality product to market faster. Finally, proactive capacity planning, rather than reactive performance issue resolution, will result in greater customer satisfaction and ultimately in greater revenue.

Note

1. Keynote Systems, Inc., "Top 10 Discoveries about the Internet," Keynote Systems, Inc., San Mateo, CA, 1998.

ABOUT THE AUTHOR

Trevor Clarke is a management consultant in the Toronto office of Deloitte Consulting. His primary focus is technology integration in the financial services and telecommunications industries. He has been involved with several testing initiatives, including the definition of a testing strategy and approach for an E-procurement company.

Chapter 44 Evaluating and Selecting Automated Testing Tools

Polly Perryman Kuver

Ten years ago, the most effective method for testing software was to bang away at the code. Occasionally, tools were built as part of the development effort to test a specific aspect of the software. A few capture/playback recorders were available, but they were cumbersome and time-consuming to use. A few debugging tools existed. That was the extent of it ten years ago.

Today, not only do commercial testing tools exist, but multiple classes of testing tools are heavily marketed throughout the software development industry. There are improved test tools for debugging programs, for regression testing, for performance testing, and for Web sites. In fact, over the past ten years, so many different test tools in so many different classifications have hit the market, that evaluating and selecting the best test tool set for an organization can became a full-time job for one or more employees for several weeks. That investment, along with the annual license fees for the tools selected, may seem steep, but the risks associated with bringing the wrong tools into an organization can be even more expensive.

To minimize risk and decrease research and evaluation time, there are two important decisions to make. The first is whether an organization has a clear picture in mind as to what the benefit of the tool will be within the organization. The second is the size of the investment the organization is committed to making in bringing one or more test tools into the organization. An understanding of the classes of test tools available today can help in making these decisions.

0-8493-1161-6/02/$0.00+$1.50

CLASSES OF TEST TOOLS

The first step in tool selection requires an understanding of the classifications of the tools. While most companies are up-front in explaining what their tool sets will and will not do, they tend to make the tools sound easier to deploy and maintain than they really are. This means that, as a consumer of the tools, it is important to know and ask the right questions. Start internally by asking yourself where test tools will have the greatest impact on quality improvement.

- Is it in the development environment? Unit testing? String testing?
- Is it in code coverage?
- Is it in system workload? Testing transaction frequency? Transaction volume? Response time?
- Is it in system functionality? Repeatability of frequently run tests?
- Is it in capturing results? Proven performance? Benchmarking? Capturing evidence of completed tests for legal and other purposes?
- Is it in defect tracking?

Knowing where the greatest benefit will come from will point to the class of tools needed.

Debuggers

Debuggers are tools used in the development environment at the unit test and string test levels. They work by allowing the developer to identify break points in the operation of the program so that the execution of the program is interrupted and the output at a specific intermediate point of operation can be checked. The checking requires the user of the tool to be able to interpret the results in machine language and to have the ability to validate the machine language results to the anticipated results. Most often, a baseline run of the program is captured, deemed correct, and then used as a point of comparison whenever any changes to the program are made.

In deciding the benefit level an organization will attain from debugging tools, it is important to consider the programming language, the number of programs, the skill level of the programmers who will be using the tool, and the frequency of planned regression testing. For instance, serious software development companies with very senior programmers who are supporting frequent multiple-releases of maintained programs on multiple platforms will benefit more from the use of debuggers than a retail operation where development and maintenance is limited to modest updates on a two- or five-year schedule.

Code Coverage Tools

Coverage tools provide an automated method for identifying what paths have been tested and what paths remain to be tested. These tools work by executing individual test cases and logging the specific paths through the code that are read to generate the output.

Coverage tools decrease edit, compile, and debug times in the development environment by providing a comprehensive picture of what has and has not been tested, how many times a particular line of code has been exercised, when select lines of code have been tested in a block of code, and lines of code that have not been tested by the test cases used. If the test cases are comprehensive, code coverage tools identify code not used by the program at all.

These types of tools — with a tool manufactured by Platinum taking the lead — have been heavily used by organizations to determine how complete the test scenarios are.

Performance Tools

Performance tools are used to measure the workload that the system is capable of handling under normal conditions and under worst-case conditions. Here, the term automated test tool is somewhat of an oxymoron. Most of these tools require some degree of programming, referred to as scripts, in UNIX, PERL, or some other proprietary language. The decision to use such tools must consider the trade-offs in training and programming time in relationship to workload benefits.

An example for putting performance testing into perspective is the retail industry. Retailers' point-of-sale software must offer near-immediate responsiveness with their price scanning, credit card swiping software, and customer look-up applications to provide their customers with acceptable service. The capability of the software to quickly process data scanned simultaneously by several users within each store serviced by the software must be tested to ensure it will handle the normal workload and worst-case volume and frequency demand.

Initially performance tools will help by identifying high-usage bottlenecks and allowing fine-tuning of the software. Eventually, the performance tools can be used to emulate a given number of users and volume and frequency of transactions. Without the aid of performance tools, testing whether the throughput will adequately handle transactions becomes nearly impossible to determine.

Implementing software without sufficient testing carries high risks. Imagine the impact at a retail store on the Friday after Thanksgiving if the software was sluggish. There would be long lines and very unhappy customers.

The same performance level does not carry the same implications for the back-end software used by retailers to set prices, maintain inventory, track shipments, record returns, complaints, etc. Many price change programs for retail chains are batch jobs that run overnight. There is some time margin allowable. This is not to say that failure of the software to recognize a temporary override of default prices with sale prices on time would not cause some amount of havoc when the store opens on sale day, but it would be a more manageable situation.

The retail example represents the two ends of a line on which it is possible to scale the weight of an organization's software performance needs from nice-to-have to critical. The point on the scale should dictate the business significance of installing and training personnel to use performance tools. Note that there are many different considerations associated with the testing of Web sites. Thus, the decisions related to use of performance tools for Web sites and Web-based products are handled differently and should not be rated on this type of scale.

Recorders

Test tools that capture and play back keystrokes, allowing black box test scenarios to be run and then rerun as needed, are called recorders. Recorders are used to ensure that testing of software functionality is a repeatable process.

There was a time when scripts and cryptic coding needed to be prepared in advance of running recorders. Today, however, most recording test tools have been simplified to the point that users click a record icon, execute a predefined test scenario involving one or more transactions, and click 'stop' when the test scenario is complete. The same scenario can then be run over and over again by replacing the recorded script. The first few times the recorder is used, if a bug is identified, both the test scenario and the steps used during the recorder setup must be analyzed to determine if the bug exists in the software or the testing process.

Many organizations today scoff at the need for recorders, opting for manual testing because they do not feel the tests will be rerun a sufficient number of times to offset the seat licensing costs for the tools. According to Boris Beizer in his book, *Black-Box Testing,* that type of thinking supports a great myth. The myth says manual test execution works. Beizer contends that the error rate of manual testing skews the test results.

The extent of data entry and processing errors is greatly reduced when the test steps are captured, analyzed, and refined, and the same test process can be run each time. Because the test can be rerun automatically, errors caused by a tester forgetting a step or a condition can be controlled and eliminated. In the process, the integrity of all future tests run is higher — because the recorder will reproduce the exact same test over and over again.

The other benefit of automated recording tools is that the test results are automatically captured and compared with the previously recorded correct answers. Evidence of the test results can be preserved for future reference, for legal matters, and most importantly, for analysis in fixing bugs and upgrading future software releases.

Capturing Results Manually

So what happens when an organization needs to test and a recording test tool is not used to automatically run and capture the test results? It only means that the organization had better have some methods and controls put into place that will ensure that a valid test is run and that results are captured in some other way. One way is to use an inexpensive tool designed to capture and store screen prints, such as Hyper Snap by Hydrionics.

Another is to use a desktop database to link to the application's database tables and copy the records in all database tables expected to be affected during the test scenario execution. The records must be copied both before and after test execution to provide sufficient information. The copied record can be pasted in a spreadsheet and the spreadsheet can be used to determine if the results are correct. The method is cumbersome, but it is not as time-consuming as it would be to try to prove the test worked without having evidence of the results.

Of course, word processing tools are a must for any testing effort. Not only are word processors needed for the construction of the test scenario, but also for providing a place for documenting test tracking and status.

Defect Tracking

Identifying and documenting defects during the testing effort is a critical part of testing. Tracking the types of defects, attributing them to the correct version of software, having sufficient information to be able to recreate the defect, and being able to report the status of outstanding and closed defects must be supported. Many organizations use home-grown tools to accomplish this. Of all of the methods and tools available for defect tracking, word processors are the least effective for this task. The best approach for a home-grown defect tracking tool is a desktop database application such as Access.

Better than a home-grown Access application are defect tracking tools that are sold with configuration management tools. The reason is simple. Defect tracking tools manufactured to be integrated with configuration management tools are sophisticated enough to help manage the testing of multiple releases, multiple platforms, and multiple versions of custom code. Shrink-wrapped defect tracking tools that are integrated with version control and release management tools provide assistance in managing the testing and subsequent releases of the software.

The integration of defect tracking tools with configuration management tools ensures that correct versions of code are tested with the correct release. This also ensures that correct versions of the software are tested and that these versions, when tested, are reported against correctly. When set up correctly, these defect tracking tools can reduce the margin for error in testing the wrong version of a program, and the probability of using the wrong data in a test is reduced.

A bigger problem associated with defect tracking is that many organizations have yet to implement a configuration management program. This issue causes difficulties in the testing process due to insufficient testing of multiple versions of code, and errors in tracking defects from one environment to another and one platform to another.

Finding bugs is the primary purpose of testing. Documenting found bugs is the only way to ensure that the software that caused the bug gets fixed. Most organizations understand this and have a process for defect tracking. Unfortunately, the professional defect tracking tools available in today's market are not widely used.

The value of professionally developed, shrink-wrapped defect tracking tools is their comprehensive recording and extensive reporting features. These tools are designed to meet the needs of varying test groups in all different types of organizations. In some instances, the home-grown defect tool developed for a project even neglects to record the application name. The information captured becomes useless for future reference. Future referencing is a necessary part of continuous process improvement.

Commercially developed defect tracking tools provide a consistent method for defect tracking across software projects. The customization features of a product designed specifically to support defect tracking jog the mind, causing it to think of what needs to be tracked. Organizations do not end up with databases full of defects without knowing which version or release the software defects were identified.

Getting the bugs out of the defect tracking process used during testing efforts is important in the big picture. If each project sets up its own rules for grading defects and recording them, it is much more difficult to improve the testing processes of the company. Commercially available defect tracking tools can be used at all testing levels from debugging through enterprise testing. And, they can be integrated with the configuration management tools at a point in time that an organization decides to install and use one.

Exhibit 1. Guidelines for Selection of Testing Tools

Score	Indication	Justification to Management
5	Major benefits result from using the test tool	Easy
4	Significant improvement in the product from using the test tool	Requires work, but worth the effort regardless of impact related to implementation
3	Identifiable benefits, but downsides to bringing in the test tool	Must emphasize benefits of bringing in the test tool
2	Few benefits	
1	Test tool not needed	

GUIDELINES FOR SELECTION

Now that the areas for improvement have been identified and the general classifications of test tools are understood, give each area a score of 1 to 5. Scoring is different from prioritization. The objective of scoring is to obtain a solid understanding of how one or more test tools will increase the value of the product being developed. A possible definition of each score is shown below and in Exhibit 1.

- A score of 5 for a defined area of development and testing indicates that major benefits would result from using a test tool in the area. Major benefits include increased productivity, increased testing without schedule impact, increased number of defects identified, and perhaps a decrease in testing budget over time. Justification to management will be easy, regardless of any initial impacts related to tool implementation.
- A score of 4 indicates that significant improvements in the product would result from using the test tool. These improvements may include all of the benefits for a score of 5 but to a lesser degree, or they may include only one or two of those benefits but to a much greater degree. Justification to management may require a little more work but would be worth it, regardless of any initial impacts related to tool implementation.
- A score of 3 indicates there are identifiable benefits but also reflects one or more downsides to bringing in a test tool in a given area. The downside might include schedule impacts while testers are trained and test tool scripts are being written, or the lack of testing processes within an organization, which would make the switch over to tool use difficult. Justification to management will have to emphasize why bringing a tool in at this point is not just awash in the testing arena.
- A score of 2 indicates there are at least a few benefits, not the least of which may be that the manufacturer is providing a significant

discount on the product as long as it is purchased at the same time as other test tool products that have been scored at 4 or higher.

- A score of 1 indicates that implementation of a test tool will not hurt anything even if there are no apparent benefits; and, besides, the manufacturer is throwing in the tool for free.
- A score of 0 indicates that the tool is not needed, cannot be used, or is already in place. One could not use it even if the manufacturer provides it at no cost.

Using this scoring scheme, if internal evaluation within an organization developing a mainframe application pointed to the areas of performance and functionality as being the areas that would yield the highest return on the investment of one or more test tools, performance tools and recorders would receive a score of 5. In that same organization, defect tracking might already be set up and running with the organization's configuration management system. Defect tracking would receive a score of 0.

Each organization should use the bulleted scoring definitions or create definitions that better reflect its own needs. What is critical is to evaluate and score the need before contacting the manufacturers. This internal evaluation and scoring will ensure the best investment of corporate dollars.

TOOLS AND TOOL SUITES

Once scoring is complete, it is time to move the evaluation process outside the organization and take a look at vendors. Many companies selling shrink-wrapped test tools offer complete sets of tools, referred to as test suites. Among the leaders in test suite offerings are Mercury Interactive Incorporated, Rational Software Corporation, and Segue. Their tools can be purchased separately or as test suites, and all of the companies have training programs and customer support programs established to ensure maximum benefits from their tools and tool suites. Exhibit 2 shows the classes of testing tools available from these three vendors. Keep in mind that the companies listed allow these tools to be purchased separately or together, and offer additional testing tool products and processes.

Test tool suites are individual test tools integrated to work together in the various areas of testing. For example, Mercury's Test Director product combines the need for defect tracking with the overall need for managing test scenarios. It does this by:

- Providing comprehensive status information about the scenarios
- Driving test scenarios using their WinRunner and XRunner recorder tools, allowing tests to be set to run at specific times and under specific conditions

Exhibit 2. Test Tools and Test Tool Classifications

	Test Tool Classifications				
Company and Product	**Debuggers**	**Code Coverage**	**Performance**	**Recorders**	**Defect Tracking**
Mercury Interactive, Inc.					
LoadRunner			X		
WinRunner				X	
Xrunner				X	
Test Director					X
Rational Software Corp.					
Rational Purify	X				
Visual PureCoverage		X			
VisualQuantify			X		
Performance Studio			X		
SQA Robot				X	
SQA Test Manager				X	
ClearQuest					X
Segue Corp.					
SilkDeveloper		X			
SilkPerformer			X		
SilkTest				X	
SilkRadar					X
LiveQuality		X	X	X	X

Rational's SQA Manager performs a similar function using its SQA Robot (to be renamed Rational Robot in the spring of 1999) recorder. Rational's defect tracking tool is ClearQuest. ClearQuest can be used as a stand-alone tool for defect tracking, or it can be used as an integrated component of Rational's ClearCase configuration management tool.

Segue's line of Silk test tools includes SilkDeveloper for code coverage, SilkPerformer for performance testing, SilkTest for capture/playback/recorder, and SilkRadar for defect tracking.

For performance testing, Mercury Interactive, Inc. offers a product called LoadRunner, readily recognized as the current leader in client/server environments. Rational offers two products with two different intents for performance testing: Performance Studio is used to test transaction workloads, and Visual Quantify is used to test the execution time and number of calls in an application.

THE TESTING PROCESS

All of the test tools available through Mercury Interactive, Inc. and Rational Software are designed to support an organization's testing processes. For example, Segue also offers a product called LiveQuality that is composed of a Producer, Realizer, and Delivery designed to provide an organization with a nearly turnkey testing process including tool usage. These companies, along with others such as Micro Focus, will assist an organization in establishing testing processes and deploying test tools to support quality goals throughout the software development life cycle. These processes and the use of tools do make a difference when the organization's needs and goals are evaluated along with the test tools.

CONCLUSION

The implementation of test tools in any organization needs to be considered carefully. Just issuing a directive that test tools will be used is not the right road to success and increased product quality. Evaluating the internal testing needs is as important as evaluating the tools themselves. With solid planning and firm commitment, bringing testing tools into the development environment will result in better overall testing and meaningful regression testing at an affordable cost.

Notes

1. Kit, Edward. *Software Testing in the Real World*, Addison-Wesley, 1996.
2. Kaner, Cem, Falk, Jack, Nguyen, Hung Quoc. *Testing Computer Software, 2nd edition*, Thompson Computer Press, 1998.
3. Stitt, Martin. *Debugging, Creative Techniques and Tools for Software Repair*, Wiley Professional Computing, 1992.
4. Beizer, Boris. *Black-Box Testing, Techniques for Functional Testing of Software and Systems*, John Wiley & Sons, 1995.
5. Lewis, William. *PDCA/Test,* Auerbach Publications, 1999.
6. Tinnirello, Paul, Ed., *Handbook of Systems Development,* Auerbach Publications, 1999.

ABOUT THE AUTHOR

Polly Perryman Kuver has more than 20 years of computer experience, including 12 in management positions. As a process engineer, her areas of expertise are national and international software engineering and documentation standards, quality assurance, configuration management, and data management. Currently, she is a consultant in the Boston, MA area.

Chapter 45
Testing Object-Based Applications

Polly Perryman Kuver

Buttons, icons, fields, menus, and windows are all objects. Each of them, by the very nature of being objects, possesses properties, methods, and events. Properties describe the object. Methods state what the object can be told to do. Events are what the object does when it is invoked. For example, the print icon in Microsoft Word is usually one-quarter inch by one-quarter inch (property). It is gray and has a picture of a yellow printer with a piece of paper on it (property). It can be told to appear on the toolbar (method), be grayed out when it is not available (method), and recognize clicks from the left mouse button (method). When clicked, it will invoke print code, causing the document to print (event).

Because the print icon is a defined object, it can be used again in other applications. Nearly all software manufacturers today take advantage of code reuse; for example, the same print icon appears across all Microsoft products from Office to Explorer and Exchange. Reusability is one benefit of object-oriented development. Maintenance is another, and the value of object-oriented development will continue to grow with technology because not many companies can stay competitive if they cannot build once, test thoroughly, and then use again and again and again.

As object-oriented development spreads and grows in the software community, techniques for testing object-based applications become more important. An understanding of objects and object classes is the first step in understanding current testing techniques and developing the skill to invent proper testing techniques.

PROPERTIES

Object properties ensure that the use of one type of object is consistent throughout an application. Take buttons, for example. Whatever the application, it is easier to use and more appealing to the eye when all of the

0-8493-1161-6/02/$0.00+$1.50

buttons available to the user are the same shape, color, and size. To accomplish this, button properties include the dimensions for width and height, color, and font properties. Each property is defined in one place for objects of a single type. When the developer wants to use this button in another software package with a green button instead of a gray button, the object properties for the button can be accessed and modified in one location, one time for the entire application. The gray button becomes a green button throughout the application with this one change. If the color of the button is to be selected by the user, the color property is made public, since any public property can be accessed by the user. In this example, the color property is made accessible to the user, allowing it to be changed by the user from an available color palette. When the object has been thoroughly tested in the first application, this type of change does not warrant or require retesting of the object at the object level.

Testing becomes a matter of checking to ensure that the correct version of the button object has been included in the new application. This testing includes checks to ensure that one of the developers did not define a button object somewhere within his code that was not affected by the single instance change. The tester will perform this as a black-box test. That is, from an end-user perspective: Does the button display when it is supposed to display? Is the button active when it is suppose to be active? This is especially needed when the button object or any other type of object is public. If a user opts to change a color, it must change everywhere or the help desk will receive a lot of unnecessary calls.

Modifying the text of an object is just as simple as changing dimensions, color, and fonts. One button object is created and defined. Since text is a unique property worded to be consistent with the action the specific button will perform, the property text can be changed to fit the function. When the object text property is changed, the object should be saved under a new name to which new methods and events will be assigned. For example, standard words for the buttons may advisedly be used throughout the application. "OK" is used instead of "Enter." In fact, "OK" has become somewhat of a *de facto* standard in the windows world, replacing what was at one time a specific command or series of commands to update records.

In some applications, "OK" does not mean update; rather, it is used to indicate continue, show me the next screen. In those cases, updating may not occur until a button saying "Update" is located and clicked. For this reason, it is important to define the application text property standard and name the object appropriately. The text on a button is not generally a property that users are allowed to change. This property is hidden.

When it comes to testing an application, it is easier to identify defects in consistency and usage when object properties are defined in system specifications. This is because the specification implicitly explains what is supposed to be happening. However, the very nature of the object-oriented design often preempts the creation and publication of formal specifications. The standards used in defining object properties are in somebody's head or on little yellow Post-Its™ stuck on and around the developer's monitor. When the application moves into testing, the Post-Its do not move to testing with the software. That may be alright if the testing is to be limited to purely black-box function testing, but what about "look and feel"? In today's market, "look-and-feel" testing is critical. Consumers place a lot of emphasis on it. It cannot be ignored. So, how is it done?

It is done by creating business-based scenarios on which end-to-end testing is planned and documented in the test plan. This accomplishes three things: it documents the scenarios as well as the strategy and scheduling for testing the object properties and that all objects were addressed during testing; it documents how each object was addressed; and it documents the criteria used to determine if the objects throughout the application met a specific level of quality.

Addressing object properties in the test plan does not have to be involved. Simple bullets or sentences can be used. Toolbar objects will all be:

- Gray in color
- Typeset in Times New Roman
- In bold print

METHODS

The development of object properties and methods go hand in hand and are often discovered simultaneously during the design phase of the project. Properties characterize an object. Methods animate the object by defining what it can do. Think about it terms of action words. Methods equal actions the object can be told to do, such as display, show, move, get, calculate.

Methods can be defined and hidden from the user, or they can be public, allowing users to select the method from a list of options. An example of this is the selection of icons to be displayed on a toolbar or on a pull-down menu. In any Microsoft product, the user can go to View/Toolbars within the application and click each of the desired icons on or off; those with a check will appear on the toolbar. In reality, the user is changing the method used by the object in the icon class from hidden to display.

Together, the properties and methods determine the boundaries or interface of an object and are defined as an instance of a class. Test scenarios that check methods, as well as how the class structure interprets the methods, must be planned for all object-based applications.

CLASSES

The combined properties and methods must be identified and recognizable to the class structure being used. That is, there must be an icon class or something equivalent to an icon class before an instance of an icon, say a print icon, can be used in an application.

The class structure is based on object linking embedded (OLE) technology and supported by the programming language used. Visual Basic, Java, and C++ each has its own techniques for handling class structures. That is, they recognize specific types of object property–method combinations and, while they may each have an icon class, a button class, and a menu class, the rules governing the inclusion of an object within the class may vary.

The value of object-oriented design and development is in the adjustments that can be made at the object level, allowing developers to make the necessary changes without touching every screen and form in an application. Variances between class structures reduce portability and increase the maintainability of objects across platforms.

This is exemplified in the case of Microsoft Java versus SUN Java where standard Java objects had to be redefined to meet the unique requirements of Microsoft's Windows-optimized Java. Internet applications, test tools, and other products constructed in standard SUN Java to take advantage of the virtual machine capabilities it offers had to be redefined, reconstructed, and retested to run on Microsoft Windows platforms. This significantly increased the workload for many software manufacturers already stretching to meet customer requests in a highly competitive market.

When a class structure for a specific family of objects does not exist, Visual Basic and other programming languages allow for the coding of instructions for recognition.

While testing for class acceptance and recognition is an important part of testing object-based applications, it is perhaps even more important to have a predefined approach for reporting and debugging class-related defects during the testing process.

EVENTS

Once the right icons and buttons are defined, tested, and placed in an object container such as a spreadsheet, document frame, or form, the code behind the scenes is connected to the objects, allowing intended application functions to occur. Test scenarios that will demonstrate "if this action, then that result" need to be developed and executed. If the user clicks on the print icon, then something will print; if the user clicks "OK," then the next form is displayed.

Testing is based on the intended object event, and object-based testing tools provide the power to exercise the event thoroughly. The reason for this is that the test criteria can be established and the steps of the test recorded in a single script. The script can then be copied and easily modified at various points to extend test coverage to any number of "what–if" conditions, based on the intended event of the object.

In traditional code or non-object-based applications, the events are actually the equivalent of program control points. Each control point triggers a subroutine, a macro, or another program. To test, each of the possible paths must be first identified and then tested. The number of "what–if" conditions is limited by the number of conditions the tester can perform in the allocated testing time.

Since testing of object-based applications can be more extensive using the testing tools available, more extensive testing can be performed in the same allocated testing time. As a result, more defects can be found, fixed, and retested. So while there may be little or no difference between object-based and traditional applications in the types of defects found, it can be faster and more effective to find the defects in an object-based system, and it is certainly more judicious to fix and retest the object-based application.

Use of a tool is not mandatory in testing object-based applications. Manual testing of object events can be conducted in the same way traditional program control points are exercised. Manual testing involves defining scenarios for all the possible paths of a program or possible paths that can occur, given various conditions for an event and exercising those paths using basic business-use scenarios. For example, if a program stores data in a database, the data can be entered from an updated payroll screen or the human resources screen, as might occur in an integrated system if an employee marries and changes the number of dependents on a W4 form and insurance coverage.

Manual testing would require separate tests for each of the data-entry screens in payroll and human resources to be defined and executed, whereas an object-based automated testing tool could be scripted to recognize the variables of the different data-entry forms and test the object-event, which updates the database. Thus, a single script will permit multiple tests to be executed in less time.

THE TESTING EFFORT

A spiral approach to design, development, and testing is a good way to optimize the benefits of object-oriented design and development. It allows for the quick turnaround required in what one executive at Sun Microsystems, Inc., termed, "Internet time." That is, keeping pace with the rapid changes in technology and meeting customer demand for products that can be easily installed, operated, and customized to fit their environments.

The spiral approach is based on a model originally developed by Barry Boehm for the U.S. Department of Defense. The model promotes and allows for the reconciliation of concurrent, related development efforts that are undertaken in the same timeframe. Thus, individual "production lines" for various objects, object-containers, and background code can be established and run at the same time. The objects, containers, and code converge during the integration phase.

When the spiral model is employed, traditional testing processes must be reviewed and revised to ensure that adequate testing occurs, but that testing does not become a bottleneck in the overall effort to complete development and get the software into production. The very first step for ensuring a successful testing effort is to invest in a software testing tool that provides object-based testing capability. The tool is essential unless an organization can really rationalize a tester using a little ruler to measure objects as they are displayed on the monitor or want to trust visual perception, judgment, and approximation as the basis for pass/fail.

Without a software testing tool, the organization would also have to be prepared to increase the testing budget by orders of magnitude because each time a change was made to an object, testing would have to begin all over again. Use of an object-based testing tool allows for the test script to be modified for reuse. The impact of development changes in the test environment is greatly minimized. The frontrunners in object-based test tools in today's market are Rationale's Robot, Mercury's WinRunner, and Seque's Silk.

Each of these products can be purchased alone or in a suite of tools. The benefit of purchasing a suite of tools is that a suite contains applications that significantly help with the organization and management of the testing effort, which is the second consideration of the testing process. Rationale's SQA Manager is an excellent example of a group of tools that support the testing process.

SQA Manager allows test scripts sequences to be defined with dependencies and it keeps track of when, who, and which scripts are run. This ensures that tests can be run to verify object properties, then methods, then events as soon as the object is developed. The same scripts or a subset of them can be reused and be scheduled to rerun when the object is placed in the object-container and again when the application is integrated.

Having the tools selected up front in the testing process ensures that the capabilities they provide can be incorporated into the test plan, thereby maximizing the power of the tool, the reuse of scripts, and the level of quality built into the product.

The test plan, although listed in third place in the testing process, is essential in building a solid testing effort. It takes the testing from beginning to end in a logical, thorough process. A good plan will allow for testing to be performed in increments and keep pace with development.

THE PLAN

The use of a testing tool does not eliminate the need to plan. Rather, it ensures that a good plan can be implemented with better, more consistent results and repeated as modules are added, modified, or deleted. For example, using the automated test tool Rationale Robot to test at the object properties and methods level would be carried out by running Object Properties and Alphanumeric test scripts. The Object Properties test will capture and compare objects.

A Robot Alphanumeric script checks for case-sensitive or case-insensitive test, numbers, or a number within a range. It will also check to see if a field is blank and allow testers to tailor the test to specific values. Again, the description should specify how the test was set up and what values were used for verification.

Validating the objects in the containers might include Window Existence scripts that literally verify that the correct window exists in memory. For example, does a pushbutton (object) appear on the dialogue box (container) as expected? These scripts can be followed by event tests that ensure that each object in the container performs as expected. The event scripts may include customized .DLL or EXE routines constructed by the development team. List scripts to determine if the alphanumeric contents of list boxes, combo boxes, and multi-line edit controls work properly. Event scripts can also be created to verify file existence, menu selections up to five submenus deep, and file comparisons.

The integration test or system test validates the functions of the application to see if they meet the end-user business needs. These scripts capture the keystrokes of the end user and can include the common wait state scripts that ensure that data populates a screen within a specified period of time or that an object is accessible when it is supposed to be during day-to-day operations. Scripts can also be set up to ensure that the edits are being performed correctly, that data has been entered in all required fields, and that pop-up windows and dialog boxes appear when they are supposed to with the correct information.

For example, one test for a purchase order application might be to ensure that the correct forms are accessed. When the type of purchase is designated as Fashion items, the series of frames, forms, or windows accessed will be different than when the type of purchase is for Staple items. The test is set up to enter all required data, including the type of purchase to be made, then click on the "OK" button. A wait state is established for the "OK" button by indicating that it is grayed out after it is clicked, making it inactive and unusable until the next form is displayed. That is, the test tool will automatically check to see if the next form is displayed as a result of clicking "OK." The tester specifies how often the checks are made (e.g., every 2 seconds for up to 30 seconds). If the correct form is not found in the 30-second period, the test fails. If the correct form is identified in that time period, the test passes. The tool determines if the form is the correct form, based on tester-defined criteria for the forms; for example, in a linked test, the banner information of the correct form, Fashion or Staple, would be specified and verified by the tool.

When the type of script is selected, it is documented in the description, along with the values and other criteria used. This documentation can be created as comments within the script rather than as a separate word processing document.

What all of this means is that by the time tests are executed to verify that data is being saved correctly, and the right window pops-up when it is supposed to, it has already been proven that the windows all have a banner or header and that the label in every banner and header will present itself with the same color.

In other words, like tests, are done with like tests and those things that in days gone by were considered merely cosmetic are identified, cleaned up, and laid to rest before an application ever gets to system test. When the same objects are used to create each of the windows, it is only necessary to test that the windows were created using the approved objects. Objects need only to be tested when a revision is made to an existing object or a new object is created.

SUMMARY

The important thing to remember in testing object-based applications is that incremental development and user involvement make the process move along swiftly and more smoothly. When an object is created, it can be viewed by the user in a prototype. Changes can be easily made as the application moves from prototype to finished production system. When testing is managed and automated, it can be repeated and elaborated upon without starting from scratch because scripts are reusable and maintainable.

Testing the functionality of an application — whether it is object-based or traditional — requires the construction of business-use scenarios mapped to system requirements. The difference in testing the two types of application is in the approach used and type of automated testing tools available. To get started:

- Define the scope of the test.
- Get an understanding of what is supposed to happen when an object event or program control is triggered.
- Create single-event scenarios (based on the object event or the program control points).
- Cover as many "if-else" conditions as time allows.
- Build scenarios that exercise as many conditions as possible.
- If a testing tools is going to be used, determine what scripts need to be created and how they can be reused by defining variable or modifying specific lines in the script.

Notes

1. Microsoft, *Visual Basic 6.0, Programmer's Guide,* Microsoft Press, 1998.
2. Rationale, *SQA Suite Documentation,* Rationale University, 1996–1997.
3. Kaner, C., Falk, J., and Nguyen Hung Quoc, *Testing Computer Software,* 2nd ed., International Thomson Computer Press, 1998.

ABOUT THE AUTHOR

Polly Perryman Kuver has more than 20 years of computer experience, including 12 years in management positions. As a process engineer, her areas of expertise are national and international software engineering and documentation standards, quality assurance, configuration management, and data management. Currently, she is a consultant in the Boston, MA area.

Chapter 46
Security Testing Is Not All the Same: A Reference Taxonomy

Jim Kates

Testing the efficacy of security systems and networks has become a thriving business for many companies today. The reasons for implementing a security-testing program are varied, and no two organizations will find exactly the same rationale applicable to them. These include:

- Customer Confidence
- Legal Protection
- New Product/System Testing
- Fiduciary Responsibility
- Privacy Laws
- Insurance Requirements
- Government Regulations
- International Cooperation
- Trade Secret Protection

As the need for increased protection heats up, a flurry of terminology is being thrown around in an effort to impress clients, but it generally ends up just confusing them. Most customers do not know or understand the significant differences among the various security-testing methods, and too many vendors rely upon that ignorance to sell their wares and services.

Consultancies and audit firms, many of which are quite new to the world of security, offer to perform "penetration tests" or "security reviews" within a client's computing environment. On the surface, these services may resemble EDP audits that have been performed in the past; however,

0-8493-1161-6/02/$0.00+$1.50

that is not always the case. Requesting an EDP audit instead of one of several security testing alternatives may be a huge and costly mistake. Without clear guidance of what these services are and how they benefit an organization, executives are often confused about which is an appropriate approach.

This chapter attempts to clarify these confusions by explaining the important differences in the way computer security controls are evaluated and tested in real-world environments.

TO TEST OR NOT TO TEST: SECURITY-WISE

One question that looms large in a company's decision about how to proceed is: "Should we test a computing system's controls and security now, or should we wait until unauthorized persons, like hackers, try to exploit us?"

Many companies face this question today, especially those that are pursuing electronic commerce endeavors. Even though the answer appears easy, testing of security is often perceived as a cost without real benefit. After all, the adage goes, "If a security system [policy, staff] works well, you will see nothing." So, why spend money for something that may never occur? Furthermore, the internal development group often assures management that its new products, applications, or systems are secure so why should it be required that controls and security features be evaluated? This apparent arrogance, however, breeds trouble and additional vulnerabilities. Integrators and developers are not security specialists, no more than a security specialist should develop custom database applications without help.

If an executive were to buy that logic, why stop there? Why not fail to require a budget, expense reductions, or other key financial controls? Testing the system's controls is a necessary sanity check to ensure that the system works as expected and to identify risks before they are exploited.

The identification of exposures before they are exploited means reduced losses and less embarrassment. Thus, when executives understand the business reasoning for testing a system control's effectiveness, it makes the decision easier. Bottom line: it is a whole lot cheaper to build and test security controls in from the beginning, rather than treat them as an afterthought. However, two fundamental questions still have to be answered before proceeding:

Is it better to use a hacker, a security consultancy or an employee to test security and process controls? What type of security tests should be performed on which systems?

WHO DOES THE TESTING?

Using a hacker to try to compromise a computing system may be more of a risk than trying to solve the original problem. The unethical and criminal nature of many hackers does not stem exclusively from their lack of character or personal disregard for the law. It often merely arises from their lack of corporate work or real-life experience, which teaches the proper use and misuse of computing systems.

Using a hacker who does not understand how a business organization functions is like asking a baseball fan to replace the starting pitcher in a game. Besides all of the negative reasons, most general-purpose hackers do not understand security concepts and organizational rules, and they are usually limited in their skill sets. Lastly, if, for whatever reason, an organization chooses to use a hacker, determine if the hacker has ever been arrested or convicted of anything. Caveat emptor.

The preferred manner, though, is to contract a security expert as a consultant, but this too, poses several issues to the executive. The first issue is the cost. Good security consultants are not cheap. Cost is almost always the reason for going to one of the other two options: hackers or nothing at all. Budget accordingly; experienced security experts charge more than less experienced, and their work product usually reflects it. The second concern is the worry of confidentiality exposure, but that is easily handled as with other contractors or consultants: through confidentiality agreements.

Real security mechanisms are rarely built into new applications or system costs. Thus, every time a security issue comes up, it appears to be an added expense. However, for the well-run organization, security expenses (like penetration tests or security reviews) are an ongoing operation that is budgeted into the overall costs of running a business.

Employees are by far the most appropriate persons to perform ongoing security tests in conjunction with the consultancy. They are experienced with the systems, they will be around longer than the consultants will and they have a vested interest in keeping the systems secure. However, many organizations do not have the luxury of hiring their own security experts to test their system controls. Most of their security employees are busy implementing the controls. One concern that does arise is whether the organization is creating its own Frankenstein, e.g., an employee trained to break into any corporate system it owns. However, as with security consultants, that risk is mitigated with strong ethical consideration of the employees and no-nonsense legal clauses stating what happens when persons extend their legal authority in a system.

CHOOSING THE RIGHT TESTS

Choosing which tests to perform is sometimes more difficult than picking whom to use for the testing. Some tests are mandated, such as corporate audits. Some are performed as a normal part of the system or the security department's work product. The right test depends on what is desired, needed, or mandated. So why is it difficult to make an informed decision on which testing methodology to implement? Because too many consultants and audit firms indiscriminately throw around techno-babble and security terms loosely and incorrectly, thereby making it difficult to understand exactly what they mean or what is being offered. Different groups may use the same terms to describe different services or different confusing terms to describe the same techniques.

The following sections explain the different ways security controls are typically evaluated. They vary in scope and objectives. Often they vary in who contracts for and receives the end report. The main differences are the depth and extent of the work and how important it is to find the root cause of discovered vulnerabilities.

As should be expected, the costs and time vary significantly. However, the largest cost difference is whether the process is reactive or proactive. Reactive costs are greater, since more resources are usually used to expedite the report and solutions. In a world of the educated consumer, a better understanding of what is wanted and what will be delivered as an end product can help reduce the overall costs. So, planning ahead saves a ton of money.

Once an organization has decided to proceed with security testing, which approach or approaches will need to be taken? The following taxonomy of security testing will examine five distinct and separate ways of evaluating controls within a computing environment. Which approach is best suited to an organization's individual needs is a decision that should be determined with a consultant. This chapter does not favor or highlight one service over another, but merely does what many consultant firms have a difficult time doing: it explains the differences in the services in a way senior management will readily understand.

Penetration Tests

This overused and misunderstood term has created a lot of hype. It has become a buzz word thrown around to describe everything from a five-minute evaluation to a several-month-long consultant assignment. "Penetration testing" or "penetration analysis" is nothing more than a phrase to describe a legitimate attempt to compromise the expected controls of a process.

Often, the process of being "penetrated" is automated, like a computing system or network. The attempt is to identify, by the system owner, if the appropriate controls are being maintained properly and work as expected. The test tries to establish whether control mechanisms can be sidestepped or manipulated in a way that would allow a greater degree of access than is expected.

The results do not focus on whether or not organizational rules are being followed, such as the frequency of password changes. During a penetration test it is not relevant how good certain controls are, or how good a job the system administrator is doing protecting the system overall. That is because the singular objective of a penetration test is the successful compromise of controls under evaluation. It is these controls that are evaluated and the basis of the report is focused. Typical types of penetration tests include:

- External source penetration
- Internal source penetration
- Targeted system penetration

Be clear, though, that limitless assaults against the organization's systems present a new set of risks, including the danger of accidental systemic collapse or other denial of service events. So, whether for penetration tests or other security testing, establish the so-called "rules of engagement" before commencing the tests.

Tiger Teams

Tiger Teaming is another one of those misused terms that needs clarification. A Tiger Team is a group of individuals who legitimately (that is, with permission) attempt to compromise a set of physical or logical controls.

Tiger Teams are similar to penetration tests; however, they permit more varied styles of attacks, specifically physical ones. They go beyond the bounds of penetration teams and may revert to disguises or other ruses to accomplish their objectives. Tiger Teams might choose to break and enter into a facility to gain access to the network resources, or pretend to be an employee, contractor, or just the water or pizza man. In any case, he gets into the facility.

In some cases, the testing of the physical controls themselves may be the goal, and therefore, physical Tiger Team assaults are the only recourse. Most commercial companies shy away from this type of dedicated attack.

Tiger Teams generally have very specific objectives in mind, whereas the Penetration Test is more generalized. Tiger Team goals may be to compromise the physical protection of a key resource or to obtain specific trade secret information — be it in electronic or physical form. Tiger

Teams also may be testing an organization's rules or expected behavior in operational security areas such as:

- Incoming and outgoing physical inspections from the facility (diskettes and CDs)
- Remote keyless entry systems
- Alarms, sensors, and response mechanisms
- Testing efficacy of specific departments, job functions, or personnel in the performance of their duties
- Physical external perimeter testing
- Computing facilities test
- Special logical application tests

Vulnerability Assessment

Vulnerability assessments are expanded penetration tests with a specific scope and objectives. Their objective is not only to identify what problems may exist within the targeted systems, but also how these problems relate to other systems or applications.

Their scope is much greater than penetration tests, which merely try to compromise the controls. The goal of these assessments is to understand the complexity of the control and determine under which circumstance these controls could be compromised, even though they may be adequately protected at the present time. (The nuclear weapons labs perform expansive, hypothetical testing continuously, as technological capability proliferates.)

Vulnerability assessment goes beyond the mere technical and includes personnel functions that oversee technical process such as:

- Excessive authorities given to, or assumed by, individuals
- Separation of duties and dual controls
- Lack of management involvement within security process

The focus of the assessment is not only on the identified weaknesses themselves, but on what really caused and is the source of the problem. Typical vulnerability assessments include:

- Intrusion monitoring and reaction capabilities
- Interenterprise connectivity
- Competitive intelligence risks
- Remote access programs
- Internet connectivity
- Electronic commerce

Once the root causes have been determined, whether the customer takes any proactive corrective measures is another issue. The better security experts will make strong cases for additional defensive postures and policy and procedures changes.

Security Review

A security review is a formal analysis of the controls within an environment that are necessary to meet good business sense and organizational requirements. Going beyond the penetration test and vulnerability study, the security review focuses on the Big Picture, not just the bits and bytes. It delves into areas to discover which factors within the environment are not meeting expected standards. It mimics a formal audit without the formal reporting mechanisms. The review process usually allows the management chain involved to correct the violations found without escalation of the findings to higher regulatory authorities.

The security review may include penetration tests, tiger teams, or a vulnerability study as part of the review process. It is this extended scope or coverage that separates it from the other processes. It may include documentation reviews, audit trail examinations, and other details that may not be within the scope of other examinations. Typical security reviews are:

- Pre-audit preparations to ensure a clean bill of health under a rigidly controlled audit
- New business applications to regulatory agencies
- Migration to other platforms or networks

Forensic Investigations

Forensic investigations usually occur after a crime has been committed, if a company believes a crime if being committed, or after a serious security violation. Forensic investigations are very structured and the scope is strictly defined. During a forensic investigation, great care is taken to preserve all of the evidence that might be useful later in any legal proceedings, and to protect the specific physical and electronic environment from corruption from accidental modification because of investigative efforts.

The forensic process is extremely laborious and time consuming. It requires highly specialized skills and tools, and a strong understanding of the process. Forensic investigations are most likely driven from outside the organizational group it is reviewing, e.g., legal. One of the most difficult questions this investigation faces is, "Whom do you trust?" The investigator does not know who is involved in the situation, from management on down, or the full extent of their efforts to help thwart the investigation. Typical forensic investigations include:

- Fraud investigations
- Criminal investigations

- Electronic intrusion investigations
- Post-merger investigations

Audits

Audits are a necessary part of the business process; however, they are not often appreciated by those subjected to them. An audit is a control mechanism within itself, which oversees other control mechanisms. The audit process is not conducted to find out which staff people are wrong, but to determine which process may be weak or need improving. The audit is very similar to a security review, except that the handling of findings and the reporting process is much more formal and rigid.

The defined scope of the audit dictates what processes are reviewed and which are included in the formal report. However, audits are not only used to find mistakes or problems. They can be used to generate logic or reasoning to justify adding manpower or technical resources that management is often reticent to provide.

Audits can explain why additional security resources are needed and the reports of those findings are sent to the appropriate management. The audit is a conventional business process found in many organizations where none of the other security tests may ever be performed. In many organizations, it is the audit group that initially starts the other process or works with the information systems area to help define the requirements. Typical audits include:

- Internal audits (internal folks)
- External audits (external auditors)
- Specialty audits (applications, pre-merger, special tasks)

CONCLUSION

Evaluating security controls is a necessary component of any effective information security program and an essential business process. Choosing from a very specific set of options of just how these controls will be evaluated is key to the results one will receive. Keep in mind that each of these approaches has distinct methods, goals, and processes, and not every one is required for every situation.

Part of the process in deciding which approach is right for any organization is based upon the status of its internal programs and business applications:

- Are the controls already implemented?
- Are they in the process of being implemented?
- Are they still in the planning stages?

There is no right answer for every business, and a wide range of criteria must be taken into consideration:

- The depth of the review desired
- The budgetary constraints
- Insurance and legal implications
- Internal skills available to the organization.
- Executive commitment to isolating security vulnerabilities in a proactive manner

When testing the efficacy of an organization's security system, take the time to do so through a quality decision-making process. By evaluating and understanding the options, an organization can make decisions more effectively.

ABOUT THE AUTHOR

Jim Kates is the chief technology officer of the Security Experts, Inc., an international security consulting firm, based in Largo, FL. He can be reached at jim@securityexperts.com.

Chapter 47
Introduction to Penetration Testing

Stephen Fried

This chapter provides a general introduction to the subject of penetration testing and provides the security professional with the background needed to understand this special area of security analysis. Penetration testing can be a valuable tool for understanding and improving the security of a computer or network. However, it can also be used to exploit system weaknesses and attack systems and steal valuable information. By understanding the need for penetration testing, and the issues and processes surrounding its use, a security professional will be better able to use penetration testing as a standard part of the analysis toolkit.

This chapter presents penetration testing in terms of its use, application, and process. It is not intended as an in-depth guide to specific techniques that can be used to test penetration-specific systems. Penetration testing is an art that takes a great deal of skill and practice to do effectively. If not done correctly and carefully, the penetration test can be deemed invalid (at best) and, in the worst case, can actually damage the target systems. If the security professional is unfamiliar with penetration testing tools and techniques, it is best to hire or contract someone with a great deal of experience in this area to advise and educate the security staff of an organization.

WHAT IS PENETRATION TESTING?

Penetration testing is defined as a formalized set of procedures designed to bypass the security controls of a system or organization for the purpose of testing that system's or organization's resistance to such an attack. Penetration testing is performed to uncover the security weaknesses of a system and to determine the ways in which the system can be compromised by a potential attacker. Penetration testing can take several forms (which will be discussed later) but, in general, a test consists of a series of "attacks" against a target. The success or failure of the attacks, and how the target reacts to each attack, will determine the outcome of the test.

0-8493-1161-6/02/$0.00+$1.50

The overall purpose of a penetration test is to determine the subject's ability to withstand an attack by a hostile intruder. As such, the tester will be using the tricks and techniques a real-life attacker might use. This simulated attack strategy allows the subject to discover and mitigate its security weak spots before a real attacker discovers them.

The reason penetration testing exists is that organizations need to determine the effectiveness of their security measures. The fact that they want tests performed indicates that they believe there might be (or want to discover) some deficiency in their security. However, while the testing itself might uncover problems in the organization's security, the tester should attempt to discover and explain the underlying cause of the lapses in security that allowed the test to succeed. Simply stating that the tester was able to walk out of a building with sensitive information is not sufficient. The tester should explain that the lapse was due to inadequate attention by the guard on duty or a lack of guard staff training to recognize valuable or sensitive information.

There are three basic requirements for a penetration test.

1. *The test must have a defined goal and that goal should be clearly documented.* The more specific the goal, the easier it will be to recognize the success or failure of the test. A goal such as "break into the XYZ corporate network," while certainly attainable, is not as precise as "break into XYZ's corporate network from the Internet and gain access to the research department's file server." Each test should have a single goal. If the tester wishes to test several aspects of security at a business or site, several separate tests should be performed. This will enable the tester to more clearly distinguish between successful tests and unsuccessful attempts.
2. *The test should have a limited time period in which it is to be performed.* The methodology in most penetration testing is to simulate the types of attacks that will be experienced in the real world. It is reasonable to assume that an attacker will expend a finite amount of time and energy trying to penetrate a site. That time may range from one day to one year or beyond, but after that time is reached, the attacker will give up. In addition, the information being protected may have a finite useful "lifetime." The penetration test should acknowledge and accept this fact. Thus, part of the goal statement for the test should include a time limit that is considered reasonable based on the type of system targeted, the expected level of the threat, and the lifetime of the information.
3. *The test should have the approval of the management of the organization that is the subject of the test.* This is extremely important, as only the organization's management has the authority to permit this type of activity on its network and information systems.

TERMINOLOGY

The following terms associated with penetration testing are used throughout this chapter to describe penetration testing and the people and events involved in a penetration test.

Tester: The person or group who is performing the penetration test. The purpose of the tester is to plan and execute the penetration test and analyze the results for management. In many cases, the tester will be a member of the company or organization that is the subject of the test. However, a company may hire an outside firm to conduct the penetration test if it does not have the personnel or the expertise to do it itself.

Attacker: A real-life version of a tester. However, whereas the tester works with a company to improve its security, the attacker works against a company to steal information or resources.

Attack: The series of activities performed by the tester in an attempt to circumvent the security controls of a particular target. The attack may consist of physical, procedural, or electronic methods.

Subject of the test: The organization upon whom the penetration test is being performed. The subject can be an entire company or it can be a smaller organizational unit within that company.

Target of a penetration test: The system or organization that is being subjected to a particular attack at any given time. The target may or may not be aware that it is being tested. In either case, the target will have a set of defenses it presents to the outside world to protect itself against intrusion. It is those defenses that the penetration test is designed to test. A full penetration test usually consists of a number of attacks against a number of different targets.

Management: The term used to describe the leadership of an organization involved in the penetration test. There may be several levels of management involved in any testing effort, including the management of the specific areas of the company being tested, as well as the upper management of the company as a whole. The specific levels of management involved in the penetration testing effort will have a direct impact on the scope of the test. In all cases, however, it is assumed that the tester is working on behalf of (and sponsored by) at least one level of management within the company.

Penetration test (or more simply the **test**): The actual performance of a simulated attack on the target.

WHY TEST?

There are several reasons why an organization may want a penetration test performed on its systems or operations. The first (and most prevalent) is to determine the effectiveness of the security controls the organization has put into place. These controls may be technical in nature, affecting the computers, network, and information systems of the organization. They may be operational in nature, pertaining to the processes and procedures a company has in place to control and secure information. Finally, they may be physical in nature. The tester may be trying to determine the effectiveness of the physical security a site or company has in place. In all cases, the goal of the tester will be to determine if the existing controls are sufficient by trying to get around them.

The tester may also be attempting to determine the vulnerability an organization has to a particular threat. Each system, process, or organization has a particular set of threats to which it feels it is vulnerable. Ideally, the organization will have taken steps to reduce its exposure to those threats. The role of the tester is to determine the effectiveness of these countermeasures and to identify areas for improvement or areas where additional countermeasures are required. The tester may also wish to determine whether the set of threats the organization has identified is valid and whether or not there are other threats against which the organization might wish to defend itself.

A penetration test can sometimes be used to bolster a company's position in the marketplace. A test, executed by a reputable company and indicating that the subject's environment withstood the tester's best efforts, can be used to give prospective customers the appearance that the subject's environment is secure. The word "*appearance*" is important here because a penetration test cannot examine all possible aspects of the subject's environment if it is even moderate in size. In addition, the security state of an enterprise is constantly changing as new technology replaces old, configurations change, and business needs evolve. The "environment" the tester examines may be very different from the one of which the customer will be a part. If a penetration test is used as proof of the security of a particular environment for marketing purposes, the customer should insist on knowing the details, methodology, and results of the test.

A penetration test can be used to alert the corporation's upper management to the security threat that may exist in its systems or operations. While security weaknesses in a system may be general knowledge or technical staff may have specific knowledge of particular threats and vulnerabilities, this message may not always be transmitted to management. As a result, management may not fully understand or appreciate the magnitude of the security problem. A well-executed penetration test can systematically uncover vulnerabilities that management was unaware existed. The presentation of concrete evidence of security problems, along with an analysis of the damage those problems can cause to the company, can be an effective wake-up call to management and spur them into paying more attention to information security issues. A side effect of this wake-up call may be that once management understands the nature of the threat and the magnitude to which the company is vulnerable, it may be more willing to expend money and resources to address not only the security problems uncovered by the test, but also ancillary security areas needing additional attention by the company. These ancillary issues may include a general security awareness program or the need for more funding for security technology. A penetration test that uncovers moderate or serious problems in a company's security can be effectively used to justify the time and expense required to implement effective security programs and countermeasures.

TYPES OF PENETRATION TESTING

The typical image of a penetration test is that of a team of high-tech computer experts sitting in a small room attacking a company's network for days on end or crawling through the ventilation shafts to get into the company's "secret room." While this may be a glamorous image to use in the movies, in reality the penetration test works in a variety of different (and very nonglamorous) ways.

The first type of testing involves the physical infrastructure of the subject. Very often, the most vulnerable parts of a company are not found in the technology of its information network or the access controls found in its databases. Security problems can be found in the way the subject handles its physical security. The penetration tester will seek to exploit these physical weaknesses. For example, does the building provide adequate access control? Does the building have security guards, and do the guards check people as they enter or leave a building? If intruders are able to walk unchecked into a company's building, they will be able to gain physical access to the information they seek. A good test is to try to walk into a building during the morning when everyone is arriving to work. Try to get in the middle of a crowd of people to see if the guard is adequately checking the badges of those entering the building.

Once inside, check if sensitive areas of the building are locked or otherwise protected by physical barriers. Are file cabinets locked when not in use? How difficult is it to get into the communications closet where all the telephone and network communication links terminate? Can a person walk into employee office areas unaccompanied and unquestioned? All the secure and sensitive areas of a building should be protected against unauthorized entry.

If they are not, the tester will be able to gain unrestricted access to sensitive company information.

While the physical test includes examining protections against unauthorized entry, the penetration test might also examine the effectiveness of controls prohibiting unauthorized exit. Does the company check for theft of sensitive materials when employees exit the facility? Are laptop computers or other portable devices registered and checked when entering and exiting the building? Are security guards trained not only on what types of equipment and information to look for, but also on how equipment can be hidden or masked and why this procedure is important?

Another type of testing examines the operational aspects of an organization. Whereas physical testing investigates physical access to company computers, networks, or facilities, operational testing attempts to determine the effectiveness of the operational procedures of an organization by attempting to bypass those procedures. For example, if the company's help desk requires each user to give personal or secret information before help can be rendered, can the tester bypass those controls by telling a particularly believable "sob story" to the technician answering the call? If the policy of the company is to "scramble" or demagnetize disks before disposal, are these procedures followed? If not, what sensitive information will the tester find on disposed disks and computers? If a company has strict policies concerning the authority and process required to initiate ID or password changes to a system, can someone simply claiming to have the proper authority (without any actual proof of that authority) cause an ID to be created, removed, or changed? All these are attacks against the operational processes a company may have, and all of these techniques have been used successfully in the past to gain entry into computers or gain access to sensitive information.

The final type of penetration test is the electronic test. Electronic testing consists of attacks on the computer systems, networks, or communications facilities of an organization. This can be accomplished either manually or through the use of automated tools. The goal of electronic testing is to determine if the subject's internal systems are vulnerable to an attack through the data network or communications facilities used by the subject.

Depending on the scope and parameters of a particular test, a tester may use one, two, or all three types of tests. If the goal of the test is to gain access to a particular computer system, the tester may attempt a physical penetration to gain access to the computer's console or try an electronic test to attack the machine over the network. If the goal of the test is to see if unauthorized personnel can obtain valuable research data, the tester may use operational testing to see if the information is tracked or logged when accessed or copied and determine who reviews those access logs. The tester may then switch to electronic penetration to gain access to the computers where the information is stored.

WHAT ALLOWS PENETRATION TESTING TO WORK?

There are several general reasons why penetration tests are successful. Most reasons are in the operational area; however, security problems can arise due to deficiencies in any of the three testing areas.

A large number of security problems arise due to a lack of awareness on the part of a company's employees of the company's policies and procedures regarding information security and protection. If employees and contractors of a company do not know the proper procedures for handling proprietary or sensitive information, they are much more likely to allow that information to be left unprotected. If employees are unaware of the company policies on discussing sensitive company information, they will often volunteer (sometimes unknowingly) information about their company's future sales, marketing, or research plans simply by being asked the right set of questions. The tester will exploit this lack of awareness and modify the testing procedure to account for the fact that the policies are not well known.

In many cases, the subjects of the test will be very familiar with the company's policies and the procedures for handling information. Despite this, however, penetration testing works because often people do not adhere to standardized procedures defined by the company's policies. Although the policies may say that system logs should be reviewed daily, most administrators are too busy to bother. Good administrative and security practices require that system configurations should be checked periodically to detect tampering, but this rarely happens. Most security policies indicate minimum complexities and maximum time limits for password, but many systems do not enforce these policies. Once the tester knows about these security procedural lapses, they become easy to exploit.

Many companies have disjointed operational procedures. The processes in use by one organization within a company may often conflict with the processes used by another organization. Do the procedures used by one application to authenticate users complement the procedures used by other applications, or are there different standards in use by different applications? Is the access security of one area of a company's network lower than that of another part of the network? Are log files and audit records reviewed uniformly for all systems and services, or are some systems monitored more closely than others? All these are examples of a lack of coordination between organizations and processes. These examples can be exploited by the tester and used to get closer to the goal of the test. A tester needs only to target the area with the lower authentication standards, the lower access security, or the lower audit review procedures in order to advance the test.

Many penetration tests succeed because people often do not pay adequate attention to the situations and circumstances in which they find themselves. The hacker's art of social engineering relies heavily on this fact. Social engineering is a con game used by intruders to trick people who know secrets into revealing them. People who take great care in protecting information when at work (locking it up or encrypting sensitive data, for example) suddenly forget about those procedures when asked by an acquaintance at a party to talk about their work. Employees who follow strict user authentication and system change control procedures suddenly "forget" all about them when they get a call from the "Vice President of Such and Such" needing something done "right away." Does the "Vice President" himself usually call the technical support line with problems? Probably not, but people do not question the need for information, do not challenge requests for access to sensitive information even if the person asking for it does not clearly have a need to access that data, and do not compare the immediate circumstances with normal patterns of behavior.

Many companies rely on a single source for enabling an employee to prove identity, and often that source has no built-in protection. Most companies assign employee identification (ID) numbers to their associates. That number enables access to many services the company has to offer, yet is displayed openly on employee badges and freely given when requested. The successful tester might determine a method for obtaining or generating a valid employee ID number in order to impersonate a valid employee.

Many hackers rely on the anonymity that large organizations provide. Once a company grows beyond a few hundred employees, it becomes increasingly difficult for anyone to know all employees by sight or by voice. Thus, the IT and HR staff of the company need to rely on other methods of user authentication, such as passwords, key cards, or the above-mentioned employee ID number. Under such a system, employees become anonymous entities, identified only by their ID numbers or their passwords . This makes it easier to assume the identity of a legitimate employee or to use social engineering to trick people into divulging information. Once the tester is able to hide within the anonymous structure of the organization, the fear of discovery is reduced and the tester will be in a much better position to continue to test.

Another contributor to the successful completion of most penetration tests is the simple fact that most system administrators do not keep their systems up to date with the latest security patches and fixes for the systems under their control. A vast majority of system break-ins occur as a result of exploitation of known vulnerabilities — vulnerabilities that could have easily been eliminated by the application of a system patch, configuration change, or procedural change. The fact that system operators continue to let systems fall behind in security configuration means that testers will continuously succeed in penetrating their systems.

The tools available for performing a penetration test are becoming more sophisticated and more widely distributed. This has allowed even the novice hacker to pick up highly sophisticated tools for exploiting system weaknesses and applying them without requiring any technical background in how the tool works. Often these tools can try hundreds of vulnerabilities on a system at one time. As new holes are found, the hacker tools exploit them faster than the software companies can release fixes, making life even more miserable for the poor administrator who has to keep pace. Eventually, the administrator will miss something, and that something is usually the one hole that a tester can use to gain entry into a system.

ABOUT THE AUTHOR

Stephen Fried is the senior manager for Global Risk Assessment and Secure Business Solutions at Lucent Technologies, leading the team responsible for determining the security threats to Lucent's internal systems and services. Stephen is a Certified Information Systems Security Professional and has been a featured speaker on information security and technology at meetings and conferences worldwide.

Chapter 48
Performing Penetration Testing

Stephen Fried

Every security professional who performs a penetration test will approach the task somewhat differently, and the actual steps used by the tester will vary from engagement to engagement. However, there are several basic strategies that can be said to be common across most testing situations.

First, do not rely on a single method of attack. Different situations call for different attacks. If the tester is evaluating the physical security of a location, the tester may try one method of getting in the building, for example, walking in the middle of a crowd during the morning inrush of people. If that does not work, try following the cleaning people into a side door. If that does not work, try something else. The same method holds true for electronic attacks. If one attack does not work (or the system is not susceptible to that attack), try another.

Choose the path of least resistance. Most real attackers will try the easiest route to valuable information, so the penetration tester should use this method as well. If the test is attempting to penetrate a company's network, the company's firewall might not be the best place to begin the attack (unless, of course, the firewall was the stated target of the test) because that is where all the security attention will be focused. Try to attack lesser-guarded areas of a system. Look for alternate entry points; for example, connections to a company's business partners, analog dial-up services, modems connected to desktops, etc. Modern corporate networks have many more connection points than just the firewall, so use them to the fullest advantage.

Feel free to break the rules. Most security vulnerabilities are discovered because someone has expanded the limits of a system's capabilities to the point where it breaks, thus revealing a weak spot in the system. Unfortunately, most users and administrators concentrate on making their systems conform to the stated policies of the organization. Processes work well when everyone follows the rules, but can have unpredictable results

0-8493-1161-6/02/$0.00+$1.50

when those rules are broken or ignored. Therefore, when performing a test attack, use an extremely long password; enter a thousand-byte URL into a Web site; sign someone else's name into a visitors log; try anything that represents abnormality or nonconformance to a system or process. Real attackers will not follow the rules of the subject system or organization — nor should the tester.

Do not rely exclusively on high-tech, automated attacks. While these tools may seem more "glamorous" (and certainly easier) to use they may not always reveal the most effective method of entering a system. There are a number of "low-tech" attacks that, while not as technically advanced, may reveal important vulnerabilities and should not be overlooked. Social engineering is a prime example of this type of approach. The only tools required to begin a social engineering attack are the tester's voice, a telephone, and the ability to talk to people. Yet despite the simplicity of the method (or, perhaps, because of it), social engineering is incredibly effective as a method of obtaining valuable information.

"Dumpster diving" can also be an effective low-tech tool. Dumpster diving is a term used to describe the act of searching through the trash of the subject in an attempt to find valuable information. Typical information found in most dumpsters includes old system printouts, password lists, employee personnel information, drafts of reports, and old fax transmissions. While not nearly as glamorous as running a port scan on a subject's computer, it also does not require any of the technical skill that port scanning requires. Nor does it involve the personal interaction required of social engineering, making it an effective tool for testers who may not be highly skilled in interpersonal communications.

One of the primary aims of the penetration tester is to avoid detection. The basic tenet of penetration testing is that information can be obtained from a subject without his or her knowledge or consent. If a tester is caught in the act of testing, this means, by definition, that the subject's defenses against that particular attack scenario are adequate. Likewise, the tester should avoid leaving "fingerprints" that can be used to detect or trace an attack. These fingerprints include evidence that the tester has been working in and around a system. The fingerprints can be physical (e.g., missing reports, large photocopying bills) or they can be virtual (e.g., system logs detailing access by the tester, or door access controls logging entry and exit into a building). In either case, fingerprints can be detected and detection can lead to a failure of the test.

Do not damage or destroy anything on a system unless the destruction of information is defined as part of the test and approved (in writing) by management. The purpose of a penetration test is to uncover flaws and weaknesses in a system or process, — not to destroy information. The actual destruction of company information not only deprives the company of its (potentially valuable) intellectual property, but it may also be construed as unethical behavior and subject the tester to disciplinary or legal action. If the management of the organization wishes the tester to demonstrate actual destruction of information as part of the test, the tester should be sure to document the requirement and get written approval of the management involved in the test. Of course, in the attempt to "not leave fingerprints," the tester might wish to alter the system logs to cover the tester's tracks. Whether or not this is acceptable is an issue that the tester should discuss with the subject's management before the test begins.

Do not pass up opportunities for small incremental progress. Most penetration testing involves the application of many tools and techniques in order to be successful. Many of these techniques will not completely expose a weakness in an organization or point to a failure of an organization's security. However, each of these techniques may move the tester closer and closer to the final goal of the test. By looking for a single weakness or vulnerability that will completely expose the organization's security, the tester may overlook many important, smaller weaknesses that, when combined, are just as important. Real-life attackers can have infinite patience; so should the tester.

Finally, be prepared to switch tactics. Not every test will work, and not every technique will be successful. Most penetration testers have a standard "toolkit" of techniques that work on most systems. However, different systems are susceptible to different attacks and may call for different testing measures. The tester should be prepared to switch to another method if the current one is not working. If an electronic attack is not yielding the expected results, switch to a physical or operational attack. If attempts to circumvent a company's network connectivity are not working, try accessing the network through the company's dial-up connections. The attack that worked last time may not be successful this time, even if the subject is the same company. This may either be because something has changed in the target's environment or the target has (hopefully) learned its lesson from the last test. Finally, unplanned opportunities may present themselves during a test. Even an unsuccessful penetration attempt may expose the possibility that other types of attack may be more successful. By remaining flexible and willing to switch tactics, the tester is in a much better position to discover system weaknesses.

PLANNING THE TEST

Before any penetration testing can take place, a clear testing plan must be prepared. The test plan will outline the goals and objectives of the test, detail the parameters of the testing process, and describe the expectations of both the testing team and the management of the target organization.

The most important part of planning any penetration test is the involvement of the management of the target organization. Penetration testing without management approval, in addition to being unethical, can reasonably be considered "espionage" and is illegal in most jurisdictions. The tester should fully document the testing engagement in detail and get the written approval from management before proceeding. If the testing team is part of the subject organization, it is important that the management of that organization knows about the team's efforts and approves of them. If the testing team is outside the organizational structure and is performing the test "for hire" the permission of management to perform the test should be included as part of the contract between the testing organization and the target organization. In all cases, be sure that the management that approves the test has the authority to give such approval. Penetration testing involves attacks on the security infrastructure of an organization. This type of action should not be approved or undertaken by someone who does not clearly have the authority to do so.

By definition, penetration testing involves the use of simulated attacks on a system or organization with the intent of penetrating that system or organization. This type of activity will, by necessity, require that someone in the subject organization be aware of the testing. Make sure that those with a need to know about the test do, in fact, know of the activity. However, keep the list of people aware of the test to an absolute minimum. If too many people know about the test, the activities and operations of the target may be altered (intentionally or unintentionally) and negate the results of the testing effort. This alteration of behavior to fit expectations is known as the Hawthorne effect (named after a famous study at Western Electric's Hawthorne factory whose employees, upon discovering that their behavior was being studied, altered their behavior to fit the patterns they believed the testers wanted to see).

Finally, during the course of the test, many of the activities the tester will perform are the very same ones that real-life attackers will use to penetrate systems. If the staff of the target organization discovers these activities, they may (rightly) mistake the test for a real attack and catch the "attacker" in the act. By making sure that appropriate management personnel are aware of the testing activities, the tester will be able to validate the legitimacy of the test.

An important ethical note to consider is that the act of penetration testing involves intentionally breaking the rules of the subject organization in order to determine its security weaknesses. This requires the tester to use many of the same tools and methods that real-life attackers use.

However, real hackers sometime break the law or engage in highly questionable behavior in order to carry out their attacks. The security professional performing the penetration test is expected to draw the line between bypassing a company's security procedures and systems and actually breaking the law. These distinctions should be discussed with management prior to the commencement of the test, and discussed again if any ethical or legal problems arise during the execution of the test.

Once management has agreed to allow a penetration test, the parameters of the test must be established. The testing parameters will determine the type of test to be performed, the goals of the tests, and the operating boundaries that will define how the test is run. The primary decision is to determine precisely what is being tested. This definition can range from broad ("test the ability to break into the company's network") to extremely specific ("determine the risk of loss of technical information about XYZ's latest product"). In general, more specific testing definitions are preferred, as it becomes easier to determine the success or failure of the test. In the case of the second example, if the tester is able to produce a copy of the technical specifications, the test clearly succeeded. In the case of the first example, does the act of logging in to a networked system constitute success, or does the tester need to produce actual data taken from the network? Thus, the specific criteria for success or failure should be clearly defined.

The penetration test plan should have a defined time limit. The time length of the test should be related to the amount of time a real adversary can be expected to attempt to penetrate the system and also the reasonable lifetime of the information itself. If the data being attacked has an effective lifetime of two months, a penetration test can be said to succeed if it successfully obtains that data within a two-month window.

The test plan should also explain any limits placed on the test by either the testing team or management. If there are ethical considerations that limit the amount of "damage" the team is willing to perform, or if there are areas of the system or operation that the tester is prohibited from accessing (perhaps for legal or contractual reasons), these must be clearly explained in the test plan. Again, the testers will attempt to act as real-life attackers and attackers do not follow any rules. If management wants the testers to follow certain rules, these must be clearly defined. The test plan should also set forth the procedures and effects of "getting caught" during the test. What defines "getting caught" and how that affects the test should also be described in the plan.

Once the basic parameters of the test have been defined, the test plan should focus on the "scenario" for the test. The scenario is the position the tester will assume within the company for the duration of the test. For example, if the test is attempting to determine the level of threat from company insiders (employees, contractors, temporary employees, etc.), the tester may be given a temporary job within the company. If the test is designed to determine the level of external threat to the organization, the tester will assume the position of an "outsider." The scenario will also define the overall goal of the test. Is the purpose of the test a simple penetration of the company's computers or facilities? Is the subject worried about loss of intellectual property via physical or electronic attacks? Is the subject worried about vandalism to its Web site, fraud in its electronic commerce systems, or protection against denial-of-service attacks? All these factors help to determine the test scenario and are extremely important for the tester to plan and execute an effective attack.

PERFORMING THE TEST

Once all the planning has been completed, the test scenarios have been established, and the tester has determined the testing methodology, it is time to perform the test. In many aspects, the execution of a penetration test plan can be compared to the execution of a military campaign. In such a campaign, there are three distinct phases: reconnaissance, attack, and (optionally) occupation.

During the reconnaissance phase (often called the "discovery" phase) the tester will generally survey the "scene" of the test. If the tester is planning a physical penetration, the reconnaissance stage will consist of examining the proposed location for any weaknesses or vulnerabilities. The tester should look for any noticeable patterns in the way the site operates. Do people come and go at regular intervals? If there are guard services, how closely do they examine people entering and leaving the site? Do they make rounds of the premises after normal business hours, and are those rounds conducted at regular times? Are different areas of the site occupied at different times? Do people seem to all know one another, or do they seem to be strangers to each other? The goal of physical surveillance is to become as completely familiar with the target location as possible and to establish the repeatable patterns in the site's behavior. Understanding those patterns and blending into them can be an important part of the test.

If an electronic test is being performed, the tester will use the reconnaissance phase to learn as much about the target environment as possible. This will involve a number of mapping and surveillance techniques. However, because the tester cannot physically observe the target location, electronic probing of the environment must be used. The tester will start by developing an electronic "map" of the target system or network. How is the network laid out? What are the main access points, and what type of equipment runs the network? Are the various hosts identifiable, and what operating systems or platforms are they running? What other networks connect to this one? Is dial-in service available to get into the network, and is dial-out service available to get outside?

Reconnaissance does not always have to take the form of direct surveillance of the subject's environment. It can also be gathered in other ways that are more indirect. For example, some good places to learn about the subject are:

- Former or disgruntled employees
- Local computer shows
- Local computer club meetings
- Employee lists, organization structures
- Job application handouts and tours
- Vendors who deliver food and beverages to the site

All this information will assist the tester in determining the best type of attack(s) to use based on the platforms and service available. For each environment (physical or electronic), platform, or service found during the reconnaissance phase, there will be known attacks or exploits that the tester can use. There may also be new attacks that have not yet made it into public forums. The tester must rely on the experience gained in previous tests and the knowledge of current events in the field of information security to keep abreast of possible avenues of attack.

The tester should determine (at least preliminarily) the basic methods of attack to use, the possible countermeasures that may be encountered, and the possible responses to those countermeasures.

The next step is the actual attack on the target environment. The attack will consist of exploiting the weaknesses found in the reconnaissance phase to gain entry to the site or system and to bypass any controls or restrictions that may be in place. If the tester has done a thorough job during the reconnaissance phase, the attack phase becomes much easier.

Timing during the attack phase can be critical. There may be times when the tester has the luxury of time to execute an attack, and this provides the greatest flexibility to search, test, and adjust to the environment as it unfolds. However, in many cases, an abundance of time is not available. This may be the case if the tester is attempting to enter a building in between guard rounds, attempting to gather information from files during the owner's lunch hour, or has tripped a known alarm and is attempting to complete the attack before the system's intrusion response interval (the amount of time between the recognition of a penetration and the initiation of the response or countermeasure) is reached. The tester should have a good idea of how long a particular attack should take to perform and have a reasonable expectation that it can be performed in the time available (barring any unexpected complications).

If, during an attack, the tester gains entry into a new computer or network, the tester may elect to move into the occupation phase of the attack. Occupation is the term used to indicate that the tester has established the target as a base of operations. This may be because the tester wants to spend more time in the target gathering information or monitoring the state of the target, or the tester may want to use the target as a base for launching attacks against other targets. The occupation phase presents perhaps the greatest danger to the tester, because the tester will be exposed to detection for the duration of the time he or she is resident in the target environment. If the tester chooses to enter the occupation phase, steps should be taken to make the tester's presence undetectable to the greatest extent possible.

It is important to note that a typical penetration test may repeat the reconnaissance/attack/occupation cycle many times before the completion of the test. As each new attack is prepared and launched, the tester must react to the attack results and decide whether to move on to the next step of the test plan, or abandon the current attack and begin the reconnaissance for another type of attack. Through the repeated and methodical application of this cycle the tester will eventually complete the test.

Each of the two basic test types — physical and electronic— has different tools and methodologies. Knowledge of the strengths and weaknesses of each type will be of tremendous help during the execution of the penetration test. For example, physical penetrations generally do not require an in-depth knowledge of technical information. While they may require some specialized technical experience (bypassing alarm systems, for example), physical penetrations require skills in the area of operations security, building and site operations, human nature, and social interaction.

The "tools" used during a physical penetration vary with each tester, but generally fall into two general areas: abuse of protection systems and abuse of social interaction. Examples of abuse of protection systems include walking past inattentive security guards, piggybacking (following someone through an access-controlled door), accessing a file room that is accidentally unlocked, falsifying an information request, or picking up and copying information left openly on desks. Protection systems are established to protect the target from typical and normal threats. Knowledge of the operational procedures of the target will enable the tester to develop possible test scenarios to test those operations in the face of both normal and abnormal threats.

Lack of security awareness on the part of the victim can play a large part in any successful physical penetration test. If people are unaware of the value of the information they possess, they are less likely to protect it properly. Lack of awareness of the policies and procedures for storing and handling sensitive information is widespread in many companies. The penetration tester can exploit this weakness in order to gain access to information that should otherwise be unavailable.

Finally, social engineering is perhaps the ultimate tool for effective penetration testing. Social engineering exploits vulnerabilities in the physical and process controls, adds the element of "insider" assistance, and combines it with the lack of awareness on the part of the subject that they have actually contributed to the penetration. When done properly, social engineering can provide a formidable attack strategy.

Electronic penetrations, on the other hand, generally require more in-depth technical knowledge than do physical penetrations. In the case of many real-life attackers, this knowledge can be their own or "borrowed" from somebody else. In recent years, the technical abilities of many new attackers seem to have decreased, while the high availability of penetration and attack tools on the Internet, along with the sophistication of those tools, has increased. Thus, it has become relatively simple for someone without a great deal of technical knowledge to "borrow" the knowledge of the tool's developer and inflict considerable damage on a target. There are, however, still a large number of technically advanced attackers out there with the skill to launch a successful attack against a system.

The tools used in an electronic attack are generally those that provide automated analysis or attack features. For example, many freely available host and network security analysis tools provide the tester with an automated method for discovering a system's vulnerabilities. These are vulnerabilities that the skilled tester may be able to find manually, but the use of automated tools provides much greater efficiency. Likewise, tools like port scanners (that tell the tester what ports are in use on a target host), network "sniffers" (that record traffic on a network for later analysis), and "war dialers" (that systematically dial phone numbers to discover accessible modems) provide the tester with a wealth of knowledge about weaknesses in the target system and possible avenues the tester should take to exploit those weaknesses.

When conducting electronic tests there are three basic areas to exploit: the operating system, the system configuration, and the relationship the system has to other systems. Attacks against the operating system exploit bugs or holes in the platform that have not yet been patched by the administrator or the manufacturer of the platform. Attacks against the system configuration seek to exploit the natural tendency of overworked administrators not to keep up with the latest system releases and to overlook such routine tasks as checking system logs, eliminating unused accounts, or improper configuration of system elements. Finally, the tester can exploit the relationship a system has with respect other systems to which it connects. Does it have a trust relationship with a target system? Can the tester establish administrative rights on the target machine through another machine? In many cases, a successful penetration test will result not from directly attacking the target machine, but from first successfully attacking systems that have some sort of "relationship" to the target machine.

REPORTING RESULTS

The final step in a penetration test is to report the findings of the test to management. The overall purpose and tone of the report should actually be set at the beginning of the engagement with management's statement of their expectation of the test process and outcome. In effect, what the tester is asked to look for will determine, in part, the report that is produced. If the tester is asked to examine a company's overall physical security, the report will reflect a broad overview of the various security measures the company uses at its locations. If the tester is asked to evaluate the controls surrounding a particular computer system, the report will most likely contain a detailed analysis of that machine.

The report produced as a result of a penetration test contains extremely sensitive information about the vulnerabilities the subject has and the exact attacks that can be used to exploit those vulnerabilities. The penetration tester should take great care to ensure that the report is only distributed to those within the management of the target who have a need to know. The report should be marked with the company's highest sensitivity label. In the case of particularly sensitive or classified information, there may be several versions of the report, with each version containing only information about a particular functional area.

The final report should provide management with a replay of the test engagement in documented form. Everything that happened during the test should be documented. This provides management with a list of the vulnerabilities of the target and allows them to assess the methods used to protect against future attacks.

First, the initial goals of the test should be documented. This will assist anyone who was not part of the original decision-making process is becoming familiar with the purpose and intent of the testing exercise. Next, the methodology used during the test should be described. This will include information about the types of attacks used, the success or failure of those attacks, and the level of difficulty and resistance the tester experienced during the test. While providing too much technical detail about the precise methods used may be overly revealing and (in some cases) dangerous, the general methods and procedures used by the testing team should be included in the report. This can be an important tool for management to get a sense of how easy or difficult it was for the testing team to penetrate the system. If countermeasures are to be put in place, they will need to be measured for cost-effectiveness against the value of the target and the vulnerabilities found by the tester. If the test revealed that a successful attack would cost the attacker U.S. $10 million, the company might not feel the need for additional security in that area. However, if the methodology and procedures show that an attack can be launched from the Internet for the price of a home computer and an Internet connection, the company might want to put more resources into securing the target.

The final report should also list the information found during the test. This should include information about what was found, where it was found, how it was found, and the difficulty the tester had in finding it. This information is important to give management a sense of the depth and breadth of the security problems uncovered by the test. If the list of items found is only one or two items long, it might not trigger a large response (unless, of course, the test was only looking for those one or two items). However, if the list is several pages long, it might spur management into making dramatic improvements in the company's security policies and procedures.

The report should give an overall summary of the security of the target in comparison with some known quantity for analysis. For example, the test might find that 10 percent of the passwords on the subject's computers were easily guessed. However, previous research or the tester's own experience might show that the average computer on the Internet or other clients contains 30 percent easily guessed passwords. Thus, the company is actually doing better than the industry norm. However, if the report shows that 25 percent of the guards in the company's buildings did not check for employee badges during the test, that would most likely be considered high and be cause for further action.

The report should also compare the initial goals of the test to the final result. Did the test satisfy the requirements set forth by management? Were the results expected or unexpected, and to what degree? Did the test reveal problems in the targeted area, or were problems found in other unrelated areas? Was the cost or complexity of the tests in alignment with the original expectations of management?

Finally, the report should also contain recommendations for improvement of the subject's security. The recommendations should be based on the findings of the penetration test and include not only the areas covered by the test, but ancillary areas might help improve the security of the tested areas. For example, inconsistent system configuration might indicate a need for a more stringent change control process. A successful social engineering attempt that allowed the tester to obtain a password from the company's help desk might lead to better user authentication requirements.

CONCLUSION

Although it seems to parallel the activities of real attackers, penetration testing, in fact, serves to alert the owners of computer and networks to the real dangers present in their systems. Other risk analysis activities, such as automated port scanning, war dialing, and audit log reviews, tend to point out the theoretical vulnerabilities that might exist in a system. The owner of a computer will look at the output from one of these activities and see a list of holes and weak spots in a system without getting a good sense of the actual threat these holes represent. An effective penetration test, however, will show that same system owner the actual damage that can occur if those holes are not addressed. It brings to the forefront the techniques that can be used to gain access to a system or site and makes clear the areas that need further attention. By applying the proper penetration testing techniques (in addition to the standard risk analysis and mitigation strategies), the security professional can provide a complete security picture of the subject's enterprise.

ABOUT THE AUTHOR

Stephen Fried is the senior manager for Global Risk Assessment and Secure Business Solutions at Lucent Technologies, leading the team responsible for determining the security threats to Lucent's internal systems and services. Lucent's Corporate Computer and Network Security Organization. He is a Certified Information Systems Security Professional and has been a featured speaker on information security and technology at meetings and conferences worldwide.

Section VIII
Solution Performance and Security

This section focuses on methods for ensuring the security and performance of solutions over the Internet. CIOs often cite both of these elements, with a focus on the former, as critical success factors in their Internet-enabled solutions. The chapters in this section examine a variety of tools and techniques to protect and enhance both these factors.

"Framework for Internet Security Planning" (Chapter 49) offers a holistic process that is aimed at improving the overall Internet chain of security. This is done through a framework for understanding the issues involved in Internet security and assessing available security options.

"Internet Security: Securing the Perimeter" (Chapter 50) examines common threats to the Internet and presents a plan to ensure that corporate assets are protected.

"Security Management for the World Wide Web" (Chapter 51) discusses the need for an underlying baseline security framework that enables organizations to successfully evolve to doing business over the Internet and using internal Intranets and other World Wide Web-based technologies. The chapter also describes a solution set that leverages existing skills, resources, and security implementations.

"Firewall Management and Internet Attacks" (Chapter 52) explores the security issues specific to firewall management, namely, choosing a firewall, laying the groundwork for a firewall, implementing a firewall, conducting firewall operations, and establishing and enforcing firewall policy and standards.

"Protecting against Hacker Attacks" (Chapter 53) provides a profile of hackers and hacker clubs, along with their methods of communication, and specific methods of information gathering and attacks. Recommended procedures and controls for countering these activities are also examined.

"Improving Performance in New Programming Environments: Java" (Chapter 54) discusses how the steps for performance improvement in Java-based applications differ from those for compiled languages. This is done through a detailed case example.

"Optimizing Web Design and Database Performance" (Chapter 55) examines strategies, tips, and tools that provide site vistors with an optimal Web experience.

Chapter 49
Framework for Internet Security Planning

Monica J. Garfield
Patrick G. McKeown

As an easy-to-use interface that supports sound, video, and graphical displays, the World Wide Web is increasingly employed by organizations of all sizes for electronic marketing and advertising, customer service, and ordering centers. This growing commercial use introduces new opportunities as well as new security risks. Many security concerns stem from flexible design techniques used to build the Internet, some of which make it difficult to identify exactly where data and requests are coming from or where outgoing data will travel.

Hackers break into computers daily to sabotage or explore mission-critical data. Formulating a plan to thwart these curious onlookers and potential computer villains is no easy task, because there are many ways unwanted intruders can attempt to gain access to a corporate computer and a range of measures available to help secure that environment.

Given the loosely controlled Internet infrastructure, the best way an organization can protect its Web environment is to provide security at the front door. Before an organization can do so, information systems (IS) managers must first ask two questions:

1. What is the organization trying to secure?
2. What price is the organization willing to pay for this level of security?

The answers to these questions provide the basis on which to formulate a security policy. This chapter presents a framework to help IS managers assess the broad range of issues involved in the creation of an Internet security plan. It does not provide the technical details needed to deploy security measures but rather a road map of the options that should be considered.

0-8493-1161-6/02/$0.00+$1.50

Exhibit 1. Internet Access Options

	Enterprise Network Connectivity	
Type of Connection	**Yes**	**No**
Direct	Full Direct Connection	Standalone Direct Connection
Indirect (through third party)	Full Buffered Connection	Standalone Buffered Connection

CONNECTING TO THE WORLD WIDE WEB

The method an organization chooses to connect to the Web plays a major role in the level of functionality it obtains and the level of risk it faces. Exhibit 1 depicts the most common ways companies gain access to the Web, each of which is associated with different degrees of flexibility, costs, and security risk.

Full Direct Connection

A full direct connection means that an organization has its own Web server directly connected to the Internet and to its enterprise network. This connection method has the greatest flexibility, the highest security risks, and potentially the highest start-up costs. It gives employees full access to the Web and the enterprise direct control over the Web site.

The actual hardware and software costs to set up a simple Web server are not high. All that is needed is a machine that can run as a server — which can be a Windows-based PC, a Macintosh workstation, or a minicomputer — plus server software. This software is typically easy to use and understand. The higher costs associated with a full direct connection result from the organization's need to protect the internal network from intruders. Securing a Web server from potential hackers requires a fairly high level of technical knowledge, because hackers are constantly improving their techniques.

Full Buffered Connection

A full buffered connection means that an organization has a Web server connected to the Internet through a third party and directly connected to the enterprise network. This type of connection is comparable to the full connection in terms of security risks but, depending on how the third-party vendor designs the Internet connection, may provide less flexibility. Although the third-party vendor may also set up most of the necessary security components, many companies believe that further security is necessary. Under this configuration, the organization must still purchase and maintain the server hardware and software.

Exhibit 2. Degree of Flexibility, Costs, and Security Risk of Internet Connection Options

	Degree		
Option	**Flexibility**	**Costs**	**Security Risk**
Full Direct Connection	High	High	High
Full Buffered Connection	Medium	Medium	High
Standalone Direct Connections	Medium	High	Low
Standalone Buffered Connections	Medium	Medium	Low

Standalone Connections

Standalone direct connections and standalone buffered connections differ from full direct connections and full buffered connections because the Internet connection is not directly tied to the enterprise network. Would-be hackers therefore cannot gain access to the company's network. Likewise, employees may not have a direct Internet connection. This option is the most secure but usually the least flexible.

Many companies are implementing standalone buffered connections, in which Internet access not linked to the enterprise network is provided by a third-party, through outsourcing. When a company outsources its Web needs, it subcontracts with another company that specializes in creating and maintaining commercial Web pages. The costs associated with this popular option vary significantly. Organizations must weigh the benefit of increased security against the disadvantages of not having direct access to the Internet. Exhibit 2 summarizes the degrees of flexibility, costs, and security risk associated with each of the four connection options.

SECURING THE NETWORK ENVIRONMENT

Securing a corporate network environment is similar to building a house. No number of amenities can make up for the lack of a well-thought-out design plan and a solid foundation. Without these, the house will always be flawed.

Security policies must also begin with a solid foundation in the form of virus protection and password integrity established before an Internet connection is obtained. Once the foundation has been laid, IS and security managers can build strong and secure protection for a corporate network by moving through five levels of security:

1. Patching and prevention of security holes
2. Encryption and authentication
3. Firewalls
4. Secure interfaces
5. Legal issues

The following sections review these levels and the options available within each.

PATCHING AND PREVENTING SECURITY HOLES

If virus protection and password integrity form the foundation of a secure environment, the patching of known security holes marks the beginning of a supporting frame. Many of these holes result from the fact that the Internet, and many of the protocols associated with it, were not designed to provide a high level of security.

One known security hole results from the UNIX operating system, which was designed by computer engineers to make their work easier to manage. The UNIX OS lets an approved user log in from anywhere at any time to administer the system. By gaining access to the root, system administrators can manipulate all files that reside on the UNIX workstation and from there enter a corporate network. Unfortunately, unauthorized users who know how to exploit these features can do the same thing. Fortunately, much of the server software and many of the operating systems can be altered to greatly improve security.

Although a knowledgeable systems administrator can patch many of the holes in the security armor of a company server or network, others are not so easily fixed and still others are as yet unknown. As a result, one of the best ways to protect mission-critical information is to move it onto other servers or networks that are not connected to the Internet.

Yet some critical information usually needs to be available on the portion of the corporate network accessible to the Internet. Several steps can be taken to improve the security of this information.

Identifying Security Holes

One way to begin to detect holes in the corporate server or network is to run a program designed to identify potential security risks. Many of these programs are controversial because they are also used by hackers. Yet it is precisely for this reason that organizations must use the programs, two of which are SATAN (Security Administrator Tool for Analyzing Networks) and Internet Scanner.

Other steps a network administrator may take include turning off unneeded UNIX functions that provide security holes and changing the default passwords. Web servers can also be set up in unprivileged mode, and the root directory should not be accessible. Sending NFS files outside the internal network should be prohibited, and sendmail and mail aliases should be restricted. If FTP services are necessary, then the network administrator should restrict writable access to FTP's home directory.

Files in the anonymous FTP should also not be writable or ownable. Restricting remote log-ins and hiding domain name services also helps secure the corporate environment.

Monitoring Hacker Activity

Once known holes are patched, network administrators need to stay on top of who may be trying to break into their computers and as well as at other Internet sites. Several mailing lists, such those run by the Computer Emergency Response Team Computer Emergency Response Team provide updates of security violations. The alert mailing list, for example, can be subscribed to with an e-mail message to request-alert@iss.net that contains the message subscribe alert. Such information is also available from Web sites.

Because only about 5 percent of all intrusions are detected and only 5 percent of these are reported, staying on top of who is trying to break into a corporate computer also requires that server logs be monitored for unusual activities. For instance, one of the new ways for hackers to break into Web sites is to put rogue code onto a Web server by overrunning a software buffer. This gives an intruder unauthorized access to the account under which the HTTP process was running. When oversights such as this are found in the software, the Web server needs to be quickly patched. Copycat hackers are only too ready to exploit the system flaws found and advertised by other hackers.

ENCRYPTION SOFTWARE AND AUTHENTICATION

Once security holes are identified and patched, IS managers should consider encryption software and authentication. Encryption programs let users encrypt their communications so that they cannot be as easily read by unauthorized parties. Using such software can be likened to locking the doors to a house or sealing an envelope. Encryption programs apply cryptographic algorithms to break down ordinary communication messages (i.e., e-mail) into unique codes that can be unlocked only by individuals who possess the unencryption key.

Encryption

Public-Key Encryption. Public-key encryption is the most popular form of encryption, largely because of the program Pretty Good Privacy (PGP). PGP, which was created by Philip Zimmermann and uses Rivest_Shamir-Adleman algorithms to encrypt messages, is freely available on various Internet sites.

The basic premise of public-key encryption is that each user creates two unique keys, one that the user keeps and a public key that the user gives to others. The user then obtains the public keys of the desired recipients of a message and uses them to encrypt a file that only the receivers can unencrypt. Most users also sign their files with a unique signature (i.e., a block of characters) that receivers can verify by applying the sender's public key to the message.

Private-Key Encryption. Private-key encryption is less popular but considered to be robust. The main advantage of this form of encryption is that it lets users exchange their keys more securely than can public-key techniques. The most popular private-key encryption software is MIT's Kerberos.

Hardware-Embedded Techniques. Some companies are moving toward encryption techniques embedded in hardware. PCMCIA cards can be manufactured with the capability to provide secrecy and authentication for the user. This technology is still in its early stages, so its usability and acceptance are uncertain.

Authentication

Various techniques, some of which are cost free and others that are encryption based, are available to verify the identity of a sender and the authenticity of a message. Authentication becomes increasingly important for ensuring that individuals ordering products over the Web are who they claim to be. Authentication methods include:

- Stipulating that a sender sign a message by citing something only the receiver and the sender would know (e.g., a discussion the sender and the recipient had the day before, a pet name, a favorite color). Obviously, this method works only when the sender and the receiver know one another.
- Using a three-way hand shake (i.e., sending a first message, having the receiver send a reply, and finally sending the actual communication).
- Using a program that creates a unique digital signature for the user. Many encryption techniques have the capability to create such signatures.
- Embedding a time stamp into an e-mail document. This method is primarily used primarily to verify when a document was mailed for legal suits and contract issues.

FIREWALLS

Firewalls are the dominant technology used to protect corporate networks from hackers. A firewall is a piece of software that lies between a company's internal network and the Internet and forms a barrier to prevent hackers from gaining access. Drawing from the analogy of home design, the designer needs to decide where to put windows and reinforced doors in the walls of a house. If a company creates a firewall without any windows, people inside the company cannot see out into the Internet and use many of its services. Thus firewall planning involves a tradeoff between user flexibility and the level of security provided for the internal network. Although no firewall is perfect in this attempt, many come close.

Once a corporation decides to put in a firewall, security personnel need to program the firewall to support the organization's security needs. A firewall can be restrictive or flexible depending on the company's goals. For instance, specific services, such as FTP, which is one of the most common ways for a hacker to break into a server, can be limited to reduce the probability of break-ins.

The primary purpose of a firewall is to look at every piece of information that is sent either into or out of the internal network. Firewalls act on a message on the basis of user identification, point of origin, file, or other codes or actions. There are four basic actions a firewall can take when it looks at a piece of information:

1. The packet of information can be dropped entirely.
2. An alert can be issued to the network administrator.
3. A message can be returned to the sender after a failed attempt to send the packet through.
4. The action can just be logged.

Several different types of firewalls protect the internal network at different network layers. The two most common types of firewalls are router-based IP level firewalls and host-based application-level firewalls.

Router-Based IP-Level Firewalls

The router-based firewall focuses on packets — the basic unit of communications within the TCP/IP, the most commonly used protocol for Internet communications. Router-based firewalls control traffic at the IP level going into or coming out of the internal network, blocking or passing along data packets depending on the packet's header. They examine the network application service requested (e.g., FTP), Telnet protocol type (e.g., TCP, UDP, ICMP), and the source and destination address of each packet that arrives at the firewall. The network administrator configures the packet-filtering firewalls to accept or reject packets according to a list of acceptable hosts, routes, or services.

Unfortunately, when a firewall is reading these packets, network performance may slow down by as much as 20 percent. Other drawbacks of router-based firewalls include:

- The firewalls do not allow for granular control of the packets.
- They are cumbersome to code and when set up incorrectly may offer a false sense of security.

- They usually do not log the actions that take place at the firewall, so the network administrator cannot monitor how hackers are attempting to break into the system.

Host-Based Application-Level Firewalls

Host-based application-level firewalls are considered more flexible and more secure than router-based IP-level firewalls. They reside on a host computer, typically a dedicated UNIX machine, PC, or Macintosh and can be configured to support elaborate network access control policies with fine granularity. Application level-firewalls control network application connections (e.g.,Telnet, FTP, SMTP) down to the individual or group level by type of action and time of action permissible. The ability to limit the time when certain functions run is particularly useful, because many renegade hackers, dubbed midnight hackers, work late at night and network administrators need to be able to restrict many of the potentially unsecured Internet functions during those hours.

One of the essential features of the application-level firewall is that it allows the network administrator to monitor a log of activities that take place at the firewall. This log can be used to identify potential breaches of security and to monitor resource usage.

A recent rash of network break-ins has been accomplished by IP-spoofing. IP-spoofing takes advantage of the UNIX OS, which erroneously presumes that anyone who logs in to a server using a previously approved TCP/IP address must be an authorized user. By altering the source IP, someone can spoof the firewall into believing a packet is coming from a trusted source. To combat this problem, many firewalls reject all packets originating from the external network and carrying an internal source IP.

SECURE INTERFACES

The secure interfaces level of security is rather sophisticated, somewhat akin to installing a new form of support beams in a house. Secure interfaces are software programs that allow for additional security checks in the network interface. Several companies offer these interfaces, most of which work with the various Web browsers as well as with Web server software. The most common secure interfaces are Netscape Communications Corp.'s SSL (secure sockets layer) and SHTP (secure hypertext transfer protocol).

SSL

SSL sits between TCP/IP and HTTP or other protocols such as SNMP or FTP. It provides privacy, authentication, and data integrity. MCI is one of the largest SSL users, employing the interface in InternetMCI. Other users include First Data Card Services (the world's largest credit-card authorization firm), First Interstate, Old Kent, Bank of America, Norwest Card Services, as well as MasterCard International.

S-HTTP

S-HTTP extends HTTP to allow both the client and the server to negotiate various levels of security based on public-key encryption and provides encryption, authentication, and digital-signature features. It can also distinguish the origin of a particular document on any server. It was created by Terisa Systems, a joint venture between RSA Data Security and Enterprise Integration Technologies. S-HTTP's strengths include its availability and flexibility.

Both the SSL and S-HTTP have been competing to become the standard secure interface for commercial sites on the Web. To head off the competition, Terisa Systems released a developers' toolkit supporting both standards. Many other secure interfaces also exist, each with its own set of features.

LEGAL ISSUES

Many companies overlook the potential legal issues associated with connecting to the World Wide Web. The press has focused attention on many of these issues, including the availability of child pornography, bootlegged software, and ease of infringement of copyright laws. IS managers should be aware of these potential dangers and take measures to protect employees and enterprises from lawsuits and loss of valuable copyrighted data.

This layer of security is comparable to household plumbing, which allows for unwanted items to be flushed away. For example, if FTP access to the server is allowed, network administrators should consider either prohibiting external users from placing files on the server or frequently purging files off the server. This guards against unwanted guests using the server as a clearing house for pirated software.

One well-publicized case of such an incident occurred at Florida State University, where unknown individuals employed a seldomly used computer as a storage facility for pirated software. It is not implausible that the owners of the server may be found liable for what resided on the computer, regardless of whether they had knowledge about it, and be brought to court on copyright infringement charges.

To curb access to sexually explicit materials, many companies are restricting access to a variety of UseNet groups. Although this practice may cut off the source of some illicit materials, users have other ways of gaining access to such materials. Companies cannot monitor the actions of all employees, but they may be able to reduce the likelihood of access to inappropriate sites by educating employees on what type of behavior will not be tolerated and aggressively enforcing such stances.

Employees also need to be educated on copyright laws. Although it is fairly well known that copying commercial, nonshareware, computer programs is illegal, other forms of copyright infringement are less obvious. Downloading a copy of a favorite song or distributing an article found on the network without permission may violate copyright laws.

Companies need to be concerned not only with what employees obtain but also with what they post outside the company. Employees may unwittingly release strategic information over the Internet, thereby jeopardizing data or potential profits. The only way to guard against such situations is through employee education that also encourages people to contact their IS manager, in-house counsel, or network administrator when they have questions.

CONCLUSION

The field of security and the threats to a corporate network will always be changing. The first step IS managers can take to secure a corporate network is to understand the range of security issues associated with Internet and Web access. The desired level of security must then be determined and security measures implemented.

Security needs to be viewed as a holistic process, because it is only as strong as its weakest link. Remaining aware of new developments in the field and continually adjusting security measures is one way of meeting the changing risks inherent on the Internet.

Some of the more recent yet still uncommon developments include HERF guns (high energy radio frequency guns) and EMPT bombs (electromagnetic pulse transformer bombs). Both of these threats can wipe out an entire data center, and the only way to be protected from them is to put corporate servers and data sources underground and secured in heavy paneling.

By monitoring server logs, staying alert to new security hazards, and altering the security system as needed, companies may be able to deter unwanted guests from visiting the corporate network. Organizations must also have adequate backup plans that speed up recovery from the potentially devastating damages resulting from a successful security breach.

ABOUT THE AUTHORS

Monica J. Garfield is a doctoral student in MIS at the University of Georgia in Athens. As a former technical assistant with the Mitre Corp., she worked on the AWACS radar program.

Patrick G. McKeown is a professor of management at the Terry College of Business at the University of Georgia and the author of *Metamorphosis: A Guide to the World Wide Web and Electronic Commerce* (New York: John Wiley, 1996).

Chapter 50
Internet Security: Securing the Perimeter

Douglas G. Conorich

The corporate community has, in part, created this problem for itself. The rapid growth of the Internet with all the utilities now available to Web surf, combined with the number of users who now have easy access through all the various Internet providers, make every desktop — including those in homes, schools, and libraries — places where an intruder can launch an attack. Surfing the Internet began as a novelty. Users were seduced by the vast amounts of information they could find. In many cases, it has become addictive.

Much of the public concern with the Internet has focused on the inappropriate access to Web sites by children from their homes or schools. A business is concerned with the bottom line. How profitable a business is can be directly related to the productivity of its employees. Inappropriate use of the Internet in the business world can decrease that productivity in many ways. The network bandwidth — how much data can flow across a network segment at any time — is costly to increase because of the time involved and the technology issues. Inappropriate use of the Internet can slow the flow of data and create the network approximation of a log jam.

There are also potential legal and public relations implications of inappropriate employee usage. One such issue is the increasing prevalence of "sin surfing" — browsing the pornographic Web sites. One company reported that 37 percent of its Internet bandwidth was taken up by "sin surfing." Lawsuits can be generated and, more importantly, the organization's image can be damaged by employees using the Internet to distribute inappropriate materials. To legally curtail the inappropriate use of the Internet, an organization must have a policy that defines what is acceptable, what is not, and what can happen if an employee is caught.

0-8493-1161-6/02/$0.00+$1.50

As part of the price of doing business, companies continue to span the bridge between the Internet and their own intranets with mission-critical applications. This makes them more vulnerable to new and unanticipated security threats. Such exposures can place organizations at risk at every level — down to the very credibility upon which they build their reputations.

Making the Internet safe and secure for business requires careful management by the organization. Companies will have to use existing and new, emerging technologies, security policies tailored to the business needs of the organization, and training of the employees in order to accomplish this goal. IBM has defined four phases of Internet adoption by companies as they do business on the Internet: access, presence, integration, and E-business. Each of these phases has risks involved.

- *Access.* In this first phase of adoption, a company has just begun to explore the Internet and learn about its potential benefits. A few employees are using modems connected to their desktop PCs, to dial into either a local Internet service provider, or a national service such as America Online. In this phase, the company is using the Internet as a resource for getting information only; all requests for access are in the outbound direction, and all information flow is in the inbound direction. Exchanging electronic mail and browsing the Web make up the majority of activities in this phase.
- *Presence.* In this phase, the company has begun to make use of the Internet not only as a resource for getting information, but also as a means of providing information to others. Direct connection of the company's internal network means that now all employees have the ability to access the Internet (although this may be restricted by policy), allowing them to use it as an information resource, and also enabling processes such as customer support via e-mail. The creation of a Web server, either by the company's own staff or through a content hosting service, allows the company to provide static information such as product catalogs and data sheets, company background information, software updates, etc. to its customers and prospects.
- *Integration.* In this phase, the company has begun to integrate the Internet into its day-to-day business processes by connecting its Web server directly (through a firewall or other protection system) to its back-office systems. In the previous phase, updates to the Web server's data were made manually, via tape or other means. In this phase, the Web server can obtain information on demand, as users request it. To use banking as an example, this phase enables the bank's customers to obtain their account balances, find out when checks cleared, and other information retrieval functions.

- *E-business.* In the final phase, the company has enabled bidirectional access requests and information flow. This means that not only can customers on the Internet retrieve information from the company's back-office systems, but they can also add to or change information stored on those systems. At this stage, the company is conducting business electronically; customers can place orders, transfer money (via credit cards or other means), check on shipments, etc.; business partners can update inventories, make notes in customer records, etc. In short, the entire company has become accessible via the Internet.

While a company can follow this road to the end, as described by IBM, they are most likely somewhere on it — either in one of the phases or in transition between them.

INTERNET PROTOCOLS

Communication between two people is made possible by their mutual agreement to a common mode of transferring ideas from one person to the other. Each person must know exactly how to communicate with the other if this is to be successful. The communication can be in the form of a verbal or written language, such as English, Spanish, or German. It can also take the form of physical gestures like sign language. It can even be done through pictures or music. Regardless of the form of the communication, it is paramount that the meaning of an element, say a word, has the same meaning to both parties involved. The medium used for communication is also important. Both parties must have access to the same communication medium. One cannot talk to someone else via telephone if only one person has a telephone.

With computers, communications over networks are made possible by what are known as protocols. A protocol is a well-defined message format. The message format defines what each position in the message means. One possible message format could define the first four bits as the version number, the next four bits as the length of the header, and then eight bits for the service being used. As long as both computers agree on this format, communication can take place.

Network communications use more than one protocol. Sets of protocols used together are known as protocol suites or layered protocols. One well-known protocol suite is the Transport Control Protocol/ Internet Protocol (TCP/IP) suite. It is based on the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model (see Exhibit 1).

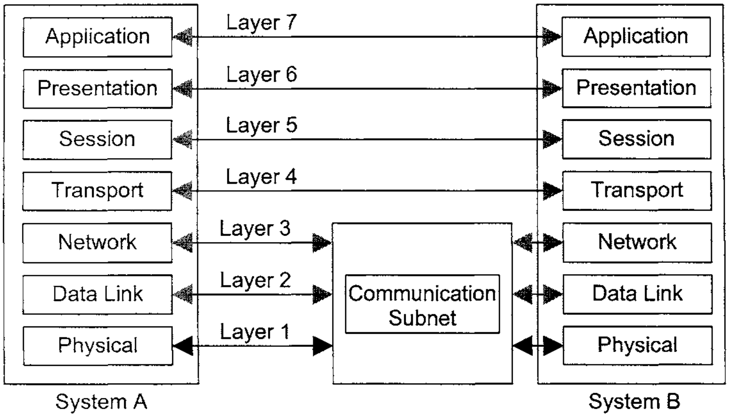
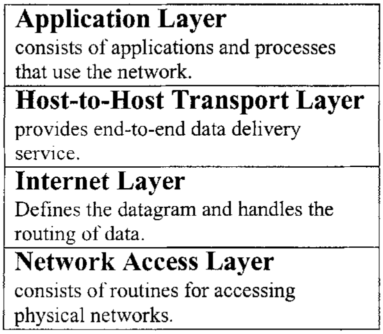


Exhibit 1. The ISO Model

The ISO Reference Model is divided into seven layers:

1. The Physical Layer is the lowest layer in the protocol stack. It consists of the "physical" connection. This may be copper wire or fiber optic cables and the associated connection hardware. The sole responsibility of the Physical Layer is to transfer the bits from one location to another.
2. The second layer is the Data Link Layer. It provides for the reliable delivery of data across the physical link. The Data Link Layer creates a checksum of the message that can be used by the receiving host to ensure that the entire message was received.
3. The Network Layer manages the connections across the network for the upper four layers and isolates them from the details of addressing and delivery of data.
4. The Transport Layer provides the end-to-end error detection and correction function between communicating applications.
5. The Session Layer manages the sessions between communicating applications.
6. The Preparation Layer standardizes the data presentation to the application level.
7. The Application Layer consists of application programs that communicate across the network. This is the layer with which most users interact.

Network devices can provide different levels of security, depending on how far up the stack they can read. Repeaters are used to connect two Ethernet segments. The repeater simply copies the electrical transmission and sends it on to the next segment of the network. Because the repeater



Application Layer
consists of applications and processes that use the network.

Host-to-Host Transport Layer
provides end-to-end data delivery service.

Internet Layer
Defines the datagram and handles the routing of data.

Network Access Layer
consists of routines for accessing physical networks.

Exhibit 2. The TCP/IP Protocol Architecture

only reads up through the Data Link Layer, no security can be added by its use.

The bridge is a computer that is used to connect two or more networks. The bridge differs from the repeater in that it can store and forward entire packets, instead of just repeating electrical signals. Because it reads up through the Network Layer of the packet, the bridge can add some security. It could allow the transfer of only packets with local addresses. A bridge uses physical addresses — not IP addresses. The physical address, also know as the Ethernet address, is the actual address of the Ethernet hardware. It is a 48-bit number.

Routers and gateways are computers that determine which of the many possible paths a packet will take to get to the destination device. These devices read up through the Transport Layer and can read IP addresses, including port numbers. They can be programmed to allow, disallow, and reroute IP datagrams determined by the IP address of the packet.

As previously mentioned, TCP/IP is based on the ISO model, but it groups the seven layers of the ISO model into four layers, as displayed in Exhibit 2.

The Network Access Layer is the lowest layer of the TCP/IP protocol stack. It provides the means of delivery and has to understand how the network transmits data from one IP address to another. The Network Access Layer basically provides the functionality of the first three layers of the ISO model.

TCP/IP provides a scheme of IP addressing that uniquely defines every host connected to the Internet. The Network Access Layer provides the functions that encapsulate the datagrams and maps the IP addresses to the physical addresses used by the network.

The Internet Layer has at its core the Internet Protocol (RFC791). IP provides the basic building blocks of the Internet. It provides:

- The datagram definition scheme
- The Internet addressing scheme
- The means of moving data between the Network Access Layer and the Host-to-Host Layer
- The means for datagrams to be routed to remote hosts
- The function of breaking apart and reassembling packets for transmission

IP is a connectionless protocol. This means that it relies on other protocols within the TCP/IP stack to provide the connection-oriented services. The connection-oriented services (i.e., TCP) take care of the handshake — the exchange of control information. The IP Layer contains the Interrnet Control Message Protocol (ICMP).

The Host-to-Host Transport Layer houses two protocols: the Transport Control Protocol (TCP) and the User Datagram Protocol (UDP). Its primary function is to deliver messages between the Aplication Layer and the Internet Layer. TCP is a reliable protocol. This means that it guarantees that the message will arrive as sent. It contains error detection and correction features. UDP does not have these features and is, therefore, unreliable. For shorter messages, where it is easier to resend the message than worry about the overhead involved with TCP, UDP is used.

The Application Layer contains the various services that users will use to send data. The Application Layer contains such user programs as the Network Terminal Protocol (Telnet), File Transfer Protocol (FTP), and Simple Mail Transport Protocol (SMTP). It also contains protocols not directly used by users, but required for system use — for example, Domain Name Service (DNS), Routing Information Protocol (RIP), and Network File System (NFS).

ATTACKS

As previously noted, TCP is a reliable messaging protocol. This means that TCP is a connection-oriented protocol. TCP uses what is known as a three-way handshake. A handshake is simply the exchange of control information between the two computers. This information enables the computers to determine which packets go where and ensure that all the information in the message has been received.

When a connection is desired between two systems, Host A and Host B, using TCP/IP, a three-way handshake must occur. The initiating host, Host A (the client), sends the receiving host, Host B (the server), a message with the SYN (synchronize sequence number) bit set. The SYN contains information needed by Host B to set up the connection. This message contains the IP address of the both Host A and Host B and the port numbers they will talk on. The SYN tells Host B what sequence number the client will start with, seq = x. This number is important to keep all the data transmitted in the proper order and can be used to notify Host B that a piece of data is missing. The sequence number is found starting at bit 32 to 63 of the header.

When Host B receives the SYN, it sends the client an ACK (acknowledgment message). This message contains the sequence number that Host B will start with, SYN, seq = y, and the sequence number of Host A incremented, the ACK, x + 1. The acknowledgment number is bits 64 through 95 of the header.

The three-way handshake is completed when Host A receives the ACK from Host B and sends an ACK, y + 1, in return. Now data can flow back and forth between the two hosts. This connection is now known as a socket. A socket is usually identified as Host_A_IP:Port_Number, Host_B_IP:Port_Number.

There are two attacks that use this technology: SYN Flood and Sequence Predictability.

SYN Flood Attack

The SYN Flood attack uses a TCP connection request (SYN). The SYN is sent to the target computer with the source IP address in the packet "spoofed," or replaced with an address that is not in use on the Internet or that belongs to another computer. When the target computer receives the connection request, it allocates resources to handle and track the new connection. A SYN_RECEIVED state is stored in a buffer register awaiting the return response (ACK) from the initating computer, which would complete the three-way handshake. It then sends out an SYN-ACK. If the response is sent to the "spoofed," nonexistent IP address, there will never be a response. If the SYN-ACK is sent to a real computer, it checks to see if it has a SYN in the buffer to that IP address. Since it does not, it ignores the request. The target computer retransmits the SYN-ACK a number of times. After a finite amount of wait time, the original SYN request is purged from the buffer of the target computer. This condition is known as a half-open socket.

As an example, the default configuration for a Windows NT 3.5x or 4.0 computer is to retransmit the SYN-ACK five times, doubling the time-out value after each retransmission. The initial time-out value is three seconds, so retries are attempted at 3, 6, 12, 24, and 48 seconds. After the last retransmission, 96 seconds are allowed to pass before the computer gives up on receiving a response and deallocates the resources that were set aside earlier for the connection. The total elapsed time that resources are in use is 189 seconds.

An attacker will send many of these TCP SYNs to tie up as many resources as possible on the target computer. Since the buffer size for the storage of SYNs is a finite size, numerous attempts can cause a buffer overflow. The effect of tying up connection resources varies, depending on the TCP/IP stack and applications listening on the TCP port. For most stacks, there is a limit on the number of connections that can be in the half-open SYN_RECEIVED state. Once the limit is reached for a given TCP port, the target computer responds with a reset to all further connection requests until resources are freed. Using this method, an attacker can cause a denial-of-services on several ports.

Finding the source of a SYN Flood attack can be very difficult. A network analyzer can be used to try to track the problem down, and it may be necessary to contact the Internet Service Provider for assistance in attempting to trace the source. Firewalls should be set up to reject packets from the external network with any IP address from the internal network.

Sequence Predictability

The ability to guess sequence numbers is very useful to intruders because they can create a short-lived connection to a host without having to see the reply packets. This ability, taken in combination with the fact that many hosts have trust relationships that use IP addresses as authentication; that packets are easily spoofed; and that individuals can mount denial of service attacks, means one can impersonate the trusted systems to break into such machines without using source routing.

If an intruder wants to spoof a connection between two computers so that the connection seems as if it is coming from B to A, using your computer C, it works like this:

1. First, the intruder uses computer C to mount a SYN Flood attack on the ports on computer B where the impersonating will take place.
2. Then, computer C sends a normal SYN to a port on A.
3. Computer A returns a SYN-ACK to computer C containing computer A's current Initial Sequence Number (ISN).
4. Computer A internally increments the ISN. This incrementation is done differently in different operating systems (OSs). Operating systems such as BSD, HPUX, Irix, SunOS (not Solaris), and others usually increment by $FA00 for each connection and double each second.

   With this information, the intruder can now guess the ISN that computer A will pick for the next connection. Now comes the spoof.

5. Computer C sends a SYN to computer A using the source IP spoofed as computer B.
6. Computer A sends a SYN-ACK back to computer B, containing the ISN. The intruder on computer C does not see this, but the intruder has guessed the ISN.
7. At this point, computer B would respond to computer A with an RST. This occurs because computer B does not have a SYN_RECEIVED from computer A. Since the intruder used a SYN Flood attack on computer B, it will not respond.
8. The intruder on computer C sends an ACK to computer A, using the source IP spoofed as computer B, containing the guessed ISN+1.

   If the guess was correct, computer A now thinks there has been a successful three-way handshake and the TCP connection between computer A and computer B is fully set up. Now the spoof is complete. The intruder on computer C can do anything, but blindly.
9. Computer C sends `echo + + >>/.rhosts` to port 514 on computer A.
10. If root on computer A had computer B in its /.rhosts file, the intruder has root.
11. Computer C now sends a FIN to computer A.
12. Computer C could be brutal and send an RST to computer A just to clean up things.
13. Computer C could also send an RST to the synflooded port on B, leaving no traces.

To prevent such attacks, one should NEVER trust anything from the Internet. Routers and firewalls should filter out any packets that are coming from the external (sometimes known as the red) side of the firewall that has an IP address of a computer on the internal (sometimes known as the blue) side. This only stops Internet trust exploits; it will not stop spoofs that build on intranet trusts. Companies should avoid using rhosts files wherever possible.

ICMP

A major component of the TCP/IP Internet Layer is the Internet Control Message Protocol (ICMP). ICMP is used for flow control, detecting unreachable destinations, redirection routes, and checking remote hosts. Most users are interested in the last of these functions. Checking a remote host is accomplished by sending an ICMP Echo Message. The PING command is used to send these messages.

When a system receives one of these ICMP Echo Messages, it places the message in a buffer, then re-transmits the message from the buffer back to the source. Due to the buffer size, the ICMP Echo Message size cannot exceed 64K. UNIX hosts, by default, will send an ICMP Echo Message that is 64 bytes long. They will not allow a message of over 64K. With the advent of Microsoft Windows NT, longer messages can be sent. The Windows NT hosts do not place an upper limit on these messages. Intruders have been sending messages of 1MB and larger. When these messages are received, they cause a buffer overflow on the target host. Different operating systems will react differently to this buffer overflow. The reactions range from rebooting to a total system crash.

FIREWALLS

The first line of defense between the Internet and an intranet should be a firewall. A firewall is a multi-homed host that is placed in the Internet route, such that it stops and can make decisions about each packet that wants to get through. A firewall performs a different function from a router. A router can be used to filter out certain packets that meet a secific criteria (i.e., an IP address). A router processes the packets up through the IP Layer. A firewall stops all packets. All packets are processed up through the Application Layer. Routers cannot perform all the functions of a firewall. A firewall should meet, at least, the following criteria:

- In order for an internal or external host to connect to the other network, it must log in on the firewall host.
- All electronic mail is sent to the firewall, which in turn distributes it.
- Firewalls should not mount file systems via NFS, nor should any of its file systems be mounted.
- Firewalls should not run NIS (Network Information Systems).
- Only required users should have accounts on the firewall host.
- The firewall host should not be trusted, nor trust any other host.
- The firewall host is the only machine with anonymous FTP.
- Only the minimum service should be enabled on the firewall in the file `inetd.conf.`
- All system logs on the firewall should log to a separate host.
- Compilers and loaders should be deleted on the firewall.
- System directories permissions on the firewall host should be 711 or 511.

THE DMZ

Most companies today are finding that it is imperative to have an Internet presence. This Internet presence takes on the form of anonymous FTP sites and a World Wide Web (WWW) site. In addition to these, companies are setting up hosts to act as a proxy server for Internet mail and a Domain Name Server (DNS). The host that sponsors these functions cannot be on the inside of the firewall. Therefore, companies are creating what has become known as the DeMilitarized Zone (DMZ) or Perimeter Network, a segment between the router that connects to the Internet and the firewall.

Proxy Servers

A proxy host is a dual-homed host that is dedicated to a particular service or set of services, such as mail. All external requests to that service directed toward the internal network are routed to the proxy. The proxy host then evaluates the request and either passes the request on to the internal service server or discards it. The reverse is also true. Internal requests are passed to the proxy from the service server before they are passed on to the Internet.

One of the functions of the proxy hosts is to protect the company from advertising its internal network scheme. Most proxy software packages contain Network Address Translation (NAT). Take, for example, a mail server. The mail from Albert_Smith@starwars.abc.com would be translated to smith@proxy.abc.com as it went out to the Internet. Mail sent to smith@proxy.abc.com would be sent to the mail proxy. Here it would be readdressed to Albert_Smith@starwars.abc.com and sent to the internal mail server for final delivery.

TESTING THE PERIMETER

A company cannot use the Internet without taking risks. It is important to recognize these risks and it is important not to exaggerate them. One cannot cross the street without taking a risk. But by recognizing the dangers, and taking the proper precautions (such as looking both ways before stepping off the curb), millions of people cross the street safely every day.

The Internet and intranets are in a state of constant change — new protocols, new applications, and new technologies — and a company's security practices must be able to adapt to these changes. To adapt, the security process should be viewed as forming a circle. The first step is to assess the current state of security within one's intranet and along the perimeter. Once one understands where one is, then one can deploy a security solution. If one does not monitor that solution by enabling some detection and devising a response plan, the solution is useless. It would be like putting an alarm on a car, but never checking it when the alarm goes off. As the solution is monitored and tested, there will be further weaknesses — which brings us back to the assessment stage and the process is repeated. Those new weaknesses are then learned about and dealt with, and a third round begins. This continuous improvement ensures that corporate assets are always protected.

As part of this process, a company must perform some sort of vulnerability checking on a regular basis. This can be done by the company, or it may choose to have an independent group do the testing. The company's security policy should state how the firewall and the other hosts in the

DMZ are to be configured. These configurations need to be validated and then periodically checked to ensure that the configurations have not changed. The vulnerability test may find additional weaknesses with the configurations and then the policy needs to be changed.

Security is achieved through the combination of technology and policy. The technology must be kept up to date and the policy must outline the procedures. An important part of a good security policy is to ensure that there are as few information leaks as possible.

One source of information can be DNS records. There are two basic DNS services: lookups and zone transfers. Lookup activities are used to resolve IP addresses into host names or to do the reverse. A zone transfer happens when one DNS server (a secondary server) asks another DNS server (the primary server) for all the information that it knows about a particular part of the DNS tree (a zone). These zone transfers only happen between DNS servers that are supposed to be providing the same information. Users can also request a zone transfer.

A zone transfer is accomplished using the `nslookup` command in interactive mode. The zone transfer can be used to check for information leaks. This procedure can show hosts, their IP addresses, and operating systems. A good security policy is to disallow zone transfers on external DNS servers. This information can be used by an intruder to attack or spoof other hosts. If this is not operationally possible, as a general rule, DNS servers outside of the firewall (on the red side) should not list hosts within the firewall (on the blue side). Listing internal hosts only helps intruders gain network mapping information and gives them an idea of the internal IP addressing scheme.

In addition to trying to do a zone transfer, the DNS records should be checked to ensure that they are correct and that they have not changed. Domain Information Gofer (DIG) is a flexible command-line tool that is used to gather information from the Domain Name System servers.

The `PING` command, as previously mentioned, has the ability to determine the status of a remote host using the ICMP ECHO Message. If a host is running and is reachable by the message, the PING program will return an “alive” message. If the host is not reachable and the host name can be resolved by DNS, the program returns a “host not responding” message; otherwise, an “unknown host” message is obtained. An intruder can use the PING program to set up a “war dialer.” This is a program that systematically goes through the IP addresses one after another, looking for “alive” or “not responding” hosts. To prevent intruders from mapping internal networks, the firewall should screen out ICMP messages. This can be done by not allowing ICMP messages to go through to the internal network or go out from the internal network. The former is the preferred method.

This would keep intruders from using ICMP attacks, such as the Ping 'O Death or Loki tunneling.

The TRACEROUTE program is another useful tool one can use to test the corporate perimeter. Because the Internet is a large aggregate of networks and hardware connected by various gateways, TRACEROUTE is used to check the "time-to-live" (ttl) parameter and routes. TRACEROUTE sends a series of three UDP packets with an ICMP packet incorporated during its check. The ttl of each packet is similar. As the ttl expires, it sends the ICMP packet back to the originating host with the IP address of the host where it expired. Each successive broadcast uses a longer ttl. By continuing to send longer ttls, TRACEROUTE pieces together the successive jumps. Checking the various jumps not only shows the routes, but it can show possible problems that may give an intruder information or leads. This information might show a place where an intruder might successfully launch an attack. A "*" return shows that a particular hop has exceeded the three-second timeout. These are hops that could be used by intruders to create DoSs. Duplicate entries for successive hops are indications of bugs in the kernel of that gateway or looping within the routing table.

Checking the open ports and services available is another important aspect of firewall and proxy server testing. There are a number of programs — like the freeware program STROBE, IBM Network Services Auditor (NSA), ISS Internet Scanner™, and AXENT Technologies NetRecon™ — that can perform a selective probe of the target UNIX or Windows NT network communication services, operating systems and key applications. These programs use a comprehensive set of penetration tests. The software searches for weaknesses most often exploited by intruders to gain access to a network, analyzes security risks, and provides a series of highly informative reports and recommended corrective actions.

There have been numerous attacks in the past year that have been directed at specific ports. The teardrop, newtear, oob, and land.c are only a few of the recent attacks. Firewalls and proxy hosts should have only the minimum number of ports open. By default, the following ports are open as shipped by the vendor, and should be closed:

- Echo on TCP port 7
- Echo on UDP port 7
- Discard on TCP port 9
- Daytime on TCP port 13
- Daytime on UDP port 13
- Chargen on TCP port 19
- Chargen on UDP port 19
- NetBIOS-NS on UDP port 137
- NetBIOS-ssn on TCP port 139

Other sources of information leaks include Telnet, FTP, and Sendmail programs. They all, by default, advertise the operating system or service type and version. They also may advertise the host name. This feature can be turned off and a more appropriate warning messages should be put in its place.

Sendmail has a feature that will allow the administrator to expand or verify users. This feature should not be turned on on any host in the DMZ. An intruder would only have to Telnet to the Sendmail port to obtain user account names. There are a number of well-known user accounts that an intruder would test. This method works even if the `finger` command is disabled.

VRFY and EXPN allow an intruder to determine if an account exists on a system and can provide a significant aid to a brute-force attack on user accounts. If you are running Sendmail, add the lines `Opnovrfy` and `Opnoexpn` to your Sendmail configuration file, usually located in /etc/sendmail.cf. With other mail servers, contact the vendor for information on how to disable the `verify` command.

```
# telnet xxx.xxx.xx.xxx
Trying xxx.xxx.xx.xxx...
Connected to xxx.xxx.xx.xxx.
Escape character is '^]'.
220 proxy.abc.com Sendmail 4.1/SMI-4.1 ready at Thu, 26 Feb 98 12:50:05 CST
expn root
250- John Doe <jdoe>
250 Jane User <juser>
vrfy root
250- John Doe <jdoe>
250 Jane User <juser>
vrfy jdoe
250 John Doe <john_doe@mailserver.internal.abc.com>
vrfy juser
250 John User <jane_user@mailserver.internal.abc.com>
^]
```

Another important check that needs to be run on these hosts in the DMZ is a validation that the system and important application files are valid and not hacked. This is done by running a checksum or a cyclic redundancy check (CRC) on the files. Because these values are not stored anywhere on the host, external applications need to be used for this function. Some suggested security products are freeware applications such as COPS and Tripwire, or third-party commercial products like AXENT Technologies Enterprise Security Manager™ (ESM), ISS RealSecure™ or Kane Security Analyst™.

SUMMARY

The assumption must be made that one is not going to be able to stop everyone from getting into a computer. An intruder only has to succeed once. Security practitioners, on the other hand, have to succeed every time. Once one reaches this conclusion, then the only strategy left is to secure the perimeter as best one can while allowing business to continue, and have some means to detect the intrusions as they happen. If one can do this, then one limits what the intruder can do.

ABOUT THE AUTHOR

Douglas G. Conorich is an Internet security analyst with IBM Corporation's Internet Emergency Response Service (ERS). Prior to his tenure at IBM, he was a principal security analyst with AXENT Technologies, Inc. He has more than 27 years of experience in the field of information security, 20 of those years working with the U.S. government as a computer security specialist in the areas of access control and authentication research.

Chapter 51
Security Management for the World Wide Web

Lynda L. McGhie
Phillip Q. Maier

Companies continue to flock to the Internet in ever-increasing numbers, despite the fact that the overall and underlying environment is not secure. To further complicate the matter, vendors, standards bodies, security organizations, and practitioners cannot agree on a standard, compliant, and technically available approach. As a group of investors concerned with the success of the Internet for business purposes, it is critical that we pull our collective resources and work together to quickly establish and support interoperable security standards; open security interfaces to existing security products and security control mechanisms within other program products; and hardware and software solutions within heterogeneous operating systems that will facilitate smooth transitions.

Interfaces and teaming relationships to further this goal include computer and network security and information security professional associations (CSI, ISSA, NCSA), professional technical and engineering organizations (I/EEE, IETF), vendor and product user groups, government and standards bodies, seminars and conferences, training companies/institutes (MIS), and informal networking among practitioners.

Having the tools and solutions available within the marketplace is a beginning, but we also need strategies and migration paths to accommodate and integrate Internet, intranet, and World Wide Web (WWW) technologies into our existing IT infrastructure. While there are always emerging challenges, introduction of newer technologies, and customers with challenging and perplexing problems to solve, this approach should enable us to maximize the effectiveness of our existing security investments, while bridging the gap to the long-awaited and always sought-after perfect solution!

0-8493-1161-6/02/$0.00+$1.50

Security solutions are slowly emerging, but interoperability, universally accepted security standards, application programming interfaces (APIs) for security, vendor support and cooperation, and multiplatform security products are still problematic. Where there are products and solutions, they tend to have niche applicability, be vendor-centric, or only address one of a larger set of security problems and requirements. For the most part, no single vendor or even software/vendor consortium has addressed the overall security problem within open systems and public networks. This indicates that the problem is very large, and that we are years away from solving today's problem, not to mention tomorrow's.

By acknowledging today's challenges, bench-marking today's requirements, and understanding our "as is condition" accordingly, we as security practitioners can best plan for security in the twenty-first century. Added benefits adjacent to this strategy will hopefully include a more cost-effective and seamless integration of security policies, security architectures, security control mechanisms, and security management processes to support this environment.

For most companies, the transition to "open" systems technologies is still in progress and most of us are somewhere in the process of converting mainframe applications and systems to distributed network-centric client-server infrastructures. Nevertheless, we are continually challenged to provide a secure environment today, tomorrow, and in the future, including smooth transitions from one generation to another. This chapter considers a phased integration methodology that, first, focuses on the update of corporate policies and procedures, including most security policies and procedures; second, enhances existing distributed security architectures to accommodate the use of the Internet, intranet, and WWW technologies; third, devises a security implementation plan that incorporates the use of new and emerging security products and techniques; and, finally, addresses security management and infrastructure support requirements to tie it all together.

It is important to keep in mind, as with any new and emerging technology, Internet, intranet, and WWW technologies do not necessarily bring new and unique security concerns, risks, and vulnerabilities, but rather introduce new problems, challenges and approaches within our existing security infrastructure.

Security requirements, goals, and objectives remain the same, while the application of security, control mechanisms, and solution sets are different and require the involvement and cooperation of multidisciplined technical and functional area teams. As in any distributed environment, there are more players, and it is more difficult to find or interpret the overall requirements or even talk to anyone who sees or understands the big picture.

More people are involved than ever before, emphasizing the need to communicate both strategic and tactical security plans broadly and effectively throughout the entire enterprise. The security challenges and the resultant problems become larger and more complex in this environment. Management must be kept up-to-date and thoroughly understand overall risk to the corporations information assets with the implementation or decisions to implement new technologies. They must also understand, fund, and support the influx of resources required to manage the security environment.

As with any new and emerging technology, security should be addressed early in terms of understanding the requirements, participating in the evaluation of products and related technologies, and finally in the engineering, design, and implementation of new applications and systems. Security should also be considered during all phases of the systems development life cycle. This is nothing new, and many of us have learned this lesson painfully over the years as we have tried to retrofit security solutions as an adjunct to the implementation of some large and complex system. Another important point to consider throughout the integration of new technologies, is "technology does not drive or dictate security policies, but the existing and established security policies drive the application of new technologies." This point must be made to management, customers, and supporting IT personnel.

For most of us, the WWW will be one of the most universal and influential trends impacting our internal enterprise and its computing and networking support structure. It will widely influence our decisions to extend our internal business processes out to the Internet and beyond. It will enable us to use the same user interface, the same critical systems and applications, work towards one single original source of data, and continue to address the age-old problem: How can I reach the largest number of users at the lowest cost possible?

THE PATH TO INTERNET/BROWSER TECHNOLOGIES

Everyone is aware of the staggering statistics relative to the burgeoning growth of the Internet over the last decade. The use of the WWW can even top that growth, causing the traffic on the Internet to double every six months. With five internal Web servers being deployed for every one external Web server, the rise of the intranet is also more than just hype. Companies are predominately using the Web technologies on the intranet to share information and documents. Future application possibilities are basically any enterprise-wide application such as education and training; corporate policies and procedures; human resources applications such as a resume, job posting, etc.; and company information. External Web applications include marketing and sales.

For the purpose of this discussion, we can generally think of the Internet in three evolutionary phases. While each succeeding phase has brought with it more utility and the availability of a wealth of electronic and automated resources, each phase has also exponentially increased the risk to our internal networks and computing environments.

Phase I, the early days, is characterized by a limited use of the Internet, due in the most part to its complexity and universal accessibility. The user interface was anything but user friendly, typically limited to the use of complex UNIX-based commands via line mode. Security by obscurity was definitely a popular and acceptable way of addressing security in those early days, as security organizations and MIS management convinced themselves that the potential risks were confined to small user populations centered around homogeneous computing and networking environments. Most companies were not externally connected in those days, and certainly not to the Internet.

Phase II is characterized by the introduction of the first versions of database search engines, including Gopher and Wide Area Information System (WAIS). These tools were mostly used in the government and university environments and were not well known or generally proliferated in the commercial sector.

Phase III brings us up to today's environment, where Internet browsers are relatively inexpensive, readily available, easy to install, easy to use through GUI frontends and interfaces, interoperable across heterogeneous platforms, and ubiquitous in terms of information access.

The growing popularity of the Internet and the introduction of the Internet should not come as a surprise to corporate executives who are generally well read on such issues and tied into major information technology (IT) vendors and consultants. However, quite frequently companies continue to select one of two choices when considering the implementation of WWW and Internet technologies. Some companies, who are more technically astute and competitive, have jumped in totally and are exploiting Internet technologies, electronic commerce, and the use of the Web.

Internet technologies offer great potential for cost savings over existing technologies, representing huge investments over the years in terms of revenue and resources now supporting corporate information infrastructures and contributing to the business imperatives of those enterprises. Internet-based applications provide a standard communications interface and protocol suite ensuring interoperability and access to the organization's heterogeneous data and information resources. Most WWW browsers run on all systems and provide a common user interface and ease of use to a wide range of corporate employees.

Benefits derived from the development of WWW-based applications for internal and external use can be categorized by the cost savings related to deployment, generally requiring very little support or end-user training. The browser software is typically free, bundled in vendor product suites, or very affordable. Access to information, as previously stated, is ubiquitous and fairly straightforward.

Use of internal WWW applications can change the very way organizations interact and share information. When established and maintained properly, an internal WWW application can enable everyone on the internal network to share information resources, update common use applications, receive education and training, and keep in touch with colleagues at the home base, from remote locations, or on the road.

INTERNET/WWW SECURITY OBJECTIVES

As mentioned earlier, security requirements do not change with the introduction and use of these technologies, but the emphasis on where security is placed and how it is implemented does change. The company's Internet, intranet, and WWW security strategies should address the following objectives, in combination or in prioritized sequence, depending on security and access requirements, company philosophy, the relative sensitivity of the companys information resources, and the business imperative for using these technologies.

- Ensure that Internet- and WWW-based application and the resultant access to information resources are protected and that there is a cost-effective and user-friendly way to maintain and manage the underlying security components, over time as new technology evolves and security solutions mature in response.
- Information assets should be protected against unauthorized usage and destruction. Communication paths should be encrypted as well as transmitted information that is broadcast over public networks.
- Receipt of information from external sources should be decrypted and authenticated. Internet- and WWW-based applications, WWW pages, directories, discussion groups, and databases should all be secured using access control mechanisms.
- Security administration and overall support should accommodate a combination of centralized and decentralized management.
- User privileges should be linked to resources, with privileges to those resources managed and distributed through directory services.

- Mail and real-time communications should also be consistently protected. Encryption key management systems should be easy to administer, compliant with existing security architectures, compatible with existing security strategies and tactical plans, and secure to manage and administer.
- New security policies, security architectures, and control mechanisms should evolve to accommodate this new technology, not change in principle or design.

Continue to use risk management methodologies as a baseline for deciding how many of the new Internet, intranet, and WWW technologies to use and how to integrate them into the existing Information Security Distributed Architecture. As always, ensure that the optimum balance between access to information and protection of information is achieved during all phases of the development, integration, implementation, and operational support life cycle.

INTERNET AND WWW SECURITY POLICIES AND PROCEDURES

Having said all of this, it is clear that we need new and different policies, or minimally, an enhancement or refreshing of current policies supporting more traditional means of sharing, accessing, storing, and transmitting information. In general, high-level security philosophies, policies, and procedures should not change. In other words, who is responsible for what (the fundamental purpose of most high-level security policies) does not change. These policies are fundamentally directed at corporate management, process, application and system owners, functional area management, and those tasked with the implementation and support of the overall IT environment. There should be minimal changes to these policies, perhaps only adding the Internet and WWW terminology.

Other high-level corporate policies must also be modified, such as the use of corporate assets, responsibility for sharing and protecting corporate information, etc. The second-level corporate policies, usually more procedure oriented typically addressing more of the "how," should be more closely scrutinized and may change the most when addressing the use of the Internet, intranet, and Web technologies for corporate business purposes. New classifications and categories of information may need to be established and new labeling mechanisms denoting a category of information that cannot be displayed on the Internet or new meanings to "all allow" or "public" data. The term "public," for instance, when used internally, usually means anyone authorized to use internal systems. In most companies, access to internal networks, computing systems, and information is severely restricted and "public" would not mean unauthorized users, and certainly not any user on the Internet.

Candidate lower-level policies and procedures for update to accommodate the Internet and WWW include external connectivity, network security, transmission of data, use of electronic commerce, sourcing and procurement, electronic mail, nonemployee use of corporate information and electronic systems, access to information, appropriate use of electronic systems, use of corporate assets, etc.

New policies and procedures (most likely enhancements to existing policies) highlight the new environment and present an opportunity to dust off and update old policies. Involve a broad group of customers and functional support areas in the update to these policies. The benefits are many. It exposes everyone to the issues surrounding the new technologies, the new security issues and challenges, and gains buy-in through the development and approval process from those who will have to comply when the policies are approved. It is also an excellent way to raise the awareness level and get attention to security up front.

The most successful corporate security policies and procedures address security at three levels, at the management level through high-level policies, at the functional level through security procedures and technical guidelines, and at the end-user level through user awareness and training guidelines. Consider the opportunity to create or update all three when implementing Internet, intranet, and WWW technologies.

Since these new technologies increase the level of risk and vulnerability to your corporate computing and network environment, security policies should probably be beefed up in the areas of audit and monitoring. This is particularly important because security and technical control mechanisms are not mature for the Internet and WWW and therefore more manual processes need to be put in place and mandated to ensure the protection of information.

The distributed nature of Internet, intranet, and WWW and their inherent security issues can be addressed at a more detailed level through an integrated set of policies, procedures, and technical guidelines. Because these policies and processes will be implemented by various functional support areas, there is a great need to obtain buy-in from these groups and ensure coordination and integration through all phases of the system's life cycle. Individual and collective roles and responsibilities should be clearly delineated to include monitoring and enforcement.

Other areas to consider in the policy update include legal liabilities, risk to competition-sensitive information, employees' use of company time while "surfing" the Internet, use of company logos and trade names by employees using the Internet, defamation of character involving company employees, loss of trade secrets, loss of the competitive edge, ethical use of the Internet, etc.

Exhibit 1. Sample Data Protection Classification Hierarchy

	Auth.	Trans. Controls	Encryption	Audit	Ownership
External Public Data				(X)	X
Internal Public Data				(X)	X
Internal Cntl. Data	X	X	(X)	X	X
External Cntl. Data	X	X	X	X	X
Update Applications	X	X		X	X

DATA CLASSIFICATION SCHEME

A data classification scheme is important to both reflect existing categories of data and introduce any new categories of data needed to support the business use of the Internet, electronic commerce, and information sharing through new intranet and WWW technologies. The whole area of nonemployee access to information changes the approach to categorizing and protecting company information.

The sample chart in Exhibit 1 is an example of how general to specific categories of company information can be listed, with their corresponding security and protection requirements to be used as a checklist by application, process, and data owners to ensure the appropriate level of protection, and also as a communication tool to functional area support personnel tasked with resource and information protection. A supplemental chart could include application and system names familiar to corporate employees, or types of general applications and information such as payroll, HR, marketing, manufacturing, etc.

Note that encryption may not be required for the same level of data classification in the mainframe and proprietary networking environment, but in “open” systems and distributed and global networks transmitted data is much more easily compromised. Security should be applied based on a thorough risk assessment considering the value of the information, the risk introduced by the computing and network environment, the technical control mechanisms feasible or available for implementation, and the ease of administration and management support. Be careful to apply the right “balance” of security. Too much is just as costly and ineffective as too little in most cases.

APPROPRIATE USE POLICY

It is important to communicate management expectation for employee use of these new technologies. An effective way to do that is to supplement the corporate policies and procedures with a more user-friendly bulletined list of requirements. The list should be specific, highlight employee expectations and outline what employees can and cannot do on the Internet,

Exhibit 2. Appropriate Use Policy

Examples of Unacceptable use include but not limited to the following:

1. Using Co. equipment, functions or services for non-business related activities while on company time; which in effect is mischarging;
2. Using the equipment or services for financial or commercial gain;
3. Using the equipment or services for any illegal activity;
4. Dial-in usage from home for Internet services for personal gain;
5. Accessing non-business related news groups or BBS;
6. Willful intent to degrade or disrupt equipment, software or system performance;
7. Vandalizing the data or information of another user;
8. Gaining unauthorized access to resources or information;
9. Invading the privacy of individuals;
10. Masquerading as or using an account owned by another user;
11. Posting anonymous messages or mail for malicious intent;
12. Posting another employee's personal communication or mail without the original author's consent; this excludes normal business E-mail forwarding;
13. Downloading, storing, printing or displaying files or messages that are profane, obscene, or that use language or graphics which offends or tends to degrade others;
14. Transmitting company data over the network to non-company employees without following proper release procedures;
15. Loading software obtained from outside the Corporation's standard company's procurement channels onto a company system without proper testing and approval;
16. Initiating or forwarding electronic chain mail.

Examples of Acceptable Use includes but is not limited to the following:

1. Accessing the Internet, computer resources, fax machines and phones for information directly related to your work assignment;
2. Off-hour usage of computer systems for degree related school work where allowed by local site practices;
3. Job related On-Job Training (OJT).

intranet, and WWW. The goal is to communicate with each and every employee, leaving little room for doubt or confusion. An Appropriate Use Policy (Exhibit 2) could achieve these goals and reinforce the higher level. Areas to address include the proper use of employee time, corporate computing and networking resources, and acceptable material to be viewed or downloaded to company resources.

Most companies are concerned with the Telecommunications Act and their liabilities in terms of allowing employees to use the Internet on company time and with company resources. Most find that the trade-off is highly skewed to the benefit of the corporation in support of the utility of the Internet. Guidelines must be carefully spelled out and coordinated with the legal department to ensure that company liabilities are addressed through clear specification of roles and responsibilities. Most companies do not monitor their employees' use of the Internet or the intranet, but find that audit trail information is critical to prosecution and defense for computer crime.

Overall computer security policies and procedures are the baseline for any security architecture and the first thing to do when implementing any new technology. However, you are never really finished as the development and support of security policies is an iterative process and should be revisited on an ongoing basis to ensure that they are up to date, accommodate new technologies, address current risk levels, and reflect the company's use of information and network and computing resources.

There are four basic threats to consider when you begin to use Internet, intranet, and Web technologies:

- Unauthorized alteration of data
- Unauthorized access to the underlying operating system
- Eavesdropping on messages passed between a server and a browser
- Impersonation

Your security strategies should address all four. These threats are common to any technology in terms of protecting information. In the remainder of this chapter, we will build upon the "general good security practices and traditional security management" discussed in the first section and apply these lessons to the technical implementation of security and control mechanisms in the Internet, intranet, and Web environments.

The profile of a computer hacker is changing with the exploitation of Internet and Web technologies. Computerized bulletin board services and network chat groups link computer hackers (formerly characterized as loners and misfits) together. Hacker techniques, programs and utilities, and easy-to-follow instructions are readily available on the net. This enables hackers to more quickly assemble the tools to steal information and break into computers and networks, and it also provides the "would-be" hacker a readily available arsenal of tools.

INTERNAL/EXTERNAL APPLICATIONS

Most companies segment their networks and use firewalls to separate the internal and external networks. Most have also chosen to push their marketing, publications, and services to the public side of the firewall using file servers and web servers. There are benefits and challenges to each of these approaches. It is difficult to keep data synchronized when duplicating applications outside the network. It is also difficult to ensure the security of those applications and the integrity of the information. Outside the firewall is simply *outside*, and therefore also outside the protections of the internal security environment. It is possible to protect that information and the underlying system through the use of new security technologies for authentication and authorization. These techniques are not without trade-offs in terms of cost and ongoing administration, management, and support.

Security goals for external applications that bridge the gap between internal and external, and for internal applications using the Internet, intranet, and WWW technologies should all address these traditional security controls:

- Authentication
- Authorization
- Access control
- Audit
- Security administration

Some of what you already used can be ported to the new environment, and some of the techniques and supporting infrastructure already in place supporting mainframe-based applications can be applied to securing the new technologies.

Using the Internet and other public networks is an attractive option, not only for conducting business-related transactions and electronic commerce, but also for providing remote access for employees, sharing information with business partners and customers, and supplying products and services. However, public networks create added security challenges for IS management and security practitioners, who must devise security systems and solutions to protect company computing, networking, and information resources. Security is a CRITICAL component.

Two watchdog groups are trying to protect on-line businesses and consumers from hackers and fraud. The council of Better Business Bureaus has launched BBBOnline, a service that provides a way to evaluate the legitimacy of online businesses. In addition, the national computer security association, NCSA, launched a certification program for secure WWW sites. Among the qualities that NCSA looks for in its certification process are extensive logging, the use of encryption including those addressed in this chapter, and authentication services.

There are a variety of protection measures that can be implemented to reduce the threats in the Web/server environment, making it more acceptable for business use. Direct server protection measures include secure

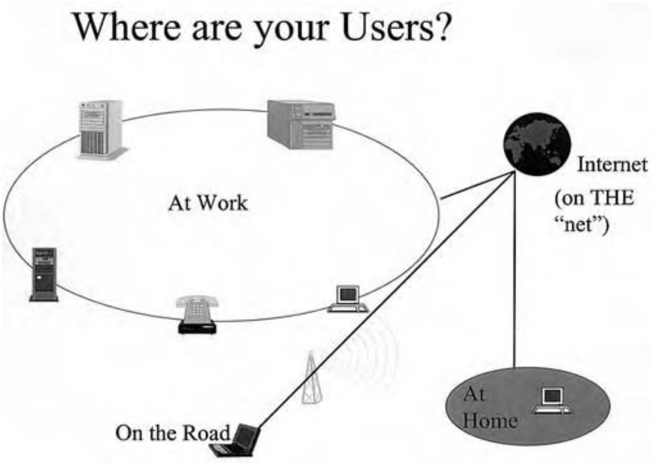


Exhibit 3. Where Are Your Users

Web server products which use differing designs to enhance the security over user access and data transmittal. In addition to enhanced secure Web server products, the Web server network architecture can also be addressed to protect the server and the corporate enterprise which could be placed in a vulnerable position due to served enabled connectivity. Both secure server and secure Web server designs will be addressed, including the application and benefits to using each.

WHERE ARE YOUR USERS?

Discuss how the access point where your users reside contributes to the risk and the security solutions set. Discuss the challenge when users are all over the place and you have to rely on remote security services that are only as good as the users' correct usage. Issues of evolving technologies can also be addressed. Concerns for multiple layering of controls and dissatisfied users with layers of security controls, passwords, hoops, etc. can also be addressed.

WEB BROWSER SECURITY STRATEGIES

Ideally, Web browser security strategies should use a network-based security architecture that integrates your company's external Internet and the internal intranet security policies. Ensure that users on any platform, with any browser, can access any system from any location if they are authorized and have a "need to know." Be careful not to adopt the latest evolving security product from a new vendor or an old vendor capitalizing on a hot marketplace.

Recognizing that the security environment is changing rapidly, and knowing that we do not want to change our security strategy, architecture, and control mechanisms every time a new product or solution emerges, we need to take time and use precautions when devising browser security solutions. It is sometimes a better strategy to stick with the vendors that you have already invested in and negotiate with them to enhance their existing products, or even contract with them to make product changes specific or tailored to accommodate your individual company requirements. Be careful in these negotiations as it is extremely likely that other companies have the very same requirements. User groups can also form a common position and interface to vendors for added clout and pressure.

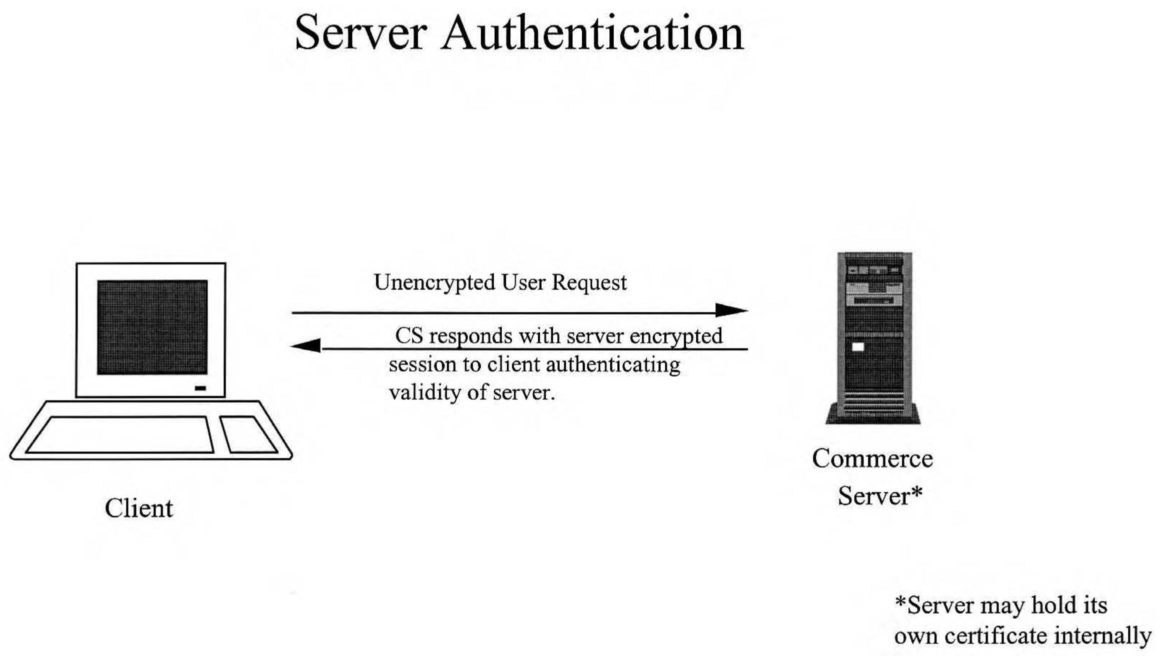
You can basically secure your Web server as much as or as little as you wish with the current available security products and technologies. The tradeoffs are obvious: cost, management, administrative requirements, and time. Solutions can be hardware, software, and personnel intensive.

Enhancing the security of the Web server itself has been a paramount concern since the first Web server initially emerged, but progress has been slow in deployment and implementation. As the market has mushroomed for server use, and the diversity of data types that are being placed on the server has grown, the demand has increased for enhanced Web server security. Various approaches have emerged, with no single *de facto* standard yet emerging (though there are some early leaders — among them Secure Sockets Layer [SSL] and Secure Hypertext Transfer Protocol [S-HTTP]). These are two significantly different approaches, but both widely seen in the marketplace.

Secure Socket Layer (SSL) Trust Model

One of the early entrants into the secure Web server and client arena is Netscape's Commerce Server, which utilizes the Secure Sockets Layer (SSL) trust model. This model is built around the RSA Public Key/Private Key architecture. Under this model, the SSL-enabled server is authenticated to SSL-aware clients, proving its identity at each SSL connection. This proof of identity is conducted through the use of a public/private key pair issued to the server validated with x.509 digital certificates. Under the SSL architecture, Web server validation can be the only validation performed, which may be all that is needed in some circumstances. This would be applicable for those applications where it is important to the user to be assured of the identity of the target server, such as when placing company orders, or other information submittal where the client is expecting some important action to take place. Exhibit 4 diagrams this process.

Optionally, SSL sessions can be established that also authenticate the client and encrypt the data transmission between the client and the server



Server Authentication

Unencrypted User Request

CS responds with server encrypted session to client authenticating validity of server.

Client

Commerce Server*

*Server may hold its own certificate internally

Exhibit 4. Server Authentication

for multiple I/P services (HTTP, Telnet, FTP). The multiservice encryption capability is available because SSL operates below the application layer and above the TCP/IP connection layer in the protocol stack, and thus other TCP/IP services can operate on top of a SSL-secured session.

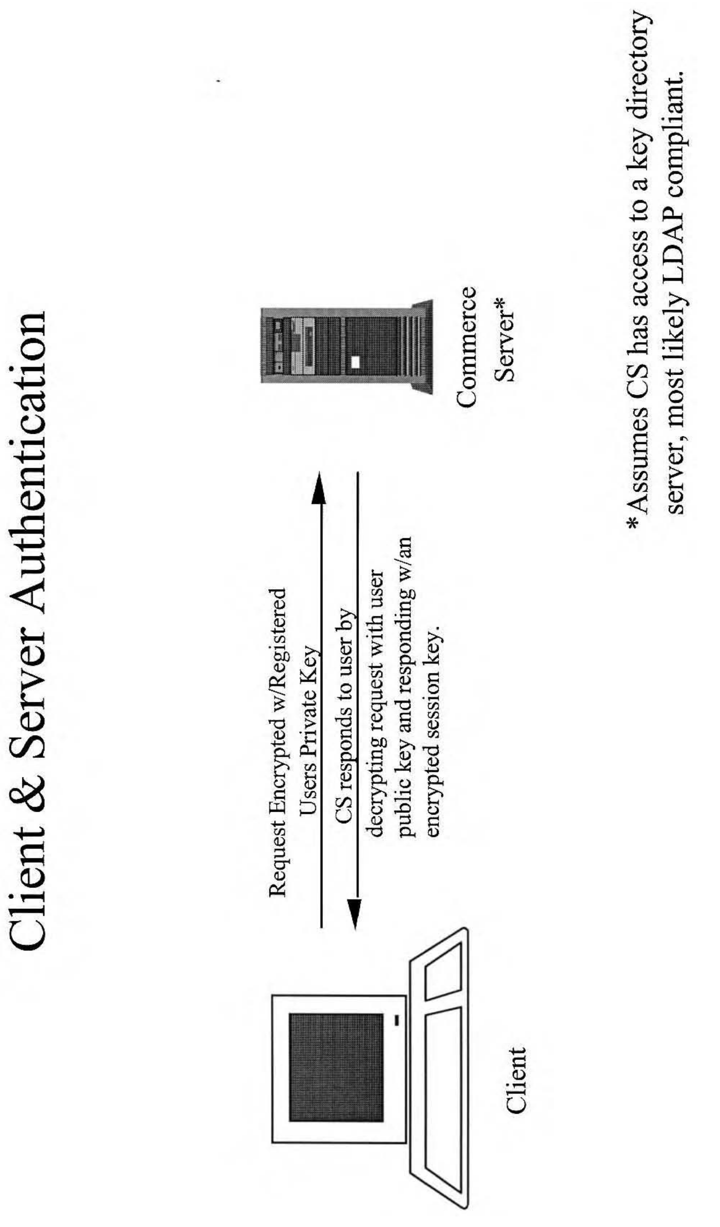
Optionally, authentication of a SSL client is available when the client is registered with the SSL server, and occurs after the SSL-aware client connects and authenticates the SSL server. The SSL client then submits its digital certificate to the SSL server, where the SSL server validates the clients certificate and proceeds to exchange a session key to provide encrypted transmissions between the client and the server. Exhibit 5 provides a graphical representation of this process for mutual client/server authentication under the SSL architecture. This type of mutual client/server authentication process should be considered when the data being submitted by the client are sensitive enough to warrant encryption prior to being submitted over a network transmission path.

Though there are some "costs" with implementing this architecture, these cost variables must be considered when proposing a SSL server implementation to enhance your Web server security. First of all, the design needs to consider whether to provide only server authentication, or both server and client authentication. The issue when expanding the authentication to include client authentication includes the administrative overhead of managing the user keys, including a key revocation function. This consideration, of course, has to assess the size of the user base, potential for growth of your user base, and stability of your proposed user community. All of these factors will impact the administrative burden of key management, especially if there is the potential for a highly unstable or transient user community.

The positive considerations for implementing a SSL-secured server is the added ability to secure other I/P services for remote or external SSL clients. SSL-registered clients now have the added ability to communicate securely by utilizing Tenet and FTP (or other I/P services) after passing SSL client authentication and receiving their session encryption key. In general the SSL approach has very broad benefits. These benefits come with the potential added burden of higher administration costs, although if the value of potential data loss is great, then it is easily offset by the administration cost identified above.

Secure Hypertext Transfer Protocol (S-HTTP)

Secure Hypertext Transfer Protocol (S-HTTP) is emerging as another security tool and incorporates a flexible trust model for providing secure web server and client HTTP communications. It is specifically designed for direct integration into HTTP transactions, with its focus on flexibility for



Client & Server Authentication

Request Encrypted w/Registered Users Private Key

CS responds to user by decrypting request with user public key and responding w/an encrypted session key.

*Assumes CS has access to a key directory server, most likely LDAP compliant.

Exhibit 5. Client and Server Authentication

establishing secure communications in a HTTP environment while providing transaction confidentiality, authenticity/integrity, and nonrepudiation. S-HTTP incorporates a great deal of flexibility in its trust model by leaving defined variable fields in the header definition which identifies the trust model or security algorithm to be used to enable a secure transaction. S-HTTP can support symmetric or asymmetric keys, and even a Kerberos-based trust model. The intention of the authors was to build a flexible protocol that supports multiple trusted modes, key management mechanisms, and cryptographic algorithms through clearly defined negotiation between parties for specific transactions.

At a high level the transactions can begin in an untrusted mode (standard HTTP communication), and "setup" of a trust model can be initiated so that the client and the server can negotiate a trust model, such as a symmetric key-based model on a previously agreed-upon symmetric key, to begin encrypted authentication and communication. The advantage of a S-HTTP-enabled server is the high degree of flexibility in securely communicating with web clients. A single server, if appropriately configured and network enabled, can support multiple trust models under the S-HTTP architecture and serve multiple client types. In addition to being able to serve a flexible user base, it can also be used to address multiple data classifications on a single server where some data types require higher-level encryption or protection than other data types on the same server and therefore varying trust models could be utilized.

The S-HTTP model provides flexibility in its secure transaction architecture, but focuses on HTTP transaction vs. SSL which mandates the trust model of a public/private key security model, which can be used to address multiple I/P services. But the S-HTTP mode is limited to only HTTP communications.

INTERNET, INTRANET, AND WORLD WIDE WEB SECURITY ARCHITECTURES

Implementing a secure server architecture, where appropriate, should also take into consideration the existing enterprise network security architecture and incorporate the secure server as part of this overall architecture. In order to discuss this level of integration, we will make an assumption that the secure Web server is to provide secure data dissemination for external (outside the enterprise) distribution and/or access. A discussion of such a network security architecture would not be complete without addressing the placement of the Web server in relation to the enterprise firewall (the firewall being the dividing line between the protected internal enterprise environment and the external "public" environment).

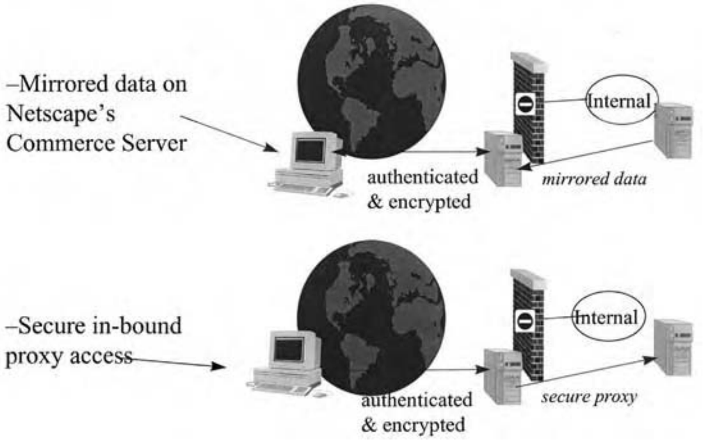

–Mirrored data on Netscape's Commerce Server
Internal
authenticated & encrypted
mirrored data
–Secure in-bound proxy access
Internal
authenticated & encrypted
secure proxy

Exhibit 6. Externally Placed Server

Setting the stage for this discussion calls for some identification of the requirements, so the following list outlines some sample requirements for this architectural discussion on integrating a secure HTTP server with an enterprise firewall.

- Remote client is on public network accessing sensitive company data
- Remote client is required to authenticate prior to receiving data
- Remote client only accesses data via HTTP
- Data is only updated periodically
- Host site maintains firewall
- Sensitive company data must be encrypted on public networks
- Company support personnel can load HTTP server from inside the enterprise

Based on these high-level requirements, an architecture could be set up that would place a S-HTTP server external to the firewall, with one-way communications from inside the enterprise "to" the external server to perform routine administration, and periodic data updates. Remote users would access the S-HTTP server utilizing specified S-HTTP secure transaction modes, and be required to identify themselves to the server prior to being granted access to secure data residing on the server. Exhibit 6 depicts this architecture at a high level. This architecture would support a secure HTTP distribution of sensitive company data, but does not provide absolute protection due to the placement of the S-HTTP server entirely external to the protected enterprise. There are some schools of thought that since this server is unprotected by the company-controlled

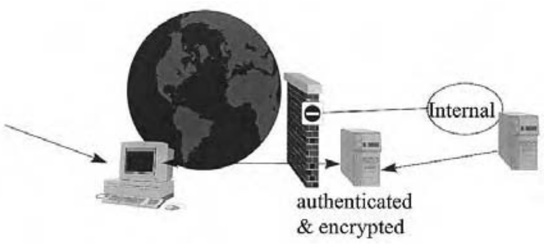


Exhibit 7. Internally Placed Server

firewall, the S-HTTP server itself is vulnerable, thus risking the very control mechanism itself and the data residing on it. The opposing view on this is that the risk to the overall enterprise is minimized, as only this server is placed at risk and its own protection is the S-HTTP process itself. This process has been a leading method to secure the data, without placing the rest of the enterprise at risk, by placing the S-HTTP server logically and physically outside the enterprise security firewall.

A slightly different architecture has been advertised that would position the S-HTTP server inside the protected domain, as Exhibit 7 indicates. The philosophy behind this architecture is that the controls of the firewall (and inherent audits) are strong enough to control the authorized access to the S-HTTP server, and also thwart any attacks against the server itself. Additionally, the firewall can control external users so that they only have S-HTTP access via a logically dedicated path, and only to the designated S-HTTP server itself, without placing the rest of the internal enterprise at risk. This architecture relies on the absolute ability of the firewall and S-HTTP of always performing their designated security function as defined; otherwise, the enterprise has been opened for attack through the allowed path from external users to the internal S-HTTP server. Because these conditions are always required to be true and intact, the model with the server external to the firewall has been more readily accepted and implemented.

Both of these architectures can offer a degree of data protection in a S-HTTP architecture when integrated with the existing enterprise firewall architecture. As an aid in determining which architectural approach is right for a given enterprise, a risk assessment can provide great input to the decision. This risk assessment may include decision points such as:

- Available resources to maintain a high degree of firewall audit and S-HTTP server audit
- Experience in firewall and server administration
- Strength of their existing firewall architecture

SECURE WWW CLIENT CONFIGURATION

There is much more reliance on the knowledge and cooperation of the end user and the use of a combination of desktop and workstation software, security control parameters within client software, and security products all working together to mimic the security of the mainframe and distributed application's environments. Consider the areas below during the risk assessment process and the design of WWW security solution sets.

- Ensure that all internal and external company-used workstations have resident and active antivirus software products installed. Preferably use a minimum number of vendor products to reduce security support and vulnerabilities as there are varying vendor schedules for providing virus signature updates.
- Ensure that all workstation and browser client software is preconfigured to return all WWW and other external file transfers to temporary files on the desktop. Under no circumstances should client server applications or process-to-process automated routines download files to system files, preference files, bat files, start-up files, etc.
- Ensure that Java script is turned off in the browser client software desktop configuration.
- Configure browser client software to automatically flush the cache, either upon closing the browser or disconnecting from each Web site.
- When possible or available, implement one of the new security products that scans WWW downloads for viruses.
- Provide user awareness and education to all desktop WWW and Internet users to alert them to the inherent dangers involved in using the Internet and WWW. Include information on detecting problems, their roles and responsibilities, your expectations, security products available, how to set and configure their workstations and program products, etc.
- Suggest or mandate the use of screen savers, security software programs, etc., in conjunction with your security policies and distributed security architectures.

This is a list of current areas of concern from a security perspective. There are options that when combined can tailor the browser to the specifications of individual workgroups or individuals. These options will evolve with the browser technology. The list should continue to be modified as security problems are corrected or as new problems occur.

AUDIT TOOLS AND CAPABILITIES

As we move further and further from the "good old days" when we were readily able to secure the "glass house," we rely more on good and sound auditing practices. As acknowledged throughout this chapter, security control mechanisms are mediocre at best in today's distributed networking and computing environments. Today's auditing strategies must be robust, available across multiple heterogeneous platforms, computing and network based, real-time and automated, and integrated across the enterprise.

Today, information assets are distributed all over the enterprise, and therefore auditing strategies must acknowledge and accept this challenge and accommodate more robust and dicey requirements. As is the case when implementating distributed security control mechanisms, in the audit environment there are also many players and functional support areas involved in collecting, integrating, synthesizing, reporting, and reconciling audit trails and audit information. The list includes applications and applications developers and programs, database management systems and database administrators, operating systems and systems administrators, local area network (LAN) administrators and network operating systems (NOS), security administrators and security software products, problem reporting and tracking systems and helpline administrators, and others unique to the company's environment.

As well as real-time, the audit system should provide for tracking and alarming, both to the systems and network management systems, and via pagers to support personnel. Policies and procedures should be developed for handling alarms and problems, i.e., isolate and monitor, disconnect, etc.

There are many audit facilities available today, including special audit software products for the Internet, distributed client/server environments, WWW clients and servers, Internet firewalls, E-mail, News Groups, etc. The application of one or more of these must be consistent with your risk assessment, security requirements, technology availability, etc. The most important point to make here is the fundamental need to centralize distributed systems auditing (not an oxymoron). Centrally collect, sort, delete, process, report, take action, and store critical audit information. Automate any and all steps and processes. It is a well-established fact that human beings cannot review large numbers of audit records and logs and reports without error. Today's audit function is an adjunct to the security function, and as such is more important and critical than ever before. It should be part of the overall security strategy and implementation plan.

The overall audit solutions set should incorporate the use of browser access logs, enterprise security server audit logs, network and firewall system authentication server audit logs, application and middle-ware audit logs, URL filters and access information, mainframe system audit information, distributed systems operating system audit logs, database management system audit logs, and other utilities that provide audit trail information such as accounting programs, network management products, etc.

The establishment of auditing capabilities over WWW environments follows closely with the integration of all external WWW servers with the firewall, as previously mentioned. This is important when looking at the various options available to address a comprehensive audit approach.

WWW servers can offer a degree of auditability based on the operating system of the server on which they reside. The more time-tested environments such as UNIX are perceived to be difficult to secure, whereas the emerging NT platform with its enhanced security features supposedly make it a more secure and trusted platform with a wide degree of audit tools and capabilities (though the vote is still out on NT, as some feel it has not had the time and exposure to discover all the potential security holes, perceived or real). The point, though, is that in order to provide some auditing the first place to potentially implement the first audit is on the platform where the WWW server resides. Issues here are the use of privileged accounts and file logs and access logs for log-ins to the operating system, which could indicate a backdoor attack on the WWW server itself. If server-based logs are utilized, they of course must be file protected and should be off-loaded to a nonserver-based machine to protect against after-the-fact corruption.

Though the server logs are not the only defensive logs that should be relied upon in a public WWW server environment, the other components in the access architecture should be considered for use as audit log tools. As previously mentioned, the WWW server should be placed in respect to its required controls in relation to the network security firewall. If it is a S-HTTP server that is placed behind (Exhibit 4) the firewall then the firewall of course has the ability to log all access to the S-HTTP server and provide a log separate from the WWW server-based logs, and is potentially more secure should the WWW server somehow become compromised.

The prevalent security architecture places externally accessible WWW servers wholly outside the firewall, thus virtually eliminating the capability of auditing access to the WWW server except from users internal to the enterprise. In this case, the network security audit in the form of the network management tool, which monitors the "health" of enterprise components can be called upon to provide a minimal degree of audit over the status of your external WWW server. This type of audit can be important when protecting data that resides on your external server from being subject to "denial of service" attacks, which are not uncommon for external devices. But by utilizing your network management tool to guard against such attacks, and monitoring log alerts on the status or health of this external server, you can reduce the exposure to this type of attack.

Other outside devices that can be utilized to provide audit include the network router between the external WWW server and the true external environment, though these devices are not normally readily set up for comprehensive audit logs, but in some critical cases they could be reconfigured with added hardware and minimal customized programming. One such example would be the "I/P Accounting" function on a popular router product line, which allows off-loading of addresses and protocols through its external interface. This could be beneficial to analyze traffic, and if an attack alert was generated from one of the other logs mentioned, then these router logs could assist in possibly identifying the origin of the attack.

Another possible source of audit logging could come from "back-end" systems from which the WWW server is programmed to "mine" data. Many WWW environments are being established to serve as "front ends" for much larger data repositories, such as Oracle databases, where the WWW server receives user requests for data over HTTP, and the WWW server launches SQL_Net queries to a back-end Oracle data base. In this type of architecture the more developed logging inherent to the Oracle environment can be called upon to provide audits over the WWW queries. The detailed Oracle logs can specify the quantity, data type, and other activity over all the queries that the WWW server has made, thus providing a comprehensive activity log that can be consolidated and reviewed should any type of WWW server compromise be suspected. A site could potentially discover the degree of data exposure through these logs.

These are some of the major areas where auditing can be put in place to monitor the WWW environment while enhancing its overall security. It is important to note that the potential placement of audits encompasses the entire distributed computing infrastructure environment, not just the new WWW server itself. In fact, there are some schools of thought that consider the more reliable audits to be those that are somewhat distanced from the target server, thus reducing the potential threat of compromise to the audit logs themselves. In general, the important point is to look at the big picture when designing the security controls and a supporting audit solution.

WWW/Internet Audit Considerations

After your distributed Internet, intranet, and WWW security policies are firmly established, distributed security architectures are updated to accommodate this new environment. When planning for audit, and security control mechanisms are designed and implemented, you should plan how you will implement the audit environment — not only which audit facilities to use to collect and centralize the audit function, but how much and what type of information to capture, how to filter and review the audit data and logs, and what actions to take on the violations or anomalies identified. Additional consideration should be given to secure storage and access to the audit data. Other considerations include:

- Timely resolution of violations
- Disk space storage availability
- Increased staffing and administration
- In-house developed programming
- Ability to alarm and monitor in real time

WWW SECURITY FLAWS

As with all new and emerging technology, many initial releases come with some deficiency. But this has been of critical importance when that deficiency can impact the access or corruption of a whole corporation or enterprise's display to the world. This can be the case with Web implementations utilizing the most current releases which have been found to contain some impacting code deficiencies, although up to this point most of these deficiencies have been identified before any major damage has been done. This underlines the need to maintain a strong link or connection with industry organizations that announce code shortcomings that impact a site's Web implementation. Some of the leading organizations are CERT, the Computer Emergency Response Team, and CIAC, Computer Incident Advisory Capability.

Just a few of these types of code or design issues that could impact a site's web security include initial issues with the Sun Java language and Netscape's JavaScript (which is an extension library of its HyperText Markup Language, HTML).

The Sun Java language was actually designed with some aspects of security in mind, although upon its initial release there were several functions that were found to be a security risk. One of the most impacting bugs in an early release was the ability to execute arbitrary machine instructions by loading a malicious Java applet. By utilizing Netscape's caching mechanism a malicious machine instruction can be downloaded into a user's machine and Java can be tricked into executing it. This does not present a risk to the enterprise server, but the user community within one's enterprise is of course at risk.

Other Sun Java language bugs include the ability to make network connections with arbitrary hosts (though this has since been patched with the following release) and Java's ability to launch denial of service attacks through the use of corrupt applets.

These types of security holes are more prevalent than the security profession would like to believe, as the JavaScript environment also was found to contain capabilities that allowed malicious functions to take place. The following three are among the most current and prevalent risks:

- JavaScript's ability to trick the user into uploading a file on his local hard disk to an arbitrary machine on the Internet
- The ability to hand out the user's directory listing from the internal hard disk
- The ability to monitor all pages the user visits during a session

The following are among the possible protection mechanisms:

- Maintain monitoring through CERT or CIAC, or other industry organizations that highlight such security risks.
- Utilize a strong software distribution and control capability, so that early releases are not immediately distributed, and that new patched code known to fix a previous bug is released when deemed safe.
- In sensitive environments it may become necessary to disable the browser's capability to even utilize or execute Java or JavaScript — a selectable function now available in many browsers.

Regarding the last point, it can be disturbing to some in the user community to disallow the use of such powerful tools, because they can be utilized against trusted Web pages, or those that require authentication through the use of SSL or S-HTTP. This approach can be coupled with the connection to S-HTTP pages where the target page has to prove its identity to the client user. In this case, enabling Java or JavaScripts to execute on the browser (a user-selectable option) could be done with a degree of confidence.

Other perceived security risks exist in a browser feature referred to as HTTP "cookies." This is a feature that allows servers to store information on the client machine in order to reduce the store and retrieve requirements of the server. The cookies file can be written to by the server, and that server, in theory, is the only one that can read back that cookies entry. Uses of the cookies file include storing user's preferences or browser history on a particular server or page, which can assist in guiding the user on his next visit to that same page. The entry in the cookies file identifies the information to be stored and the uniform resource locator (URL) or server page that can read back that information, although this address can be masked to some degree so multiple pages can read back the information.

The perceived security concern is that pages impersonating cookies-readable pages could read back a user's cookies information without the user knowing it, or discover what information is stored in a user's cookie file. The threat depends on the nature of the data stored in the cookie file, which is dependent on what the server chooses to write into a user's cookie file. This issue is currently under review, with the intention of adding additional security controls to the cookie file and its function. At this point it is important that users are aware of the existence of this file, which is viewable in the Macintosh environment as a Netscape file and in the Win environment as a cookies.txt file. There are already some inherent protections in the cookie file: one is the fact that the cookie file currently has a maximum of 20 entries, which potentially limits the exposure. Also, these entries can be set up with expiration dates to they do not have an unlimited lifetime.

WWW SECURITY MANAGEMENT

Consider the overall management of the Internet, intranet, and WWW environment. As previously mentioned, there are many players in the support role and for many of them this is not their primary job or priority. Regardless of where the following items fall in the support infrastructure, also consider these points when implementing ongoing operational support:

- Implement WWW browser and server standards.
- Control release and version distribution.
- Implement secure server administration including the use of products and utilities to erase sensitive data cache (NSClean).
- Ensure prompt problem resolution, management, and notification.
- Follow industry and vendor discourse on WWW security flaws and bugs including CERT distribution.
- Stay current on new Internet and WWW security problems, Netscape encryption, Java, cookies, etc.

WWW SUPPORT INFRASTRUCTURE

- WWW servers accessible from external networks should reside outside the firewall and be managed centrally.
- By special approval, decentralized programs can manage external servers, but must do so in accordance with corporate policy and be subjected to rigorous audits.
- Externally published company information must be cleared through legal and public relations departments (i.e., follow company procedures).
- External outbound http access should utilize proxy services for additional controls and audit.
- WWW application updates must be authenticated utilizing standard company security systems (as required).
- Filtering and monitoring software must be incorporated into the firewall.
- The use of discovery crawler programs must be monitored and controlled.
- Virus software must be active on all desktop systems utilizing WWW.
- Externally published information should be routinely updated or verified through integrity checks.

In conclusion, as information security practitioners embracing the technical challenges of the twenty-first century, we are continually challenged to integrate new technology smoothly into our existing and underlying security architectures. Having a firm foundation or set of security principles, frameworks, philosophies and supporting policies, procedures, technical architectures, etc. will assist in the transition and our success.

Approach new technologies by developing processes to manage the integration and update the security framework and supporting infrastructure, as opposed to changing it. The Internet, intranet, and the World Wide Web is exploding around us — what is new today is old technology tomorrow. We should continue to acknowledge this fact while working aggressively with other MIS and customer functional areas to slow down the train to progress, be realistic, disciplined, and plan for new technology deployment.

ABOUT THE AUTHORS

Lynda McGhie is director of Information Security for Lockheed Martin Corporation in Bethesda, MD.

Phillip Q. Maier is a member of the Secure Network Initiative at Lockheed Martin in Sunnyvale, CA.

Chapter 52
Firewall Management and Internet Attacks

Jeffery J. Lowder

Network connectivity can be both a blessing and a curse. On the one hand, network connectivity can enable users to share files, exchange e-mail, and pool physical resources. Yet network connectivity can also be a risky endeavor if the connectivity grants access to would-be intruders. The Internet is a perfect case in point. Designed for a trusted environment, many contemporary exploits are based on vulnerabilities inherent to the protocol itself. According to a recent dissertation by John Howard on Internet unauthorized access incidents reported to the Computer Emergency Response Team (CERT), there were 4567 incidents between 1989 and 1996, with the number of incidents increasing each year at a rate of 41 to 62 percent. In light of this trend, many organizations are implementing firewalls to protect their internal networks from the untrusted Internet.

LAYING THE GROUNDWORK FOR A FIREWALL

Obtaining management support for a firewall prior to implementation can be very useful after the firewall is implemented. When a firewall is implemented on a network for the first time, it will almost surely be the source of many complaints. For example:

- Organizations that have never before had firewalls almost always do not have the kind of documentation necessary to support user requirements.
- If the firewall hides information about the internal network from the outside network, this will break any network transactions in which the remote system uses an access control list and the address of the firewall is not included in that list.
- Certain types of message traffic useful in network troubleshooting (e.g., PING, TRACEROUTE) may no longer work.

0-8493-1161-6/02/$0.00+$1.50

All of these problems can be solved, but the point is that coordination with senior management *prior to* installation can make life much easier for firewall administrators.

Benefits of Having a Firewall

So how does one obtain management support for implementation of a firewall? The security practitioner can point out the protection that a firewall provides: protection of the organization's network from intruders, protection of external networks from intruders within the organization, and protection from "due care" lawsuits. The security practitioner can also list the positive benefits a firewall can provide:

- *Increased ability to enforce network standards and policies.* Without a firewall or similar device, it is easy for users to implement systems that the Information Services (IS) department does not know about, that are in violation of organizational standards or policies, or both. In contrast, organizations find it very easy to enforce both standards and policies with a firewall that blocks all network connections by default. Indeed, it is not uncommon for organizations to discover undocumented systems when they implement such a firewall for the first time.
- *Centralized internetwork audit capability.* Because all or most traffic between the two networks must pass through the firewall (see below), the firewall is uniquely situated to provide audit trails of all connections between the two networks. These audit trails can be extremely useful for investigating suspicious network activity, troubleshooting connectivity problems, measuring network traffic flows, and even investigating employee fraud, waste, and abuse.

Limitations of a Firewall

Even with all of these benefits, firewalls still have their limitations. It is important that the security practitioner understand these limitations because if these limitations allow risks that are unacceptable to management, it is up to the security practitioner to present additional safeguards to minimize these risks. The security practitioner must not allow management to develop a false sense of security simply because a firewall has been installed.

- *Firewalls provide no data integrity.* It is simply not feasible to check all incoming traffic for viruses. There are too many file formats and often files are sent in compressed form. Any attempt to scan incoming files for viruses would severely degrade performance. Firewalls have plenty of processing requirements without taking on the additional responsibility of virus detection and eradication.

- *Firewalls do not protect traffic that is not sent through it.* Firewalls cannot protect against unsecured, dial-up modems attached to systems inside the firewall; internal attacks; social engineering attacks; or data that is routed around them. It is not uncommon for an organization to install a firewall, then pass data from a legacy system around the firewall because its firewall did not support the existing system.
- *Firewalls may not protect anything if they have been compromised.* Although this statement should be obvious, many security practitioners fail to educate senior management on its implications. All too often, senior management approves — either directly or through silence — a security posture that positively lacks an internal security policy. Security practitioners cannot allow perimeter security via firewalls to become a substitute for internal security.
- *Firewalls cannot authenticate datagrams at the transport or network layers.* A major security problem with the TCP/IP is that any machine can forge a packet claiming to be from another machine. This means that the firewall has literally no control over how the packet was created. Any authentication must be supported in one of the higher layers.
- *Firewalls provide limited confidentiality.* Many firewalls have the ability to encrypt connections between two firewalls (using a so-called virtual private network, or VPN), but they typically require that the firewall be manufactured by the same vendor.

A firewall is no replacement for good host security practices and procedures. Individual system administrators still have the primary responsibility for preventing security incidents.

FIREWALLS AND THE LOCAL SECURITY POLICY

Cheswick and Bellovin (1994) define a firewall as a system with the following set of characteristics:

- All traffic between the two networks must pass through the firewall.
- Only traffic that is authorized by the local security policy will be allowed to pass.
- The firewall itself is immune to penetration.

Like any security tool, a firewall merely provides the capability to increase the security of the path between two networks. It is the responsibility of the firewall administrator to take advantage of this capability; and no firewall can guarantee absolute protection from outside attacks. The risk analysis should define the level of protection that can be expected from the firewall; the local security policy should provide general guidelines on how this protection will be achieved; and both the assessment and revised policy should be accepted by top management prior to firewall implementation.

Despite the fact that, according to Atkins et al.,[1] all traffic between the two networks must pass through the firewall, in practice this is not always technically feasible or convenient. Network administrators supporting legacy or proprietary systems may find that getting them to communicate through the firewall may not be as easy as firewall vendors claim, if even possible. And even if there are no technical obstacles to routing all traffic through the firewall, users may still complain that the firewall is inconvenient or slows their systems down. Thus, the local security policy should specify the process by which requests for exceptions[1] will be considered.

As Bellovin[2] states, the local security policy defines what the firewall is supposed to enforce. If a firewall is going to allow only authorized traffic between two networks, then the firewall has to know what traffic is authorized. The local security policy should define "authorized" traffic, and it should do so at a somewhat technical level. The policy should also state a default rule for evaluating requests: either all traffic is denied except that which is specifically authorized, or all traffic is allowed except that which is specifically denied.

Network devices that protect other network devices should themselves be protected against intruders. (If the protection device were not secure, intruders could compromise the device and then compromise the system[s] that the device was supposed to protect.)

FIREWALL EVALUATION CRITERIA

Choosing the right firewall for an organization can be a daunting task, given the complexity of the problem and the wide variety of products from which to choose. Yet the following criteria should help the security practitioner narrow the list of candidates considerably.

- *Performance.* Firewalls always impact the performance of the connection between the local and remote networks. Adding a firewall creates an additional hoop for network packets to travel through; if the firewall must authenticate connections, that creates an additional delay. The firewall machine should be powerful enough to make these delays negligible.
- *Requirements support.* A firewall should support all of the applications that an organization wants to use across the two networks. Virtually all firewalls support fundamental protocols like SMTP, Telnet, FTP, and HTTP; strong firewalls should include some form of circuit proxy or generic packet relay. The security practitioner should decide what other applications are required (e.g., Real Audio, VDOLive, S-HTTP, etc.) and evaluate firewall products accordingly.
- *Access control.* Even the simplest firewalls support access control based on IP addresses; strong firewalls will support user-based access

control and authentication. Large organizations should pay special attention to whether a given firewall product supports a large number of user profiles and ensure that the firewall can accommodate increased user traffic.

- *Authentication.* The firewall must support the authentication requirements of the local security policy. If implementation of the local security policy will entail authenticating large numbers of users, the firewall should provide convenient yet secure enterprisewide management of the user accounts. Some firewalls only allow the administrator to manage user accounts from a single console; this solution is not good enough for organizations with thousands of users, each of whom needs his or her own authentication account. Moreover, there are logistical issues that need to be thought out. For example, suppose the local security policy requires authentication of all inbound telnet connections. How will geographically separated users obtain the proper authentication credentials (e.g., passwords, hard tokens, etc.)?
- *Physical security.* The local security policy should stipulate the location of the firewall, and the hardware should be physically secured to prevent unauthorized access. The firewall must also be able to interface with surrounding hardware at this location.
- *Auditing.* The firewall must support the auditing requirements of the local security policy. Depending on network bandwidth and the level of event logging, firewall audit trails can become quite large. Superior firewalls will include a data reduction tool for parsing audit trails.
- *Logging and alarms.* What logging and alarms does the security policy require? If the security policy dictates that a potential intrusion event trigger an alarm and mail message to the administrator, the system must accommodate this requirement.
- *Customer support.* What level of customer support does the firewall vendor provide? If the organization requires 24-hour-a-day, 365-days-a-year technical support, is it available? Does the vendor provide training courses? Is self-help online assistance, such as a Web page or a mailing list, available?
- *Transparency.* How transparent is the firewall to the users? The more transparent the firewall is to the users, the more likely they will be to support it. On the other hand, the more confusing or cumbersome the firewall, the more likely the users are to resist it.

FIREWALL TECHNIQUES

There are three different techniques available to firewalls to enforce the local security policy: packet filtering, application-level gateways, and circuit-level gateways. These techniques are not mutually exclusive; in practice, firewalls tend to implement multiple techniques to varying extents. This section defines these firewall techniques.

Exhibit 1. Sample Packet Filter Configuration

Rule Number	Action	Local Host	Local Port	Remote Host	Remote Port
0	Allow	WWW server	80	*	*
1	Deny	*	*	*	*

Exhibit 2 Packet Filter Configuration to Allow Telnet Access from <machine room> to <www-server>

Rule Number	Action	Local Host	Local Port	Remote Host	Remote Port
0	Allow	WWW server	80	*	*
1	Allow	WWW server	23	<machine room>	*
2	Deny	*	*	*	*

Packet Filtering

Packet filters allow or drop packets according to the source or destination address or port. The administrator makes a list of acceptable and unacceptable machines and services, and configures the packet filter accordingly. This makes it very easy for the administrator to filter access at the network or host level, but impossible to filter access at the user level (see Exhibit 1).

The packet filter applies the rules in order from top to bottom. Thus, in Exhibit 1, rule 0 blocks all network traffic by default; rule 1 creates an exception to allow unrestricted access on port 80 to the organization's Web server.

But what if the firewall administrator wanted to allow telnet access to the Web server by the Webmaster? The administrator could configure the packet filter as shown in Exhibit 2. The packet filter would thus allow telnet access (port 23) to the Web server from the address or addresses represented by <machine room>, but the packet filter has no concept of user authentication. Thus, unauthorized individuals originating from the <machine room> address(es) would be allowed telnet access to the WWW server, while authorized individuals originating from non-<machine room> address(es) would be denied access. In both cases, the lack of user authentication would prevent the packet filter from enforcing the local security policy.

Application-Level Gateways

Unlike packet filters, application-level gateways do not enforce access control lists. Instead, application-level gateways attempt to enforce connection integrity by ensuring that all data passed on a given port is in accordance with the protocol for that port. This is very useful for preventing transmissions prohibited by the protocol, but not handled properly by the remote system. Consider, for example, the Hypertext Transmission Protocol (HTTP) used by WW servers to send and receive information, normally on port 80. Intruders have been able to compromise numerous servers by transmitting special packets outside the HTTP specification. Pure packet filters are ineffective against such attacks because they can only restrict access to a port based on source and destination address, but an application gateway could actually prevent such an attack by enforcing the protocol specification for all traffic on the related port.

The application gateway relays connections in a manner similar to that of the circuit-level gateway (see below), but it provides the additional service of checking individual packets for the particular application in use. It also has the additional ability to log all inbound and outbound connections.

Circuit-Level Gateways

A circuit-level gateway creates a virtual circuit between the local and remote networks by relaying connections. The originator opens a connection on a port to the gateway, and the gateway in turn opens a connection on that same port to the remote machine. The gateway machine relays data back and forth until the connection is terminated.

Because circuit-level gateways relay packets without inspecting them, they normally provide only minimal audit capabilities and no application-specific controls. Moreover, circuit-level gateways require new or modified client software that does not attempt to establish connections with the remote site directly; the client software must allow the circuit relay to do its job.

Still, circuit relays are transparent to the user. They are well suited for outbound connections in which authentication is important but integrity is not.

See Exhibit 3 for a comparison of these firewall techniques.

DEVELOPING A FIREWALL POLICY AND STANDARDS

Reasons for Having Firewall Policy and Standards

There are a number of reasons for writing formal firewall policies and standards, including:

- Properly written firewall policies and standards will address important issues that may not be covered by other policies. Having a generic corporate policy on information systems security is not

Exhibit 3. Advantages and Disadvantages of Firewall Techniques

Firewall Technique	Advantages	Disadvantages
Packet filtering	Completely transparent Easy to filter access at the host or network level Inexpensive: can use existing routers to implement	Reveals internal network topology Does not provide enough granularity for most security policies Difficult to configure Does not support certain traffic Susceptible to address spoofing Limited or no logging, alarms No user authentication
Application-level gateways	Application-level security Strong user access control Strong logging and auditing support Ability to conceal internal network	Requires specialized proxy for each service Slower to implement new services Inconvenient to end users No support for client software that does not support redirection
Circuit-level gateways	Transparent to user Excellent for relaying outbound connections	Inbound connections risky Must provide new client programs

good enough. There are a number of specific issues that apply to firewalls but would not be addressed, or addressed in adequate detail, by generic security policies.

- A firewall policy can clarify how the organization's security objectives apply to the firewall. For example, a generic organizational policy on information protection might state that, "Access to information is granted on a need-to-know basis." A firewall policy would interpret this objective by stating that, "All traffic is denied except that which is explicitly authorized."
- An approved set of firewall standards makes configuration decisions much more objective. A firewall, especially one with a restrictive configuration, can become a hot political topic if the firewall administrator wants to block traffic that a user really wants. Specifying the decision-making process for resolving such issues in a formal set of standards will make the process much more consistent to all users. Everyone may not always get what he or she wants, but at least the issue will be decided through a process that was adopted in advance.

Policy and Standards Development Process

The following process is recommended as an efficient, comprehensive way to develop a firewall policy. If the steps of this process are followed in order, the security practitioner can avoid making time-wasting oversights and errors in the policy. (See also Exhibit 4.)

Exhibit 4. Policy Development Process

1. Risk analysis
2. Identify list of topics to cover
3. Assign responsibility for policy
4. Define the audience
5. Write the policy
6. Identify mechanisms to foster compliance
7. Review

1. *Risk analysis.* An organization should perform a risk analysis prior to developing a policy or a set of standards. The risk analysis will not only help policy-makers identify specific issues to be addressed in the document itself, but also the relative weight policy-makers should assign to those issues.
2. *Identify list of topics to cover.* A partial listing of topics is suggested under Policy Structure later in this article; security policy-makers should also identify any other relevant issues that may be relevant to the organization's firewall implementation.
3. *Assign responsibility.* An organization must define the roles and responsibilities of those accountable for administering the firewall. If necessary, modify job descriptions to reflect the additional responsibility for implementing, maintaining, and administering the firewall, as well as establishing, maintaining, and enforcing policy and standards.
4. *Define the audience.* Is the policy document intended to be read by IS personnel only? Or is the document intended to be read by the entire organization? The document's audience will determine its scope, as well as its degree of technical and legal detail.
5. *Write the policy.* Because anyone can read the document, write without regard to the reader's position within the organization. When it is necessary to refer to other organizational entities, use functional references whenever possible (e.g., Public Relations instead of Tom Smith, Public Relations). Be sure to list a contact person for readers who may have questions about the policy.
6. *Identify mechanisms to foster compliance.* A policy is ineffective if it does not encourage employees to comply with the policy. Therefore, the individual(s) responsible for developing or maintaining the policy must ensure that adequate mechanisms for enforcement exist. These enforcement mechanisms should not be confused with the clause(s) of a policy that specify the consequences for noncompliance. Rather, enforcement mechanisms should include such administrative procedures as awareness and training, obtaining employee signatures on an agreement that specifies the employee has read and understands the policy and will comply with the intent.

7. *Review.* New policies should be reviewed by representatives from all major departments of the organization — not just IS personnel. A special effort should be made to resolve any disagreements at this stage: the more low- and mid-level support that exists for a policy, the easier it will be to implement that policy.

After the policy has been coordinated with (and hopefully endorsed by) department representatives, the policy should be submitted to senior management for approval. It is extremely important that the most senior-level manager possible sign the policy. This will give the IS security staff the authority it needs to enforce the policy.

Once the policy is adopted, it should be reviewed on at least an annual basis. A review may have one of three results: no change, revisions to the policy, or abandoning the policy.

Policy Structure

A policy is normally understood as a high-level document that outlines management's general instructions on how things are to be run. Therefore, an organizational firewall policy should outline that management expects other departments to support the firewall, the importance of the firewall to the organization, etc. The structure of a firewall policy should look as follows:

- *Background.* How does the importance of the firewall relate to overall organizational objectives (e.g., the firewall secures information assets against the threat of unauthorized external intrusion)?
- *Scope.* To whom and what does this policy apply?
- *Definitions.* What is a firewall? What role does it play within the enterprise?
- *Responsibilities.* What resources and respective responsibilities need to be assigned to support the firewall? If the default configuration of the firewall will be to block everything that is not specifically allowed, who is responsible for requesting exceptions? Who is authorized to approve these requests? On what basis will those decisions be made?
- *Enforcement.* What are the consequences for failing to meet the administrative responsibilities? How is noncompliance addressed?
- *Frequency of review.* How often will this policy be reviewed? With which functions in the organization?
- *Policy coordinator.* Who is the point of contact for this policy?
- *Date of last revision.* When was this policy last revised?

Firewall Standards

Firewall standards can be defined minimally as a set of configuration options for a firewall. (Although firewall standards can and should address more than mere configuration issues, all firewall standards cover at least this much.) Exhibit 5 presents a sample outline for firewall standards. Because all firewalls come with default configurations, all firewalls have default standards. The job of the security practitioner is to draft a comprehensive set of standards governing all aspects of firewall implementation, usage, and maintenance, including, but not limited to:

- Protection of logs against unauthorized modification
- Frequency of logs review
- How long logs will be retained
- When the logs will be backed up
- To whom the alarms will be sent

Legal Issues Concerning Firewalls

If firewall audit trails need to be capable of being presented as evidence in a court of law, it is worthwhile to provide a "warning banner" to warn users about what sort of privacy they can expect. Many firewalls can be configured to display a warning banner on telnet and FTP sessions. Exhibit 6 shows an example of such a warning.

FIREWALL CONTINGENCY PLANNING

Firewall Outage

What would be the impact on an organization if the firewall was unavailable? If the organization has routed all of its Internet traffic through a firewall (as it should), then a catastrophic hardware failure of the firewall machine would result in a lack of Internet connectivity until the firewall machine is repaired or replaced. How long can the organization tolerate an outage? If the outage were a catastrophic hardware failure, do you know how you would repair or replace the components? Do you know how long it would take to repair or replace the components?

If the organization has a firewall, the odds are that a firewall outage would have a significant impact on that organization. (If the connection between the two networks was not important to the organization, why would that organization have the connection and protect it with a firewall?) Therefore, the security practitioner must also develop contingency plans for responding to a firewall outage. These contingency plans must address three types of failures: hardware, software, and evolutionary (failure to keep pace with increasing usage requirements).

In the case of a hardware failure, the security practitioner has three options: repair, replacement, or removal. Firewall removal is a drastic measure that is not encouraged, it drastically reduces security while disrupting any user services that were specially configured around the firewall

Exhibit 5. Sample Outline of Firewall Standards

I. Definition of terms
II. Responsibilities of the firewall administrator
III. Statement of firewall limitations
  a. Inability to enforce data integrity
  b. Inability to prevent internal attacks
IV. Firewall configuration
  a. Default policy (allow or deny) on network connections
  b. Physical location of firewall
  c. Logical location of firewall in relation to other network nodes
  d. Firewall system access policy
    1. Authorized individuals
    2. Authentication methods
    3. Policy on remote configuration
  e. Supported services
    1. Inbound
    2. Outbound
  f. Blocked services
    1. Inbound
    2. Outbound
  g. Firewall configuration change management policy
V. Firewall audit trail policy
  a. Level of granularity (e.g., we will have one entry for each FTP or HTTP download)
  b. Frequency of review (e.g., we will check the logs once a day)
  c. Access control (e.g., access to firewall audit trails will be limited to the following individuals)
VI. Firewall intrusion detection policy
  a. Alarms
    1. Alarm thresholds
    2. Alarm notifications (e.g., e-mail, pager, etc.)
  b. Notification procedures
    1. Top management
    2. Public relations
    3. System administrators
    4. Incident response teams
    5. Law enforcement
    6. Other sites
  c. Response priorities (e.g., human safety, containment, public relations)
  d. Documentation procedures
VII. Backups
  a. Frequency of incremental backups
  b. Frequency of system backups
  c. Archive of backups (e.g., we will keep backups for one year)
  d. Off-site backup requirements
VIII. Firewall outage policy
  a. Planned outages
  b. Unplanned outages
    1. Reporting procedures
IX. Firewall standards review policy (e.g., this policy will be reviewed every six months)

Exhibit 6. Sample Warning Banner

Per AFI 33-219 requirement:

Welcome to USAFAnet
United States Air Force Academy

This is an official Department of Defense (DoD) computer system for authorized use only. All data contained on DoD computer systems is owned by DoD and may be monitored, intercepted, recorded, read, copied, or captured in any manner and disclosed in any manner by authorized personnel. THERE IS NO RIGHT TO PRIVACY ON THIS SYSTEM. Authorized personnel may give any potential evidence of crime found on DoD computer systems to law enforcement officials. USE OF THIS SYSTEM BY ANY USER, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES EXPRESS CONSENT TO THIS MONITORING, INTERCEPTION, RECORDING, READING, COPYING, OR CAPTURING, AND DISSEMINATION BY AUTHORIZED PERSONNEL. Do not discuss, enter, transfer, process, or transmit classified/sensitive national security information of greater sensitivity than this system is authorized. USAFAnet is not accredited to process classified information. Unauthorized use could result in criminal prosecution. If you do not consent to these conditions, do not log in!

(e.g., Domain Name Service, proxies, etc.). Smaller organizations may choose to repair their hardware because it is cheaper, yet this may not always be an option and may not be quick enough to satisfy user requirements. Conversely, access can be restored quickly by swapping in a "hot spare," but the cost of purchasing and maintaining such redundancy can be prohibitive to smaller organizations.

Significant Attacks, Probes, and Vulnerabilities

To be effective, the firewall administrator must understand not only how attacks and probes work, but also must be able to recognize the appropriate alarms and audit trail entries.

There are three attacks in particular with which every Internet firewall administrator should be familiar.

Internet Protocol (IP) Source Address Spoofing. IP Source Address Spoofing is not an attack itself. It is a vulnerability that can be exploited to launch attacks (e.g., session hijacking). First described by Robert T. Morris in 1985 and explained in more detail by Steven Bellovin in 1989, the first known use of IP Source Address Spoofing was in 1994. Since then, hackers have made spoofing tools publicly available so that one need not be a TCP/IP expert in order to exploit this vulnerability.

IP Source Address Spoofing is used to defeat address-based authentication. Many services, including rlogin and rsh, rely on IP addresses for authentication. Yet, as this vulnerability illustrates, this form of authentication is extremely weak and should only be used in trusted environments. (IP addresses provide identification, not authentication.) By its very nature, IP allows anyone to send packets claiming to be from any IP address. Of course, when an attacker sends forged packets to a target machine, the target machine will send its replies to the legitimate client, not the attacker. In other words, the attacker can send commands but will not see any output. As described below, in some cases, this is enough to cause serious damage.

Although there is no way to totally eliminate IP Source Address Spoofing, there are ways to reduce such activity. For example, a packet filter can be configured to drop all outbound packets that do not have an "inside" source address. Likewise, a firewall can block all inbound packets that have an internal address as the source address. However, such a solution will only work at the network and subnet levels. There is no way to prevent IP Source Address Spoofing within a subnet.

TCP Hijacking. TCP Hijacking is used to defeat authenticated connections. It is only an attack option if the attacker has access to the packet flow. In a TCP Hijacking attack, (1) the attacker is located logically between the client and the server, (2) the attacker sends a "killer packet" to the client, terminating the client's connection to the server, and (3) the attacker then continues the connection.

Denial of Service. A strength of public networks like the Internet lies in the fact that anyone can create a public service (e.g., a Web server or anonymous File Transfer Protocol [FTP] server) and allow literally anyone else, anonymously, to access that service. But this unrestricted availability can also be exploited in a denial-of-service attack. A denial-of-service attack exploits this unrestricted availability by overwhelming the service with requests. Although it is relatively easy to block a denial-of-service attack if the attack is generated by a single address, it is much more difficult — if not impossible — to stop a denial-of-service attack originating from spoofed, random source IP addresses.

There are two forms of denial-of-service attacks that are worth mentioning: TCP SYN Attack and ICMP Echo Flood.

1. TCP SYN Attack. The attacker floods a machine with TCP "half-open" connections, preventing the machine from providing TCP-based services while under attack and for some time after the attack stops. What makes this attack so significant is that it exploits an inherent characteristic of TCP; there is not yet a complete defense to this attack.

Under TCP (used by Simple Mail Transfer Protocol [SMTP], Telnet, HTTP, FTP, Gopher, etc.), whenever a client attempts to establish a connection to

Exhibit 7. Normal TCP Handshake

Client		Server
SYN	------------->	Server
<----------------		SYN-ACK
ACK	------------->	
Client and server may now exchange data		

a server, there is a standard "handshake" or sequence of messages they exchange before data can be exchanged between the client and the server. In a normal connection, this handshake looks similar to the example displayed in Exhibit 7.

The potential for attack arises at the point where the server has sent an acknowledgment (SYN-ACK) back to the client but has not yet received the ACK message. This is what is known as a half-open connection. The server maintains, in a memory, a list of all half-open connections. Unfortunately, servers allocate a finite amount of memory for storing this list, and an attacker can cause an overflow by deliberately creating too many partially open connections.

The SYN Flooding is easily accomplished with IP Source Address Spoofing. In this scenario, the attacker sends SYN messages to the target (victim) server masquerading a client system that is unable to respond to the SYN-ACK messages. Therefore, the final ACK message is never sent to the target server.

Whether or not the SYN attack is used in conjunction with IP Source Address Spoofing, the effect on the target is the same. The target system's list of half-open connections will eventually fill; then the system will be unable to accept any new TCP connections until the table is emptied. In some cases, the target may also run out of memory or crash.

Normally, half-open connections timeout after a certain amount of time; however, an attacker can generate new half-open connections faster than the target system's timeout.

2. Internet Control Message Protocol (ICMP) Echo (PING) Flood. The PING Flood Attack is where the attacker sends large amounts of ICMP ping requests from an intermediary or "bounce" site to a victim, which can cause network congestion or outages. The attack is also known as the "smurf" attack because of a hacker tool called "smurf," which enables the hacker to launch this attack with relatively little networking knowledge.

Like the SYN attack, the PING Flood Attack relies on IP Source Address Spoofing to add another level of indirection to the attack. In a SYN attack with IP Source Address Spoofing, the spoofed source address receives all of the replies to the PING requests. While this does not cause an overflow on the victim machine, the network path from the bounce site to the victim becomes congested and potentially unusable. The bounce site may suffer for the same reason.

There are automated tools that allow attackers to use multiple bounce sites simultaneously. Attackers can also use tools to look for network routers that do not filter broadcast traffic and networks where multiple hosts respond.

Solutions include:

- Disabling IP-directed broadcasts at the router
- Configuring the operating system to prevent the machine from responding to ICMP packets sent to IP broadcast addresses
- Preventing IP source address spoofing by dropping packets that contain a source address for a different network

CONCLUSION

A firewall can only reduce the risk of a breach of security; the only guaranteed way to prevent a compromise is to disconnect the network and physically turn off all machines. Moreover, a firewall should always be viewed as a supplement to host security; the primary security emphasis should be on host security. Nonetheless, a firewall is an important security device that should be used whenever an organization needs to protect one network from another.

The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. government.

Notes

1. Atkins, Derek et al. *Internet Security Professional Reference,* 2nd edition, New Riders, Indianapolis, IN, 1997.
2. Bellovin, Steven M. *Security Problems in the TCP/IP Protocol Suite, Computer Communications Review,* 19:2, April 1989, pp. 32–48. Available on the World Wide Web at ftp://ftp.research.att.com/dist/-internet_security/ipext.ps.Z

References

Bernstein, Terry, Anish B. Bhimani, Eugene Schultz, and Carol Siegel. *Internet Security for Business,* John Wiley & Sons, New York, 1996.

Cheswick, W.R. and Bellovin, S.M. *Firewalls and Internet Security: Repelling the Wily Hacker,* Addison-Wesley, Reading, MA, 1994.

Garfinkel, Simson and Spafford, Gene. Practical Unix & Internet Security, Sebastopol, CA, 1995.

Huegen, Craig A. The Latest in Denial of Service Attacks: ‘Smurfing’, Oct. 18, 1998. Available on the World Wide Web at http://www.quadrunner.com/~chuegen/smurf.txt.

Howard, John D. An Analysis of Security Incidents on the Internet 1989–1995, Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 1997.

Morris, Robert T. A Weakness in the 4.2BSD Unix TCP/IP Software, *Bell Labs Computer Science Technical Report #117,* Feb. 25, 1985. Available on the World Wide Web at ftp://ftp.research.att.com/-dist/internet_security/117.ps.Z.

Wood, Charles Cresson. Policies from the Ground Up, *Infosecurity News,* March/April 1997, pp. 24-29.

ABOUT THE AUTHOR

Jeffrey J. Lowder is chief of network security element at the United States Air Force Academy, Colorado Springs, CO.

Chapter 53
Protecting against Hacker Attacks

Ed Norris

The problem of people breaking into computer and telephone systems is not new. Such activities have occurred since the introduction of these technologies. As new technologies become available, it can be expected that new methods of obtaining unauthorized access to these technologies will be developed.

To protect their systems against unauthorized intrusion, security practitioners need to understand hackers: who they are, what motivates them to break into systems, and how they operate. This chapter examines these issues, describes specific hacking techniques, and recommends actions that should be taken to reduce exposure to hacker attacks.

WHAT IS A HACKER?

The term *hacker* means different things to different people. The author of the book *Prevention and Prosecution of Computer and High Technology Crime* defines hackers as computer criminals and trespassers who view illegal computer access as an intellectual challenge and a demonstration of technical prowess. *The New Hacker's Dictionary* offers a more benign definition: "A person who enjoys learning the details of computer systems and how to stretch their capabilities as opposed to most users of computers who prefer to learn only the minimum amount necessary." Indeed, many people consider themselves hackers, yet would never attempt to gain unauthorized access to a computer or telephone system.

Other terms are also used to refer to hackers. For example, *phreaks* are hackers that target telephone systems. The terms *computer intruder* and *cracker* are also commonly used for computer hackers. This modifies the definition from *The New Hacker's Dictionary*, adding that some hackers may attempt to gain unauthorized access to computer and telephone systems to achieve their goals.

0-8493-1161-6/02/$0.00+$1.50

A Hacker Profile

In the early 1980s, the hacker profile was of a highly intelligent, introverted teenager or young adult male who viewed hacking as a game; most were thought to be from middle- and upper-class families. Like most stereotypes, this profile has proved to be wrong. In reality, hackers can be very smart or of average intelligence, male or female, young or old, rich or poor. Recent hacker arrests and convictions have taught hackers that, whatever they may have thought in the past, hacking is not a game.

To succeed, hackers need three things: motive, opportunity, and means. The motive may be increased knowledge, a joy ride, or profit. Many IS practitioners have the opportunity and means to hack systems but lack the motive.

The opportunity to hack systems has increased greatly over the years. Today, computer systems can be found everywhere. Hackers do not need state-of-the-art equipment to hack systems — used equipment is inexpensive and adequate to the task. Most companies allow some type of remote access by means of either dial-up lines or connections to external networks. For a relatively small monthly fee, anyone can have access to the Internet. Unfortunately, many corporations provide opportunities to access their systems by failing to provide adequate security controls. And many hackers believe that the potential for success outweighs the possible penalties of being caught.

The means of attack is limited only by the imagination and determination of the hacker. A basic law of hacking can be summarized as "delete nothing, move nothing, change nothing, learn everything."

Some hackers target such entities as corporations, security and law enforcement personnel, and other hackers. Kevin Mitnick allegedly electronically harassed a probation officer and FBI agents who got in his way. Some hackers target organizations for political reasons. For example, the Chaos Computer Club supports Germany's Green Party. Others target any machine that runs an operating system capable of executing a particular virus, worm, or Trojan horse.

The estimates of the number of people involved in hacking vary greatly. Estimates range from about one hundred serious hackers to hundreds of thousands. No one really knows the number of people involved. Suffice it to say there are enough hackers to warrant taking precautions to prevent unauthorized access.

Hackers often use aliases, such as Shadow Hawk 1, Phiber Optik, Knight Lightning, Silent Switchman, Dark Avenger, and Rock Steady. These aliases allow them to remain anonymous, yet retain a recognizable identity. And they can change that identity at any time simply by choosing another handle.

For example, Shadow Hawk 1 is known to have also used the handles Feyd Rautha, Captain Beyond, and Mental Cancer. Changing handles is intended to confuse security personnel as to the identity of the hacker, which makes it more difficult to monitor hacker activity. A hacker may also want the targeted organization to think that several people are attacking the target. Security practitioners need to be aware of these methods of operation in order to understand the identity and the true number of hackers involved in an attack.

Hacker Clubs

Some hackers and phreaks belong to such hacker clubs as Legion of Doom, Chaos Computer Club, NuKE, The Posse, and Outlaw Telecommandos. These clubs give a sense of companionship, although most members never physically meet. More importantly, they help members work as a team toward common goals. By bringing together unique technical skills of individual hackers (e.g., specialties in UNIX or TCP/IP), these teams can achieve goals that might be out of reach for an individual hacker. Some hackers may also view membership in hacker clubs as demonstrating to the hacker and security communities that they are skilled members of an elite.

Hacker clubs come and go. The more willing the members are to contribute to club activities, the longer such clubs remain active. Hack-Tic and the Chaos Computer Club have been in existence for a relatively long time, whereas such groups as MAGIK (Master Anarchists Giving Illicit Knowledge) and RRG (Rebels Riting Guild) lasted only a very short time.

Hacker Publications

Some hacker and phreak clubs produce publications. For example, the Legion of Doom produces *Phrack,* Phalcon/Skism produces *40Hex*, and the Chaos Computer Club produces *Chaos Digest*. These publications supply hackers with technical information as well as provide a social function. Some hacker publications can be received by means of electronic mail over the Internet; others are sent through the postal system. Some book stores and magazine stands sell *2600 The Hacker Quarterly*, which has been published for ten years. This publication periodically publishes a list of addresses of other hacker publications.

To keep informed of hacker interests and activities, security practitioners with access to the Internet should subscribe to the nonhacker publication *Computer underground Digest.*This electronic digest covers general issues related to information systems, and also covers hacker- and security-related topics. It often provides pointers to other sources of hacker information. Searching these sources can help the security administrator learn about the ways hackers obtain knowledge. *Computer underground*

Digest can be subscribed to by sending an electronic mail message to listserv@vmd.cso.uiuc.edu; the message should be sub cudigest your-name.

Hacker Conventions

Some hacker groups sponsor hacker conventions. For example, the Chaos Computer Club sponsors the Chaos Congress, Hack-Tic sponsors Hacking at the End of The Universe, and Phrack and the Cult of the Dead Cow sponsor HoHoCon. These conventions are held in the United States and Europe. Hackers and phreaks, as well as security and law enforcement personnel, are featured speakers. The conventions are open to all interested parties.

Most of these conventions serve primarily as venues for hackers to brag, swap stories, and exchange information. They tend not to be highly organized; most substantive information is exchanged in hotel rooms and lobbies. There have been a few raids and arrests at some conventions.

Bulletin Boards and Newsgroups

Hackers and hacker clubs primarily communicate by means of bulletin board systems. It is estimated that there are about 1,300 underground bulletin boards in the U.S. The information found on bulletin board systems is usually current and state of the art. Timeliness of this information is important; hacker techniques described in print publications are usually already well know by the time they appear in print, and these published methods may no longer work.

Even old information can be valuable, however. The LOD (Legion of Doom) Communications is selling old bulletin-board system archives. Although these message bases are from the mid-1980s, many organizations are still being successfully attacked using methods described in those files.

It can be difficult to gain access to an underground bulletin board. A hacker has to have been active for some time and shared valid information in the hacker community. Some bulletin boards even require background checks and references. As new members become more trusted among their peers, their level of access to sensitive information increases. Aboveground hacker bulletin boards usually grant access to anyone, and may even invite security professionals to join the communications exchange. The telephone numbers for this type of bulletin board are sometimes published in the Internet newsgroup alt.bbs.Internet newsgroups are similar to bulletin board systems in that they allow people a vast forum for communication. Currently there are three active hacker newsgroups: alt.2600, de.org.ee, and zer.t-netz.blueboxing. The *Computer underground Digest* is also posted in the newsgroup comp.society.cu digest. The alt.2600newsgroup is very active; as of June 1994, it had approximately 15,000 messages representing 25M-bytes of information from December

1993 through June 1994.Two other newsgroups that sometimes have information relevant in this arena are alt.wired and the Electronic Freedom Foundation s comp.org.eff.talk.

METHODS OF ATTACK

As technology has changed, so too have the challenges facing security professionals. It is important to keep informed of the many methods of attack used by hackers. The following sections describe some of the most popular techniques.

Social Engineering

Social engineering is a term used to describe techniques for getting someone to do something for an unauthorized individual. Hackers and phreaks are usually very adept at the art of social engineering. For example, a hacker may simply ask someone for help. He or she may not get the intended information, but the person asked will often provide at least some useful information. In one case, for example, an intruder told a guard that he needed access to an office because he had a report due on Monday; he went on to complain about having to work on the weekend. The guard sympathized and let the intruder in without asking for his company identification or any other job-related information. The more sincere and knowledgeable the hacker sounds to the person being targeted, the more likely the hacker will get what he or she wants.

Knowledge can also be gained by many other methods, for example, by reading newspapers, magazines, and annual reports. An annual report might disclose the projects a corporation is pursuing and the people who are working on those projects. Reading Internet newsgroups may provide the hacker with a good idea of certain corporate projects; it may also inform the hacker which computer system is being used on which projects. Postings in newsgroups usually contain much useful information; telephone books or information can provide the hacker with a telephone number for the company. By assembling these various pieces of information, the hacker can appear as if he or she were an employee of the targeted company.

One hacker publication recommended that hackers write out a script before calling a target. The script should include the initial explanation, plus answers to questions that they might be asked.

Exhibit 1 illustrates how Internet newsgroups can reveal information useful to hackers. The newsgroup article in Exhibit 1 tells the reader that John Smith is a geologist in the MIS department of Eastern Mining Corporation and that the company is located in New York. The phone number to reach John is 212-555-1234. John is using the computer system emcmis and

Exhibit 1. Gathering Information from a Newsgroup Message

Newsgroups:	comp.programming
From:	jds@emcmis.emc.com (John Smith (Geologist — MIS))
Subject:	USGS Code ????
Reply-To:	jds@emcmis.emc.com
Organization:	Eastern Mining Corporation
Date:	Fri, 6 May 1994 01:23:41 GMT

Does anyone out there know if it is possible to gain access to the USGS source code that I have seen mentioned in journal, using the net. If it is available on the net, can anyone provide me with a site (or even better a list of sites) that I can access using ftp.

Thanks in advance …
John

John D. Smith Eastern Mining Corporation
jds@emcmis.emc.com Phone: 212-555-1234
New York, NY

his user ID is jds. Eastern Mining has access to the Internet and has (at least) outgoing FTP service as well as news and incoming mail access. Finally, John's current project involves geological surveys. There is enough information in this one message for someone to start a social engineering attack.

To help limit disclosure of at least some of this information, the security manager might implement an application-level firewall to standardize the mail addresses to the corporation and other key access information.

Employees should also be made aware of social engineering and the ways to combat it. Because anyone can be the subject of such attacks, awareness programs should be directed to general employees as well as systems operators and help desk personnel.

Employees should also be informed as to whom to notify if they believe someone is asking for information or requesting them to perform a task that is suspect. External requests for corporate information should be passed to a trained public relations department staffperson. Requests to perform a task should not be carried out until that person can verify who the caller is and if the caller is authorized to request that the task be performed. If the verification cannot be completed successfully, someone in the organization should be notified of the attempted intrusion. Often, what appears to be an isolated attempt at unauthorized access is part of a larger social engineering attack. If such an attack is detected, advisories should be sent to employees warning them to be on guard.

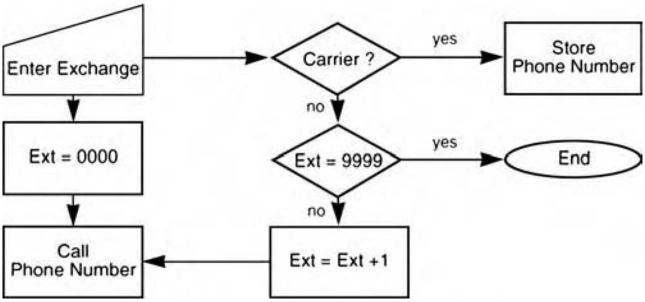


Exhibit 2. Logic of Simple War Dialer Program

DUMPSTER DIVING

Dumpster diving is a term used to describe the searching of garbage for information. Hackers frequently search dumpsters located outside of buildings for such information sources as operator logs, technical manuals, policies, standards, company phone books, credit card numbers, and dial-in telephone numbers, all of which might aid the hacker in gaining access to the organization.

The security administrator should make employees aware that confidential material should be disposed of in accordance with the organization's security standards. After material is destroyed, it may be recycled as part of the regular recycling program.

Hardware and Software Tools

Hackers also rely on hardware and software tools for carrying out attacks. For example, phreaks use hardware devices to generate tones that allow them to navigate the various telephone switches and gain free phone access. These devices, referred to as boxes, are known by their color. For example, the red box is used to generate the coin tones used by pay phones. The newest type of red box makes use of Hallmark greeting cards that allow users to record a message; the phreaks use them to record coin tones. This is not the first time an innocent device has been used for illicit purposes; a whistle given away in Captain Crunch cereal boxes was used to generate the 2600 tone.

Many software tools are available on the Internet or on bulletin board systems. The most infamous is the war dialer. War dialers scan a telephone exchange looking for modem tones. When one is found, the modem phone number is saved. Software tools do not need to be complicated to be useful to hackers. Exhibit 2 illustrates the logic for a simple war dialer that requires only a few lines of code.

Although it is not possible to stop attempts to connect to a modem, unauthorized successful connections are very easy to stop. Modem access can be protected by means of strong passwords, tokens, or other mechanisms. The protection mechanism should have the ability to log access attempts. If the organization is experiencing many unsuccessful access attempts, it may be the target of a hacker or hacker group.

A war dialer expects a quick answer to its call. Some war dialers can be thwarted by increasing the number of rings before the modem answers the call.

One problem that has become widespread is use of unauthorized modems. Internal modems can be purchased for under $30, well within the price range of an average employee. They are very easy to install in a PC workstation and can be used with the office phone line. Using a war dialer, the security manager should conduct a periodic check of telephone numbers that belong to the corporation. A check against the list of authorized modems will detect use of any unauthorized modems.

A password cracker is another popular tool. With this approach, a hacker downloads a targeted password file (e.g., UNIX's/etc/passwdor OpenVMS's SYS$SYSTEM:SYSUAF.DAT) to his or her computer and then attempts to crack the passwords locally. The hacker can do this without triggering any alarms or having to run through the log-in sequence. Six-character UNIX-based passwords have been cracked in less than an hour. If only lowercase letters are used in the password, the password can be cracked in less than one minute. It should be recognized that what is considered a strong password for today's technology may not be adequate a year from now.

Many intrusions are successful because of weak passwords. The security administrator can run a password cracker program against system authorization database s in order to find vulnerable passwords. The security manager should schedule and conduct such checks periodically. Anyone whose password is cracked should be instructed on how to select effective passwords. These persons should also be reminded that failure to do so may jeopardize corporate assets.

Network sniffer software has also been used on the Internet to capture user IDs and passwords; TCP/IP packets were scanned as the packets passed through a node that a hacker already had under his control. (Some security professionals argue that use of Kerberos on the network cures the network sniffer problem, but this only protects the password when it travels between the Kerberos daemon and slave.)

The underlying problem presented by this hacker attack has to do with how the hacker was able to gain control of one or more of the network nodes. Many corporations that have connected to the Internet fail to implement any security measures to counter the additional risk of public access. Any organization that plans to connect to the Internet should first install a firewall.

A firewall is a collection of components placed between two networks. A firewall has the following properties:

- All traffic in both directions must pass through the firewall.
- Only authorized traffic, as defined by the local security policy, is allowed to pass.
- The firewall itself is immune to penetration.

A firewall allows the organization to block or pass access to the internal and external networks based on application, circuit, or packet filtering.

In summary, the security manager should be familiar with the types of hardware and software tools used to attack computer and communications systems. By searching the Internet, he or she should be able to find the same tools that hackers are using. These tools can be used to verify that the organization is adequately protected against them.

Reverse Intent

Reverse intent refers to a phenomenon in which an object that is intended to perform an action is used to perform the opposite action. For example, a deliberate reverse intent message might state: "This product is not to be used to increase the octane in gasoline." The message is intended to warn us that use of the product is prohibited for the purpose of increasing octane levels, but it also discloses that the product is capable of boosting the octane rating.

Hackers and phreaks can use such messages to their advantage. Computer Emergency Response Team (CERT) advisories and Computer Incident Advisory Capability (CIAC) information bulletins are intended to notify people of security problems. They contain information about a given product, the damage that can occur from use or misuse of the product, the solution, and additional information. As illustrated by Exhibit 3, this information can be useful to hackers. In this exhibit, it is reported that Sun Solaris V2.x and SunOS V5.x have a security problem which gives local users the ability to gain root (full privilege) access. The local user can execute the expreserve utility which gives access to system files. If the computer system does not have expreserve disabled or the system administrator has not installed the patch solution provided by the vendor, the security of the system may be compromised. If a hacker has access to a Sun workstation, the hacker can find out how to exploit this security exposure, either on his or her own or with the help of others.

Exhibit 3. Example of Reverse Intent

The Computer Incident Advisory Capability
INFORMATION BULLETIN
Solaris 2.x expreserve patches available

July 1, 1994 0900 PDT
Number D-18
PROBLEM: The expreserve utility allows unauthorized access to system files.
PLATFORM: Sun workstations running Solaris 2.0, 2.1, and 2.2
(SunOS 5.0, 5.1, and 5.2)

DAMAGE: Local users can gain root access.
SOLUTION: Disable expreserve immediately, then install patch from Sun.

Hackers are quick to notify each other of these types of announcements. For example, a CERT advisory dated August 14, 1990 appeared in *Network Information Access* the following day.

It is important that the appropriate department within the organization receive security problem notification from software vendors. (Such notification should not be made to the purchasing department simply because it signed the check for the software.) CERT and CIAC information can be received by means of electronic mail over the Internet. Some vendors have their own advisory mailings over the Internet (e.g., *Hewlett-Packard Security Bulletin*).

The information security manager should develop an action plan for installing security patches. The security manager should also conduct a postmortem after the installation of security patches, noting which actions completed without problems and which actions did not. The action plan can then be adjusted to fix deficiencies and to reflect changes to the business environment.

In addition to obtaining information from advisories, hackers also seek out such sources of information as the system security manuals (or sections of other manuals) provided by software vendors with their products. These books are intended to instruct the system administrator on how to secure the product. Supplemental computer manuals found in almost every book store are another source for the hacker. A security manual might state: “Do not disable high-water marking on disk volumes.” This statement tells a hacker that if one or more disks have disabled high-water marking, there is a potential problem to be exploited. In this case, the hacker may discover the art of disk scavenging and access the information contained in unallocated blocks.

Some professional organizations advocate sharing published security standards among their members. But it can be difficult to control the distribution of these standards, and they can also be used with reverse intent. The standards tell how a corporation secures its business. Many standards contain such sensitive information as group names, employee names and titles, phone numbers, electronic mailing address, and escalation procedures. A standard in the hands of a hacker becomes a powerful tool for social engineering.

It is impossible to stop reverse intent. Security practitioners must be aware of information that is available to hackers and ensure that they act appropriately according to the intent of the information. When an advisory or other piece of information reaches the security administrator, he or she should try to gauge how a hacker might use this information and modify his or her actions accordingly.

SECURITY MONITORING AND REVIEW

To stop a hacker, the security administrator must know when the hacker is knocking at the door or has already entered the system. Waiting until something has gone wrong may be too late. Auditing a system is more than turning on every auditable event, however. The security administrator must monitor enough events to be able to detect an attack, but not so many that the audit information becomes unmanageable. Too much data tends not to be analyzed properly and exceptions to normal behavior become more difficult to detect.

One of the first things a hacker attempts to do is delete the audit trail. Novice hackers may stop the audit processes and delete the entire audit database; experienced hackers remove only their records from the database s. If warranted, audit information should be printed directly to a hardcopy device or to a write-once storage device. The data should be analyzed on a regular basis with follow-up done on any suspect activity. Most hacker intrusions produce a few knocks on the door before a successful penetration takes place. It is easier to keep a hacker out than to recover after a successful intrusion.

RECOMMENDED COURSE OF ACTION

It is important to understand how hackers navigate throughout the electronic world and how they attack systems. Security practitioners also need to understand the threats they pose to the organization and implement appropriate security controls to counter those risks. The existing security program should also be monitored to ensure that it continues to be able to counter the risks created by hackers.

Bibliography

1. Arkin, S., et al. Prevention and Prosecution of Computer and High Technology Crime. New York: Matthew Bender, 1991.
2. Steele, G. and Raymond, E. *The New Hacker's Dictionary.* Cambridge, MA: MIT Press, 1991.
3. Goldstein, E. "Crime Waves" 2600 *The Hacker Quarterly* (Spring 1994).
4. Dispater, "Phrack Pro-Phile on Shadow Hawk 1." *Phrack* (November 39: June 26, 1992).
5. Johnson, J. "Dark Side" Hacker Seen as "Electronic Terrorist." *Los Angeles Times* (January 8, 1989).
6. Milligan, B. "The Organized Hackerhood." *InfoSecurity News* (July/August 1994).
7. Gilboa, N. "Paranoia Strikes Deep: Breaking into the Well." *Gray Areas* (Spring 1994).
8. LOD Communications, *Computer underground Digest* (No. 5.64: August 22, 1993).
9. Black Death, "Social Engineering." *P/H/A Newsletter* (No. 2: July 26, 1990).
10. Madsen, J. "A New World Record in Password Checking Has Been Set." alt.security newsgroup (August 18, 1993).
11. Cheswick, W. and Bellovin, S. *Firewalls and Internet Security.* Reading, MA: Addision-Wesley, 1994.
12. Dredd, J. Network Information Access (No. 45: August 15, 1990).

ABOUT THE AUTHOR

Ed Norris, CISSP, is a senior security consultant for Digital Equipment Corporation. He consults on a wide range of security issues, from strategy to analysis and implementation of security programs. He is an active member of the Information Systems Security Association and the National Computer Security Association.

Chapter 54

Improving Performance in New Programming Environments: Java

Paul J. Jalics
Donald Golden

Java is an effort by Sun Microsystems to bring the 30+-year-old C language into the next century by cleaning it thoroughly of all its sloppiness — including most of the object-oriented (OOP) features of C++ — and providing it with the built-in functionality needed today, which is so sorely missing in most older languages.

The most striking feature of Java programs is that they are executed with the help of a software component called the Java Virtual Machine (JVM), which executes Java machine instructions that are defined as part of the Java language. Thus, when a Java source program prog1.java is compiled, the compiler generates a prog1.class file that contains only JVM machine instructions. To execute the Java program, one calls upon the JVM interpreter. Sun's compiler is named javac, and its JVM interpreter is named java. Thus, to compile and execute prog1, one might type javac prog1.java, which generates prog1.class, which is then executed by typing java prog1.

Java executables (.class files) are totally portable from any architecture computer to any other, and from any operating system to any other. This is one reason that most Internet browsers have the JVM implemented so that sophisticated actions can be implemented by downloaded Java .class

0-8493-1161-6/02/$0.00+$1.50

files. The JVM checks each class file before loading it for execution to make sure that it is well behaved and obeys the basic rules of the Java language. Hopefully, the security features of Java will be found sufficient to prevent downloaded class files from doing damage to the machines running the Java-enabled Internet browsers.

There is another twist in Java code management: some JVMs include a “just-in-time” compilation feature. Instead of the JVM interpreting the instructions of a method directly, it translates or compiles the Java code into the native Pentium machine instructions (on PCs at least) the first time a given method is executed. The resulting Pentium code is put into a cache, so that subsequent executions of that method during the current program’s execution will be found in the cache and executed at maximum speed.

INTRODUCTION TO PERFORMANCE IMPROVEMENTS

Program performance is often considered synonymous with the speed with which a program solves a specific problem. The speed, in turn, is influenced by a number of factors, some of which are programmer controlled, while others depend on the hardware and software environment of the program’s execution. While it may not be necessary to tune all programs for performance efficiency, there frequently exist one or more critical programs where such tuning is essential.

Performance has received too little attention in the programming workplace. Reasons for this lack of emphasis include the still-immature nature of the discipline, and the rapid performance gains realized through faster hardware. However, program performance is and will continue to be a software issue because, given the increasingly complex nature of software products, the desired throughput of a program is not always realizable through hardware advances alone. As in engineering, where performance standards are usually part of a product’s specifications, a program’s performance may become an integral part of a mature software engineering process.

Speed of execution is only one of the dimensions of a program’s performance. Other performance factors include memory usage, code portability, and readability. Unfortunately, tuning a program for improvement in speed of execution can lead to compromises in other dimensions of performance. However, speed of execution (hereafter referred to as performance) is not a critical consideration for all programs. For example, performance is irrelevant if a program generates results faster than they can be used. Also, for many other programs, it may be more cost-effective to use a faster hardware platform, if one is available or can be acquired, than to spend the effort required in tuning; this becomes increasingly common as labor costs continue to increase and hardware costs continue to decrease. However, there still exists a small percentage of programs, say 10 percent, for which performance is important. These programs are called critical programs.

A program can be considered critical for any of several reasons. For example:

- The user may need the output shortly after the input is available, and a short delay may be life-threatening or lead to serious economic loss. Many realtime operations fall into this category when sensor-generated data (as in process control, and command and control systems) needs to be analyzed and results fed back to an appropriate decision-maker. Computer control of a car engine is an example of this type of system.
- A variation on the first type of critical program occurs when the program is part of an interactive system in which the user's productivity would be adversely affected if there is a wait for the system to respond to input.
- In some systems, hardware upgrades are not feasible for economic or technical reasons, yet the performance goals still must be achieved within the existing configuration.
- The system may need to process very large volumes of data. No matter how fast a program runs, a sufficient volume of data can make it run too long.

Fortunately, in most cases, a program's execution time is not uniformly distributed across its statements. Experience has shown that a small part of a program, typically 10 percent or less, largely determines its performance. Therefore, most of the performance improvements are achievable by concentrating on the 10 percent and ignoring the remaining 90 percent of the code. This 10 percent will be called the critical part of a program. There is certainly no implication that non-critical parts of a program cannot yield performance improvements. Such improvements are possible, but would lead to diminishing returns on the labor invested. In other words, a disproportionate amount of labor may be needed for small improvements in execution time.

A PROCEDURE FOR IMPROVING PERFORMANCE

Four steps for improving program performance include:

1. Measuring the performance of the initial program
2. Identifying the critical parts of the program

3. Improving the performance of the critical parts
4. Testing the modified program, remeasuring the performance, and comparing it to the initial performance; if not satisfied, go back to step 2

Step 1: Measuring a Program's Initial Performance

Measuring the execution time of a program can be as simple as using a stopwatch to note the start and finish times of a program. Although a very crude method, it may be adequate in cases where the program executes long enough (e.g., 60 seconds or more) so that manual time-keeping does not lead to too much error in the calculated elapsed time. For example, a human being typically can respond to an event in a tenth of a second or less. This means that human inaccuracy should introduce an error of less than 0.2 percent in timing a 60-second program.

A more precise technique is to let the program compute its own execution time by recording the start and finish times through calls to the host operating system. Current system time is available from all operating systems. Exhibit 1 shows a Java program that reports on its own execution performance measured in elapsed milliseconds.

Step 2: Identifying Critical Program Components

One can identify critical components of a program through manual inspection. For example, inner loops of a program can contribute more significantly to the execution time. However, proper identification of such code is difficult even for experienced programmers who know the program well. Correct identification would require the programmer to be aware of details such as the amount of work done by various statements and the relative frequency of data access from the input files.

A better approach is to use special tools, called profilers, to profile the performance characteristics of the program. A profiler evaluates an executable program (a collection of class files). The user describes what areas of the program are to be profiled; that is, which methods from the application, and which Java system methods. The test program is then run under control of the profiler, which collects information about the execution, including the number of times a procedure is executed, and the execution time spent in each method included in the profiling. The resulting data can then be used to identify the program's critical parts. Note that profiling has the disadvantage of increasing execution time dramatically — sometimes by a factor of 5 to 1000 or more.

In the authors' experiments, Sun's Java Workshop 2 Profiler tool was used to identify the critical parts of the program at the method level, although some other profilers can also give information at the statement

Exhibit 1. Measuring the execution time of a Java Program

```
–> CULLER PROGRAM LISTING <– Thu Feb 04 10:24:51 1999
1. fig1.java 1.1 public class NHScheckByteArray {
2. fig1.java 2.2 public static void main(String args[]) { long counter=0;
3. fig1.java 3. System.gc(); // run garbage collector synchronously
4. fig1.java 4. time1 = System.currentTimeMillis(); //get start time
5. fig1.java 5. for(int I=0; I++; I< 100000 ) /loop being measured
6. fig1.java 6. counter +=5000; // body of loop
7. fig1.java 7. long time2 = System.currentTimeMillis(); //get ending time
8. fig1.java 8. System.out.println((time2 - time1)
9. fig1.java 9. + " milliseconds to add 5000 to counter 100,000 times. ");
10. fig1.java 10.1 }
11. fig1.java 10.0 }

### CULLER CROSS REFERENCE LISTING

add 9
args 2
being 5
body 6
class 1
collecto 3
counter 2 6 9
currentT 4 7
```

level. Typically, the results of profiling are stored in a data file for later analysis. These results can then be viewed in the Java Workshop profiler in three basic ways:

1. Display the time spent in each of the methods profiled
2. Display the cumulative time spent in each method profiled, which also includes the time spent in methods called from that method
3. Display the number of times the various methods were called

First Profiling Results. Exhibit 2 shows the initial performance profile of the program jxreff.java, which is discussed below. In this exhibit, only the last entry refers to a user-written method; all remaining methods are system methods and cannot be modified by the user.

Note that the time spent in a method is not directly related to the number of times the method is called. As can be seen from Exhibit 2, just two methods (String.toLower and StringBuffer.append) account for 50 percent of the execution time excluding input/output, in spite of the fact that the program used hundreds of system methods.

Exhibit 2. Partial Profile Results

Method	# of Times Executed	Amount of Time Spent in the Method
String.toLowerCase	447,126	192,999 milliseconds
StringBuffer.append	2,666,970	192,549 milliseconds
Character.toLowerCase	2,652,386	75,516 milliseconds
StringBuffer.ensureCapacity	2,676,565	61,871 milliseconds
StringBuffer.copyWhenShared	2,681,003	54,220 milliseconds
String.<init>	454,642	51,308 milliseconds
String.toLowerCase	447,126	28,902 milliseconds
StringBuffer.toString	454,642	21,009 milliseconds
StringBuffer.<init>(I)	454,958	19,098 milliseconds
StringBuffer.<init>	449,289	18,955 milliseconds
sym_table.find_sym	2.179	17,555 milliseconds
Object.<init>	915,011	15,518 milliseconds
String.equals	450198	15,103 milliseconds
String.compareTo	223,562	12,392 milliseconds
…		
identifier.tell_identifier	18,560	2,946 milliseconds
…		

Step 3: Improving the Performance of a Program

Once the critical program components have been identified, code changes can be implemented to improve performance. The discussion of these code changes will follow in the next section.

Testing the Modified Program. A modified program needs to be tested to verify that it continues to execute correctly, and that its performance has improved. If the execution time is not reduced, it will be necessary to return to the premodification state of the program. Therefore, it is essential to retain older versions of the program until changes are successfully tested. If the performance of the revised program improves without meeting the overall goals, one needs to go back to step 2 to identify anew the critical parts of the modified program. Note that the critical parts of the modified program can be substantially different from those of the previous version. Thus, a new profiling run is needed after each improvement to see where the most fruitful area of scrutiny for future performance improvements is located.

IMPROVING JAVA PERFORMANCE USING AN EXAMPLE APPLICATION: CXREFF/JXREFF

Cxreff.cpp is a sample application consisting of 330 source lines of C++. The program reads the file(s) named on the command line parameters, and produces a line-numbered listing of the lines of each file preceded by the file name, with indication as to the level of curly bracket nesting ({,}) whenever that changes. Finally, a cross-reference listing of all symbols encountered is produced, including line numbers in the program in which they were encountered. The output shown in Exhibit 1 is taken from a report produced by cxreff.cpp.

The same program was also implemented in Java and stored in the file jxreff.java. By comparing the results of executing cxreff.cpp and jxreff.java, one can see how closely C/C++ execution performance is to Java, as well as study the process of improving Java execution performance.

The programs were executed using an input file of 326 kilobytes of source text. The Java program was compiled with Microsoft Visual J++ 6.0 into .class files, and executed with the Visual J++ JVM, the Sun JDK 2.0 JVM, and the VisualCafe 3.0 JVM. The J++ compiler can also create an executable file (jxreff.exe). However, experiments indicate that this executable file does not execute any faster than executing the class files under the J++ JVM.

Other experiments indicated that programs compiled by the Sun JDK compiler and the Visual Café compiler generate the same execution performance as those compiled by J++. Thus, the main focus of performance is on the JVM, and the experiments demonstrate that choosing the JVM to execute the .class files is more important than the choice of Java compiler. The main interest in all cases is the execution speed of the program with the specific JVM.

Rather than simply showing execution times, a performance factor was computed using the execution time of cxreff.cpp (the C++ version of the program) as a base. The performance factor was produced by measuring the execution time of cxreff.cpp and dividing that into the execution times of the various Java programs to produce the performance factor. Thus, the execution time for cxreff.cpp was 12 seconds, while the initial execution time for jxreff.java using the executable (.exe) program produced by Microsoft Visual C++ 5.0 was 642 seconds. Therefore, the performance factor was 53.5. Obviously, the lower the ratio, the better the performance.

Base Version: The Original Performance Results

cxreff.cpp using Microsoft Visual C++ 5.0:	12 seconds;	the original program
cxreffMap.cpp using Microsoft Visual C++ 5.0:	7 seconds;	same as the program above but uses the C++ STL **map** hashtable mechanism to improve performance

Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 53.5 Java/C++
Microsoft Visual J++ 6.0 JVM using .class: 53.5 Java/C++
Sun JDK 2.0 JVM using .class: 32.6 Java/C++
Symantec Visual Café 3.0 JVM using .class: 46.9 Java/C++

These results are very disturbing. C/C++ programs are compiled into native code and executed directly, so they are expected to be as fast as one can get. Java, on the other hand, is executed by the JVM by interpreting the Java machine instructions. A degradation of a factor of 10 is expected for interpretation, but a factor of 53.5 is very high. Note, however, that the Sun JDK 2.0 has a much lower performance factor at 32.6. Nevertheless, a performance factor of 32.6 is still too high in many cases. What can be done? Knowledgeable programmers have been able to improve the performance of any program using some well-known techniques (see References 1 through 8), and these techniques have been applied to the Java programs.

Steps in Improving the Performance of jxreff.java

1. Try Collapsing the Most Frequently Executed Method in the Application. Looking at the jxreff program and the results from the profiler, the most frequently executed method in jxreff.java is tell_identifier, which is called for every character input. Since this tell_identifier method is called from only one place in find_sym, it was collapsed into the find_sym method, thereby saving the call overhead and the return. The results are puzzling because they show some improvement for J++ and the Sun JDK 2.0, and show a substantial increase in execution time for Visual Café 3.0.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 52.1 Java/C++
Sun JDK 2.0 JVM using .class: 31.8 Java/C++
Symantec Visual Café 3.0 JVM using .class: 60.2 Java/C++

Note that the profiler output in Exhibit 2 indicates that, by far, the largest amount of time is spent in the Java system classes for String, StringBuffer, and Character, which are called from the jxreff application. How these system library classes are used needs to be examined and improved.

2. Symbol Table Comparisons Too Slow: Call C++ strcmpi Function from Java Using Java Native Interface. Looking at the jxreff program and the results from the profiler in Exhibit 2, a great deal of time seemed to be spent on comparing symbols in a case-insensitive manner using the toLowerCase methods for both operands. Since C++ has a good case-insensitive string compare, the Java code was interfaced to call the C++ strcmpi subroutine. The results were disastrous for the J++ program, which aborted. The Sun JDK 2.0 had a slight improvement, while Visual Café 3.0 had a substantial one.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: aborted
Sun JDK 2.0 JVM using .class: 30.2 Java/C++
Symantec Visual Café 3.0 JVM using .class: 37.5 Java/C++

3. Symbol Table Comparisons Still Too Slow: Try to Use Collator Class Instead. The above was not satisfactory for two of the three platforms, so another alternative was sought. The problem was still the same: a great deal of time seemed to be spent comparing symbols in a case-insensitive manner using the toLowerCase methods for both operands. This was modified to use the Collator class provided by Java using PRIMARY strength for case-insensitive and NO_DECOMPOSITION for efficiency. Also, the collation keys were used to order the symbols in the symbol table. Compare the results to the original ones in the base version, and then to step 2 above. These changes improved the performance dramatically.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 14.4 Java/C++
Sun JDK 2.0 JVM using .class: 2.7 Java/C++
Symantec Visual Café 3.0 JVM using .class: 3.5 Java/C++

4. Symbol Table Comparisons Still Too Slow: Minimize Conversions to Lower Case. The profiler still indicated that, as in the original program profile in Exhibit 2, String.toLowerCase used the largest amount of computer time. To reduce converting symbols to lower case for the comparison (using String.toLowerCase method), the symbols in the symbol table were all stored in lower case, and new symbols just scanned were immediately converted to lower case using a quick method that subtracted 32 from uppercase symbols. The performance results show J++ a little slower, but Sun and Visual Café improved further and significantly.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 16.7 Java/C++
Sun JDK 2.0 JVM using .class: 1.6 Java/C++
Symantec Visual Café 3.0 JVM using .class: 1.7 Java/C++

5. Symbol Table Comparisons Still Too Slow: Eliminate toLowerCase Using a Mapping Table. This version of jxreff tried to improve efficiency by creating a mapping table to convert uppercase characters to lowercase characters. The table provides a lowercase value simply by indexing into it with the original value of the character. This is used to make comparison strings case neutral prior to using the String class compare to method, and eliminates the need to use the toLowerCase method of the String class. This change brought a modest improvement for J++ and little change for the other two.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 13.2 Java/C++
Sun JDK 2.0 JVM using .class: 1.5 Java/C++
Symantec Visual Café 3.0 JVM using .class: 1.7 Java/C++

6. Reading Input File Too Slow: Use ReadFully Method to Read the Whole File in One Read. The profiler now indicated that much time was spent in reading the input file in cross_referencer.get_line. The code was modified to read the input file in one I/O call using Java's ReadFully I/O method. This caused practically no change for any of the systems, probably because input buffering was already adequate in each JVM.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 13.1 Java/C++
Sun JDK 2.0 JVM using .class: 1.5 Java/C++
Symantec Visual Café 3.0 JVM using .class: 1.7 Java/C++

7. String Comparison Too Slow: Write Own Case-Insensitive String Compare Using Character Arrays. Much time was still spent in the string compare routine, so the use of the String class in the application was eliminated in favor of char arrays. A Java case-insensitive string compare function was written using char arrays rather than Strings. This caused a very substantial reduction in execution time in every platform, and huge reduction in J++.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 1.5 Java/C++
Sun JDK 2.0 JVM using .class: 1.2 Java/C++
Symantec Visual Café 3.0 JVM using .class: 1.3 Java/C++

8. Linked-List Processing Too Slow: Use Java HashTable Class. This application is still about symbol table management, so the Java HashTable collection class was used rather than the previous linked list of symbols. The results are dramatic. Java is faster than the base C++ program by a significant 40 to 50 percent for all but J++, which is still 20 percent slower than C++.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 1.2 Java/C++
Sun JDK 2.0 JVM using .class: 0.5 Java/C++
Symantec Visual Café 3.0 JVM using .class: 0.6 Java/C++

This Java version should also be compared to the cxreffMap.cpp, which, like the Java HashTable, speeds up symbol table access using hashing. Even with this tougher comparison, the performance of Java is faster than cxreffMap.cpp for both the Sun JVM and Visual Café; however, it is still twice as slow using the J++ JVM.

New Performance Factor: jxreff/cxreffMap:
Microsoft Visual J++ 6.0 JVM using .EXE: 2.0 Java/C++
Sun JDK 2.0 JVM using .class: 0.9 Java/C++
Symantec Visual Café 3.0 JVM using .class: 1.0 Java/C++

9. Symbol Table Processing Still Slow: Use Java TreeSort Class. The Java TreeSort class was used to replace the previous linked list of symbols (instead of the Java HashTable class used in step 8 above). The results are even more dramatic than in step 8 and show that Java is faster than C++ by a significant 25 to 60 percent.

New Performance Factor: jxreff/cxreff:
Microsoft Visual J++ 6.0 JVM using .EXE: 0.7 Java/C++
Sun JDK 2.0 JVM using .class: 0.4 Java/C++
Symantec Visual Café 3.0 JVM using .class: 0.5 Java/C++

This Java version should also be compared to the cxreffMap.cpp which, like the Java TreeSort, speeds up symbol table access using hashing. Even with this tougher comparison, the performance of Java is faster than cxreffMap.cpp using the Sun JVM and the Visual Café, but 30 percent slower using the J++ JVM.

New Performance Factor: jxreff/cxreffMap:
Microsoft Visual J++ 6.0 JVM using .EXE: 1.3 Java/C++
Sun JDK 2.0 JVM using .class: 0.7 Java/C++
Symantec Visual Café 3.0 JVM using .class: 0.9 Java/C++

More Improvements If Needed. The goal of obtaining a performance better than the base C++ program has now been reached. One could, however, continue with the process and improve the Java program still further. Any program that runs so much faster than the original C++ would be very acceptable to most developers and their managers. A summary of the results from the above experiments is shown in Exhibit 3.

Potential For Java Performance Improvements Is Much Greater Than C++

The results demonstrate the great potential for execution-time performance improvements of Java programs. The reader may well ask, "Could one have done the same with the C++ cxreff.cpp program?"

The answer is certainly, "Yes." However, the C++ program was written to be as efficient as possible, and C++ code generation is already very good. Thus, cxreff.cpp is a good comparison vehicle. Note also that making Java improvements often has more dramatic results because the underlying interpretation by the JVM (or execution time just-in-time compilation

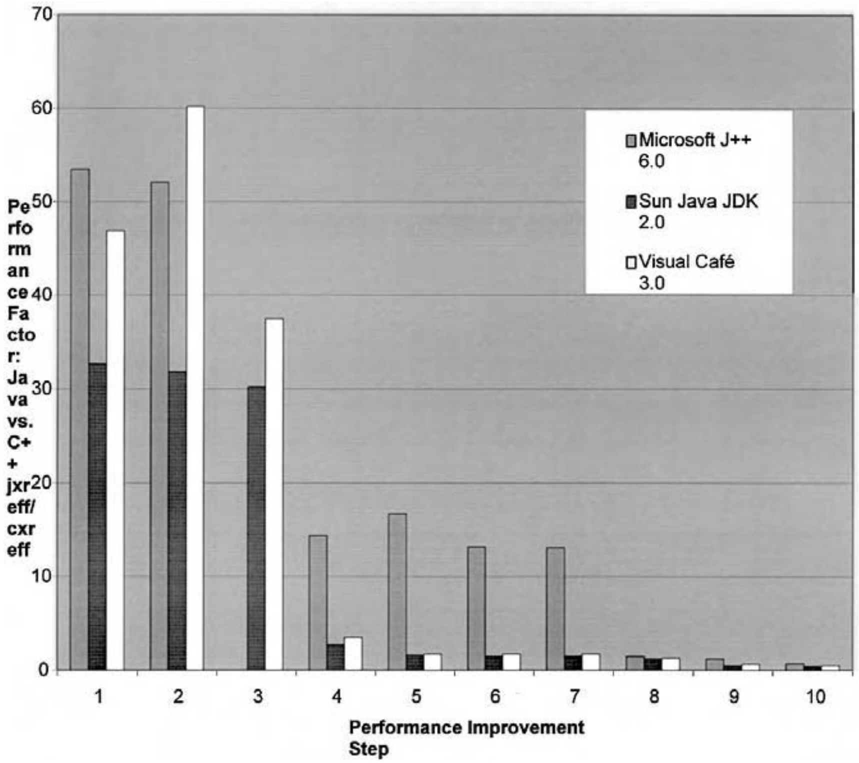


Exhibit 3. Performance Improvement of jxreff Java Application

results) is inherently less optimized than C++ code. So, in some sense, Java code is ripe for the performance pickings.

CONCLUSION

Java is a programming language with a great deal of promise. Computer science is still in its infancy, so we all have to get used to learning new tools on a continuous basis. Java is certainly not the ultimate tool, but it is one that is far superior to many previous tools. Java programs may be slower than C/C++, but speed of execution is not the major concern in a majority of applications. Furthermore, Jalics[1] has shown through a series of measurements that Java performance can be on the same order of magnitude as C/C++.

In this chapter, the results are extended to show that in many cases Java programs can be as fast or faster than corresponding C++ programs if the programmer spends some effort in improving that performance.

Thus, in this case study, the Java application originally took 5250 percent more time to execute than C++, but after the improvements, the same program took 30 percent less time than the corresponding C++ program. Also, note that for a better optimizing JVM (Sun JDK 2.0), Java initially took only 3160 percent more time than C++, but the improved program took 60 percent less time than C++.

The choice of the Java Virtual Machine is critical to Java program performance. The Microsoft JVM took from 1.25 to 10.44 times as long as the Sun JVM to execute the same program.

Programmer choices of Java language features have a tremendous impact on program performance. Calling C/C++ procedures from Java (using the Java Native Interface [JNI]) is relatively slow and should only be attempted for procedures that do a lot of work in a single call.

Some Java systems have program profilers available. These can pinpoint the most critical sections of Java code in a program, and the user can choose from a variety of techniques to improve these critical sections of code.

There might also be the possibility of an improved implementation in the future of some Java system classes like String, StringBuffer, and Character, which could improve the performance of many programs. This would likely have to be left to the Java developers because these classes are at the core of Java and are used in the Java system software and therefore are not easily changed.

While most of the performance improvement techniques are similar to ones that can be used with C++, Java also has some advanced language features like the Collator class, HashTable class, etc. that can bring better performance.

This case study program has demonstrated that Java applications may be incredibly slower than corresponding C++ ones when the programmer does not pay attention to performance factors when the program is written. However, through the use of Java profilers and by insights into Java performance characteristics, one can easily identify the critical sections of a given program and improve their performance substantially. One can experiment with alternate data structures, advanced language features and library routines, alternate algorithms and data types, etc. All of these techniques take expertise, time, and patience, however.

ACKNOWLEDGMENTS

This project started as a research course (CIS 695) for graduate student William Zhukovsky at the Computer & Information Science department of Cleveland State University. Mr. Zhukovsky was tireless in his efforts to

learn about performance improvement techniques and specifically to squeeze more and more performance out of the case study programs.

References

1. Jalics, P. and Misra, S., Java and C++: Similarities, Differences, and Performance, *Systems Development Management,* 21(6), February 1998.
2. Jalics, P. and Misra, S., Performance Improvement Techniques, *Systems Development Management,* 21(6), December 1996.
3. Jalics, P. and Misra, S., Measuring Program Performance, *Systems Development Management,* 21(6), December 1996.
4. Jalics, P. and Blake, B., Benchmarking C++ Performance, *Systems Development Management,* 21(3), June 1996.
5. Jalics, P. and Blake, B., Performance of Object-Oriented Programs: C++, *Systems Development Management,* 21(3), June 1996.
6. Jalics, P. and Blake, B., An Assessment of Object-Oriented Methods in C++, *Journal of Object-Oriented Programming,* May 1996.
7. Jalics, P. and Misra, S., Improving C Program Performance: Techniques and Examples, *Handbook of Systems Management and Support,* Auerbach Publishers, June 1993.
8. Performance Evaluation of Computer Systems, *Macmillan Encyclopedia of Computers,* Macmillan Publishing Co., 2, 1991, pp. 764–769.

Chapter 55
Optimizing Web Design and Database Performance

Srinivas Padmanabharao

The last few years have seen an explosion in the number of Web sites that can be surfed by users, as well as a correspondingly large increase in the business volumes that are conducted over the Internet. Many factors affect the end-user experience, including the speed of the connection, the nature of the transaction, the appearance of the screen, and content that stretches beyond the type of business conducted on a Web site. Out of this list of factors, Web site performance is increasingly proving to be a key competitive advantage (or disadvantage) that firms can use to win in the marketplace.

Most Web sites can usually be looked upon as consisting of two main components, which are: (i) the front end that the user interacts with (e.g. Web pages); and (ii) everything else that includes business components and data repositories that power the Web site. It is vital that businesses pay due attention to both components in order to improve the end-user experience and the performance of the application.

This work examines some of the issues that help determine the full end-user experience. It also explores some of the strategies that businesses can employ to improve Web site design and database performance. It then reviews some tools that are available in the market to evaluate the performance of Web sites.

FACTORS INFLUENCING THE END USER EXPERIENCE

This section outlines some factors that go into determining the full, integrated end-user experience.

0-8493-1161-6/02/$0.00+$1.50

Content

Content is undoubtedly one of the most important factors influencing end-user satisfaction with a Web site. For example, it can be an extremely frustrating experience for an end user to discover erroneous information shown on a Web page. There are three Web content qualities that must be met, namely: accuracy, appropriateness and scope.

Accuracy involves the presentation of error-free information that is current and updated as frequently as necessary. While it may serve as a medium of advertising for the company's products and services it must try and present a balanced and objective view, especially at Web sites that are supposed to act as sources of information.

Appropriateness of a Web site involves the use of concepts and information that is relevant to the target group of users. Use of appropriate language and content is especially important given that children are increasing becoming consumers of the Web. Only information relevant to the business objective of the organization and supporting information must be presented on a Web site.

Scope involves the presentment of information that is sufficient in scope to adequately cover the topic or message for the intended audience. This could be for the purchasing of goods and services or for the dissemination of information. Companies can enhance the value of their Web sites by presenting information that is not easily attainable from other sources.

Technical Competence

The technical competence of a Web site directly impacts the end-user experience. This involves navigational and presentation approaches.

A Web site must be intuitive and easy to navigate. It must not be overloaded with graphics and must have reasonable download times. It must be complete with no broken links and must adhere to standard formats. It must provide logical options for printing and downloading selected components such as white papers and product information.

Presentation involves leveraging good graphic design principles where the use of images and multimedia is functional and not merely decorative. Appropriate text size, uncluttered screen displays and captions, labels or legends for all visuals must be used. Information must be presented in a manner to stimulate imagination and curiosity. At the same time product advertising must not be intrusive to users.

Trust

A Web site must be able to gain the confidence of the consumer and establish trust as a basis for all transactions. Hackers that hit Web sites with too many messages or erroneous data heighten security concerns for all users of the Web. Hence for companies that intend to do business on the Web this issue is of key concern. Companies must publish and adhere to their privacy policy on the Web. They must not solicit any information from the user that is not directly relevant to complete the business transaction. This information must not be used to support e-mail or other Web marketing campaigns without the permission of the users. Web sites must also employ adequate technology to prevent the misuse of information collected from users.

WEB SITE DESIGN CONSIDERATIONS

A Web site is not just a company's address on the Web. It is a home. It is the only "face" of the company that an online consumer may ever see. Hence the importance of good design in putting up a Web site cannot be underestimated. This section reviews some of the key points that companies should consider while designing their "homes' on the Web."

The Business Objective

The Web site must be a part of an overall corporate strategy to acquire and retain customers. In this respect it is important to clearly establish the business objective for building a Web site before one goes about designing it. This has a direct and immediate impact on the content of a Web site. However, it also impacts factors such as technology used, investment made, and the time to completion.

Companies could choose to make their Web-sites the primary modes of information dissemination, e.g., Autobytel acts as a single source of information on all models of cars. Other companies could make them the primary modes of transacting business and generating revenue, e.g., Amazon wants the customer to complete the purchase of the books online. Companies could use the Web site as a launch pad to generate excitement about a new product or service they are planning to introduce and use it to gauge the market potential of the service or product.

Ergonomics

The issue of ergonomics in Web site design is becoming increasingly important. This involves a company paying due attention to factors such as the visual appearance of the Web pages, the presentation of content in the Web pages through the use of text/images/video and audio, the ease with which a user can navigate around the Web site and how intuitively the Web site is organized. Companies will also have to pay attention to the task of developing and deploying a whole variety of Web site aids. The most basic of these aid tools includes the "Site Map" and "Search" functions. However companies need to go beyond this and use this as an opportunity to build a more interactive experience with the user through more Web tools. Examples of such tools include – Banks providing mortgage calculators online and Amazon automatically providing you with a list of books that users who bought the book you are interested in also bought themselves.

Size of the Site

The size of a Web site must be determined by various factors and issues such as the business objective, the expected number of customers/the volume of traffic the Web site is expected to generate, and the number of transactions expected (if so designed). While deciding on the size of the Web site it is also important to evaluate the content of your Web site and appropriately size the servers and other hardware that is required to run the Web site. The overall systems infrastructure is crucial to the overall performance of the Web site. Due attention must also be paid to the bandwidth needs of the Web site and that an appropriate hosting solution chosen (e.g. either in-house or through a specialized vendor).

Investment in Building the Site

Gone are the days when companies could quickly put up a couple of Web pages and claim to have become an E-business. The whole act of putting up a quality Web site requires a significant investment of time and money. Companies are increasingly putting senior management in direct charge of such activities in order to demonstrate their commitment to a strong product. Companies also have to evaluate whether they already have the content required to build the Web site or if it needs to be developed from scratch. These are separate from the obvious costs involved in hiring programmers, building the infrastructure to host the site (or an agreement with a Web hosting service), software tools needed to build and power the Web site like Web servers, security mechanisms (firewalls), and billing systems. In addition, companies will also need to invest in integrating the Web site with their existing back office systems to achieve true seamless business integration.

Investment in Maintaining the Site

A static Web site is a sign of a dead or stagnant business. Web sites need to be constantly monitored, reviewed, and updated with fresh content on a regular basis. While the frequency of this varies based on the nature of the site (ESPN may need to update it many times a day while FORD may only revise it once a month) the need to plan for this and incorporate this into the business process, as determined by the business objective, cannot be overly emphasized. Ongoing maintenance can also include training requirements for the staff maintaining the Web sites, cost of upgrading and maintaining the infrastructure including the hardware and software, and the cost of communications. Such recurring expenditures must be included in the budget.

DATABASE REQUIREMENTS IN E-BUSINESS

A well-designed Web site offers a good start for attracting potential customers to your business. Most commercial sites handle volumes of information that need the use of a database at the back end to both hold corporate information and to track customer-related information. This places enormous demands on the database servers to support applications that have an unpredictable number of transactions, potentially large data flow, and high performance requirements. All these factors taken together are unprecedented.

Availability

While availability of the database server and its "up-time" has always been a key concern for IS managers at corporations, the 365 × 24 × 7 nature of the Web places unprecedented demands. There cannot be any *"scheduled maintenance downtimes,"* let alone any unplanned outages. Competitors are only a click away and a customer who cannot use your Web site is a lost customer.

Security

Web sites are open around the clock and are also open to users from all over the world. All information needed by customers must be accessible via the Web site. This presents a security nightmare from the IS managers viewpoint. These requirements can be an open invitation to hackers and people with malicious intent to come in and create havoc at your Web site. Databases, containing the core value of your business, must be protected from such elements. Adequate security architecture must be build across the key elements of the database server. If the data is spread across multiple instances of the database then critical customer and business information, like credit cards and financial details, need extra security when compared to data that is not quite so strategic (e.g., product catalogs).

Scalability

In the earlier days of databases corporate IS managers had a pretty good handle on issues like number of users, number and type of transactions/queries and rate of data growth. However in the world of E-business these are all variables which are difficult to predict with accuracy. There could be millions of concurrent users on the Internet. In addition the IS manager does not have the option of balancing the load across over the day and the night. Hence the scalability of the database and its ability to manage the varying load patterns across time is crucial to ensuring acceptable performance levels from an end-user perspective.

Integration

The use of the Internet for commercial transactions requires that databases be able to talk to other databases to exchange information. This means that databases may soon be required to send, receive and store a messaging mechanism that is standardized across the Web. The Extensible Markup Language (XML) is becoming the communication language of the Web. Hence databases may soon be required to process XML statements.

Huge Amounts of Data Processed in Real Time

There is an increasing trend toward storing all the user actions occurring in a Web site in a data warehouse where it is then analyzed and mined to better understand customer behavior, customer and product profitability, and other E-business issues. Such data can grow exponentially and the overall size of the data warehouses soon becomes a difficult issue to deal with. E-business is also focused on a faster moving world. The information flowing into the data warehouse, both continuously and in large volumes, must be exploited more or less immediately, and then be relayed to operational systems for further processing.

STRATEGIES TO IMPROVE DATABASE PERFORMANCE

E-business poses significant new challenges to IS managers. This section examines some options that can be used to improve database performance.

Architecture

In order to improve the database performance from the end-user perspective the first issue that IS managers may need to address is the overall IT application architecture. In order to achieve and address the twin challenges of scalability and performance, it is a good idea to explore the use of a three-tier architecture involving the use of a middleware product.

From a database perspective E-businesses tend to drive the centralization of data. However the use of multiple, distributed databases should be explored. The best approach may turn out to be to distribute the data across multiple databases based on an analysis of the nature of transactions being carried out against the entire spectrum of data. If most visitors to the Web site just want details on products and services offered by the business then it may be appropriate to dedicate a database to this and hold customer demographic information in a separate database.

Sizing

The size of databases is constantly increasing. These days E-business databases will routinely begin to approach the terabyte range and enter into the realm of what are commonly called "Very Large Databases." This poses two challenges. The first is to be able to estimate the rate of growth and ensure that appropriate processes to manage this volume of data are put in place. This may involve regular archiving of information into a backup data warehouse while keeping the "live" database within manageable limits. The second challenge is to determine and obtain the appropriate hardware for the size of the database expected. Significant investments in hardware to help improve performance must not be ruled out and all limits from a hardware perspective, such as increasing the RAM or number of processors, must be explored.

Data Organization

The organization of the data within the database is the most important factor influencing the performance of the database. This involves building a list of data requirements, translating the same into a data model, normalizing the data model and implementing the normalized structure in the database. However it must be cautioned that non-normalized databases are found in commercial applications and this may actually enhance performance as per the application requirements. Another key factor that may need to be considered is the skew in the data that one expects to store. In a relational database, data is "skewed" when the distinct values in a column are not uniformly distributed over the table's rows. Expected instances of skew in data must be duly considered while designing SQL queries and creating indexes.

Database Design

Finally the traditional issues that must be considered during database design must not be ignored. These include but are not limited to the following:

- *Simplicity.* An attempt must be made to keep the queries focused and simple. Many tasks may be better accomplished at the client end rather than at the server side.
- *Get rid of useless indexes.* While having appropriate indexes has always been emphasized, lately more emphasis has been placed on removing (or concatenating the other way round) indexes which may not be used.
- *Pay attention to nested scans.* On an enormous table, indexes are rarely lightweight themselves, even if they seem small by comparison. A full index scan, and even a range scan can significant performance implications.

EVALUATING WEB SITE PERFORMANCE

It is becoming increasingly clear that companies will need to be able to constantly monitor the performance of their Web sites and fine tune all aspects of the Web site to ensure a better end-user experience. This section reviews a couple of such tools, but does not provide a recommendation on which tool to actually use. This is an assessment that you must do for your own unique situation.

FireHunter

This is a product from Agilent technologies, a subsidiary of Hewlett Packard corporation. The Firehunter family of products provides E-businesses with critical capabilities for proactive management of the performance and availability of business-critical services. It supports the monitoring of service-levels and can be used to maintain a strict control over the end user experience. These powerful capabilities can help a business gain immediate revenue increases due to increased control over the business. It can be used to monitor, manage and report on your basic Internet services, such as mail, news and Web, as well as your value-added services, such as Web-hosting and E-commerce.

Jyra

Jyra provides companies with an E-commerce performance assessment solution that can access, browse, and log onto sites on the Internet in the same way as any live user or E-commerce customer, thus simulating an end-user experience. Jyra can provide a graphical representation of customer's experience when purchasing from a site displaying what the customer experience is and management reports pointing out real peaks and lost business due to poor performance. Jyra can also be used to gather information the most immediate cause of poor performance, be it network, load balancing systems, firewalls, persistent connection failures, server capacity, or bandwidth capacity.

CONCLUSION

The rapid growth of E-business is posing significant challenges to IS managers in providing acceptable levels of performance from their Web applications. Adopting a holistic approach that involves examining both the front and back ends (and everything in between) of a Web-based application will help IS managers achieve their objectives. Businesses will need to adapt their infrastructure to successfully compete in this environment. Firms that can provide their customers with superior performance will succeed in the next millennium.

ABOUT THE AUTHOR

Srinivas Padmanabharao is a consultant with Deloitte Consulting. He specializes in E-business solutions and technical architecture.

Section IX
Advanced Topics

The chapters in this section explore some advanced Internet topics:

"Cellular Digital Packet Data: An Emerging Mobile Network Service" (Chapter 56) examines how the CDPD can support the needs of the mobile workforce. This is a method of transmitting data over cellular voice networks.

"TN3270 and TN5250 Internet Standards" (Chapter 57) focuses on providing key Internet standards activity for integration of host SNA terminal applications.

"Knowledge Portal Classification Technology: Mapping Knowledge Landscapes" (Chapter 58) discusses how to create searchable category headers — taxonomies — that are used to navigate enterprise repositories in diverse applications including those available over the Web. The chapter examines a suite of tools, Corporate Portal Categorization Solutions, to determine how these types of architectures can supply knowledge workers with a map to the knowledge landscapes.

Chapter 56
Cellular Digital Packet Data: An Emerging Mobile Network Service

Nathan J. Muller

Cellular digital packet data (CDPD) is a data-over-cellular standard for providing LAN-like service over cellular voice networks. CDPD employs digital modulation and signal processing techniques, but it is still an analog transmission. The CDPD infrastructure employs existing cellular systems to access a backbone router network that uses the IP to transport user data. Personal digital assistants, palmtops, and laptops running applications that use IP can connect to the CDPD service and gain access to other mobile computer users or to corporate computing resources that rely on wireline connections.

Because CDPD leverages the existing $20 billion investment in the cellular infrastructure, carriers can economically support data applications and avoid the cost of implementing a completely new network, as most competing technologies would require. CDPD also offers a transmission rate that is four times faster than most competing wide area wireless services, which are limited to 4.8K b/s or lower.

CDPD FUNDAMENTALS

Unlike circuit-switched schemes, which use dialup modems to access the cellular network, CDPD is a packet-switched technology that relies on wireless modems to send data at a raw speed of 19.2K b/s. Although CDPD piggybacks on top of the cellular voice infrastructure, it does not suffer from the 3-KHz limit on voice transmissions. Instead, it uses the entire 30-KHz radio frequency channel during idle times between voice calls.

0-8493-1161-6/02/$0.00+$1.50

Using the entire channel contributes to CDPD's faster and more reliable data transmission.

Underlying Technologies

CDPD is in fact a blend of digital data transmission, radio technology, packetization, channel hopping, and packet switching. This technology lets the cellular network carry the 1s and 0s of binary digital code more reliably than is usually possible over cellular voice networks.

Digital Transmission Technology. Digital transmission technology is reliable and more resistant to radio interference than analog transmission technology. The digital signals are broken down into a finite set of bits, rather than transmitted in a continuous waveform. When signal corruption occurs, error-detection logic at the receiving end can reconstruct the corrupted digital signal using error correction algorithms. Digital technology also enables processing techniques that compensate for signal fades without requiring any increase in power.

Digital Cellular Radio Technology. Digital cellular radio technology is used for transmitting data between the user's mobile unit and the carrier's base station.

Packetization. Packetization divides the data into discrete packets of information before transmission. This approach is commonly used in wide area and local computer networks. In addition to addressing information, each packet includes information that allows the data to be reassembled in the proper order at the receiving end and corrected if necessary.

Channel Hopping. Channel hopping automatically searches out idle channel times between cellular voice calls. Packets of data select available cellular channels and go out in short bursts without interfering with voice communications. Alternatively, cellular carriers may also dedicate voice channels for CDPD traffic.

Packet Switching. Packet switching, using the IP, accepts data packets from multiple users at many different cell sites and routes them to the next appropriate router on the network.

APPLICATIONS FOR CDPD

The wireless-industry consortium that funded the development of the CDPD specification includes Ameritech Cellular, Bell Atlantic Mobile, Contel Cellular Inc., GTE Mobilnet, Inc., McCaw Cellular Communications, Inc., NYNEX Mobile Communications, AirTouch (formerly PacTel Cellular), and Southwestern Bell Mobile Systems. Three principles guided their efforts: that emerging CDPD recommendations could be deployed rapidly, economically, and in conjunction with technology already available in the marketplace.

More specifically, the consortium's stated objectives include:

- Ensuring compatibility with existing data networks
- Supporting multiple network protocols
- Exerting minimum impact on end systems; existing applications should operate with little or no modification
- Preserving vendor independence
- Ensuring interoperability among service providers without compromising their ability to differentiate offerings with service and feature enhancements
- Allowing subscribers to roam between serving areas
- Protecting subscribers from eavesdropping

Emerging Class of Remote Users. CDPD allows traditional wireline networks to reach a new class of remote user: the roaming mobile client. With the establishment of a wireless link to the cellular carrier's CDPD network, remote users can operate their terminals as if they were located on the desktop in a branch office. Mobile workers, for example, can regain much of the productivity they lose while away being from their main offices by using CDPD to send and receive e-mail from computers or personal digital assistants.

Another application example is a debit card. Commuters could purchase a debit card to run through a card-reading device on a bus or another transit system and the fare would be deducted automatically from the card's total. That fare information could be transmitted to a central processing center in less than a second for just a few cents. CDPD could also be used by service providers to monitor and control devices such as traffic lights, alarm systems, kiosks, vending machines, and automated teller machines.

Service Pricing. As an overlay to the existing analog cellular infrastructure, CDPD networks are easy and economical for carriers to set up and operate. Carriers estimate that it costs only five percent over the initial cost of a cell site to upgrade to CDPD. Cell sites typically cost about $1 million to set up, including the cost of real estate.

Users are the beneficiaries of CDPD's resulting economies and efficiencies. For many applications, initial CDPD service pricing is competitive with that of the proprietary analog wireless services of ARDIS and RAM Mobile Data. Exhibit 1 indicates thatCDPD is best suited for transaction-oriented applications. Although these services might prove too expensive for heavy database access, the use of intelligent agents can cut costs by minimizing connection time.

Exhibit 1. CDPD Services

	Per-User Pricing	
Application	**Bell Atlantic Mobile Systems**	**GTE Mobilnet**
Database Inquiry	$23 to $27/month	$20 to $28/month, 25 sessions a day, 5 days a week
Electronic Mail	$40 to $60/month	$45 to $60/month, 14 messages a day, 5 days a week
Dispatch	$13 to $17/month	$10 to $20/month, 1 to 2 jobs per hour, 9 hours a day, 5 days a week
Alarm Monitoring	$13 to $17/month	$10 to $20/month, 1 transaction per hour, 24 hours a day, 7 days a week
Field Service	$23 to $27/month	$16 to $22/month, 20 transactions a day, 5 days a week

Note: Estimated per-user prices are based on sample applications and usage figures. All prices subject to change without prior notice.

Benefits to Mobile Users. Because CDPD uses the existing voice-oriented cellular network and off-the-shelf hardware for implementation, it is cost-effective. There are, however, additional benefits to users besides economy. These benefits include:

- *Efficiency.* CDPD transmits both voice conversations and data messages using the same cellular equipment. Using a single device, it is a versatile and efficient way to communicate. The digital data does not disrupt or degrade voice traffic, and vice versa.
- *Speed.* With a maximum channel speed of 19.2K b/s — a four-fold increase over competing mobile radio technologies — CDPD is the fastest wireless technology available on the WAN.
- *Security.* With encryption and authentication procedures built into the specification, CDPD offers more robust security than any other native wireless data transmission method, preventing casual eavesdropping. As with wireline networks, users can also customize their own end-to-end security.
- *Openness.* Because CDPD is an open, nonproprietary standard, it promotes low equipment costs and broad availability of hardware and software.
- *Flexibility.* Because CDPD units use existing cellular radio technology, they are capable of transmitting data over both packet- and circuit-switched networks, allowing applications to use the best method of communication.
- *Reliability.* Because CDPD uses existing equipment on the network (i.e., routers), as well as time-tested protocols based on TCP/IP, the

highest quality of wireless data service is assured. CDPD also provides excellent penetration within buildings.

- *Worldwide Reach.* CDPD can be used in conjunction with existing cellular systems around the world. These systems already serve 85 percent of the world's cellular users.

Because CDPD allows the network to operate more efficiently by providing digital packet data over the voice network, carriers also realize maximum flexibility, simplified operations and maintenance, and cost savings. Carriers can offer enhanced messaging services such as multicast, cellular paging, and national short-text messaging. CDPD allows portable access to a variety of information services.

In effect, CDPD extends client/server-based applications from the LAN environment into the wireless arena. This extension provides nearly limitless possibilities for future wireless data services.

EQUIPMENT REQUIREMENTS

CDPD is not without its problems. Even though CDPD takes advantage of the existing circuit cellular voice infrastructure to send data at up to 19.2K b/s, existing cellular modems cannot be used on CDPD-based networks. Modems designed for CDPD networks are still larger and more expensive than those designed for circuit cellular.

CDPD-only modems cost about $500; modems that handle both CDPD and circuit cellular run about $1,000. When the cost ofCDPD modems drops to the $200 range, expense will no longer be a barrier. Also, carriers are considering subsidizing the cost of CDPD modems, the way they currently do with cellular phones, when users sign up for service.

NETWORK ARCHITECTURE AND PROTOCOLS

The CDPD specification defines all the components and communications protocols necessary to support mobile communications.Exhibit 2 shows the main elements of a CDPD network.

Mobile Data-Intermediate Systems

The backbone router, also known as the Mobile Data-Intermediate System (MD-IS), uses the location information derived from the mobile network location protocol (MNLP) to route data to the mobile units, which are referred to as Mobile-End Systems (M-ES). Information on the link between the backbone router and an MDBS is transmitted using a data link layer (DLL) protocol. Communications on the other side of the backbone router are handled using internationally recognized protocols. This ensures that standard, off-the-shelf systems can be used in the network infrastructure

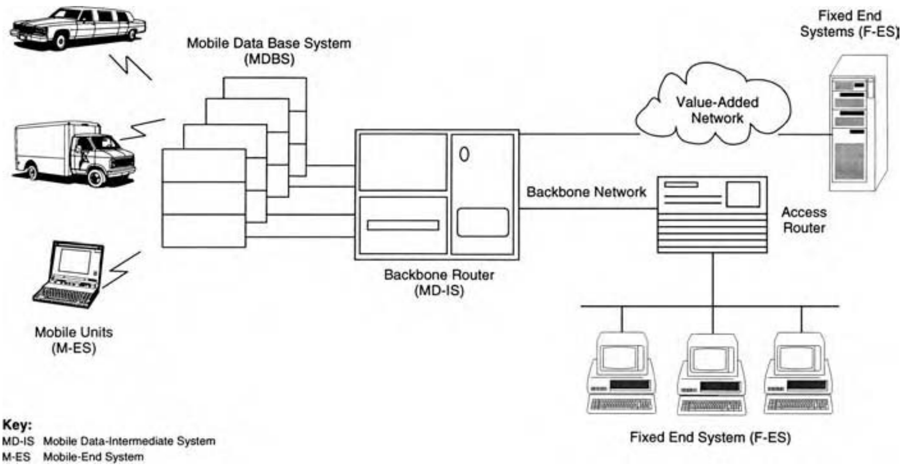


Exhibit 2. CDPD Network Architecture

and that computer systems currently in use can be accessed by CDPD networks without modification.

Mobile Database Systems

The mobile database system (MDBS) provides the relay between the cellular radio system and the digital data component of the CDPD network. The MDBS communicates with the mobile units through radio signals. Up to 16 mobile units in a sector can use the same cellular channel and communicate as if they were on a LAN. This communications technique is known as digital sense multiple access (DSMA). After the MDBS turns the cellular radio signal into digital data, it transmits the data stream to its backbone router, typically using frame relay, X.25, or the Point-to-Point Protocol.

Mobile-End Systems

Although the physical location of a M-ES, or mobile unit, may change as the user's location changes, continuous network access is maintained. The CDPD specification stipulates that there will be no changes to protocols above the network layer of the seven-layer Open Systems Interconnection (OSI) model, so that applications software will operate in the CDPD environment. At the network sublayer and below, mobile units and backbone routers cooperate to allow the equipment of mobile subscribers to move transparently from cell to cell, or roam from network to network. This mobility is accomplished transparently to the network layer and above.

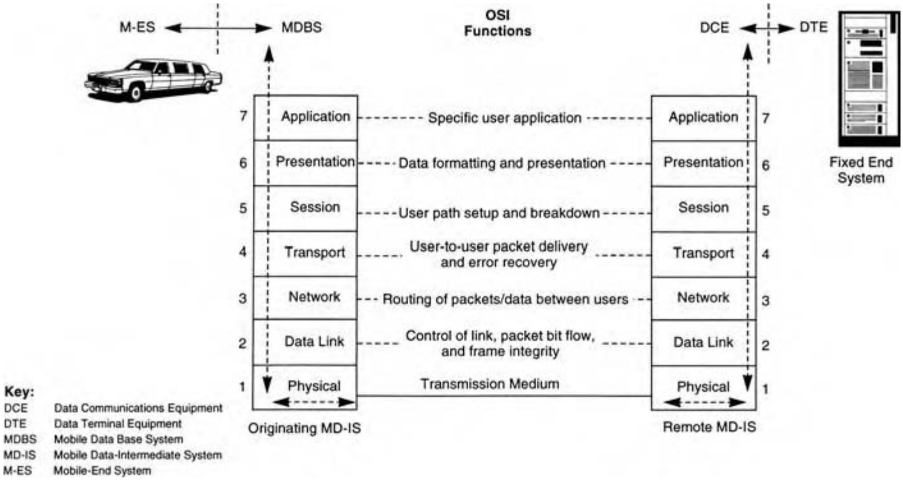

M-ES
MDBS
OSI Functions
DCE
DTE
7 Application - Specific user application - Application 7
6 Presentation - Data formatting and presentation - Presentation 6
5 Session - User path setup and breakdown - Session 5
4 Transport - User-to-user packet delivery and error recovery - Transport 4
3 Network - Routing of packets/data between users - Network 3
2 Data Link - Control of link, packet bit flow, and frame integrity - Data Link 2
1 Physical - Transmission Medium - Physical 1
Originating MD-IS
Remote MD-IS
Fixed End System
Key:
DCE Data Communications Equipment
DTE Data Terminal Equipment
MDBS Mobile Data Base System
MD-IS Mobile Data-Intermediate System
M-ES Mobile-End System

Exhibit 3. The OSI Reference Model

OSI Protocols. The recommendations of the CDPD consortium were designed using the OSI reference model (see Exhibit 3). The model not only provides a structure to the standardization process, but also offers recommendations regarding protocols available for use in the CDPD network.

Network Layer Protocols. The CDPD overlay network may use either the OSI Connectionless Network Protocol (CLNP) or IP at the network layer. These protocols have virtually the same functionality: Both interpret device names to route packets to remote locations.

IP has been used for more than ten years and is one of the most popular protocols today. Its inclusion in the CDPD specification is intended to accommodate the vast number of networked devices already using it.

Application Layer Protocols. Applications required to administer and control CDPD networks use OSI-defined protocols. OSI-defined application-layer protocols are widely accepted and have been tested to ensure robust, open communications among CDPD service providers. The use of these protocols provides a level playing field for manufacturers of the CDPD infrastructure equipment. Therefore, service providers can be confident that the various network elements will communicate together and that no single manufacturer can exert undue influence on the market.

Examples of OSI protocols that operate at the application layer and can be implemented for CDPD network administration and control are explained as follows:

- The Common Management Information Protocol is the object-oriented management standard for OSI networks developed by the International Standards Organization (ISO).
- The X.400 message handling system is a global messaging standard recommended by the International Telecommunications Union-Telecommunications Standards Section (ITU-TSS, formerly known as the International Telegraph and Telephone Consultative Committee) that defines an envelope, routing, and data format for sending e-mail between dissimilar systems.
- X.500 directory services are a standard for directory services recommended by the ITU/ISO that operate across multiple networks used to convey e-mail. It allows users to look up the e-mail addresses of other users they wish to communicate with.

MOBILITY MANAGEMENT

Traditionally, the network address of the end system has been used to determine the route used to reach that end system. CDPD is unique in allowing mobile units to roam freely, changing their subnetwork point of attachment at any time — even in midsession.

To find the best route for transmitting data to an end system, CDPD mobility management definitions describe the creation and maintenance of a location information database suitable for real-time discovery of mobile unit locations. Three network entities — the mobile units, the home backbone router, and the serving backbone router — participate in mobility management.

Mobile units are responsible for identifying their unique network equipment identifiers (NEIs) or network layer addresses to the CDPD network. As the mobile unit moves from cell to cell, it registers itself with the new serving backbone router. Each NEI is permanently associated with a home backbone router. The serving backbone router notifies the home backbone router of a mobile unit when it registers itself in the new serving area. Mobility management makes use of two protocols: the Mobile Network Registration Protocol (MNRP) and theMobile Network Location Protocol (MNLP).

Mobile Network Registration Protocol. MNRP is the method mobile units use to identify themselves to the network. This information is used to notify the network of the availability of one or more NEIs at a mobile unit. The registration procedure includes the information required by the network for authenticating the user's access rights.

MNRP is used whenever a mobile unit is initially powered up and when the mobile unit roams from cell to cell. In either case, the mobile unit automatically identifies itself to the backbone router so its location can be known at all times.

Mobile Network Location Protocol. MNLP is the protocol communicated between the mobile serving function and mobile home function of the backbone routers for the support of network layer mobility. MNLP uses the information exchanged in MNRP to facilitate the exchange of location and redirection information between backbone routers, as well as the forwarding and routing of messages to roaming mobile units.

INFORMATION PROTECTION

To facilitate the widespread acceptance of CDPD by cellular service providers, the specifications define methods for ensuring the security of customer information, while still providing an open environment for mobile users. Cellular service providers are legitimately concerned about protecting information about their subscriber base from each other, yet the nature of the service dictates that carriers exchange information with one another to provide subscribers with full mobility.

For example, when a user who is usually served by Carrier X in Chicago roams to the Carrier Z service area in Boston, Carrier Z must be able to find out whether that user is authorized to use the network. To do that, Carrier Z queries the Carrier X database about the user's access rights using the NEI. Carrier X provides a simple "yes" or "no" response. The details concerning the identity of the user, types of service the user has signed up for, rates being charged, and amount of network usage are all protected.

CDPD NETWORK BACKBONE

The internal network connecting the backbone routers (i.e., MD-ISs) must be capable of supporting CLNP and IP. The backbone routers terminate all CDPD-specific communications with mobile units and MDBS, producing only generic IP and connectionless network protoc ol (CLNP) packets for transmission through the backbone network.

M-ES Protocols

As noted, the requirement that mobile units support IP is meant to ensure that existing applications software can be used in CDPD networks with little or no modification. However, new protocols below the network layer have also been designed for CDPD. These protocols fall into two categories: those required to allow the mobile unit to connect locally to an MDBS, and those required to allow the mobile unit to connect to a serving backbone router and the network at large.

Digital sense multiple access is the protocol used by the mobile unit to connect to the local MDBS. Digital Sense, Multiple Access is similar to the carrier sense multiple access (CSMA) protocol used in Ethernet. Digital Sense, Multiple Access is a technique for multiple mobile units to share a single cellular frequency, much as Carrier Sense, Multiple Access allows multiple computers to share a single cable. The key difference between the two, apart from the data rate, is that CSMA requires the stations on the cable to act as peers contending for access to the cable in order to transmit, whereas in DSMA the MDBS acts as a referee, telling a mobile unit when its transmissions have been garbled.

A pair of protocols permit communications between the mobile unit and the backbone router. The mobile data link protocol (MDLP) uses Media Access Control framing and sequence control to provide basic error detection and recovery procedures; the subnetwork dependent convergence protocol (SNDCP) provides segmentation and head compression.

In addition to segmentation and header compression for transmission efficiency, other important features of SNDCP include encryption and mobile unit authentication. While the cellular network provides a certain amount of protection against eavesdropping because of its channel-hopping techniques, the applications expected to be used on the CDPD network require definite security. Competing businesses must have the confidence that their information cannot be seen by competitors. SNDCP encryption uses the exchange of secret keys between the mobile unit and the backbone router to ensure that there can be no violation of security when transmitting over the airwaves. The authentication procedure guards against unauthorized use of a network address.

TRANSPARENT OPERATION

Complete mobility is one of the key goals of CDPD networks. Because applications software must be able to operate over the network, the network itself must make any required operational changes transparently.

For example, the mobile units must automatically identify themselves to the network using the MNRP protocol, which recognizes the network addresses of mobile units whenever subscribers power on their computers or move to a new cell.

Data sent to a mobile unit is always sent through its home backbone router — another example of transparent operation. The home backbone router maintains an up-to-date table of the locations of the mobile units it is responsible for, thus making it possible to send connectionless data transmissions to a roaming mobile unit at any time. The home backbone router sends the data to the current serving backbone router. This scheme ensures that data reaches an end system regardless of its location, while keeping internal routing table updates to a minimum.

A connectionless service is one in which a physical connection need not be established in order to transmit data because the network is always available. In this scheme, each block of data is treated independently and contains the full destination host address. Each packet may traverse the network over a different path. A connection-oriented service, on the other hand, requires a destination address in the first packet only. Subsequent packets follow the path that has been established.

Sending Data from a Mobile Unit

Registration Procedure. Before a mobile unit can begin transmission, it enters into a dialogue, called the registration procedure, with the backbone router serving the area in which it is currently located. This dialogue identifies the mobile unit's OSI network layer address to the CDPD network. The serving backbone router tells the home backbone router responsible for that mobile unit that it is requesting service. The home backbone router authenticates the mobile unit, checking such things as the user's access rights and billing status. The registration procedure must be performed whenever the mobile unit is first powered on, or roams to a new serving backbone router.

Once the registration and authentication procedures are completed, the mobile unit begins sending data. The mobile unit is now on what appears to be a LAN connecting all such units operating within the cell of the telephone network. The LAN is really a single set of transmit and receive frequencies shared by the mobile units that access this cellular LAN using the digital sense multiple access technique.

The cells, or Digital Sense, Multiple Access LANs, are interconnected by the backbone routers in much the same way that routers connect Ethernet or token ring LANs. The serving backbone router examines the data sent by the mobile units, looking for the destination address. By comparing the destination address with those in its tables, the backbone router can send the data to the appropriate destination by the best path available (see Exhibit 4). The user can now log on to the portable computer, access shared services such as CompuServe, or send information directly to other roaming mobile units. When sending data from a mobile unit to other computers, the CDPD network must only ensure that the user is allowed to transmit. Once the user is authenticated, data is sent in a manner similar to the way it is sent in current LAN internetworks.

Sending Data to a Mobile Unit

On the return path, when data is sent to the mobile unit, the CDPD network must be prepared to deal with mobile units that are actively mobile — moving in a car, for example. In this case, it is likely that the mobile unit would move from one serving backbone router to another during the session. The CDPD network accommodates the roaming

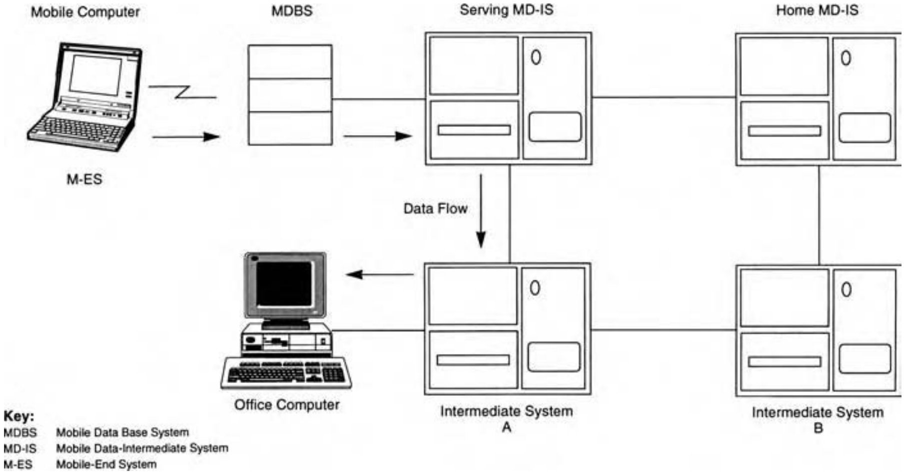


Exhibit 4. Potential Data Paths

mobile unit by always sending its data to its home backbone router. The home backbone router always advertises itself as the destination router for the mobile units it serves.

Redirect Procedure. The home backbone router knows the current location of the mobile unit because of the registration procedure. When sending information to a mobile unit, the home backbone router encapsulates it into frames using the ConnectionLess Network Protocol protocol and sends them to the address of the current serving backbone router. Once the data arrives at the serving backbone router, it is de-encapsulated into its original form to be sent to the mobile unit. This method of handling data transmissions at the home backbone router is called the redirect procedure (see Exhibit 5).

The redirect procedure takes advantage of the identification done during the registration procedure. The registration procedure serves two purposes:

1. To authenticate the user's access rights
2. To identify the current location of the user

The redirect procedure uses this information to minimize network overhead. The alternative, in which all the backbone routers would update their global routing tables whenever a mobile unit moved, would saturate the network with overhead traffic. The CDPD network permits full mobility, but without imposing an undue burden on the network infrastructure.

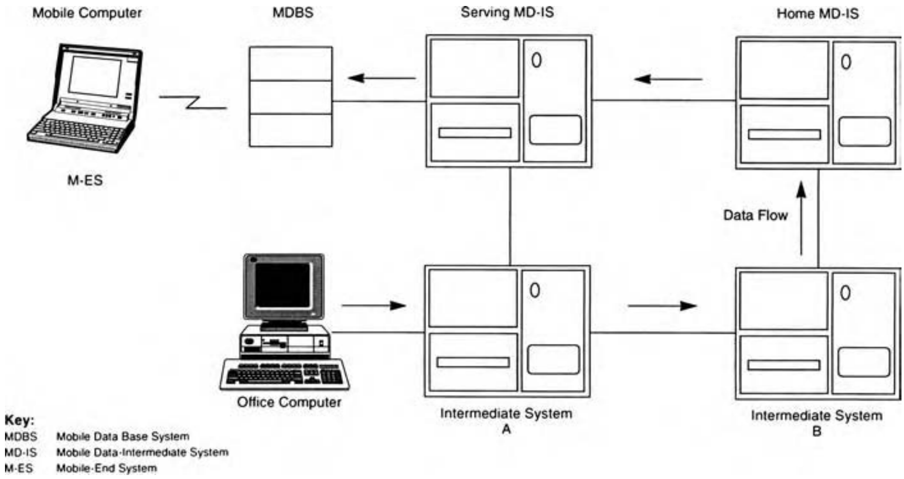


Exhibit 5. The Redirect Procedure

IS THERE A MASS MARKET FOR CDPD?

Industry analysts have estimated that the wireless data market would be worth $10 billion by the year 2000, providing service to about 13 million mobile data workers. Bell Atlantic Mobile, an early provider of CDPD-based services, predicted that as much as a fifth of its cellular revenues could come from data services by the start of the new millenium.

The eventual availability of low-cost CDPD modems does not guarantee a mass market for CDPD. For this to happen, commonly used applications must be adapted to the technology using application programming interfaces (APIs). APIs are required to optimize new and existing applications for use over relatively low-bandwidth wireless links with their high overhead and delay. After overhead is taken into account, the wireless CDPD link will top out at 14.4K b/s. The average throughput falls between 9K b/s and 12K b/s, depending on the number of errors and retransmissions.

Although CDPD is ideal for vertical niche markets such as fleet dispatch and field service, the more popular applications include e-mail, facsimile, and Remote Database access. Several toolkits are available to give new and existing applications the capability to run over CDPD networks.

To improve application performance over low-bandwidth wireless links, middleware that uses intelligent agents is now available that allows laptop users to query a corporate database using a software agent at the corporate site. If the user does not want to wait for a response to a query, or the connection is lost, the agent collects the information and sends it over the wireless network when the user makes the next connection.

CONCLUSION

Mobile users who are already committed to wireless data services are among the early users of CDPD service. As the price of CDPD modems fall, coverage increases, and more applications become optimized for CDPD, the technology will have even wider appeal.

CDPD networks are appealing because they offer seamless nationwide availability; work with the vast installed base of computers, applications, and data networks; and make use of existing private and public network infrastructures, encompassing all products and user equipment. The ultimate success of CDPD is, of course, closely tied to industry efforts to standardize its implementation. A universal standard for cellular packet data would facilitate terminal capability, allow users to roam between service areas, and simplify the introduction of wireless data services.

ABOUT THE AUTHOR

Nathan J. Muller is an independent consultant in Huntsville, AL, specializing in advanced technology marketing and education. He has more than 22 years of industry experience and has written extensively on many aspects of computers and communications. He is the author of eight books and more than 500 technical articles. He has held numerous technical and marketing positions with such companies as Control Data Corp., Planning Research Corp., Cable & Wireless Communications, ITT Telecom, and General DataComm, Inc. He holds an M.A. in social and organizational behavior from George Washington University.

Chapter 57
TN3270 and TN5250 Internet Standards

Ed Bailey

Why is access to System/390 and AS/400 terminal applications using the Internet so important? There are three basic reasons.

1. These terminal applications represent a large percentage of the data and logic of a typical enterprise.
2. There is a huge installed base of users who are currently accessing these applications productively.
3. New devices and end systems are being enabled with technology to access these applications.

ENTERPRISE DATA AND LOGIC

For larger enterprises, 70 percent or more of their mission-critical data resides on mainframes. While System/390 and AS/400 terminal applications are typically referred to as legacy applications, they collectively represent a vast majority of the business infrastructure upon which our economy rests today. And with a solid foundation of business logic that represents a history of investment, users of these applications rely heavily on their ability to access them to perform daily business transactions.

USER PRODUCTIVITY AND CONFIDENCE

Most of these terminal applications emerged in the late 1970s and early 1980s during the evolution from batch to online processing. Brought about by the introduction of the computer terminal and System Network Architecture (SNA), online processing gave rise to more direct access to enter, view, and alter information by the end user (see Exhibit 1). This empowered the individual and allowed productivity to soar. The ongoing reliability and availability of these terminal-oriented applications fostered confidence in the Information Technology (IT) professional and the end user to commit to completing ever-increasing volumes of transactions interactively.

0-8493-1161-6/02/$0.00+$1.50

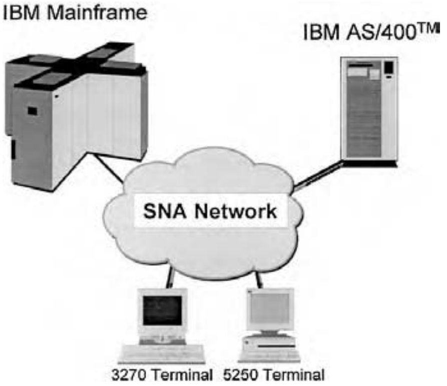


Exhibit 1. Terminal-Oriented Applications

NEWER PLATFORMS AND DEVICES

Since their introduction, personal computers have been used to emulate the terminal functions without requiring change to the host application. This emulation, along with new personal computing paradigms for downsized solutions and client/server distributed platforms, has not displaced System/390 and AS/400 terminal applications. Rather, the technology that these approaches of the 1990s represented has propelled end-user productivity to new heights. Greater end-user mobility has also contributed to this technology, which enables the use of much smaller yet powerful devices such as laptops and PDAs (see Exhibit 2).

With more power at their fingertips, the demand by users for access to System/390 and AS/400 terminal applications remains high as even newer and more innovative approaches are developed to manipulate and present the information that these applications maintain.

The most recent contributor to this demand has been the increased number of users served by the advancement of the Internet and World Wide Web. Viewed as the open global information highway, the Internet is expected to provide worldwide access between the information provider and the consumer. Electronic commerce (E-commerce) has quickly become the model for conducting business involving the Internet in the next millennium. It then stands to reason that if one intends to participate in E-commerce, then System/390 and AS/400 terminal applications must be accessible to end users using the Internet. See Exhibit 3.

Basic to the Internet is use of Transmission Control Protocol/Internet Protocol (TCP/IP). Standards have been developed to enable access to SNA terminal applications using TCP/IP. These standards are TN3270 and

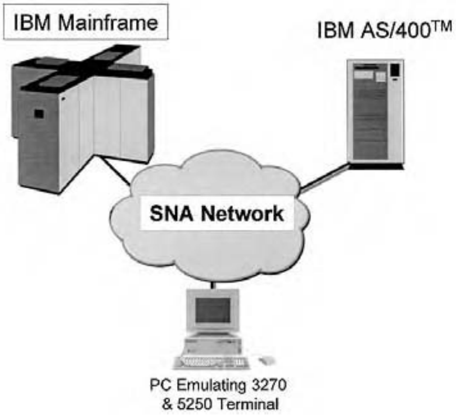


Exhibit 2. PC Emulation of Terminal Functions

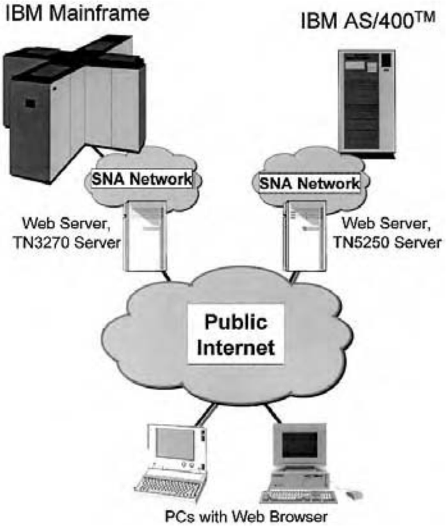


Exhibit 3. Internet Access to SNA Applications

TN5250. Client and server implementations of TN3270 and TN5250 provide access without requiring change to the host application. See Exhibit 4. These standards are actively evolving to adopt additional Internet technologies associated with Web browsers and programming. The remainder of this article will address the TN3270 and TN5250 standards for using the Internet to access System/390 and AS/400 terminal applications.

HOW DOES STANDARDIZATION HELP?

The Internet consists of processors that are interconnected via transmission links. Some of the processors are designed to do a special task such as establishing a path or routing the information along to the next destination, while other processors provide services to end users such as security or information retrieval. Typically, the processor that the end user operates directly is called the client, and the processor that contains the application is called the server. To access a particular application, an end-user request may pass from the client through multiple processors before reaching the correct server. The manner in which these processors communicate is through the use of protocols.

Left to chance, it would be very unlikely that any of the independent implementations of hardware and software would operate well — if at all — with other implementations. This creates a multitude of business and technical challenges for IT professionals and their end users. Therefore, consumers expect developers of Internet products and services to follow specifications, known as standards produced by the Internet Engineering Task Force (IETF). The IETF is the organization that facilitates open participation in producing Internet standards and promotes interoperability among various Internet components from different sources. The benefit of greater interoperability among particular implementations is higher confidence in their use and the use of the Internet overall. More information on the IETF is available at the http://www.ietf.org Web site.

PROTOCOLS

A closer examination of the various components of TN3270 and TN5250 allows one to better understand the benefits of standardization. One begins with the protocol. Simply stated, the protocol is the manner in which two components establish and maintain communications. As observed in Exhibit 4, there are two network protocols involved to allow end users to access SNA applications via the Internet: TCP/IP and SNA (Systems Network Architecture). Any enterprise with 3270 or 5250 terminal applications will have SNA protocol. Also, any enterprise with access to the Internet will have TCP/IP. Enabling users to be connected to the Internet using TCP/IP and have access to SNA applications requires two key components, referred to as servers and clients (Exhibit 4).

Servers

A server has the important role of initiating, managing, and terminating the SNA flows required by the OS/390 and AS/400 applications business logic. This is depicted as "A" in Exhibit 5. A server can include the entire protocol stack or use the programming interface provided by an existing stack. The SNA resources assigned to the server are used to support

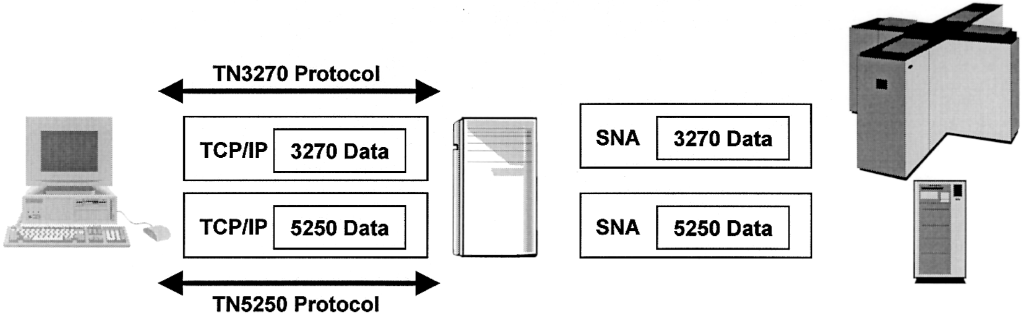


Exhibit 4. Internet Protocol to Access SNA Applications

requests from the end user. A server may reside on the same processor as the application or on a different processor.

The server passes the SNA datastream received from the terminal application to the end user by supporting the TCP/IP flows to and from the client. Requests and responses flow to and from the user on the TCP/IP connection maintained by "B" in Exhibit 5. A server may provide the TCP/IP stack or rely on the use the programming interface of an existing stack. The role of "C" is to pass the datastream between "A" and "B." Early implementations of TN3270 and TN5250 servers conveyed only the datastream and very little information to the client about the SNA connection. The latest specifications include many more options for passing additional information to the client about the SNA connection and resources.

Clients

The client maintains the graphical end-user interface (GUI) and connection to the server (see Exhibit 6). Component "D" is responsible for initiating, maintaining, and terminating the connection to the server using TCP/IP. Just like the server, it can include the TCP/IP stack or use the programming interface of an existing TCP/IP stack. Component "E" sends and receives the datastream with the SNA terminal application and provides the GUI to the end user. The latest specifications include more options that enable the client to provide the user with many more choices for presentation of the information.

Because multiple client and server implementations exist in the market and the client implementation can be obtained from a different source than the source for the server implementation, one can readily see how interoperability could be an issue without the benefit of standards. Keep in mind that standards evolve and do not always address every unique circumstance. Therefore, they allow for options that are left to the discretion of the implementers. Although a TN3270 or TN5350 client or server makes a claim of support for the standard, it is the set of options that should be examined closely against the requirements to determine the best solution for a business. Options for supporting such technologies as security, management, and programming should receive particular attention.

INTEROPERABILITY

The IETF charters working groups to address particular Internet problems or requirements. The TN3270E Enhancements Work Group was such a work group chartered under the Applications Division. This work group has produced a number of specifications for enhancing TN3270 and

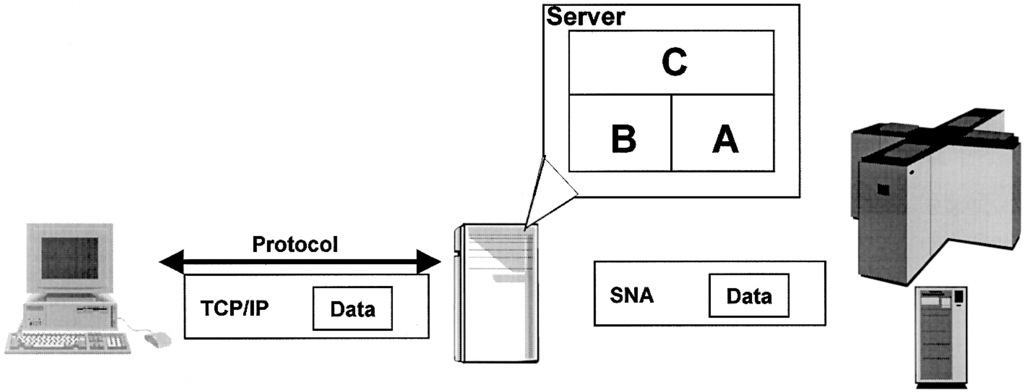


Exhibit 5. TN3270 and TN5250 Server

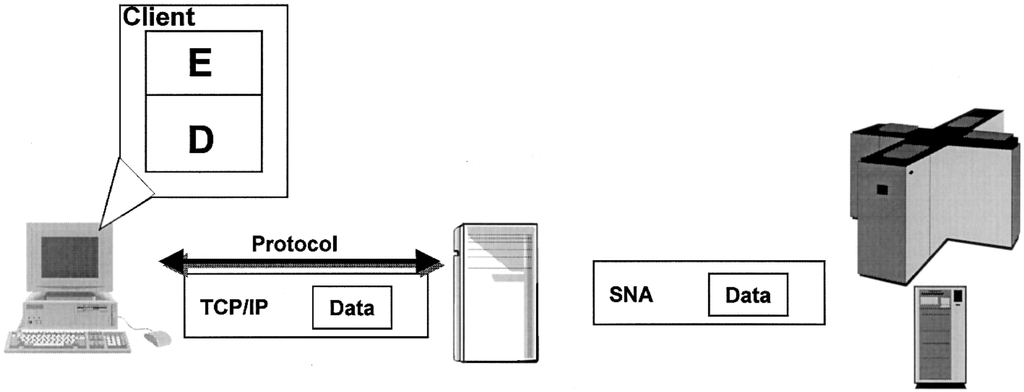


Exhibit 6. TN3270 and TN5250 Client

TN5250. In support of the implementation of these specifications, the work group has conducted a number of interoperability tests.

Basic interoperability testing focused on:

- How well the server connected with the SNA application
- How well the server connected with the client
- How well the client connected with the server
- How well the client displayed the information

Results from the interoperability have been the consistent interpretation of the specifications, the increase in the number of new clients and servers, and the solidification of the protocol on which to base additional enhancements.

Enhancements

As end users access System/390 or AS/400 SNA terminal applications from the Internet, certain characteristics, which existed in their prior access, are expected. The work group has defined specifications to support these characteristics, which include:

1. Security (encryption, authentication, authorization). This specification addresses the application level security based on the Transport Layer Security (TLS) standard and Secure Sockets Layer (SSL).
2. Management (configuration, response time monitoring). Two MIB specifications here address the configuration and response time monitoring for service-level management.
3. Performance (service location, session balancing, caching). This specification addresses the use of the Service Location Protocol (SLP) standard to identify services dynamically and learn their current workload factor.

Current efforts are under way by a number of vendors to implement these new specifications. These capabilities will enable IT professionals to deliver consistent levels of service to their end users when using the Internet.

Programming

With the growing popularity of E-commerce, independent software vendors (ISVs) want to deploy more Internet-ready applications. Most ISVs traditionally rely on the ability to build their services on top of clients and servers. Users want to integrate information from a variety of sources and formats with the data available from SNA terminal applications. Satisfying the need of the ISV and end user in this problem space requires new programming capabilities extensible to the Internet user. This leads to the use of object classes, object interfaces, and transform services. The support for Hypertext Markup Language (HTML), Java, and XML in browsers requires new programming interfaces, which provide programmatic interaction with the SNA terminal application.

SUMMARY

The use of the Internet to access SNA terminal applications is considered to be rejuvenation as it opens up new markets and expands the reach of end users. TN3270 and TN5250 specifications are designed to support the needs of these applications. Interoperability and standardization have established confidence in their use. Migration to these technologies by the enterprise results in:

- No change to the terminal application
- Reduced SNA networking complexities
- Consistent user interface and tools
- E-commerce enabling

When selecting an implementation of a TN3270 or TN5250 client or server, always request interoperability information for the implementation from the provider. This information should state how well this implementation operates with similar implementations from other sources. An implementation that has not been well tested with other implementations may be lacking in capability and leave requirements unsatisfied after installation. If the provider is unwilling to produce such information, one should ask oneself the question, “Do I really want my mission-critical applications to depend on this implementation?”

INTERNET INTEGRATION

The TN3270 and TN5250 standards established by the IETF paved the way for the next generation of technologies for Internet integration of the System/390 and AS/400 SNA terminal applications. Comparing the role of the protocols to that of a typical package delivery service, the emphasis is on preservation of the package or datastream (sometimes referred to as the payload) from end to end. This gives the most consistency in behavior for presentation of information to the end user when compared to the original terminal or emulator. See Exhibit 7. When the application creates the datastream (1), it remains intact and unchanged as it travels to GUI (6). Although use of the Internet is made to carry the datastream to the end user, this does not support integration of the information with other Internet applications.

The Web browser has introduced a new behavior that can enhance the presentation of the information to the end user. The use of a browser may require transformation of the data stream into another format — HTML and XML technologies use transformations. Java, on the other hand, does

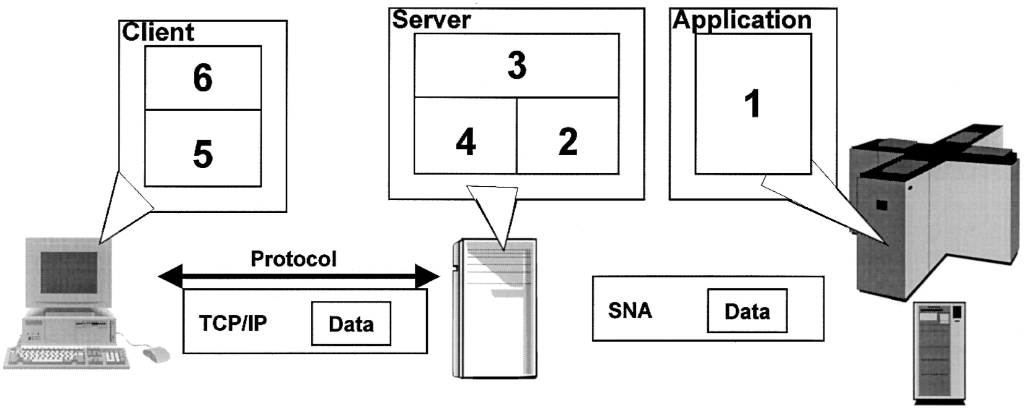


Exhibit 7. Datastream Flow

not require a transformation of the datastream. As an applet, it is designed to correctly process the original datastream. Typically, the transform occurs at point 3 in the flow in Exhibit 7. However, some designs may actually not perform the transform until point 6.

One can see that careful thought should be given to where the transform occurs. Performance and function can be significantly affected. The use of objects to assist in processing the datastream is advancing the use of transforms. This is the momentum behind such efforts as the Open Host Interface Objects (OHIO) specification to establish an industry standard. Placing object interfaces into point 3 or 6 greatly enhances the capabilities of vendors and users to integrate the information from SNA terminal applications with other Web and Internet bases while maintaining function and performance. All the excitement of E-commerce has brought a number of new players to the market, providing SNA terminal application integration into the Internet. How consistent these implementers are in developing and using transforms and objects can impact E-commerce deployment within an enterprise. One should know where and how the transform is achieved before committing one's business to a particular implementation.

LOOKING FORWARD

Many of today's business processes (e.g., inventory, finance, claims, manufacturing, shipping) lack integration with the Internet due to a previous inability to offer services to end users that they needed and were accustomed to using. Slow progress is partly attributed to insufficient security and manageability. Although still maturing, recent enhancements in the market have produced better solutions that enable enterprises to speed up Internet integration of their host terminal applications and enter into E-commerce.

ABOUT THE AUTHOR

As an IBM senior engineer, **Ed Bailey** has more than 25 years of experience in Information Technology. Ed has extensive knowledge in host applications, client/server computing, networking, and data communications. With degrees in computer science and business economics, Ed uses his knowledge and experience to develop emerging technologies. Currently, Ed is chairman of the TN3270 Work Group at the Internet Engineering Task Force.

Chapter 58
Knowledge Portal Classification Technology: Mapping Knowledge Landscapes

Michael J.D. Sutton

An emerging technology platform, often referred to as automatic classification systems or Corporate Portal Categorization Solutions (CPCS), has matured as an application category in the past 12 to 18 months. A CPCS uses a tool to *automatically* organize the subject categories of the institutional memory of an enterprise, and consequently permit the navigation of the resulting knowledge landscapes with relative ease.

Our collective experience with the World Wide Web and the Internet has exposed us to the limitations of search engines. These tools furnish a basic searching method based on simple keywords (e.g., "entertainment") or phrases (e.g., "information science"). The searches result in aggravation for the knowledge worker because of the classical information science problem of *recall* versus *precision*. Simple search expressions result in high recall (250,000 hits), but low precision (who is going to browse more than 50 or 100 "hits"?). The result is significant noise levels, which work against intelligent use of tools to find documents that fit the goal of the knowledge worker's query.

Most information experts (from Professors Paul Strassmann and James Martin to John Thorp of DMR and Dr. Neil Burk) have articulated the traditional challenge of inventorying and navigating corporate information and knowledge assets (institutional memory). Because information contained in

0-8493-1161-6/02/$0.00+$1.50

databases is estimated to characterize between 15 and 20 percent of the knowledge assets of an organization, structured queries recall, at most, 20 percent of the overall assets of the organization's memory. Even with 100 percent precision within this recall set, knowledge workers are blindsided by their inability to easily navigate and locate text-based information in the remaining 80 percent of an institution's memory.

CHALLENGES OF TYPICAL PORTAL TECHNOLOGY

Portals rely on the design and deployment of topic maps — often called knowledge maps or taxonomies — to facilitate the classification of representative categories for the purpose of providing navigational landmarks to the knowledge worker. In general, this is "hard-wired" by a portal designer within a dashboard metaphor. Thus, as new categories are identified or emerge, reprogramming of the portal interface is necessary. Often the knowledge worker is not presented with a category on the dashboard that fits their need, and they must revert to a Web site search engine, which relegates them back to their initial *recall* versus *precision* problem.

In the past two years, a number of textually and graphically based applications have been developed and deployed in an attempt to alleviate this problem. The common application labels used to describe software fitting this function include:

- Automatic classification tools
- Automated categorization software
- Business intelligence or competitive intelligence software
- Knowledge discovery tools
- Knowledge mapping tools
- Text mining or text analysis applications

CLASSICAL RESPONSES TO USEFUL NAVIGATIONAL APPROACHES

Traditionally, individuals trained as librarians and information scientists played the role of classifier. These professionals single-handedly controlled the information gateways to the enterprise. Finding aids were painstakingly developed that directed the knowledge worker to locations where text-based information assets could be found. Corporate repositories containing documents, records, reports, studies, books, journals, audio clips, video, microform, and photograph/slides were within the reach of the inquisitive knowledge worker. Yet, most repositories were nondigital and the process of browsing and retrieval took place manually.

The introduction of Enterprise Document Management Systems (EDMSs) in the past decade opened the door to faster searching of the institutional memory; but without sophisticated knowledge organization

Exhibit 1. Vendor/Product Information

CPCS Vendor	Product	Web Site
Autonomy Corp.	Autonomy Application Suite	http://www.autonomy.com
Cartia	ThemeScape Product Suite	http://www.cartia.com
CIRI Lab Inc.	Auto-Classifier and K-Books	http://www.ciri.net
IBM Corp.	Intelligent Miner Family	http://www-4.ibm.com/software/data/iminer
Invisible Worlds	Blocks Architecture Suite	http://www.invisible.net
Inxight	Inxight Eureka and Tree Studio	http://www.inxight.com/
Semio Corp.	Semio Taxonomy & Semio Map	http://www.semio.com/

schemes, the knowledge worker was restricted to a few fields of metadata (i.e., title, author, date created, etc.) and again relegated to using basic text search and retrieval expressions. The limitations and constraints of most text search engines cannot be overcome even by practiced use of Boolean operators in the search query.

The knowledge worker spent an inordinate amount of time wading through the lakes and oceans of unstructured, text-based corporate information. Moreover, many librarians and information professionals who had not been able to demonstrate "added value" to the enterprise were laid off or fired. Consequently, no one within the enterprise who understood the rules and processes of categorization and classification was available to develop and deploy knowledge maps for the enterprise.

EMERGENCE OF CORPORATE PORTAL CATEGORIZATION SOLUTIONS

Now, however, this situation may have been reversed — or, if not reversed, at least abated. Anecdotal evidence suggests that the implementation of a CPCS can dramatically alter the satisfaction and effectiveness associated with the portal search and retrieval functions for both customers as well as employees. Numerous products have surfaced in the marketplace to fill this obvious information management gap. The products form a spectrum of different solution types and architectures. Exhibit 1 presents a brief list of the coordinates for a number of the products.

BACKGROUND INFORMATION ON THE CPCS VENDORS

The following profiles of the vendors and products outline the spectrum of differences in the way the tools furnish navigational landmarks through a repository or information collection. The paucity of any background material was a function of the lack of vendor or product information contained on the company's Web site.

Autonomy Corporation

Autonomy Corporation is a publicly traded company calling itself an infrastructure firm. It has dual headquarters in Cambridge, U.K., and San Francisco, CA; and its regional offices are located throughout Australia, Europe, North America, and Scandinavia. Founded in 1996, it gained its success and capital from the reputation and tools acquired through Cambridge Neurodynamics Ltd., which was founded by Dr. Michael Lynch in 1991.

Autonomy can claim a very visible line of clients, encompassing many financial, government, police and defense organizations, and telecommunications firms such as Alcatel, Associated Press, Australian Air Force, BAE Systems, Barclays Bank, British Aerospace, Lucent Technologies, Merrill Lynch, Novartis, Shell International, and Unilever. It can count as its partners numerous high-tech and systems integration firms, including CapGemini-Ernst Young, KPMG, Oracle Corp., PricewaterhouseCoopers, Sun Microsystems, Sybase, and Unisys.

The foundation of the Neurodynamics' technology is neural networks–pattern recognition. Combined with this technique is Bayesian Probability Theory and Shannon's Information Theory. The Bayesian inference technique is based on a mathematical model for selecting significant concepts. Shannon's model is used to decrease "noise" across a collection of documents so that only important concepts are classified.

An additional advantage for this product is that through its Dynamic Reasoning Engine, the repository content can be analyzed as well as the search patterns of the knowledge workers. Thus, the application can be both reactive and proactive to the knowledge worker's subject matter needs. Autonomy has a broad range of off-the-shelf application products for knowledge management, E-commerce, online content publishing, and "active" browser services:

- ActiveKnowledge: conducts real-time analysis of objects and proposed links as they are created
- Autonomy Answer: automates the process of responding to customer queries
- Autonomy Application Builder: allows developers to integrate other applications through an API
- Autonomy i-WAP: provides access through WAP browsers
- Autonomy Navigator: furnishes intelligent textual category management and retrieval
- Autonomy Portal-in-a-Box: provides an information portal for the enterprise
- Autonomy Server: the pattern recognition retrieval engine

- Autonomy Update: monitors registered event changes from sites, news feeds, chat streams, and repositories
- Autonomy Visualizer: the graphical navigation applet
- Kenjin: examines and responds to the contextual content of a browser initiated search; delivers additional proposed links

Cartia Inc.

Cartia Inc. was founded in 1996 but evolved from a remarkable lineage through research and development history in the U.S. intelligence community associated with a project to visualize the content of massive text bases. Cartia is a privately held corporation headquartered in Bellevue, WA. Cartia supports a number of well-known clients, including British Gas, Ford, Philips Electronics, and Texaco. It does not have partners in the traditional sense, but does support NewsMaps.com (http://www.newsmaps.com), a massive collection of items from current news stories, news groups, and discussion forums that demonstrates the cartographic capabilities of Cartia's product, ThemeScape.

In 1993–1994, Cartia grew out of the text visualization requirements of the intelligence and defense communities as the Galaxies Visualization Environment (GVE). The Americans used it to analyze Iraqi message traffic during the second mobilization by the Iraqis along the Kuwait border. From GVE, the next-generation product was born — Spatial Paradigm for Information Retrieval and Exploration (SPIRE). SPIRE was used to describe the peaks and valleys in a geographical metaphor for large text bases of documents. The peaks would suggest coherent themes, while proximity of the peaks indicated topical content that was similar. Although constructed in 1994–1996, it is still used in a number of government agencies for tracking terrorist groups.

A parallel initiative to SPIRE called WebTheme used the visualization component of SPIRE and focused its sources of information to the Web. It was the first at visualizing and organizing Web searches based on the Web page's content, and is still being used by NASA. Between 1996 and 1997, a group of researchers from the SPIRE team formed their own firm, ThemeMedia, and commercialized the SPIRE product as a new off-the-shelf product called SPIREX — still in use by corporations such as Texaco and British Petroleum. Finally, in 1998, ThemeMedia transformed itself through a corporate identity change into Cartia Inc. The product ThemeScape was launched as a method to represent an information space through a geographical metaphor.

The key concept used throughout the ThemeScape product suite is document harvesting. Text-based document repositories are transformed through contextual analysis, statistical and natural language filtering algorithms. The outcome is a self-organizing, interactive spatial and conceptual information map. In two-dimensional space, the topological representation preserves the concept of neighborhoods, showing related themes elevated and close together and dissimilar themes at a relative distance from each other.

Cartia offers a suite of four integrated application products:

1. ThemeServer for content recognition, map generation, and map distribution
2. ThemeScape WebManager for Web viewer distribution and map access control
3. Theme Publisher for file and Web harvesting, and map publishing
4. ThemeScape Web Viewer for viewing and interacting via topic, keyword, and zoom functionality with the published maps

The maps permit the knowledge worker to gain quick insight into the relationships between documents' content and topics based on their elevation and distance.

Ciri Labs Inc.

Ciri Labs is a low-profile player in the knowledge landscape product genre. Ciri Labs is a privately held company headquartered in Hull, Québec, Canada, near the Canadian Silicon Valley North. Ciri Labs employs an estimated 25 people, has no sales offices, and works exclusively through partner and OEM relationships. Ciri Labs' partner and customer base are not reported on its Web site.

Ciri Labs manufactures Auto-Classifier and a unique product and interface called kBooks. Auto-Classifier is a CPCS that automatically organizes documents, e-mail, and correspondence into subject categories. Documents are indexed by Auto-Classifier by means of knowledge classes or taxonomies that are categories induced from the material within a document collection.

The knowledge worker uses the kBook Knowledge Workbench to process a sample document that contains relevant subject matters. kBook initially builds a Knowledge Tree representing the subject categories within the document. Then a query is generated from the Knowledge Tree and launched against specified document collections. Documents containing similar subject categories are identified and retrieved. This process can be iteratively executed with the resultant document or document sets. The Knowledge Tree serves as a conceptual "table of contents" of the document representing the significant subjects within the document.

IBM Corporation

IBM's Intelligent Miner for Text is implemented at a number of client sites where huge text repositories of technical, newspaper, and financial information exist, including IBM's own DB2 Product and Service Online Technical Library, Risk Publication's FinanceWise, Southern New England Telephone, Süddeutsche Zeitung, Frankfurter Rundschau, and Frankenpost. IBM partners with numerous systems integration firms.

IBM's Intelligent Miner for Text evolved from the data warehouse and mining offerings of IBM, specifically IBM's Intelligent Miner for Data. Many of the same customers with a data warehouse problem also expressed frustration with text mining. The Intelligent Miner for Text is a toolkit comprised of two major components:

- Text analysis tools encompass:
  - Feature Extraction tool: used to extract topics, terms, names, relationships, acronyms, and concepts
  - Categorization tool: used to assign objects (such as documents or queries) to predefined categories in a taxonomy
  - Summarization tool: used to automatically create condensed summaries from the sentences contained in a document
  - Clustering tool: used to segment a document collection into topical clusters based on a hierarchical arrangements or binary relationships, and present the relationships visually
- Web searching tools encompass:
  - IBM Text Search Engine tool: used to execute in-depth document analysis during indexing using a set of linguistic and content analysis algorithms
  - NetQuestion Solution tool: used to index and search documents on single or multiple servers
  - IBM Web Crawler tool: used to crawl through Internet sites and collect documents by following selected links

Because IBM has taken a toolkit approach instead of the suite approach, the customer would engage a systems integrator to select the tool that satisfies the particular business problem and build an application. The Clustering tool actually furnishes the visual interface to view the relationship of clusters within a collection.

Invisible Worlds

Invisible Worlds, founded in August 1998 by Internet visionaries Carl Malamud and Marshall T. Rose, is located in Petaluma, CA, a 40-minute drive from San Francisco. The company is privately held and is funded by Softbank, El Dorado Ventures, Reuters' Greenhouse Fund, and private investors. Invisible Worlds, which employs approximately 50 professionals, has built a new class of infrastructure tool that is founded on the strength of the open-source software movement as a universal solution for metadata management.

Invisible Worlds offers an architecture and new product line based on an emerging — but uncopyrighted — system of organizing and mapping metadata into blocks, BXXP (Blocks eXtensible eXchange Protocol). The protocol uses XML as its underlying data structure. The company is currently testing its product on large text bases, including the business intelligence repositories at an Internet publisher, legal information stored in the repositories of outside law firms contracted by a large automobile manufacturer, and a CRM system at a software service provider firm.

Inxight

Inxight is a privately held company headquartered in Santa Clara, CA, that employs an estimated 100 employees. Inxight is a late-1990s spin-off of Xerox Corporation and presently has U.S. sales offices in Boston, Washington, D.C., Chicago, New York, Atlanta, Austin, and a U.K. sales office in London. Inxight has created a strong partner and customer base, to include AltaVista, Aurigin, Battelle, Comshare, Entreon, FocusEngine Inc., Hewlett-Packard, Infoseek, Inktomi, Interactive Edition, Lotus Corporation, Microsoft, Netrocity, Picturesafe, Sentius, SoftQuad, Sovereign Hill, Verity, VIT, *Wall Street Journal,* WebTrends, and YellowBrix, Inc.

Inxight Eureka is a graphical navigational tool that transforms large arrays of tabular data into a visual map of patterns and outlines. The map permits the knowledge worker to explore correlations and build hypotheses to explain the observed patterns. Eureka was formerly known as Table Lens. Table Lens uses a patented Focus+Context feature that furnishes a knowledge worker with interactive access to the “trees” (details of underlying data) while simultaneously viewing the “forest,” the corpus of all the data records. Consequently, the knowledge worker can aggregate all the data records while screening out the details.

Inxight Categorizer is a powerful knowledge management engine that automatically identifies and classifies MS Office documents, e-mail, Adobe PDF, or news feed information into a taxonomy of subject categories. This approach surpasses traditional manual- and rule-based indexing and categorizing approaches. Inxight Categorizer includes linguistic and language support in 12 languages.

Inxight Categorizer employs a “categorization by example” process that compares a new document with a large training set collection of manually coded documents. Inxight uses a patented linguistic analysis technology to select similar documents from the training set and, from these examples, infer the probable coding for the new document. Inxight Categorizer also assigns a confidence value to each coded document based on its ability to identify similar training documents. Coded "low-confidence" documents can then be routed and reviewed manually.

Inxight Tree Studio uses a patented visual display (called Star Tree maps) to quickly identify and organize information into nodes for navigation and retrieval. The interactive visual rendition of a Web site enables a knowledge worker to effectively navigate a Web site without encountering dead ends. To compose a Star Tree map, a wizard is dispatched to search the selected Web site or information collection and identify all of the nodes to be mapped. Then Inxight Tree Studio generates a hyperbolic visual display of these nodes that can be published and interactively manipulated.

Semio Corporation

Semio is a private venture-funded company founded in 1996 with offices located in the United States and Europe. Semio Taxonomy uses a patented technology that identifies and categorizes key concepts from the content contained in very large text repositories. It offers improvements over traditional approaches, such as AI (artificial intelligence), neural networks, or conventional keyword indexing applications.

Semio's core engine is based on years of research by Dr. Claude Vogel and is called SEMIOLEX. The patented technology draws its power from both linguistic processing and the field of computational semiotics. The process extracts lexical derivatives from the cluster nodes and automatically places them into categories. Semio Map leverages the previously created taxonomy to furnish the knowledge worker with a visual interface of the underlying source concepts, which may have been extracted from Lotus Notes, Documentum, or XML repositories.

Semio has acquired a significant partner base, which includes Epicentric Inc., InfoImage, Inso, Lockheed Martin, Plumtree, Project Performance Corp., Primix Solutions, and Sequoia Software Corp. Since June 1999, Semio's flagship product, Semio Taxonomy, has been in use throughout many Fortune 1000 companies encompassing research and development, and telecommunications firms (Applied Materials, AT&T, Cambridge Technology Partners, Cisco, DuPont, HighWire Press of Stanford University, IDG, Proctor & Gamble, SmithKline Beecham, and the U.S. Postal Service).

POTENTIAL PAYBACK AND ROI

CPCS facilitates a knowledge worker's navigational capability for customer E-business, relationship management applications, web publishing, and EDMSs. The application, while increasing the *precision* of retrieval, also dramatically reduces browsing where the traditional search engine *recall* function may have presented a large set of information assets.

The vendors, of course, have claimed significant return on investment (ROI) for implementing this kind of technology and display numerous customer testimonials on their Web sites; but an independent study or research initiative has yet to be published that objectively proves the value of the technology. Potential vendor-reported tangible benefits include the usual "suspects" for increased efficiency and effectiveness:

- Cost reductions
- Enhanced productivity
- Improved decision-making

Controlled experimentation with the technology by enterprises on portals for access by customers or employees could facilitate the identification of pragmatic "value-added" benefits. There appears to be enough truth in the marketing hype to make a pilot worthwhile in an enterprise.

Although the products cannot easily be compared among themselves, pricing in this product category runs the gamut from $200 to $1000+/seat. Annual software support contracts can be hefty and will depend on corporate size and number of server sites. Each opportunity should be evaluated as a separate business case. Cost/benefit issues will encompass:

- Application training and support
- Conversion planning, approach, cost, and complexity
- Document volumes
- Forecast knowledge worker usage
- Heterogeneous information sources
- Installation design
- Planned infrastructure platform
- Repository types and sizing
- Scalability requirements
- Taxonomy setup

A caveat worth mentioning is the problem associated with testing a concept-based CPCS product:

> How do you know that the product has accurately and comprehensively classified all the documents that comprise your different information collections?

Because of the sophisticated algorithms, fuzzy set theory, heuristics, and inference engines involved with this product category, your enterprise may be challenged to prove that the product actually works by your auditors. If you "lose" a document that actually remains in your repository but is "undiscovered," doubt may grow that this kind of tool actually works. However, at this stage, no known major embarrassments or omissions have been reported.

CONCLUDING REMARKS

Anecdotal stories abound from the CPCS vendors testifying to increased sales, revenue, or market share, but your enterprise will need to develop a suite of metrics that can prove or disprove this marketing promise. If one starts with a prototype, one should be able to quickly isolate any issues or concerns regarding accuracy as well as cost/benefit. Minimize risk in this new software category while trying to strategically position the tool for use in mission-critical customer support, employee-based knowledge retrieval, EDMS, and business intelligence applications.

A considerable breadth currently exists in the functionality of the products and approaches of the CPCS vendors. A company cannot expect to "throw" technology at the problem represented by these vendors and find an instant solution. Identify a requirement for this form of knowledge mapping and evaluate these products for the functionality that fits. Then one will be in a position to successfully assess and deploy what appears to be a unique and revolutionary approach to navigating the knowledge landscape.

ABOUT THE AUTHOR

Michael J.D. Sutton has returned to McGill University in Montréal, Québec, Canada, to pursue his Ph.D. in the Knowledge Management field within the Graduate School of Library and Information Studies. Formerly, he was the senior director, Knowledge and Document Management, in the international consulting firm of CBSI Canada Inc. Mr. Sutton has participated as an advisory member on numerous standards and specifications bodies, including two international ISO committees: ISO 11730-Forms Interface Management System and ISO 8879-Standard Generalized Markup Language.

About the Editor

Sanjiv Purba holds a Bachelor of Science degree from the University of Toronto and has more than 16 years of relevant information technology experience. He is practice director for Microsoft Consulting Services — E-Solutions in Canada. Prior to joining Microsoft, Mr. Purba was a senior manager with Deloitte Consulting and leader of its object-oriented practice.

Mr. Purba has extensive industry experience, with a focus on the financial and retail industry sectors. As a consultant, Mr. Purba has gained relevant experience in other industries, such as telecommunications, travel, and tourism, manufacturing, and entertainment. He has served in a variety of roles in large organizations, including developer, senior developer, business analyst, systems analyst, team leader, project manager, consultant, senior architect, senior manager, director, and acting vice president.

Mr. Purba is the author of six information technology (IT)-related textbooks. He is also editor of *High-Performance Web Databases* and the *Data Management Handbook,* both published by Auerbach. Mr. Purba has authored more than 100 IT articles for *Computerworld Canada, Network World, Computing Canada, DBMS Magazine, Hi-Tech Career Journal (HTC),* and the *Toronto Star.* Mr. Purba is a past editor of *ITOntario,* a publication of the Canadian Information Processing Society (CIPS). He has also written fantasy and science fiction graphic novels.

Mr. Purba is a regular speaker at industry symposiums on technical and project management topics. He has lectured at universities and colleges for the past 16 years, including Humber College, the University of Toronto, and Ryerson Polytechnic University. He recently hosted an IT forum on a television program in the Toronto area.

Prior to joining Deloitte Consulting, Mr. Purba ran his own computer consulting business, Purba Computer Solutions, Inc., during which time he consulted with Canadian Tire, Sun Life Assurance Company of Canada, and IBM in a variety of roles, including senior architect, facilitator, and project leader.

Mr. Purba also served as a senior architect and senior consultant with Flynn McNeil Raheb and Associates, a management consulting firm, for five years prior to owning his own business. During this time, he consulted with such organizations as IBM, ISM, The Workers Compensation Board, Alcatel, and the Ministry of Education.

Mr. Purba enjoys weightlifting, aerobics, karate, tae kwon do, tae bo, movies, and charity work.

Index

A

Abandon rate, 358
Abstract Windowing Toolkit, 502–503
Acceptable use policy
  activation of, 70
  approval of, 66
  committee for, 70–72
  definition of, 65
  description of, 65
  e-mail, 67
  employee acceptance and familiarity with, 66, 70
  netiquette descriptions, 70
  newsgroups, 68
  penalties for violating, 69, 71–72
  posting of, 70
  purpose of, 72
  scope of, 66–70
  World Wide Web resources use, 67–68
  writing of, 65–66
Access control lists, 46
Active Server Page, 330–331
ActiveX
  characteristics of, 294–295, 315–316, 381–382
  Java Beans versus, 384
ADEX, *see* Ad Exchange
Ad Exchange, 464
Agents, 31
American National Standards Institute, 231
Analog transmissions, 266–267
Apache Web Server, 498–499
Applet-based emulation
  description of, 322–323
  user interface rejuvenation options for, 323–329
Application aggregators platform enablers, 31
Application engine, 170–171
Application layer protocols, 745–746
Application program interface
  cellular digital packet data, 751
  CORBA, 383
  DCOM, 383–384
  description of, 330–331
  Java
   data connectivity, 540–541
   description of, 534
   interface definition language, 540
   management, 542
   message service, 540
   security, 541
   server, 542
   transaction services, 542
Application servers, 393, 397–399
Application service developers, 30
Application service managers, 30
Application service providers
  applications for
   E-procurement, 33–34
   host applications integrated with Web services, 34
   supply chain management, 33–34
   Web services, 34–35
  benefits of, 30
  challenges for, 35–36
  description of, 29, 97
  increase in, 31
  industry of, 31–33
  market for, 31
  .NET architecture, 32–33
  outsourced services, 33
  point-of-responsibility approach by, 36
  roles of, 30–31
  security concerns, 36
  skepticism regarding, 35
  wireless data access using, 194–196
Application tiers, for Web sites
  business tier, 99–100
  client tier, 99
  data tier, 100
  presentation tier, 98–99
Appropriate use policy, 660–662
AS/400
  description of, 753–755
  TN3270 standard
   client, 760
   description of, 754–755
   enhancements of, 758, 761
   server, 759

- TN5250 standard
  - client, 760
  - description of, 754–755
  - server, 759
- ASP, *see* Application service provider
- Asynchronous transfer mode, wireless
  - advantages of, 226, 227
  - applications for, 226, 239–240
  - architecture, 228–230
  - Arielnet system, 239–240
  - ATM wireless access communication system project, 234–236
  - bandwidth, 227–228
  - connections, 226
  - description of, 225–226, 228
  - hardware
    - handheld communications terminal, 238
    - schematic diagrams of, 236–238
    - wireless interface processor, 238–239
  - interfaces for, 226
  - international joint ventures for, 236
  - network demonstrator, 231–234
  - prototypes, 230
  - reasons for, 226–228
  - remote data collection, 239
  - standards, 230–231
  - summary overview of, 240
- ATM, *see* Asynchronous transfer mode
- ATM wireless access communication system project, 234–236
- Audit, 596, 672–676
- Audit trail, 709
- Authentication
  - client/server, 667–668
  - description of, 273–275, 630
  - firewall support of, 685
  - secure socket layer, 665, 667
- Automatic call distributor, 141, 143–144
- Autonomy Corporation, 768–769
- AWACS, *see* ATM wireless access communication system project
- AWT, *see* Abstract Windowing Toolkit

B

- Baan Data Navigator Plus, 9
- Banks
  - E-commerce, 432
  - electronic bill presentment and payment role of, 121
- Basket-to-buy rate, 359
- B2B exchanges, *see* Business-to-business exchanges
- Biller Integration System, 130–131
- Blackberry devices, 211
- Bluetooth, 160, 215
- Bridge, 641
- Browser, *see* Web browser
- Business(es)
  - E-commerce requirements, 6–7
  - wireless technology impact on, 164–168
- Business tier, 99–100
- Business-to-business exchanges
  - antitrust considerations, 24
  - benefits of, 19–20
  - business considerations for, 23–25
  - buyers
    - benefits for, 19–20
    - costs for, 22
  - characteristics of, 13–14
  - costs of, 20–22
  - customer/supplier relationships, 25
  - definition of, 13
  - disadvantages of, 22
  - distribution logistics considerations, 24
  - estimated number of, 14
  - evaluation of, 25–26
  - evolution of, 16–17
  - factors that affect, 22–23
  - fees of, 20–22
  - firewall use, 24
  - functions performed by, 13, 16–17
  - horizontal, 17
  - implications of, 23–25
  - industries that utilize, 13
  - manager considerations for, 25–26
  - mechanism of action, 14
  - relationship management effects, 24–25
  - software packages, 18
  - suppliers
    - benefits for, 20
    - costs for, 20, 22
    - evaluation of, 23
  - supply chain automation, 19
  - technology used, 18–19
  - third-party control of, 13
  - types of, 17–18
  - vertical, 17–18
- Business units
  - description of, 335
  - financial, 342
  - interviewing of, 338–341
  - questionnaire for, 340–341

requirements of, 341–344
value added back to, 345, 348
Button objects
properties of, 579–580
testing of, 580

C

C++
C compilers, 513
characteristics of, 513–514
Java versus, comparisons between
features, 514–523
performance, 523–528, 721–722
operating system interface with, 514
Call-back, 145–146
Call center, Internet-enabled
connection methods for
call-back, 145–146
call-through, 146–147
description of, 144–145
switched connection, 148
description of, 141–142, 144
interactive voice response, 141, 144
questions to consider, 148–149
Call-through, 146–147
Capacity planning testing, 566–567
Carrier sense multiple access, 747
Cartia Inc., 769–770
CDMA, *see* Code Division Multiple Access
CDPD, *see* Cellular digital packet data
Cell, 255
Cellular digital packet data
applications for, 740–743
companies involved in developing, 740
definition of, 739
description of, 172
equipment requirements for, 743
fundamentals of, 739–740
information protection, 747
infrastructure, 739
mass market appeal of, 751–752
mobile units
data sent from, 749
data sent to, 749–750
description of, 748
redirect procedure, 750–751
mobile user benefits, 742–743
mobility management, 746–747
network architecture
mobile database system, 744
mobile data-intermediate system, 743–744
mobile-end system, 744
network backbone, 747–748
pricing, 741–742
principles of, 740–741
protocols, 745–746
reliability of, 742–743
remote users, 741
security of, 742
Cellular phones
authentication measures for, 273–275
costs of using, 157
memory storage in, 218
signal triangulation, 220
transmission methods for
analog, 266
digital, 266–272
wireless application protocol-enabled, 218–219
Cellular Telecommunications Industry Association, 247–248
Certificate authority, 419–422, 429
Certificate repository, 422–423
CGI, *see* Common gateway interference
Channel, 190
Channel hopping, 740
Chargeback systems
benefits of, 73–74
client/server applications, 79–80
contracts for, 84
costs of, 75
customer reactions to, 75, 84–85
description of, 73
design considerations for, 85
disadvantages of, 74–75
implementation of, 85–86
Internet use, 80–81
local area networks
billing methods, 77–78
considerations for, 76
hardware, 76–77
services included, 76–77
software, 77–79
mainframe operations, 75–76
programming services
fixed charges, 82–83
hourly rate, 81–82
maintenance and support of system, 83
rate comparisons, 84
reasons for not using, 74–75
reasons for using, 73–74

- resources use and, 74, 85
- simplicity of, 85

Circuit-switched networks, 156, 160

Ciri Labs Inc., 770

Click-through rate, 358

Click-to-basket rate, 359

Client(s), 758

Client/server applications
- authentication, 667–668
- chargeback systems for, 79–80
- components for, 380–381
- impediments, 390

Client-side processing, 294–295

Client tier, 99

Code coverage tools, 571

Code Division Multiple Access, 270–271

COM
- description of, 330, 381–382
- development of, 395
- history of, 391

Command_Rule, 473–474, 487–488

Commercial mobile radio service, 247

Common gateway interference, 295

Common object request broker architecture, *see* CORBA

Communication
- Internet protocols for, 9–10
- personal communication service, *see* Personal communication service

Component(s)
- acquiring of, 317–318
- application integration with, 314, 379
- business, 311–312, 314, 378–379
- classification of, 377–378
- client/server, 380–381
- configuration management, 318
- data access, 314
- definition of, 311, 377–378
- designing of, 384–386
- framework, 312, 314, 378–379
- granularity of, 378
- integration of, 318
- life cycle of, 386–387
- maintenance of, 318–319
- purchasing of, 318
- reuse potential, 387–388
- software, 377
- standards
  - ActiveX, 381–382
  - COM, 381–382
  - CORBA, 382–384
  - DCOM, 383–384
  - Java Beans, 384
- subsystem, 314
- technical, 378–379
- user interface, 314

Component-based development
- alternatives to, 317
- architecture of, 313–314
- benefits of, 312
- challenges for, 317–319
- client/server architecture application of, 316
- components, *see* Component(s)
- definition of, 311
- description of, 386
- middleware, 316
- models, 315–316
- network computing architecture application of, 316–317
- object-oriented technology foundation of, 316
- testing time benefits of, 312
- weaknesses of, 313

Component object model, *see* COM

Computer emergency response team advisories, 707–708

Computer Fraud and Abuse Act (1986), 58

Connectionless network protocol, 747

Consolidator model of electronic bill presentment and payment, 123–125

Contract law, 55–56

Conversion rate, 358

Cookies, 438, 677–678

Copyright laws, 634

CORBA
- application program interfaces used by, 383
- characteristics of, 301–302, 393–394
- DCOM versus, 382–383, 395–396
- definition of, 382
- distributed computing use of, 394
- elements of, 382
- Enterprise JavaBeans, 539
- history of, 391, 393
- limitations of, 405
- operating principles of, 394

Corporate portal categorization solutions
- benefits of, 774
- costs of, 774
- description of, 765
- emergence of, 767

return on investment by using, 773–774
summary overview of, 776
vendors who provide
Autonomy Corporation, 768–769
Cartia Inc., 769–770
characteristics of, 767
Ciri Labs Inc., 770
IBM, 771
Invisible Worlds, 771–772
Inxight, 772–773
Semio Corporation, 773
CPCS, *see* Corporate portal categorization solutions
Criminal law, 57–58
CRM, *see* Customer relationship management
Cross-sell rate, 358
CTIA, *see* Cellular Telecommunications Industry Association
Customer(s)
changing nature of, 105
electronic bill presentment and payment benefits for, 121, 128–129
familiarity with, 107
loyalty of, 108, 128–129
multimedia interactions with, 142
privacy of, 357–358
Customer relationship management
benefits of, 360
changes in, 107–108
definition of, 111
design elements of, 114–115
electronic
advantages of, 112
business applications for, 116
description of, 111
designing of, 115–116
E-commerce application of, 116
enhancements of, 114
examples of, 114–115
model of, 113
implementation of, 108–110
initiatives for, 108–109, 111
integration with Web site, 360
Internet effects on, 107
misconceptions of, 115
software
description of, 106–107
failure of, 111–112
strategy for, 109
success in, 108–109, 112
technical issues associated with, 112
technology for, 107–108
touch points, 112
Cyclic redundancy check, 650

D

Data
billing, 125
encryption of, 11–12
security for, 11–12
Database management systems, XML in, 456–457
Data interchange standards association, 130
Data tier, 100
DBMS, *see* Database management systems
DCOM
application program interfaces used by, 383–384
CORBA versus, 382–383, 395–396
description of, 301, 382
limitations of, 405
operating principles of, 395–396
server components, 395
Debuggers, 570
Deck, 219–220
Defense message system, 428
Demilitarized zone
definition of, 646
testing of, 647–650
Development, component-based
alternatives to, 317
architecture of, 313–314
benefits of, 312
challenges for, 317–319
client/server architecture application of, 316
components, *see* Component(s)
definition of, 311
middleware, 316
models, 315–316
network computing architecture application of, 316–317
object-oriented technology foundation of, 316
testing time benefits of, 312
weaknesses of, 313
Digital cellular radio technology, 740
Digital certificates, 438
Digital signature, 55–56, 421–422
Digital wallet, 438
Direct-sequence spread spectrum, 258
DISA, *see* Data interchange standards association

Distributed component object model, *see* DCOM
Distributed computing
- application servers for, 397–399
- approaches for, 393
- CORBA for, *see* CORBA
- DCOM for, *see* DCOM
- history of, 389–392
- objects, 392–393
- summary overview of, 398–399
- vendor-supplied, 391

Distributed interNet Architecture, 102–103
DMZ, *see* Demilitarized zone
Document object model, 466
Document-type declaration, 465
Document-type definition, 453, 465, 476–478
DOM, *see* Document object model
Domain
- definition of, 385
- design of, 385
- implementation of, 385
- specification of, 385
- validation of, 386
- verification of, 385

Domain Commerce, 449
Domain name server
- description of, 646
- services offered, 648

DSSS, *see* Direct-sequence spread spectrum
DTD, *see* Document-type definition
Dumpster diving, 610, 705
Dynamic hypermedia engine, 488

E

Eavesdropping, 268–269
EBPP, *see* Electronic bill presentment and payment
E-business
- characteristics of, 134
- definition of, 91, 134, 354–355
- description of, 639
- factors that affect, 355–356
- growth of, 353
- information access, 354–355
- interpersonal communication services, 355
- privacy considerations, 357–358
- requirements for, 92
- routes for, 91–92
- self-services, 354
- shopping services, 355
- virtual enterprises, 355
- Web site development for, *see* Web site

E-commerce
- bank involvement in, 432
- business requirements, 6–7, 335, 341–344
- business units
  - description of, 335
  - financial, 342
  - interviewing of, 338–341
  - questionnaire for, 340–341
  - value added back to, 345, 348
- challenges to, 432–433
- costs of, 360–361
- customer relationship changes associated with, 107–108
- economic advantages obtained by, 432
- electronic customer relationship management use, 116
- growth of, 333, 353–354, 431, 754
- hardware for
  - mainframes, 447
  - microchips, 446–447
  - servers, 447–450
- infrastructure
  - business requirements and, 341–344
  - buying versus building of, 335–338, 348, 350–351
  - definition of, 333–334
  - E-services for constructing, *see* E-services
  - public key, 428
  - schematic diagram of, 354
  - service attributes, 419–420
- operating systems
  - Linux, 497–499
  - Microsoft Windows NT, 450–451
  - Novell, 451
- overview of, 452
- privacy considerations, 357–358
- product selection, 8
- revenues of, 142, 431–432
- security of, *see also* Security
  - certificate authority, 420–422, 429
  - certificate repository, 422–423
  - commercial exchange services, 427
  - description of, 11–12, 434–435
  - electronic postmark, 424
  - encryption techniques, 434–435
  - return receipt, 425–426
  - software for, 435–436
  - storage and retrieval services, 426–427
  - transactions, 434–436

sell-side, 19
services provided, *see* E-services
site for, *see* Web site
technical requirements, 7–8
technologies for, 427–428
terminology associated with, 438–439
vendor ratings, 436–437
Web site for, *see* Web site
e-CRM, *see* Electronic customer relationship management
EDGE, *see* Enhanced Data Rates for Global Evolution
Edge, 160
EDI, *see* Electronic data interchange; Electronic document interchange
E-human resources
case study examples of, 135–137
description of, 134
globalization effects, 139
trends that affect, 137–139
EJB, *see* Enterprise JavaBeans
Electromagnetic pulse transformer bombs, 634
Electronic bill presentment and payment
billers' role in, 121–123
bill presentment, 127
consolidator model of, 123–125
cost savings associated with, 129
customer care, 127–128
customer service benefits of, 128–129
definition of, 120
description of, 119
direct biller model of, 122–123
emergence of, 120
example of, 120
IFX, 130–131
open financial exchange, 129–130
participants in, 121–122
reasons for implementation of, 128–129
stages of, 125–128
summary overview of, 131
technology involved in, 129–131
Electronic business XML, 464
Electronic customer relationship management
advantages of, 112
business applications for, 116
description of, 111
designing of, 115–116
E-commerce application of, 116
enhancements of, 114
examples of, 114–115
model of, 113
Electronic data interchange
definition of, 438
description of, 433, 460
extranet use of, 446
Electronic document interchange
business requirements for, 6–7
description of, 5
disadvantages of, 5–6
legal considerations, 55
standards, 5
technical requirements for, 7–8
Electronic postmark, 424
Elliptic curve, 436
e-mail
acceptable use policy for, 67
archiving of, 67
privacy, 67, 428
secure, 428
Enclaves
benefits of, 46–47
components of, 45
definition of, 40
description of, 44
enterprise network and, 45
implementation process for
deployment, 50
information gathering, 48
overview of, 47–48
planning, 48–49
prototyping, 49
refinement, 50–52
limitations of, 47
principles of, 44–45
resource allocation benefits of, 47
summary overview of, 51
Encryption
data, 11–12, 660
PCT, 438
private-key, 435, 630
public-key, 435, 629–630
RSA, 438
security uses of, 278, 434–435
Enhanced Data Rates for Global Evolution, 272
Enterprise document management systems, 766–767
Enterprise JavaBeans, 301–302
Enterprise systems
enclaves and, 45
security for, 11
Entrust Technologies, 435–436
E-procurement systems, 19

E-services
  business requirements assessment, 341–344
  buying versus building of, 335–338, 348, 350–351
  description of, 334
  goal of, 334
  identifying of, 344–345
  importance of, 334
  in-house development of, 350
  measurement methods for, 337
  outsourcing of, 334, 338, 348
  prioritizing of, 337
  quantifying of, 344–345
  types of, 354–355
  Web hosting, 345–347
Exchanges
  antitrust considerations, 24
  benefits of, 19–20
  business considerations for, 23–25
  buyers
    benefits for, 19–20
    costs for, 22
  characteristics of, 13–14
  costs of, 20–22
  customer/supplier relationships, 25
  definition of, 13
  disadvantages of, 22
  distribution logistics considerations, 24
  estimated number of, 14
  evaluation of, 25–26
  evolution of, 16–17
  factors that affect, 22–23
  fees of, 20–22
  firewall use, 24
  functions performed by, 13, 16–17
  horizontal, 17
  implications of, 23–25
  industries that utilize, 13
  manager considerations for, 25–26
  mechanism of action, 14
  relationship management effects, 24–25
  software packages, 18
  suppliers
    benefits for, 20
    costs for, 20, 22
    evaluation of, 23
  supply chain automation, 19
  technology used, 18–19
  third-party control of, 13
  types of, 17–18
  vertical, 17–18
Export control regulations, 58–59
Extensible markup language, *see* XML
Extensible stylesheet language, *see* XSL
Extranet security, 43–44
Extreme programming, 305–308
Eyeballs, 358

F

FDMA, *see* Frequency division multiple access
Federal interest computers, 58
Firehunter, 732
Firewall, *see also* Gateway
  access control, 684–685
  actions performed, 631
  attacks prevented by
    denial of service, 694–696
    Internet protocol source address spoofing, 693–694
    TCP/IP hijacking, 694
    TCP SYN, 694–695
  authentication, 685
  benefits of, 682
  business-to-business exchange use of, 24
  characteristics of, 683, 707
  contingency planning, 691, 693–696
  criteria for, 646, 684–685
  definition of, 438, 630, 646, 683
  description of, 11, 101, 681–682
  gateways
    application-level, 686–688
    circuit-level, 687–688
  host-based application-level, 632
  implementation of, 102
  legal issues regarding, 691, 693
  limitations of, 682–683
  network traffic through, 683–684
  outage, 691, 693
  packet filtering, 686, 688
  policies and standards
    definition of, 690–691
    development of, 688–690
    employee compliance with, 690
    reasons for having, 687–688
    risk analysis before creating, 690
    sample, 692
    structure of, 690
  programming of, 631
  protocols supported, 684
  purpose of, 631
  removal of, 691, 693
  router-based IP-level, 631–632
  schematic diagram of, 10

security policy for setting parameters of, 683–684
summary overview of, 696
transparency of, 685
Forensic investigations, after security compromises, 595–596
Frequency division multiple access, 267
Frequency hopping, 258

G

Gateway, *see also* Firewall
application-level, 686–688
circuit-level, 687–688
definition of, 641
description of, 170–171
session services for connecting to, 184
wireless application protocol
architectures, 279
definition of, 278
at host, 281–284
security issues regarding, 278–279
at service provider, 279–281
General Packet Radio Service, 272
Gentran Server, 8–9
Geographic portability, 252
Global Systems for Mobile Communications, 269–270
GPRS, *see* General packet radio service
Graphical user interface
autoGUI schemes, 322
description of, 313, 368
Java, 514
GSM, *see* Global Systems for Mobile Communications

H

Hackers
attack methods
dumpster diving, 705
hardware, 705
network sniffer software, 706
password cracker, 706
reverse intent, 707–709
social engineering, 703–704
software, 705–707
war dialers, 705–706
bulletin board use by, 702–703
clubs, 701
conventions, 702
definition of, 699
description of, 662
handles used by, 700–701
monitoring of, 629
newsgroups, 702–703
penetration testing by, 606
prevention of, 709
profile of, 700–701
prosecution of, 63–64
publications for, 701–702
security testing using, 590–591
synonyms for, 699
targets of, 700
HACL, *see* Host access class library
Hailstorm, 34–35
Handheld communications terminal, 238
Handheld device markup language, 200–201, 212
Handheld devices, *see also* Wireless Internet; Wireless technology
application development and networks, 213–214
database access, 213–214
security issues for, 284–285
types of, 210–211
Handshake, 642
HDML, *see* Handheld device markup language
High energy radio frequency guns, 634
High-level language application program interface, 321
Host access class library, 327–328
Host-based application-level firewalls, 632
Host Integration Server, 34
Host On-Demand, 327
Host-to-host transport layer, 642
HTML
definition of, 464
description of, 93, 97, 469
example of, 193–194, 464
extensible markup language versus, 219–220
history of, 454
user interface, 322–323
wireless Web development using, 192–194
XML versus, 454, 462, 469–470
HTTP
content type, 404
extension framework, 404–405
headers, 404
M-, 405
request methods, 404
secure, 428
servers, 498–499

S-HTTP, 428, 633, 667, 669
SOAP use of, 401–402
Human resources
E-HR
case study examples of, 135–137
description of, 134
globalization effects, 139
trends that affect, 137–139
Internet-based technology applications, 135–137
Hypertext markup language, *see* HTML

I

IBM, 771
ICMP, *see* Internet control message protocol
iDEN, *see* Integrated digital enhanced network
IEEE 802.11 standard, 258–260
IFX, 130–131
IIS, *see* Internet information server
i-Mode, 265
Impression, 358
Information systems, hypertext-enabled
architecture for, 470–471
definition of, 471
mapping rules for, 471–476
master wrapper, 471
objects of, 472–473
prototype, 476–488
wrapper, 471
XML, 480–481, *see also* XML
XSL, 482–487
Information technology
importance of, 29
outsourcing of, 30
Injunctions, 61–62
Integrated development environment, 327, 329
Integrated Digital Enhanced Network, 272
Intellectual property, 56
Interactive voice response, 141, 144
Interface definition language, 540
International standards organization open system interconnection reference model, 639–640
International traffic in arms regulations, 58–59
Internet, *see also* World Wide Web
adoption phases by company, 638–639
call centers enabled by
connection methods for
call-back, 145–146
call-through, 146–147
description of, 144–145
switched connection, 148
description of, 141–142, 144
questions to consider, 148–149
chargeback systems for use of, 80–81
client authentication, 419–422
company integration of, 638–639
connection methods
full buffered, 626
full direct, 626
standalone, 627
customer relationship management using
changes associated with, 107
live interaction, 142–143
description of, 445
export control regulations, 58–59
graphics-oriented nature of, 158
growth of, 353, 445, 754
human resources applications, 135–137
integration, 762–763
international traffic in arms regulations, 58–59
legal considerations for
contract law, 55–56
criminal law, 57–58
digital signatures, 55–56, 421–422
intellectual property, 56
jurisdiction issues, 54–55
overview of, 53–55
product liability, 57
sentencing guidelines, 58
torts and negligence, 56–57
liability issues
case study examples of, 59–60
criminal, 62
hackers, 63–64
injunctions for, 61–62
lawyer, 62–63
monetary damages for, 61
organizations versus individuals, 60–61
product, 61
publisher, 60
remedies for, 61–62
phases of, 656
security, *see* Security
wireless, *see* Wireless Internet
Internet control message protocol
description of, 645–646
PING flood attack, 695–696
Internet engineering task force, 435, 756
Internet information server, 96–98

Internet Inter-ORB Protocol, 391, 394
Internet protocol-automatic call distributor
  automatic call distributor, 143–144
  interactive voice response, 144
  voice data, 143
Internet protocol source address spoofing, 693–694
Internet server application programming interface, 295
Internet service providers
  E-services provided by, 334–335
  Web hosting by, 335
Interoperability
  description of, 758, 761
  personal communication system, 244–245
  testing of, 761
  wireless local area network, 257
Interworking, 245
Invisible Worlds, 771–772
Inxight, 772–773
IP-ACD, *see* Internet protocol-automatic call distributor
ISAPI, *see* Internet server application programming interface
IVR, *see* Interactive voice response

J

Java
  Abstract Windowing Toolkit, 502, 506
  applets, 294–295, 506–507
  application
    development of, 510–512, 533
    performance improvements
      cxreff.cpp, 716–718
      description of, 712–713
      jxreff.java, 718–721
      procedure for, 713–716
      speed of execution, 712–713
  application program interfaces
    data connectivity, 540–541
    description of, 534
    interface definition language, 540
    management, 542
    message service, 540
    security, 541
    server, 542
    transaction services, 542
  architecture neutrality of, 501–503
  characteristics of, 514
  consumer and embedded devices that use, 508–509
  C++ versus, comparisons between
    features, 514–523
    performance, 523–528, 721–722
  description of, 711, 762, 764
  development kit, 504–505, 510
  directory interface, 540
  enterprise server platform
    architecture, 538
    description of, 537–538
    JavaBeans, 538–539
  error/exception handling features of, 504
  executables, 711
  foundation classes, 506
  graphical user interface, 514
  history of, 501, 533
  integrated development environment, 510–511
  JavaBeans
    ActiveX versus, 384
    characteristics of, 315, 384, 397, 534
    customization of, 535
    enterprise, 538–539
    event handling, 535
    history of, 397
    introspection, 536
    persistence, 535
    property control, 534–535
    runtime versus design time, 536–537
    security of, 536
  JDBC, 505
  Jini connection, 509–510
  just-in-time compilation, 712
  Microsoft versus Sun, 582
  naming, 540
  object-oriented nature of, 503–505
  overview of, 511, 531, 722–723
  performance of
    assessments of, 523–529
    CPU, 531
    improvements in, 529–530
    input/output, 530–531
  platform for, 508
  RMI
    characteristics of, 396–397
    description of, 505
    limitations of, 405
  Runtime Environment, 511
  security of, 541, 676–677
  servlets, 507–508
  Swing classes, 506
  Virtual Machine, 397, 502, 508, 523, 533, 711, 723
JavaScript, 93, 97

JDBC, 505
JDK, *see* Java, development kit
Jini, 509–510
Just-in-time compilation, 529, 712
JVM, *see* Java, Virtual Machine
Jyra, 732

K

Kernel, 175

L

LAN, *see* Local area network
Legal considerations
  contract law, 55–56
  criminal law, 57–58
  digital signatures, 55–56, 421–422
  intellectual property, 56
  jurisdiction issues, 54–55
  overview of, 53–55
  product liability, 57
  sentencing guidelines, 58
  torts and negligence, 56–57
Liability
  acceptable use policies for, *see* Acceptable use policy
  case study examples of, 59–60
  criminal, 62
  hackers, 63–64
  injunctions for, 61–62
  lawyer, 62–63
  monetary damages for, 61
  organizations versus individuals, 60–61
  product, 61
  publisher, 60
  remedies for, 61–62
Linux
  applications for, 496–497
  description of, 493
  E-commerce uses of, 497–499
  features of, 495–496
  hardware requirements, 494–495
  installing of, 495
  obtaining of, 494
  prevalence of, 493
Local area networks
  chargeback systems for
    billing methods, 77–78
    considerations for, 76
    hardware, 76–77
    services included, 76–77
    software, 77–79
  wireless
    advantages and disadvantages of, 256–257
    cells of, 255–256
    characteristics of, 254–255
    configuration of, 255–256
    costs of, 257
    definition of, 254
    description of, 225, 260–261
    IEEE 802.11 standard, 258–260
    interoperability of, 257
    power consumption of, 257
    range/coverage of, 256
    reliability of, 256
    scalability of, 257
    security of, 257
    simplicity of, 257
    technology options for
      infrared, 258
      narrowband, 257–258
      spread spectrum, 258
    throughput, 256
Local exchange carriers, personal communication service interconnection with, 249–250
Logical request model, 171
Look-to-click rate, 359
Lookups, 648

M

Magic WAND project, 232
Mainframes
  chargeback systems for, 75–76
  description of, 447
  E-commerce use of, 447
  IBM, 448–449
Maintenance contract, 83
Medium access control layer, 258
Merced, 447
Mercury Interactive, 565, 577
Microchips, 446–447
Microconversion, 359
Microsoft
  Active Server Page, 330–331
  Component Object Model, 330
  transaction server, 97–98
  Windows NT, 450–451
Mobile commerce, 180
Mobile data-intermediate system, 743–744
Mobile data link protocol, 748
Mobile network location protocol, 747
Mobile network registration protocol, 746

Mobile Web Software Development Kit, 202, 204–206
Modified prototype method
  description of, 298–299
  extreme programming versus, 305–308
  traditional prototyping methods versus, 305
MPM, *see* Modified prototype method
MW-SDK, *see* Mobile Web Software Development Kit

N

National computer security association, 663
Navigation
  classic approaches, 766–767
  data display, 373
  Web site, 413
NCSA, *see* National computer security association
.NET architecture, 32–33
Net.Commerce, 433–434
NetWare, 451
Network(s)
  integrated nature of, 41–42
  interoperability of, 244–245
  interworking of, 245
  local area, *see* Local area networks
  personal communication service interconnections, 249–250
  security threats, 42–43
Network access layer, 641
Network guardians, 40, 44–46
Network node interface, 226
Newsgroups
  acceptable use policy for, 68–69
  defamation messages in, 68–69
  hacker use of, 702–703
911 emergency services, personal communication service provision of, 253–254
North American Numbering Plan number, 251
Numbers
  for personal communication service, 251–252
  portability of, 251–252

O

Object-based applications
  classes, 582
  description of, 579
  events, 582–583
  methods, 581–582
  properties of, 579–581
  testing of
    integration, 585
    manual, 583
    plan for, 585–586
    scripts, 585–586
    spiral approach, 584
    summary overview of, 586–587
    tools, 583–584
Object linking and embedding automation, 383–384
Object management group, 391
Object-oriented programming
  component-based development, 316
  description of, 513
  Java, *see* Java
Object request broker, 380
Object_Rule, 474–475, 487–488
ODC, *see* Open distributed computing
One big network paradigm
  description of, 41–42
  security considerations, 42
Open distributed computing, 392
Open financial exchange, 129–130
Open host interface objects specification, 764
Openwave, 214, 282–283
Oracle Internet Commerce Server, 434
ORB, *see* Object request broker
Organizational liability
  for acts of employees, 60
  description of, 60
Outsourcing, of E-services, 334, 338, 348

P

Packet filtering, 686, 688
Packetization, 740
Packet-switched networks, 156, 160
Packet switching, 740
Palm Os, 174–175, 210
Palm Pilot
  challenges associated with, 209
  history of, 209
  synchronization of, 209
Password cracker, 706
PCIA, *see* Personal Communications Industry Association
PCS, *see* Personal communication service
PCT encryption, 438
PDAs, *see* Personal digital assistants

PDC, *see* Personal digital communications
Penetration testing
  definition of, 599
  description of, 592–593
  destruction of information during, 610–611
  dumpster diving, 610, 705
  electronic, 604–605, 617–618
  goals of, 618
  hackers, 606
  indications for, 599
  management approval of, 612
  operational infrastructure, 604
  parameters for, 613
  performance of, 614–618
  phases of
    attack, 615–616
    occupation, 616–618
    reconnaissance, 614–615
  physical infrastructure, 603–604, 616–617
  plan for, 611–614
  purpose of, 600, 610
  reasons for performing, 602–603
  requirements for, 600
  results reporting, 618–620
  social engineering, 617, 703–704
  success of, 605–607
  switching of tactics for, 611
  terminology associated with, 601
  testing strategies for, 609–611
  time limit setting for, 613
  tools used, 616–617
  vulnerabilities assessment, 606–607
Performance tools, 571–572
Perimeter network
  definition of, 646
  testing of, 647–650
Permanent virtual connection, 226
Personal communication service
  definition of, 244
  description of, 211, 243, 260
  enhanced 911 emergency services, 253–254
  FAA tower review process for, 248
  growth of, 245–246
  interoperability, 244–245
  interworking, 245
  local exchange carriers interconnected with, 249–250
  market forecasts for, 243, 245–246
  mutual compensation rights for carriers, 250–251
  numbering issues for, 251–252
  public service obligations, 252
  radiofrequency emissions standard, 248–249
  services included, 244
  site acquisitions
    on federal property, 247–248
    issues regarding, 246–247
  subscription rates, 246
  universal service, 253
  vision of, 243
Personal Communications Industry Association
  description of, 243
  enhanced 911 emergency services, 253–254
  Federal Aviation Administration discussions with, 248
  history of, 244
  surveys conducted by, 245
Personal digital assistants
  future of, 177
  networks used by, 156
  operating systems for, 174–177
  Palm Os, 174–175
  Windows CE, 175–177
Personal Digital Communications, 272
Personal finance management systems, 120
PHP, 498
Phreaks, *see* Hackers
Physical layer, 258–259
Pico-cells, 228
PocketPCs, 210–211
Portals
  challenges associated with, 766
  corporate portal categorization solutions, *see* Corporate portal categorization solutions
  description of, 766
PostgreSQL, 497–498
Privacy enhanced mail, 428
Private-key encryption, 435, 630
Product liability, 57, 61
Profilers, 714
Protocol(s)
  application layer, 745–746
  connectionless network, 747
  definition of, 639
  description of, 756
  Internet control message
    description of, 645–646
    PING flood attack, 695–696
  message format, 639
  mobile network location, 747

mobile network registration, 746
wireless application, *see* Wireless application protocol
wireless communication, 168
wireless datagram, 185–187
wireless session, 183–184
wireless transaction, 184–185
Protocol translation, 170
Proxy servers, 647
Public-key encryption, 435, 629–630
Public-key infrastructures, 47, 428
Public service, personal communication service's role in, 252
Publisher liability, 60

R

Radiofrequency emissions standard, 248–249
Radview Software, 565–566
Rational Software, 577
Recorders, 572–573
Relationship-navigation analysis, 488
Relationship-navigation-rule analysis, 488
Remote computing, speed of, 159
Remote method invocation, *see* Java, RMI
Repeaters, 640–641
Return receipt, 425–426
Reverse intent, 707–709
Router-based firewalls, 631–632
Routers, 641
RSA encryption, 438
RSW Software, 566

S

Scripting, 329–330
Search engines, 356
Secure electronic transaction, 435, 439
Secure enterprise proxy, 282–283
Secure hypertext transfer protocol, *see* S-HTTP
Secure sockets layers
authentication, 665, 667
description of, 101, 439, 632–633
wireless transport layer security versus, 276
Security
acceptable use policy, *see* Acceptable use policy
appropriate use policy, 660–662
architecture for, 669–671
authentication, 273–275, 630
call-through connections, 147
cellular digital packet data, 742
certificate authority, 420–422, 429
certificate repository, 422–423
commercial exchange services, 427
context for, 40–41
data classification scheme, 11–12, 660
description of, 11–12
electronic postmark, 424
encryption
data, 11–12
PCT, 438
private-key, 435, 630
public-key, 435, 629–630
RSA, 438
security uses of, 278, 434–435
enterprise network, 42
enterprise systems, 11
extranet, 43–44
firewall, *see* Firewall
flaws in, 676–678
gateway, 170, 278–284
guidelines for, 678
hacker activity monitoring, 629
holes, 628–629
infrastructure, 678
internal and external applications, 662–664
Internet, 43–44
Java application programming interface, 541
JavaBeans, 536
legal issues, 633–634, 661–662
levels of, 659
network guardians, 40, 44–46
objectives, 657–658
overview of, 39–40, 653–655
policies for, 40, 42, 658–659
return receipt, 425–426
server, 628–629, 665–666
S-HTTP, 428, 633, 667, 669–671
storage and retrieval services, 426–427
testing of
audits for, 596, 672–676
considerations for, 596–597
description of, 564, 589–590
forensic investigations, 595–596
hackers for, 591
penetration, *see* Penetration testing
persons who perform, 591
reasons for, 589–590
security experts for, 591
security review, 595
test selection, 592

Tiger Teams, 593–594
vulnerability, 594–595
threats in, 39, 662
tools for, 41
types of, 11
user considerations, 664
Web applications, 296, 309
Web browser
description of, 664–665
secure socket layer, 665, 667
settings, 672
wireless Internet
authentication, 273–275
confidentiality, 275–277
description of, 272–273
malicious codes, 277
viruses, 277
wireless technology, 158–159
Security review testing, 595
Segue Software, 565, 577–578
Semio Corporation, 773
Sendmail, 650
Sequence predictability attack, 644–645
Servers
Apache Web Server, 498–499
auditability of, 674
authentication of, 665–666
description of, 756–757
domain name, 646
E-commerce, 447–450
Hewlett Packard, 449–450
IBM, 448–449
proxy, 647
security of, 628–629, 665
S-HTTP, 671
Sun Microsystems, 450
Server-side processing, 295
Service provider portability, 252
SET, *see* Secure electronic transaction
SGML, *see* Standard generalized markup language
Shopping cart, 439
Short messaging service, 160
S-HTTP, 428, 633, 667, 669–671
SIM, *see* Subscriber identity module
SMIL, *see* Synchronized multimedia integration language
SNA, *see* Systems network architecture
SOAP
definition of, 401
description of, 403–404
envelope, 407
fault codes, 406–407
HTTP use by, 401–402
importance of, 402–403
improvements in, 408
industry support for, 405
limitations of, 402
message, 406–408
technical elements of, 401–402
technologies that compete with, 405
XML messaging, 405–406
Social engineering
hacker use of, 703–704
penetration testing, 617
SQL server, 98
SSL, *see* Secure sockets layers
Staged delivery, 297–298
Standard generalized markup language, 461
Subscriber identity module, 269–270
Supply chain
automation of, 19
business-to-business exchange benefits for, 19
description of, 14
schematic diagram of, 15
Supply Chain Management, 33–34
Switched virtual connection, 226
Switch service providers, 252
Synchronized multimedia integration language, 464
SYN Flood attack, 643–644
System/390
description of, 753–755
TN3270 standard
client, 760
description of, 754–755
enhancements of, 758, 761
server, 759
TN5250 standard
client, 760
description of, 754–755
server, 759
Systems network architecture, 756

T

T-9, 160
TCP/IP
architecture of, 641
attacks on
sequence predictability, 644–645
SYN Flood attack, 643–644
description of, 639
hijacking, 694

Internet control message protocol, 645–646
layers of, 641–642
standards, 754–755
three-way handshake used by, 642–643
TDMA, *see* Time Division Multiple Access
Telcos, 31
Testing
ad hoc performance of, 550, 555
automated
description of, 549
development of, 551
scripts for, 551–552
summary overview of, 559
test library for, 551–552, 557
benchmarks, 566
capacity planning, 566–567
capture and playback tools, 551, 557–558
current practices for, 549–551
debugging, 558, 574
defect tracking, 573–575
definition of, 550
end-to-end, 564
end user's role in, 554, 557–558
growth forecasts, 566–567
integration, 563
object-based applications
integration, 585
manual, 583
plan for, 585–586
scripts, 585–586
spiral approach, 584
summary overview of, 586–587
tools, 583–584
penetration, *see* Penetration testing
performance, 562, 564
planning, 563–566
regression, 564
scope of, 563
security
audits for, 596, 672–676
considerations for, 596–597
description of, 564, 589–590
experts for, 591
forensic investigations, 595–596
hackers for, 591
penetration, *see* Penetration testing
persons who perform, 591
reasons for, 589–590
security review, 595
test selection, 592
Tiger Teams, 593–594
vulnerability, 594–595
stress, 564
system analyst's role in, 552–555
system developer's role in, 555–556
system management's role in, 556–557
test organization's role in, 552
tools for
code coverage, 571
debuggers, 570
description of, 570
performance, 571–572
recorders, 572–573
selection guidelines, 575–576
suites, 576–577
types of, 576–577
vendors who offer, 576–577
unit, 563
usability, 564
Web-based
description of, 561
difficulties associated with, 562
3G wireless networks
definition of, 160
description of, 167, 177–178, 211, 271
market introduction of, 177
packet-switched nature of, 178
transmission speed for, 271–272
types of, 272
Tiger Teams, 593–594
Time Division Multiple Access, 268–269
TN3270 standard
client, 760
description of, 754–755
enhancements of, 758, 761
server, 759
TN5250 standard
client, 760
description of, 754–755
server, 759
Touch points, 112
Transmission control protocol, 276
Transport control protocol/Internet protocol, *see* TCP/IP
Transport layer security
definition of, 276, 435
encryption, 439
2G wireless networks, 271
2Roam, 212–213

U

UMTS, *see* Universal Mobile Telecommunications System
Unit test, 555

Unit testing, 563
Universal Mobile Telecommunications System, 272
Universal service, 253
Up-sell rate, 358
UP.Simulator, 214
Usability, of Web applications
  approaches for, 368
  case study, 369–371
  costs of, 368–369
  data display navigation, 373
  data input considerations, 374–376
  description of, 367–368
  design considerations, 371, 725–732
  destructive actions confirmed, 374
  error correction, 376
  function actuation, 374
  information presentation considerations, 372–373
  instructions, 373
  interfaces for, 372–376
  misconceptions regarding, 368–369
  naming of functions, 373–374
Usenet groups, 633
User agent, 181
User datagram protocol, 642
User interface
  component-based development, 314
  rejuvenation of
    for applet-based thin-client emulators, 323–329
    application program interface, 330–331
    description of, 321–322
    levels of, 322–323
    scripting, 329–330
    for 3270/5250-to-HTML conversion, 323, 329–331
  wireless Internet, 190–192, 215
  wireless technology, 179, 215
User network interface, 226

V

Viruses, 277
Vtables, 383
Vulnerability assessments and testing, 594–595, 647

W

WAP, *see* Wireless application protocol
War dialers, 705–706
WDP, *see* Wireless datagram protocol
Web applications
  advantages of, 308, 656–657
  client-side processing, 294–295
  complex, 292
  component-based development of, *see* Component-based development
  components of, 293–294
  content-rich maintenance, 296
  cross-platform capability, 308
  definition of, 292
  development of
    architectural layout, 300–302
    challenges associated with, 295–297
    component-based, *see* Component-based development
    deve-maintenance cycle, 303–304
    history of, 389–392
    initial versions, 302–303
    methodology, 297–304
    staged delivery, 297–298
    steps involved in, 361–362
    system analysis and design, 298–299
    user involvement in, 307–308
  disadvantages of, 309
  interactive, 292
  layout of, 292–293
  legacy systems, 296–297
  load balancing, 296
  maintenance of, 304
  overview of, 291–292
  prototyping model, 297–298
  scability of, 295–296
  security of, 296, 309
  server-side processing, 295
  static, 292
  summary overview of, 309
  testing of, 304–305
  usability of
    approaches for, 368
    case study, 369–371
    costs of, 368–369
    data display navigation, 373
    data input considerations, 374–376
    description of, 367–368
    design considerations, 371
    destructive actions confirmed, 374
    error correction, 376
    function actuation, 374
    information presentation considerations, 372–373
    instructions, 373

interfaces for, 372–376
misconceptions regarding, 368–369
naming of functions, 373–374
Web browser
description of, 664–665, 762
secure socket layer, 665, 667
Web Clipping wireless architecture, 171–173
Web hosting
description of, 334–335, 344–345
E-services necessary for, 345–347
Web site
accuracy of, 726
advertising on, 358–359
application tiers for
business tier, 99–100
client tier, 99
data tier, 100
presentation tier, 98–99
business objective stated in, 727
conceptual application model for, 100–101
confidentiality issues, 275–277
confidential transactions performed using, 101
consumer confidence in safety of, 726–727
content of, 298, 726
core foundation technology for
components of, 97
Internet information server, 96–98
SQL server, 98
transaction server, 97–98
corporate requirements for, 95–96
costs of, 360–361, 728
customer relationship management package integrated with, 360
customer service, 142, 360
databases
data organization, 731
design of, 731
performance improvements, 730–731
requirements for, 729–730
size of, 731
designing of, 93–96, 356–357, 371–372, 413–415, 433–434, 725–732
development of
in-house, 411–412
outsourced, 412–413
downloading considerations, 356
E-commerce, 416
ergonomics of, 727–728
firewall for, 101–102
graphical animations on, 413
informational, 416
information on, 415
interface design for, 94–96
layout of, 94
maintenance of, 728–729
marketing considerations, 358–359
navigation of, 413
page content for, 94
performance evaluations, 732
pornographic, 637
presentation of, 93–94
privacy considerations, 357–358
repeat visits of, 355–356
requirements for, 356
sales channels, 359
scope of, 726
search engines, 356
security for, 101, 729
size of, 728
software for designing, 433–434
statistical information tracking, 415
systems integrated into, 359–361
technical competence of, 726
technology requirements for, 94–95
traffic analysis, 358
trust in, 726–727
types of, 415–417
usability of
approaches for, 368
case study, 369–371
costs of, 368–369
data display navigation, 373
data input considerations, 374–376
description of, 367–368
design considerations, 371, 725–732
destructive actions confirmed, 374
error correction, 376
function actuation, 374
information presentation considerations, 372–373
instructions, 373
interfaces for, 372–376
misconceptions regarding, 368–369
naming of functions, 373–374
user interface considerations, 355–356, 371–372
WebSuite, 8–9
WebTrust, 436–437
Windows CE operating system, 175–177
Wireless application environment, 181–186
Wireless application protocol
architecture, 168–171, 180–187, 212
components of, 217

definition of, 157, 160, 196
description of, 211–212
development of, 217
gateway, *see also* Gateway
architectures, 279
definition of, 278
at host, 281–284
security issues regarding, 278–279
at service provider, 279–281
indications, 196
security measures
description of, 276
encryption, 278
gateway, 278–284
Web Clipping wireless architecture versus, 173–174
Web sites for, 215

Wireless application protocol forum, 276
Wireless application server provider, 212
Wireless architecture model
communication infrastructure, 169–170
communication protocol, 168–169
description of, 168
gateway, 170–171
logical request model, 171
Web Clipping, 171–173

Wireless asynchronous transfer mode, for data centers
advantages of, 226, 227
applications for, 226, 239–240
architecture, 228–230
Arielnet system, 239–240
ATM wireless access communication system project, 234–236
bandwidth, 227–228
connections, 226
description of, 225–226, 228
hardware
handheld communications terminal, 238
schematic diagrams of, 236–238
wireless interface processor, 238–239
interfaces for, 226
international joint ventures for, 236
network demonstrator, 231–234
prototypes, 230
reasons for, 226–228
remote data collection, 239
standards, 230–231
summary overview of, 240

Wireless ATM, *see* Wireless asynchronous transfer mode

Wireless carriers, mutual compensation for, 250–251
Wireless communication protocols, 168
Wireless datagram protocol, 185–187
Wireless interface processor, 238–239
Wireless Internet, *see also* Handheld devices
applications available using, 265–266
confidentiality issues, 275–277
description of, 263, 284
development of
application service provider, 194–196
data access methods, 194–196, 211–212
data entry considerations, 191
description of, 189
handheld device markup language, 200–201
HTML use, 192–194
indications for, 190
Mobile Web Software Development Kit, 202, 204–206
user interface requirements, 190–192, 215
wireless access versus synchronization, 190
wireless application protocol, 196
wireless markup language, 196–197
XML, 201–202
XSL, 201–202
digital transmission methods
Code Division Multiple Access, 270–271
eavesdropping, 268–269
encoding techniques for, 268–272
Global Systems for Mobile Communications, 269–270
interception of, 266–267
providers of, 267
Time Division Multiple Access, 268–269
estimated use of, 264
growth of, 264–265
limitations of, 264–265
security issues
authentication, 273–275
confidentiality, 275–277
description of, 272–273
malicious codes, 277
viruses, 277
services available, 265–266

Wireless local area network
advantages and disadvantages of, 256–257

cells of, 255–256
characteristics of, 254–255
configuration of, 255–256
costs of, 257
definition of, 254
description of, 260–261
IEEE 802.11 standard, 258–260
interoperability of, 257
power consumption of, 257
range/coverage of, 256
reliability of, 256
scalability of, 257
security of, 257
simplicity of, 257
technology options for
infrared, 258
narrowband, 257–258
spread spectrum, 258
throughput, 256
Wireless markup language
definition of, 182, 196, 218, 464
description of, 212, 217, 456
document creation using, 219–223
emulator, 223
example of, 197–198, 221
graphics supported, 218–219
indications for, 196–197
letters, 222
overview of, 218–219
servers, 223
terminology associated with, 219
Wireless markup languageScript
description of, 182–183
example of, 197–200
goal of, 197
Wireless networks
access technologies for, 179, 211
database access methods, 213–214
integration platforms for, 178
2G, 271
3G
definition of, 160
description of, 167, 177–178
market introduction of, 177
packet-switched nature of, 178
Wireless notebooks, 160
Wireless session protocol, 183–184
Wireless technology
advantages of, 210
benefits of, 163, 210
business impact of, 164–168
central processing units for, 178
commerce benefits of, 180
compatibility issues, 156–157
costs of using, 157–158
data access methods, 194–196, 211–212
database access, 213–214
description of, 155–156
enhanced 911 emergency services available by, 253–254
future of, 177–180, 215–216
handheld devices, *see* Handheld devices
lack of suitable content for, 158
lexicon associated with, 160
limitations of, 210
location information using, 165
mobility of, 163–164
network providers for, 167–168
personal productivity benefits of, 165
productivity gains associated with, 165
security issues, 158–159
service providers for, 167–168
speed of, 159
terminology of, 160
transmission rates, 191
types of, 155
user interfaces for, 179, 215
value of, 163–164
wireless application protocol, 168–171
wireless architecture model, 168–171
Wireless telephony application, 181
Wireless transaction protocol, 184–185
Wireless transmission layer security, 159
Wireless transport layer security
description of, 185, 276
secure sockets layers versus, 276
WLAN, *see* Wireless local area network
WML, *see* Wireless markup language
WMLScript, *see* Wireless markup languageScript
World Wide Web, *see also* Internet
acceptable use policy for, 67–69
applications, *see* Web applications
connection methods
full buffered, 626
full direct, 626
standalone, 627
description of, 625, 655
HTML document browsing on, 470
human resources application, 137, 140
marketing terminology, 358
secure, 428, 672, *see also* Security
server, 293–294
wireless application protocol-enabled applications, 212
WSP, *see* Wireless session protocol

WTLS, *see* Wireless transport layer security
WTP, *see* Wireless transaction protocol

X

X.400, 746
X.500, 746
XLink, 466
XML
- accuracy of, 461
- associated standards, 467–468
- benefits of, 461–462, 469–470
- characteristics of, 461–462
- coding of, 463
- context of, 454
- database management systems support of, 456–457
- data exchange using, 455–456
- definition of, 453
- description of, 16, 18–19, 201–202, 453
- document-type declaration, 465
- document-type definition, 453, 465, 476–478
- electronic business (ebXML), 464
- electronic publishing use of, 454–455
- history of, 454, 459
- HTML versus, 219–220, 454, 462, 469–470
- hypertext support using, 470
- information systems communication using, 456, 460
- linking language, 466
- namespaces, 467
- parser, 464
- path language, 467
- sample document, 463, 480–481
- SOAP use of, 405–406
- standards, 457–458, 463
- summary overview of, 458
- tags used, 462–463
- value of, 454–456, 459–460
- vocabularies, 458, 463–465
- Web searching use of, 455

XP, *see* Extreme programming
XPath, 467
XPointer, 467
XSL, 454–455, 467, 476, 482–487
XSLT, *see* XSL transformations
XSL transformations, 467

Z

Zone transfer, 648

INTERNET TECHNOLOGY

Architectures for E-Business Systems

Editor **Sanjiv Purba**

THE AUERBACH BEST PRACTICES SERIES

As dot.com companies grapple with rigid market conditions and we keep hearing how the big technology players are being punished on Wall Street, it becomes easy to think of the Internet as a fad. The Internet frenzy may have subsided, but interest in the Internet as a business and marketing tool is still strong. It will continue to impact organizations and create opportunities.

Sooner or later every organization will use the Internet for some facet — large or small — of its business. **Architectures for E-Business Systems: Building the Foundation for Tomorrow's Success** provides complete coverage of best practices and architecture applications. The book gives hands-on details to the IT manager faced with the daunting task of transitioning 40 years worth of computing detritus supporting a brick-and-mortar operation into an online business — melding the walk-in customer with the surf-in customer. It highlights strategy and planning, E-enabled business solutions, wireless and mobile business solutions, project development approaches, E-enabled architecture and design, toolkits, testing, performance, and security.

Many brick-and-mortar companies looking to grow their businesses through the Internet will find numerous new opportunities. With its focus on strategic and tactical knowledge **Architectures for E-Business Systems: Building the Foundation for Tomorrow's Success** shows you how to successfully build and deploy Internet applications that stand up to the rigors of today's demanding business environment.

FEATURES

- Details architectural strategies for building industrial-strength Internet and E-business applications
- Recommends Web development tools and environments for building mission-critical Internet applications
- Discusses the Internet languages an IT organization should focus on for its applications; their current status and future trends
- Includes strategies that will harness the resources of new applications and old legacy systems
- Shows how to verify the latest security technologies and strategies have been effectively implemented to provide industrial-strength authentication, encryption, hacker-attack protection, and privacy

Sanjiv Purba is Practice Director for Microsoft Services E-Solutions, Canada. Prior to that he was Senior Manager with Deloitte Consulting and national leader of the object-oriented and E-business practices for financial institutions. Sanjiv is a frequent public speaker and a regular contributor to IT publications.

AU1161

ISBN 0-8493-1161-6

90000

9 780849 311611

AUERBACH PUBLICATIONS

www.auerbach-publications.com