Supporting E-Learning with Technologies for Electronic Documents

Edited by
Alistair Mclean & Leila Alem

AACE

Supporting E-Learning with Technologies for Electronic Documents

Edited by
Alistair Mclean & Leila Alem

AACE

Association for the Advancement of Computing in Education
www.aace.org

Supporting E-Learning with Technologies for Electronic Documents

Articles

Supporting E-Learning with Technologies for Electronic Documents (ISBN 1-880094-70-3) is published by the Association for the Advancement of Computing in Education (AACE), an international, educational, nonprofit organization. Published by: AACE, PO Box 1545, Chesapeake, VA 23327-1545, USA

757-366-5606; Fax: 703-997-8760; E-mail: info@aace.org
Website: http://www.aace.org

Preface

Special Issue: Supporting E-Learning with Technologies for Electronic Documents

This special issue explores how to complement traditional e-learning techniques with technologies for electronic documents in order to assist e-learning. We use the phrase "technologies for electronic documents" to mean processes and algorithms that operate on unprocessed electronic content and, in general, the research undertaken in these areas assumes that the domain of application is open, changing and unstructured.

Techniques being developed in the information retrieval and knowledge management communities, for example, include mechanisms to describe the semantics of document fragments and relationships between documents, tools for automatically deriving and visualising the concepts within a document and between documents, authoring tools for adding metadata to documents, the use of task and user models to improve the relevance of documents retrieved by search engines.

In asking this question, we received papers ranging from those that address core document modelling issues to papers that discuss how to use analysis of users' interaction with electronic content in a collaborative environment. The papers encompass the following themes: document reuse, concept and task modelling, collaboration between learners and dealing with dynamic open environments.

There are clear synergies between research occurring in the fields of electronic document technologies and AIED. We hope this special issue will encourage further work at the boundary between the two areas.

ALISTAIR MCLEAN & LEILA ALEM

Supporting Learning and Information Sharing in Natural Resource Management with Technologies for Electronic Documents

LEILA ALEM AND ALISTAIR McLEAN
CSIRO ICT[1], Australia
leila.alem@csiro.au
Alistair.McLean@voxsurf.com

Community participation is central to achieving sustainable natural resource management. A prerequisite to informed participation is that community and stakeholder groups have access to different knowledge sources, are more closely attuned to the different issues and viewpoints, and are sufficiently equipped to understand and maybe resolve complex issues (salinity, ecosystem stability, erosion, grazing, nutrients, etc.). Our project objective is to research and develop methods and technologies for supporting learning and information sharing among community members. The backbone of such an environment is an open, dynamic and evolving knowledge map composed of documents, knowledge models (ecosystems models, data models) and people information. Users are able to navigate through electronic resources via the evolving knowledge map and to gain greater understanding of the relationships, issues, relevant information sources and people, thus enabling them to hold more informed discussions. This paper describes our research framework and our initial implementation and evaluation in the context of a case study in Douglas Shire: a sugar cane area in far north Queensland.

"Sustainability is better seen as a measure of the relationship between the community as learners and their environment rather than an externally designed goal to be achieved," (Sriskandarajah, Bawden, & Packham, 1991).

In Australia the water crisis is worsening. There is a major problem with drought and our use of natural capital is not sustainable. The community is crying out for sustainable solutions. This has lead to one of the biggest consultation exercises in natural resource management (NRM) undertaken in Australia involving government, industry, the community, Co-operative Research Centres (CRCs), universities and Commonwealth Scientific & Industrial Research Organisation (CSIRO). All these stakeholders took part in the formulation of the business case of the CSIRO Healthy Country Flagship Program (CSIRO, 2003).

The Healthy Country Flagship Program's aim is to achieve sustainable NRM through informed participation by engaging community and stakeholders' groups. A prerequisite to efficient, constructive participation is that community and stakeholders groups have access to different knowledge sources, are more closely attuned to the different issues and viewpoints, and are sufficiently equipped to understand (and maybe resolve) complex issues (salinity, ecosystem stability, erosion, grazing, nutrients, etc.).

Traditional support for knowledge sharing and learning approaches has mostly focused on documents sharing. These approaches have been successful only in very specific and constrained environments where the task people are engaged in is well defined, the people are collaboratively working towards a common goal and people are within a similar practice. In contrast, decision making in natural resource management takes place among various communities (social, cognitive and political), with different practices (farmers, tourist operators, state and federal regulatory agencies, etc.), engaged in ill-defined, complex tasks with conflicting goals using various information types (databases, documents, decision support systems, ecosystem models, etc.). In particular, from our research in this area, we have noted that:

- Little knowledge generated in science is directly impacting the practices and decision making within the communities. NRM plans and strategies need to be continually adapted to reflect new scientific knowledge.
- Land managers are central to achieving sound management of land, water and vegetation resources and to addressing critical issues such as salinity, yet they do not always have the required information to make sound decisions about the management of natural resources.
- Local and indigenous knowledge is not always taken into account. This leads to reduced community ownership of local problems, and little adoption of new methods and policies (Productivity Commission Report, 2003).
- There is little understanding of other stakeholders' views and issues.

It is clear that support for knowledge sharing is key to searching for sustainable NRM solutions. We have taken the approach that by improving the access of the stakeholders to relevant information, supporting idea sharing, model exploration, information annotation (i.e., with local knowledge), and providing an editing mechanism to capture new knowledge, we will alleviate some of the issues identified above.

Supporting Learning and Knowledge Sharing

As global competition based on knowledge intensive products/services rapidly increases, many organizations are seeking ways to harness knowledge through business strategies and Information & Communication Technology. Computer networks, Internet and Intranet, e-mail, bulletin boards, groupware, workflow, news groups, data warehousing, decision support systems, Lotus Notes etc. have already become important media for knowledge creation, sharing, and transmission (Liebowitz, 1999; Macintosh, 1994; O'Leary, 1997). These tools are core to knowledge management. Research in technologies for supporting learning and knowledge sharing often uses a combination of:

- The data/document approach using databases and document repositories with associated data mining and search engine facilities.
- The knowledge-based approach uses ontologies representing knowledge models (Gandon, 2001; Decker, Erdmann, Fensel, & Studer, 1999), cases representing past experiences (Simon & Granbastien, 1995), lessons learned representing current practices (Alem, 1998).
- The people finder approach with associated yellow pages, expertise finding systems (McLean, Vercoustre, & Wu, 2004; Craswell, Hawking, Vercoustre, & Wilkins, 2001), peer helper technologies (McCalla, Greer, Kumar, Meagher, Collins, Tkatch, & Parkinson, 1997). The expertise finding capability has been coupled with lessons learned corporate memory (Alem & McLean, 2003).
- The collaborative approach using computer supported co-operative work, video conferencing and mind mapping technologies. Work by knowledge models (a conceptual model of the domain and a meta model describing the terminology structure) have been used for supporting collaborative work (Kethers, von Buol, Jarke, & Rudolf Repges, 1998).
- The community centered approach uses interaction through online communities of practices (COPs) and communities of interest (COIs), chat rooms, or bulletin board technologies (Preece, 2002; Brown, van Dam, Earnshaw, Encarnacao, Guedj, Preece, Shneiderman, & Vince 1999; Walker and McCown, 2003).

The limitations of the database, document and knowledge based approaches include operating in a very constrained environment where the tasks people are engaged in are well defined, with people who are collaboratively working towards a common goal within a similar practice (automotive engineers, aerospace engineers, offshore oil operators). They commonly work within one organization whose leaders are supportive of knowledge sharing.

Furthermore, the knowledge based approach is often very labour intensive. Building the ontology, maintaining it, and manually annotating documents requires a great deal of work. There is a need for a more cost-effective (light weight) approach. Also, as far as we know, the knowledge based approach has dealt mostly with representing and exploiting ontologies and lessons learned models. We do not know of any work using this approach that represents and exploits more physical models (of ecosystems, for example).

Finally, the community-based approach is often restricted to supporting one specific practice, for example, farmers (Walker, Cowell, & Johnson, 2001) or health practitioners (Preece, 2002). As far as we know, little has been done in linking the community centred approach with the data/document and knowledge based approach.

It is also significant that the impact of knowledge sharing is generally not evaluated. Important questions need to be addressed, such as whether knowledge sharing led to learning, better community understanding, and better environment management.

These needs suggest that supporting learning and knowledge sharing in NRM requires an integrated approach, combining a community-centred approach with data/document and knowledge-based approaches, supporting not only document sharing, but also idea and view sharing and collective exploration of ecosystem models in an open, evolving and networked information environment.

Our Proposed Research Framework

Our aim is to support individual participation as well as group participation, in a knowledge sharing and learning environment where relationships between participants are being developed (through collaboration models and social networks), resources are being used (documents, web resources, models), new information/knowledge is created and shared (collaborative design of NRM plans and strategies, collaborative design of biophysical models) and learning is supported (e-learning).

Our framework (Figure 1) incorporates the three dimensions: information (documents), discussion (forum) and knowledge models (domain knowledge, user models, biophysical models, simulation models), all dynamically evolving through users' participation (as suggested by the large arrow).

Central to our approach is the notion of designing lightweight models and leveraging on these models for supporting learning and knowledge sharing. This approach is more cost effective than the knowledge intensive approach such as that of (Gandon, 2001). A knowledge model in a form of a knowledge map is composed of a set of topics (concepts), and associations between the topics (is-a can-contribute-to, can-reduce, etc.). The knowledge map is used as the basis for designing a structured and semantically rich navigation space and is also used for generating advanced queries of the docu-

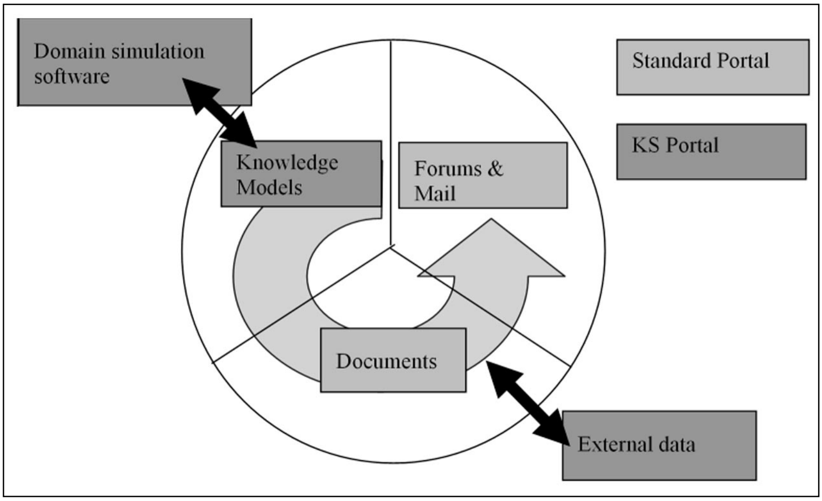

Figure 1. The knowledge sharing framework

ment collection. We can associate with each concept a set of query words and other search operators that are to be used to retrieve external (i.e., web) information when the user is examining this concept. The query can be created manually, by an expert or the participants themselves, or by the system, through analysis of the knowledge map. We argue that the combination of model-driven navigation and an advanced search facility provides access to relevant information to a specific topic in the context of the topic's relations and associations with other topics. In doing so, the environment supports learning about the domain.

For our research we examined water quality issues in the Wet Tropics coastal region in North Queensland (Douglas Shire). We created a knowledge map for this domain that includes the following high level concepts: *land use, land practices, effect on pollution* and *water quality issues*. For example (see Figure 2), *horticulture* is a type of land use in the Wet Tropics; *use of chemical* and *irrigation* are current practices in *horticulture; herbicides, pesticides* and *fertilizers* are types of *use of chemicals*; the *use of chemicals* contributes to *contaminant runoff* which then is a *water quality issue*.

Consider a user is interested in finding out about the impact of horticulture practices in the Wet Tropics. The user can navigate the map, find related water quality issues as well as open relevant documents related to the use of chemicals. Relevant documents are made accessible and presented in a form that increases the user's awareness about how information relates to each other.

We also promote the notion that the environment should be open and able, where appropriate, to take advantage of external information. One example is

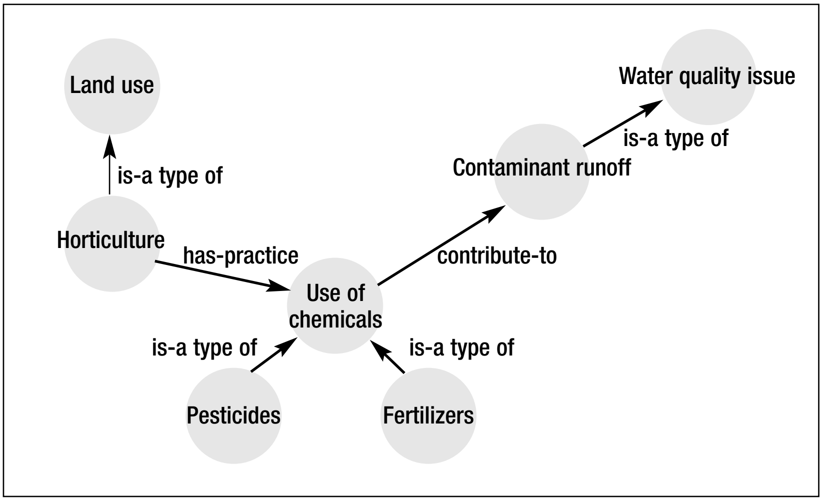

Figure 2. An example knowledge map

simply using external web resources that are well known sources of domain specific knowledge such as Department of Sustainable Environment (DSE), Catchment Management Authority (CMA), Department of Primary Industry (DPI), etc. Thus information delivered to a particular user in response to a query may not just come from information captured within the environment, but also from external web sites and databases. This helps to ensure that the information gathered and presented to the user is up-to-date. Another example is to use a people finding tool, such as that described in (McLean et al., 2004) that automatically extracts evidence of expertise for individuals and can be used to recommend relevant people as well as information. In this way, we hope to encourage further discussion amongst the participants.

Our knowledge sharing environment extends classical portal architectures in the following three ways:

- We add an explicit knowledge model level to support better access to and understanding of information.
- We open the portal to external web sources that are dynamically added to the portal.
- We offer a participation component that elicits people expertise, roles, networking and involvement in discussion.

The next session describes our initial implementation, in the context of our engagement with Douglas Shire community.

Our Initial Implementation

The project objective is to electronically capture and deliver information to targeted members of the DS community (mostly MAS and DSC) in a way that helps understanding of water quality issues and water quality monitoring data. Our research hypotheses were 1) access to information and data provides insights and hence may lead to better decisions and 2) sharing of existing information and knowledge among the various members of the DS community will support shared understanding and hence may lead to improved participation.

Our initial implementation is aimed at delivering relevant information to a user through leveraging domain knowledge encoded in the knowledge network. For the network we gathered information from domain users and domain literature and used a topic map representation[2]. Our knowledge map contains over 150 entities and 250 relationships[3]. The top level map is shown in Figure 3(a) and a sample graphical representation of part of the actual map is shown in Figure 3(b). We also harvested and indexed document data from a set of around 25 websites amounting to about 15 Gigabytes of data. Each entity had a default query that could be used against this document set; however, for many entities we manually crafted queries that were more specific. We provided some textual description for some of the entities which gave a short summary of the concept and the system generated a concept description that not only used this text but also described how a particular entity was related to other entities. Finally we also manually attached documents, people, and database records to certain entities where we were confident that the attached document was highly relevant. This provided an initial information space in which the users could browse and is illustrated in

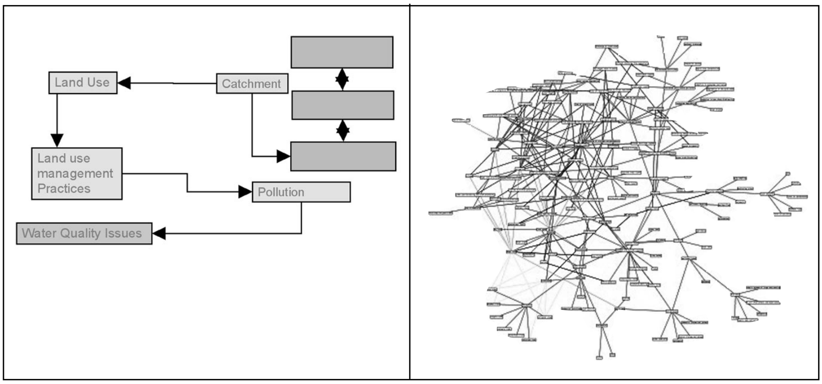

Figure 3. (a) Top level representation of the knowledge map (left) (b) example knowledge map (right)

Figure 4. In this figure the left-hand panel shows the knowledge map and is primarily used as a navigation panel. The top right panel provides a natural language description of the current node and its relationship to other entities in the network. This description is generated automatically. The central panel shows all the data objects that have been manually attached to this particular node. These data objects can be documents, web pages, database records, people, images, and in the future video. The bottom right panel shows the results of the query for this entity on the harvested data in a standard search page format. If the user decides that a particular result is of high relevance he can attach that document to the entity by clicking on the image link next to the result. The user can also refine the search query in this pane.

A user can also annotate a particular entity by right-clicking on the current concept name in the navigation panel, or by right-clicking on the link of an attached document on the attached-document panel. This allows the user to select an option to annotate the item and he can then provide a comment or note. In this way, we can support the capture of local knowledge and the collective creation of new knowledge.

We also provide a map-based entry to the browser. There are 12 water quality stations in Douglas Shire at various river sites. These stations are

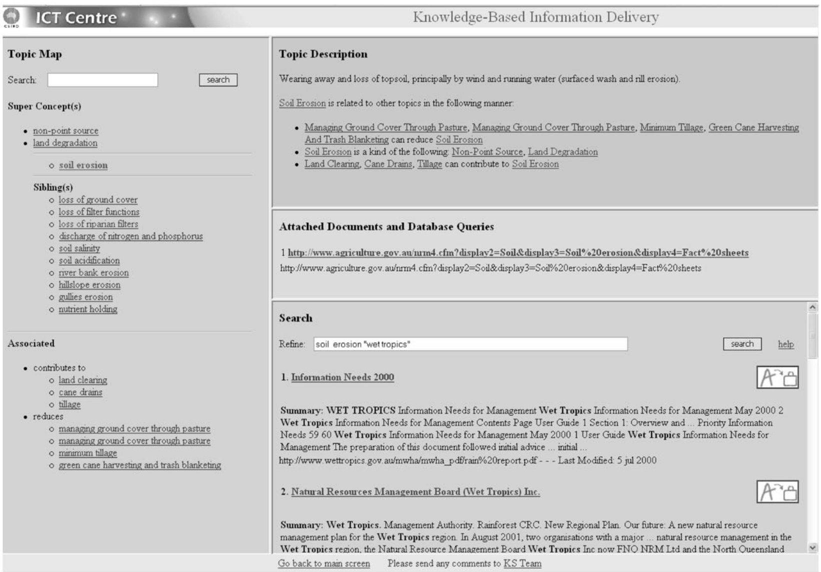

Figure 4. The browser screen of our implementation. (See the text for a full description.)

measuring water quality as part of another project in which CSIRO is participating. We are able to access web services that allow us to query the location of the sites. We use this to dynamically create a map of Douglas Shire showing each water quality station that is active. We then map the locations to relevant concepts in the knowledge map. If, for example, a water quality station is situated in the middle of a sugar cane farming area, we allow the user to select the "sugar cane" concept from a drop-down list of relevant topics when he clicks on the water quality station on the map (see Figure 5).

Figure 5 also shows the main screen of the editor. The editor mode allows users to add, edit or delete new concepts and new relationships, link concepts by a relationship, add/remove contacts and add/remove attached documents.

In a collaborative environment with many users, the idea is that this open, web-based information space will support the capture of knowledge from different users. Although we have not implemented linked discussions yet, we hope that the browser can provide an entry point into informed discussion about natural resource management issues.

Evaluation of the Prototype

There are three dimensions of our approach that could be evaluated:

- Learning: how effective the approach is in terms of improving end users' level of understanding of water quality issues and their relations to land uses and land practices.
- Capture: how effective the system is in terms of supporting the authoring of the Knowledge Map: adding/removing concepts, adding/removing links, adding/removing information resources (documents, data, spatial maps).
- Sharing: how effective the approach is in terms of supporting document sharing, models sharing, experience sharing.

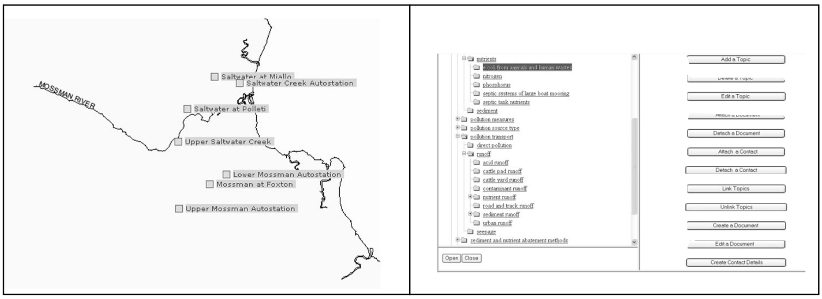

Figure 5. (a) The map-based entry into the knowledge map browser showing water quality measuring stations in the Douglas Shire (left)
(b) The main editor page (right)

The evaluation we are reporting on here examined the first dimension only. The evaluation of our approach has been performed with 16 students from the Department of Arts at Monash University. The evaluation consisted of the following steps:

- Assess the student's computer skills, domain knowledge and spatial awareness (using student profile questionnaire).
- Provide time for the student to become familiar with the knowledge portal.
- Ask the student to answer five review questions.
- Assess the student's overall learning experience using the knowledge portal (using an exit-questionnaire).

Then, for each of the tasks and for each student:

- Assess the student's prior knowledge (using pre-knowledge evaluation questionnaire).
- Assess the student's post-knowledge (using post-knowledge evaluation questionnaire).
- Capture the student's log.

The five questions were:

Q1: Explain the nature of the impact of pig-digging on water quality in Douglas Shire.

Q2: What factors affect water quality in the Douglas Shire, and how do they relate to each other?

Q3: What practices minimize horticultural impact on the environment?

Q4: List causes of water system degradation in Douglas Shire.

Q5: What are "sea grass meadows" or "sea grass beds," and what are the consequences of sea grass meadow degradation?

The students were asked to answer the five questions in a different order (Figure 6) to compound the factor related to learning about the system.

An expert was asked to assess learning gain based on student's pre- and post-answers to questions using the following score: 0 no change between

Student 1, 2, 3,4	Q1Q2Q3Q4Q5
Student 5, 6,7	Q5 Q1Q2Q3Q4
Student 8,9,10	Q4Q5 Q1Q2Q3
Student 11,12,13	Q3 Q4Q5 Q1Q2
Student 14,15,16	Q2 Q3 Q4Q5 Q1

Figure 6. Order of questions

pre- and post-answer, 25% positive change, 50% positive change, 75% positive change, and 100% positive change. Figure 7 shows the students' overall learning gain per question.

The learning gains of students with low initial knowledge level are shown in Figure 8.

Figure 9 compares the average learning gain for the three different groups: students with a high level of domain knowledge (series1), all stu-

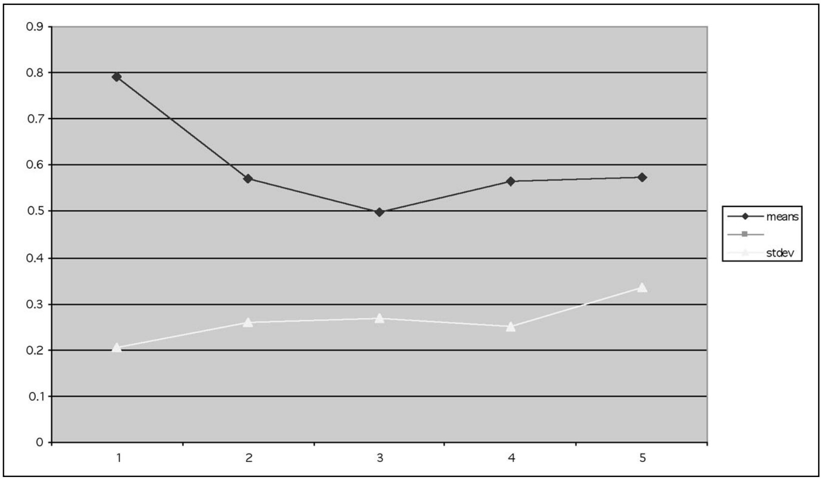

Figure 7. Overall learning gain per question

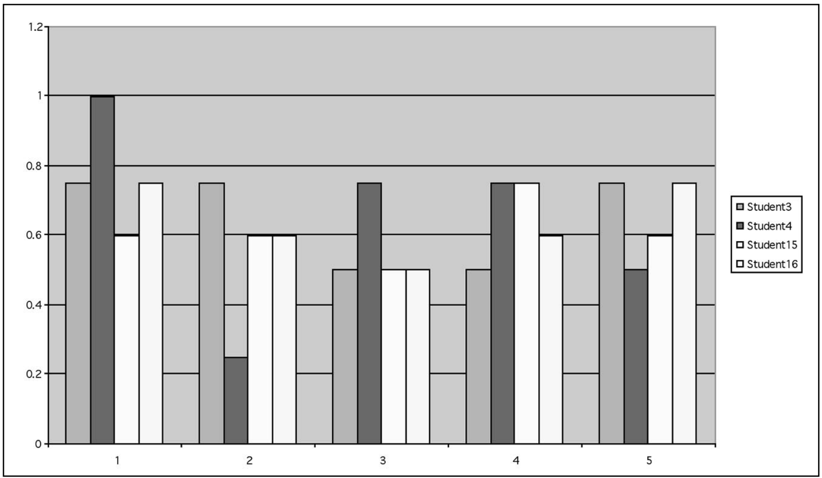

Figure 8. Learning gains of student with low knowledge level

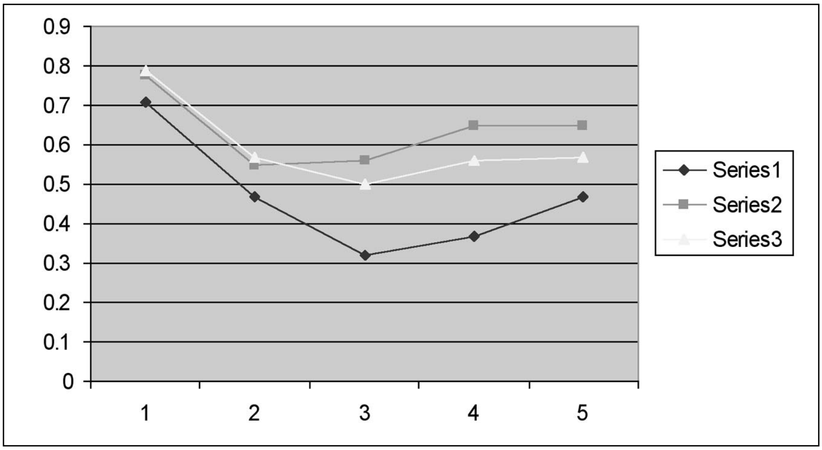

Figure 9. Average learning gain by groups

dents (series2), and students with a low level of domain knowledge (series3). The figure shows that students with a high level of domain knowledge learned less than students with a low level of domain knowledge, and students with a low level of domain knowledge have similar learning gains to the average of the whole group. We compared the distribution of learning gains per question by using the two-tailed paired t-test. Statistically significant differences between the two groups occurred when the result of the t-test was less then 0.05. We obtained: t-test (Low, All) = 0.173477 and t-test (High, Low) = 0.016521. Thus, we can see that the High group has learning gains that are statistically different from the Low group, but the Low group could not differentiate the learning gains from the whole group average.

CONCLUSION

In this paper we have proposed a research framework for supporting learning and knowledge sharing by making use of technologies for electronic documents. The backbone of such an environment is an open, dynamic and evolving knowledge map composed of documents, knowledge models (ecosystems models, data models) as well as people. Users are able to navigate through electronic resources via the evolving knowledge map and to gain greater understanding of the relationships, issues, relevant information sources and people, thus enabling them to hold more informed discussions. We have presented our initial implementation and evaluation in the context of a case study in Douglas Shire: a sugar cane area in far North Queensland. Initial analysis of our evaluation data indicates that our environment supports learning for different profiles of learners. Learners with a low level of domain knowledge have better learning gain than learners with a high level domain knowledge.

Our next step is to further analyse our evaluation data and explore the various factors that may explain learning gains. Our second step is to assess with members of the community in Douglas Shire, the extent to which our technology is effective in terms of supporting capture and sharing. But as stated by Huysman & DeWit (2002):

One should not fall into the known trap of assuming that it is the use of these technologies that stimulates people to communicate and share knowledge. The first thing to be addressed is the question of how to stimulate a need to share knowledge among a group of people. It is only when this need is satisfied that physical and electronic spaces are used for knowledge sharing purposes.

When such a need exists, we can envisage other tools that may extend the knowledge sharing space. These tools include a people finder tool that helps to locate people with the required expertise together with some evidence of the expertise. A facilitator can use this tool to solicit an expert to engage in a current discussion. Whereas a person can currently be manually attached to a concept, a people finder tool uses the document and data, both within the portal and pointed to by the portal, as evidence for people's expertise and finds people automatically. We can also envisage support for collaborative creation of new models as a result of the discussion. As we place our emphasis more on collaborative understanding and sharing rather than information access and creation, we can imagine tools that support group awareness, measure the level of participation, analyse the participation process, and even visualise credibility based on expertise, reputation, contribution and social network analysis.

References

Alem, L. (1998). *Learning in the workplace: Initial requirements of a lessons learned centred corporate memory*. Proceedings of the Intelligent Tutoring Systems: 4th International Conference, San Antonio, Texas.

Alem, L., & McLean, A. (2003). *Technologies for electronic documents to support learning*. Paper presented at the AI-ED Workshop, Sydney, Australia.

Brown, J., van Dam, A., Earnshaw, R., Encarnacao, J., Guedj, R., Preece, J., Shneiderman, B., & Vince, J. (1999). Human-centred computing, online communities and virtual environments. *ACM SIGCHI Interactions, 6*(5), 9-16.

Craswell, N., Hawking, D., Vercoustre, A., & Wilkins, P. (2001, April). P@NOPTIC Expert: Searching for experts not just for documents. *Proceedings of Ausweb 2001*, 21-25.

CSIRO (2003). Healthy Country, A Case for CSIRO Investment. Internal Report.

Decker, S., Erdmann, M., Fensel, D., & Studer, R. (1999). Ontobroker: Ontology based access to distributed and semi-structured information. (R. Meersman, Z. Tari, & S. Stevens, Eds.). *DS-8: Semantic Issues in Multimedia Systems*. Berlin: Kluwer Academic Publishers.

Gandon, F. (2001). Experience in ontology engineering for a multi-agents corporate memory system. *Proceedings of the Workshop on Ontologies and Information Sharing, IJCAI 2001, Seventeenth International Joint Conference on Artificial Intelligence,* 119-122.

Huysman, M., & DeWit, D. (2002). *Knowledge Sharing in Practice.* Norwell, MA: Kluwer Academic Publishers.

Kethers, S., von Buol, B., Jarke, M., & Rudolf Repges, R. (1998). Lessons learned from cooperative terminology work in the medical domain. *Proceedings of MedInfo 1998*, 628-632.

Liebowitz, J. (1999). *Knowledge Management Handbook*. Boca Raton, FL: CRC Press.

Macintosh, A. (1994). Corporate knowledge management: State of the art review. *Proceedings of the International Symposium on the management of industrial and corporate knowledge (ISMICK94)*, 26-27.

McCalla, G., Greer, J., Kumar, V., Meagher, P., Collins J., Tkatch, R., & Parkinson, B. (1997). A peer help system for workplace training. *Proceedings of Artificial Intelligence in Education*, IOS Press.

McLean, A., Vercoustre, A., & Wu, M. (2004). *Combining structured corporate data and document content to improve expertise finding*. Submitted for publication.

O'Leary, D.E. (1997). The Internet, Intranets and the AI Renaissance. *Computers, 30*(1), 71-78.

Preece, J. (2002). *Online communities: Designing usability and supporting sociability.* Hoboken, NJ: John Wiley & Sons, Inc.

Productivity Commission (2003). *Industries, land use and water quality in the Great Barrier Reef catchment.* Research Report, Canberra.

Simon, G., & Grandbastien, M. (1995). Corporate knowledge: A case study on the detection of metallurgical flaws. *Proceedings of ISMIK'95*, France, 43-52.

Sriskandarajah, N., Bawden, R.J., & Packham, R.G. (1991). Systems agriculture: A paradigm for sustainability. *Association for Farming Systems Research-Extension Newsletter, 2*(3), 1-5.

Walker, D.H., Cowell, S.G., & Johnson, A.K.L. (2001). Integrating research results into decision making about natural resource management at a catchment scale. *Agricultural Systems, 69*, 1-2, 85-98.

Walker, D., & McCown, B. (2003). *A community of practice approach to organizational learning on engaging with communities for improved natural resource management.* CSIRO, Internal Report.

Acknowledgement

We would like to thank Anne-Marie Vercoustre and Michael Flower for their contribution to this work. Special thanks to Professor Jim Peterson, Director of the Centre for GIS at Monash University, Xiaoye Liu, and the 16 students who participated in the evaluation of our learning environment.

Notes

[1]Commonwealth Scientific & Industrial Research Organisation Information and Communication Technology
[2]See http://www.topicmaps.org
[3]We concentrated our effort in creating links around certain concepts.

Designing Agents for Feedback Using the Documents Produced in Learning

CAT KUTAY AND PETER HO
University of New South Wales, Sydney, Australia
ckutay@cse.unsw.edu.au
peterh@cse.unsw.edu.au

This article reports on research, which is based on the premise that the main aim of teaching is to provide appropriate feedback to students as they learn. Human tutors make this process easier by asking students what they are thinking about a topic, then relate the answer to this stage or context of their learning. Hence, the first Computer Supported Learning systems were based on a tutorial question-answer format. Since then research has branched out into Learner Modelling and Intelligent Agents to support learning in more open systems. This article looks at computers emulating mentors who analyse the student's documented activities to provide feedback. The activities are analysed using a methodology that looks at the Human-Computer-Human Interface, and a pattern structure is developed, which is based on an ontology of group learning. Agents are designed and implemented using this structure to analyse synchronous and asynchronous group learning processes and to provide feedback. The ontology used in this research is based on the structure provided by Activity Theory where technology plays the role of mediator in the context of student actions.

Computers have been used to assist learning in many domains. This work is an attempt to develop a process of codifying student learning needs focusing on the documents produced in a group-based project course, into rules for agent support and a structure for learner modelling. The actions of groups of students in an online synchronous system are analysed for patterns, which signify the feedback approach to be taken. The ontology developed by Barros, Verdejo, Read, and Mizoguchi (2002) is used for the pattern structure, based on Activity Theory (Nardi, 1996) format for representing user actions in the learning system.

The subjects are students involved in developing software engineering specifications in groups, which also may work separately as individuals and bring their completed work to the next group session. The logistical overheads involved in communicating, recording and sharing files, and discussion between group members pose a hindrance to learning. In addition, there are the problems of facilitating, tracking, and managing group discussion and outcomes. As a result, without appropriate supervision, it is common for groups to develop specifications that do not meet the course requirements, while clearly they did undertake much design work and development effort. Last, time constraints and locality of group members work against having regular collocated meetings to provide the synergy needed to discuss and review their work.

Based upon these needs, a tool called Intertac-I (Kutay, 2003a) was developed to facilitate storage of files, workflow management, and group communication. The tool enables concurrent planning, editing, drawing, and discussing (Figure 1). This in itself was insufficient as no feedback was provided to further enhance Intertac-I. The documents produced by the system include the project documents, discussion histories, logs of interactions, and agent's rules. In the next version, user modelling data of configuration selections and the history of agent analysis will be added.

The aim of this work is to develop agents which use data mining techniques (Papatheodorou, Vassiliou, & Simon, 2002) on the documents produced to provide appropriate feedback. Intertac-I logs provide the data collection and preprocessing.

This article describes the pattern extraction and development of agents to provide postprocessing in the form of feedback. This feedback is designed to motivate and guide students towards experiences that enable: (a) the generation of the desired conceptions involved in the course; (b) elaboration of these conceptions; and (c) an ability to differentiate between different conceptions. The aim is to enable students to develop a desired depth of understanding and a range of skills in the learning domain.[1]

While focusing on the computer response to student actions, the research methodology places the human interaction in the centre of the analysis with the computer included as a mediating agent, which takes action in response to the history of student interaction. This approach is termed Human-Computer-Human (HCH) as the computer interface has been subjugated to the human's shared interface.

Experiments were conducted with students using Intertac-I, and the results of these case studies were used to develop rule-based agents that are a feature of the second version (Intertac-II). The constructivist principles of learning as enumerated by Savery & Duffy (1995) were used to specify the activity types for agents. These activities include:

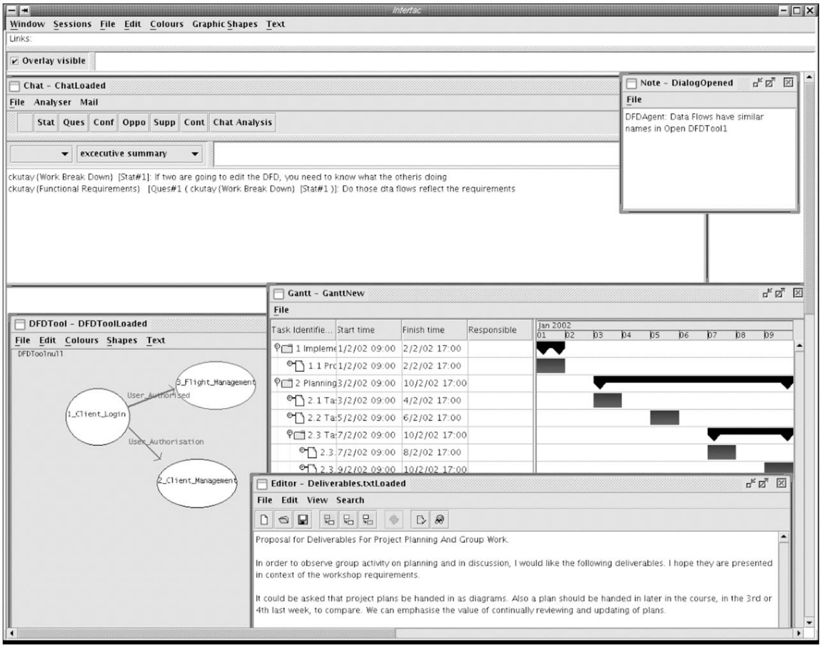

Figure 1. Intertac interface

1. scaffolding particularly in planning and design;
2. feedback which is instantaneous with programmed text or through e-mail to the tutor;
3. alternative views to provide different approaches and contexts; and
4. reflection support by providing opportunities.

This article is an analysis of the experimental results and a presentation of the pattern language. This language and its underlying ontology for learning and interaction, provide the structure for the implementation of agents in Intertac-II.

THE EXPERIMENT

Collaborative activity has been posed as having a cognitive advantage in learning through the joint activity (Dillenbourg, Baker, Blaye, & O'Malley, 1996). Collaborative learning has been analysed to understand and interpret the collaborative process to assess the conditions and elements for effective learning (Dillenbourg, Baker, Blaye, & O'Malley). This article explores how some of these learning processes are enacted and develops agents for supporting them, as a method of enhancing learning in online groups.

For this study, students were asked to work in groups of three or four and engage in one of two different specification projects (one easy, the other more complex). Students with little software engineering experience were given the easier project that required them to develop diagrams of a recipe for an Asian meal. The meal involved interdependent courses and various "hardware" implements which could be included or not, depending on how students decided to treat them. Meanwhile students with more software engineering experience were given a more complex task of producing a specification document for a proposed software system described on a commercial web site.

Students were required to use Intertac-I during the course of the experiment. The specifications produced are both textual and diagrammatic in format. While the present Intertac system cannot combine the two into a single document format, the students can use the output to develop a final document in Latex which is a scripting system for document preparation. The data collected using Intertac-I included snapshots of the files at regular intervals plus information on the activities that students were involved in, such as: the opening and closing of tools, contributions to discussion, and the editing changes to diagrams and documents. This data provides a dynamic picture of events or actions.

The data of students' actions is stored in a knowledge base, which forms the aspects of the learning ontology used – configuration elements and historical analysis of process in User Models, processing criteria for the agents in Agent Rule files, the present state of the system in log file. The aspects of each action that can be extracted from Intertac-I log files are: (a) the user initiating the action; (b) the proximity of an object or text to other objects or text; (c) duration or time of the change; (d) conceptual context of the interchange; (e) oppositional forms such as addition and deletions of an object; and (f) user notifying completion of an activity or stage. This knowledge base can be mined for information on the process used by individuals or the group as a whole.

The tools are linked through the Intertac-I system so data is collected that enables agents to:

1. analyse common threads running between the tools;
2. check the timing of document production stages as compared to the planned time-line;
3. compare documents to a template where appropriate;
4. analyse for adherence to rules of the domain; and
5. analyse of interactions and participation within discussion and across all tools.

RESEARCH FOUNDATIONS

The research uses a case study approach to document the range of activities undertaken by students. The groups were both self-selected and selected on the basis of various features including: range of experience; years of study; familiarity with domain; and experience in group work. The approach taken is to extract and analyse the activities and tasks in processes used by students in their learning and then identify significant actions in these process.

Since students are working on an ill-formed problem, it is not possible to provide the sort of specific questions, which a computer-based tutorial methodology could be used to analyse. In an alternative approach (Constantino-González, Suthers, & Escamilla de los Santos, 2003), students develop public as well as private designs that can be compared to determine advice. In this present work only a single design is developed by the group, or by an individual with the group commenting. Access to students' responses to contributions by others could, in the foreseeable future, come from language recognition tools.

For the present, group interactions and their learning approach are analysed on the basis of repeated processes. Again similar work has been done specifically on interaction patterns in text-based dialogue (Booth, 1992; Constantino-González & Suthers, 2001; McManus & Aiken, 1995; Martin, Rodden, Rouncefield, Sommerville, & Viller, 2001), however, these approaches relied on language parsers, which are not used in this work.

There has been much less research done in analysing the patterns in processes used in editing documents and diagrams.[2] The research described here covers patterns of student actions, such as interactions through the dialogue system, work patterns in the interplay between contributions to different applications, and patterns of approaches to learning in the areas of planning, diagrams, and document writing.

On the basis of findings from this research, an Implementation Pattern Language is developed to describe the feedback appropriate for various interactions and learning approaches that were isolated. While these interactions and learning approaches are themselves too disparate to warrant synthesis into a pattern format, the data analysis and implementation of feedback based on these patterns forms a coherent pattern language (Kutay, 2003b) based on a formalism similar to the learning formalism developed by Mühlenbrock, Tewissen, & Hoppe (1998). This approach is similar to that taken in developing patterns to assist in HCI design (Bayle et al.).

Implementation Patterns are necessarily an informal presentation of the concepts and processes of the agents and provide the metadata or ontology of learning in this domain. The ontology is developed in a structure that supports a process of translation from the instructor's definition of the learning objectives and the desired activities and processes of students, to an agent support language. A later section describes the ontology used to translate those requirements into agents.

RESULTS

The experiments were used to specify an ontology for the domain and the synchronous environment.[3] The ontology covers tools and tasks as the context, the source of information as stored in log files and user models and rules for interpretation as used by agents.

The interaction and learning processes that are selected for study are two sources of information on the process: increasing depth of learning; and stage in learning; and two tasks in learning in this domain: efficiency in document production; and efficiency of learning.

The aim of the work is to provide feedback through agents by entering comments in the discussion window or a separate "mentor" window. To develop this feedback, the processes and actions are studied for computer observable patterns. Given the broad scope of interactions that fall under the process and tasks areas previously described, it is necessary to restrict the review to particular aspects that concerned the lecturers in the courses using Intertac-I. The following brief discussion of the results of the investigation into user activity is included to provide an explanation of where the patterns derive from the experimental results.[4]

Depth of Learning

The aim of this section is to provide an analysis of how groups can be distinguished as to their depth of approach to their learning or depth of their conceptions. Some aspects of these differences are only visible if the students choose to use the full attributes of the tools supplied by Intertac, so they may not be able to be analysed in all groups.

One feature of groups working on the projects is the changing focus of discussion and of their work. Even when dealing with a small design, students often talk about other issues, especially as they wait from one person to do some editing. When they do have difficulty with their design they will discuss many issues in quick succession, not fully resolving them. Sometimes they will link these changing foci and build their knowledge progressively, while other groups just appear to search around without direction. Within each project, there can be a limited number of conceptual and data foci for any design and by analysing what the students did in terms of these concepts, a picture emerged of their process for approaching design.

As the students develop or change the way they link or navigate these foci, so does the depth of their learning. The aim is to initiate or develop these links in the group through feedback. The difficulty expressed by students in this experiment and also found by Wood (2001), is the need for this feedback to be pitched at the correct level of understanding, or injected at the correct time in their work.

To track the conceptual development of students, the agents need to provide an accurate way to connect their discussion and actions with each concept of the domain. Intertac-I provides a list of topics in discussion similar to the approach used in CHAOS (Simone, 1994), where they provide a lexicon of terms or jargon from the domain. The lexicon is used to develop agents to support conversational grounding by comparing the history of a user's contributions with the lexicon, and can be extended to include the Data Dictionary as used in E-R representations (Maiden, Cisse, Perez, & Manuel, 1998). The patterns derived from these activities are:

- **Lexicon.** It is important that the system be able to track the use and development of domain concepts in the learning. From the prior analysis, it is proposed that by providing a summary lexicon, which students can link with their contributions to discussion, students are encouraged to focus their conversation by grounding it in the course.
- **Concept extensions.** The instructor in their course plan will develop different approaches to the important concepts as examples or questions to ask students. These can be implemented as agents with the aim of encouraging students to link their understanding of a concept to a different focus on the concept.
- **Discussion feedback.** The students can also receive support for how much and what interchange of conversation has been linked to each lexical word or phrase. For instance has there been much explanation, disagreement, argument, decisions made, and so forth.

Stage of Learning

Another part of the course plan is the timing of the introduction of concepts in the seminars, which are held each week to support the projects, and experience of which aspect of the project the student will be undertaking each week during the session. Again this information can be derived from a course plan by the instructor. This leads to the formulation of the following patterns:

- **Template course plan.** The timing of interjections in discussion can be assisted by the use of a course plan that provides a time-line for the expected development of various concepts. These are compared to discussion to verify if they are being taken up by the groups.
- **Group planning schedule.** The course template is available for students to edit. The template provides a guideline or structure for their own group planning and is stored in the knowledge base separately for analysis of stages in the project development.[5]

Example of Process: Approach to Conception of Requirements

The conception of requirement is visible through the changing of requirements; deleting requirements; talking about changes; related changes to DFDs; informing the client[6] of changes to the requirements in the specification; and relating the requirements to the DFDs. In order of depth of conception, the approaches to specification development from requirements can be analysed as follows:

- **Null.** Few changes to the requirements, and these are not linked to discussion topics.
- **Explain change requirements.** Changes to requirements are temporally linked to the design of that section of the DFD or B specification. Usually a change in the removal of a requirement. The group often forgets to include reasons for this change for the client. Also changes can be made late in the workshop course (**Justify Change Pattern**).
- **Template course plan.** Changes are made during the design stage, and may be linked to an explanation in the document. Changes tend to dimish after this stage, but late changes are frequently a problem for maintaining coherency within the separate components of the grousp documents.
- **Context design.** Changes to the requirements start in the first session and include some additions as well as deletions. There may be a strong relation between the wording of the requirements and the processes and data of the DFD to trace consistency of late changes, or this link may be provided in the integration section required in the project report.

The main finding that arises from this analysis is that certain processes could be implemented by agents to support deeper conceptions. For example, when requirements are removed, there should be some explanation in the document relating to this removal, or change, using Explain Change Requirements. However, usually any sign of variation in depth of approach must be analysed using more than one basic pattern.

Efficiency of Document Production

Many documents or diagrams produced for the projects are required to have a set structure. While this can be presented in templates, sometimes it is useful for students to have their work analysed according to these rules automatically, rather than repeat the same errors in both their mid-session and final reports. However these rules and also templates change with the domain so must be set up by individual instructors.

Students also recommended that options be set up for them to view each document by its structure alone. That is, the user can select to view the headings and summary only of each section, or headings and descriptions of the role of that section. A template, which includes headings and a description of the role of that part of the document is included as the document template for the course. Students can then edit this directly, with the extra descriptive information removable for displaying and printing a normal view.

The areas where students needed support are:

- **Template.** To provide document description a template can be included with the suggested sections and a description of the role of each section. The information is formatted in an XML style language that enables the document editor to hide or display information as selected.
- **Document rule checker.** To improve document design the course documents provide certain expectations for the layout of the document in the form of a template, which can also be encoded as automatic document advice agents. If the students are to be marked on these aspects, they should have them reinforced during their learning.
- **View make and use.** To improve consistency, the students can link their document by *threads* of subjects or concepts, which they can display in separate user-selected views.
- **Changes.** To improve document construction when students are continually changing the document, either when offline or using other software, a summary of these changes can be displayed for group comment. This ensures that other members of the group are made aware of the main changes between versions. A versioning difference list can be generated when in group session, to provide information to generate queries for the students about alterations and present these as part of group discussion.
- **Diagram rule checker.** This study used Data Flow Diagrams (DFD) in the document, as an example of a diagrammatic form used in the students' projects. However in each discipline different diagrammatic formats are used to represent various parts of planning or design. The rules of each diagrammatic form can be linked to the system through agents, rather than requiring that they be programmed into the tool each time a new diagrammatic format is selected.

Efficiency of Learning

Improving learning also involves linking users to resources that help them change their view of their own understanding of the domain. Diagrams is an area of design that requires feedback in the form of alternative suggestions, since it is difficult for the computer to analyse if a design is good or bad, beyond whether it fits the rules of that format of diagram. In terms of diagrammatic rules, the feedback is uniform for all diagrams so it can be easily implemented by rule-based agents, as done for Entity Relationship diagrams in the software COLER (Constantino-González, & Suthers, 2000). With other aspects of the design, such as the taking of alternative design approaches, the feedback will depend on the design that students have developed, so this feedback is difficult to automate.

The three generic areas that patterns are found can be defined as:

- **Design.** The main request of users is for examples from previous projects. The course projects change each session so the requirement is to find the similarity between each current project and linking them to similar products from previous session projects. Not only does the software have to find resources that are similar to the design being developed by this group, but also has to find this similarity in a different project context. These alternative designs are presented in a way that encourages the user to consider why their design/document differs, and how it is dependent on the context they have assumed. This is similar to other work in distance learning where web courses use hyperlinks to alternative approaches to a problem.
- **Interaction.** During the entire session online, the users are interacting through the various tools. Often they make poor use of the tools to seek and gain answers to questions, or discuss differences. In particular users are often inexperienced in the steps required to resolve conflict or to even acknowledge and use conflict constructively. Some basic analysis can be made of their use of speech tokens to describe their intention in contributions[7] to the discussion, plus their actions in other tools.
- **Learning depth.** Similarly the users are often inexperienced in learning course material of any significant depth. Students have been encouraged to learn for assessment and avoid the extra work required to extract meaning from their courses. The main aim and design motivation of the workshop courses, and Intertac, is to motivate and encourage students into a deeper approach and conception of their learning.

IMPLEMENTATION PATTERNS

This article is about the design of Implementation Patterns. It must be noted that these are different to the interface patterns developed in HCI (Schümmer, 2002) as this article looks at the HCH interactions, and are different from the interaction pattern studies in other CSCW systems (McManus & Aiken, 1999), in that the next step of feedback implementation is included in the pattern structure. Also the patterns of approach to learning and the interaction patterns are now combined into one pattern system. These patterns are combined as the conversation or HCH interaction is as significant in the learning process (Pask, 1975) as the concept generation patterns.

Implementation Structure

Since Activity Theory is the analysis used to derive the patterns, an ontology based on this approach is used to describe the pattern structure, using the aspects that came out of the analysis. The aspects of Activity Theory that are important in the analysis are shown in Table 1.

Table 1
Activity Theory Structure for Implementation Patterns

Activity or Task	Activity enacted in the learning
Rules	Rules of analysis of the knowledge concepts or approach skills
Information Source	Source for the individual knowledge or group grounded knowledge which is either the knowledge base (log), historical analysed data and configuration data (user model) or rules (agents rules)
Temporal	Temporal extension of the activity (Akhras & Self, 2000).

Each pattern is initially assigned a weight, that will be altered by the User Model Agents to be developed in the next version, Intertac-III. The weight determines the agent that acts when there may be more than one agent that has achieved its condition to act.

The aim of developing patterns for learning content is to provide learning objectives in a format that can be easily translated into agent rules. The patterns developed in this analysis are just an extension of the diagrammatic and text structure rules. The structure for the patterns is shown in Table 2.

Table 2
Implementation Pattern Structure

Name	To provide easy reference.
Activity Type	See Figure 2.
Information Source of data	User or Group, and the Tool this pattern is related to.
Rules Outline	Problem to be solved or skill/concept learned by feedback.
Focus	The learning aspects or interaction aspects that are the focus of this pattern.
Information Source of Conditional	What data interplay signifies this pattern is achieved.
Action	Solutions to problem and processes to follow for feedback. May involve a series of steps, if the first does not get desired action by group or user, try next in order.
Information Source of Goal	Desired end result or response for each action.
Example	Practical and specific.
Weight	Initially assigned on basis of pattern complexity.
Semantics	Role this feedback plays in the pattern language semantics.

At present these patterns have been manually extracted and then processed by agents, rather than any automated analysis and learning of patterns from the student actions. This is the limitation of any such rule-based system.

As with any pattern development, it is desirable to analyse the interconnection between patterns. This connection is made through the activity type, in that only one activity should be implementing feedback at any one time. However the role of the patterns in the overall language can be used to select combinations or sequence of feedback. Where two patterns are complementary, they can support each other, where they are contradictory, they negate each other so should not be taking action together.

Pattern Language Structure

The nature of the patterns and the aspects for which they are patterns, form the structure of the pattern language. A pattern language is the semantics of how the patterns related to each other, or are distinct. Alexander, Ishikawa, and Silverstein (1977) used language in reference to architectural patterns in two senses. First, they talked about the shared design language. In the case of an implementation language, this is not always shared between domains. However, this research looks at a range of domains including workflow, document construction, and discussion interactions, providing a broad sample of patterns. The second sense is in terms of an organising principle which facilitates the use of a language. Clearly in implementation of the agents in this work, this structure is important for coordinating a multi-agent system.

The first division in activity type for the patterns is between those that deal with group interactions and those that deal with learning (Figure 2). These patterns can deal with individual or group processes.[8] Most Learning Patterns and all Interaction Patterns fall into the latter category, so unless specified otherwise, patterns are assumed to be for the group process. When users are working alone group and individual become synonymous.

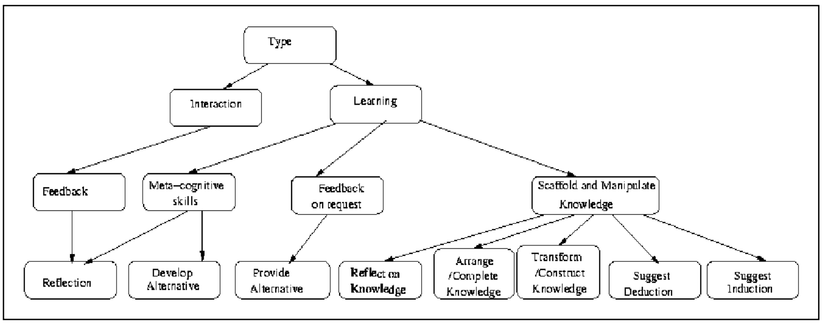

Figure 2. Activity types of patterns showing basic ontology

The next division is between agents that scaffold the manipulation of knowledge, provide feedback or encourage the development of learning skills or meta-cognition, such as reflection. Manipulation Agents are again divided into agents that generate actions that can be classified as scaffolding for manipulating conceptual knowledge by arrangement, transformation, deducing, or inducing structure. Meta-cognition agents are divided into agents that generate procedural skills or actions to encourage reflection, or encourage other group processes.

GENERIC AGENT PATTERN EXTRACTION

Intertac-II is an extension of the Intertac-I software, using the component-based design to insert agents that implement rules for each application, or as a link between applications, such as taking an overview of a user's contribution. The design of the Intertac-II agents already requires a rule language and structure that can enable the coding of rules from many domains (for instance team work, document formatting, diagram design and requirements engineering). The patterns provide an outline for translating learning needs in a course into an agent process. The next step is to develop a generic agent for each pattern.

The article now looks at the most significant patterns in detail. First, the Design Patterns, which are the most difficult to extract, then the Interaction and Learning Patterns, which are the most generic patterns.

Design Patterns

Design patterns are used to search files from projects in previous years for specific features to be used as alternative examples for students. In this research, the diagrammatic analysis relies on selecting the basic aspects of each drawing primitive and looking for patterns of similar designs. Since the projects differ between years, the design features that are similar are more likely to be structural issues rather than the entire design. The Design Pattern can be used both to check for similar design aspects in previous years and also for differences between an older version of the student group's design and the most recent version. To search files for specific features the agent needs to analyse aspects such as:

- **Visual issues**, such as joining data too close to each other, is a problem. Alternative better spaced designs can be displayed.
- **Keyword searches** for keywords missing from the DFD processes and data flows can be done. Unfortunately, the keywords can change between projects in each workshop, so it will be hard to analyse across years, except when dealing with common system processes, such as "login" processes or the designing of time into the system.

- **Design steps**; some steps in the process of diagrammatic designs are handled badly by students, such as combining sections of a highly detailed level and then moving the detail to a lower level of the design. The two diagrams in this process, before and after, can be displayed from the context of another project, hence abstracted from the details of the particular processes of the student's design.

Similarly, documents from previous years can be searched for changes in patterns. The searched documents also will be from a different software project than the present one. The patterns observed between distinct projects which would be worth noting for search categories are:

- **Length of section**; where user's sections appear too short on some aspect, display a longer example.
- **Use of sections**; if important sections such as Nonfunctional Requirements are missed, then the students could be shown an example to see what this section would cover.
- **Expert sections**; where sections are generally handled badly in the course, such as sections on integration of aspects of the design document, then an example of a skilled report could be displayed.

The Design Pattern uses a unification algorithm on the graphical or textual representation of the data in historical files compared to the present files. Files are searched on the basis of similarity in any one of the previously mentioned criteria, looking for a list of alternatives designs in any second criteria. The alternatives can then be displayed for the student group.[9] The Design Pattern is used to enable the following search types:

- **Alternative Design Pattern – ADP**; search for alternatives to the present users' design.
- **Change Design Pattern – CDP**; search for changes in design between versions of a design.
- **Context Design Pattern – TDP**; search for related aspects of one design throughout the document by keyword search and trace this aspect.

The Design Pattern is described by the structure shown in Table 3.

Table 3
Design Pattern Structure

Name	Design Pattern.
Activity Type	Alternative/Changed/Traced Document/Diagram.
Features	Search specifications.
Weight	Significance of this difference.

Interaction Patterns

These are the patterns, which analyse how people interact in the discussion using extra data from students' editing strategies on other tools. The actions that are analysed for patterns are additions, deletions and moves in editing, and the interchange of Tokens in the discussion. The actions are analysed for:

1. time between actions;
2. number of actions of any type;
3. length of the action, such as the edit or the discussion contribution;
4. the user who takes the action and if this changes; and
5. concurrent use of keywords from lexicon.

These patterns are similar to the many previous examples of interaction analysis in Computer Support for Collaborative Learning (CSCL) systems, but are included here to provide a complete pattern language for Collaborative Systems, and enable data from other tools to be included in the interaction analysis. Also, despite the sparcity of information available from text-based dialogue in this simple system, a wide range of interactions can be supported with this added data from actions in other tools.

Learning Patterns

Learning Patterns are those that relate to the depth of learning of a concept or an approach to learning. At present the course timetable enables agents to select the stage the group has achieved to assess the learning depth, or to assess the knowledge that is available to the users to date. Hence this is purely an assessment of the knowledge that is possibly available for synthesis into the users' own understanding.

However this is a very simplistic approach and this research involved a more comprehensive analysis of patterns of the depth of learning. In accordance with the findings of Booth (1992) this research found that:

- **Depth of conception** is usually attained through exposure to a greater variety of uses of the concept; and
- **Depth of approach** usually involves the ability to develop an overview of the learning, which is combined with more detailed knowledge.

Hence the Implementation Patterns provide increasing complex representations or experiences of conceptions (Concept Extension Patterns) and ones that monitor bottom-up designs (Complexity Patterns in DFDs) or encourage top-down designs (View Patterns on Documents).

VERIFICATION OF AGENT SYSTEM

The next stage in the agent development process is to verify the validity of the patterns that are implemented through their feedback. This verification will consider the aspects of Constructive Learning Environments that were adopted as the goals of this work:

- **Scaffolding**; by developing User (or Group) Models that enable the tracking of students configuration selection (such as role) and overt response to advice and the actions that follow any advice (Kutay & Ho, 2003) some analysis of the scaffolding effect can be examined. Students may respond to such agent advice and it is important that the agents rules are reapplied soon afterwards to verify the effect of the feedback, if any, on the patterns observed.
- **Alternatives**; another important aspect to verify is the search agents. This will involve running the agents on documents produced by students to verify that the Design Patterns extracted in searches are valid comparisons or alternatives.
- **Feedback**; a study should be made of the feedback categories that are received during the course of a workshop and how these relate to the resultant document and design produced by the group. This will verify if design problems are missed in the feedback or feedback is made that is not helpful.
- **Reflection**; during the workshops the students can be interviewed about their approach to learning software design, their approach to working in groups remotely, and their conceptions of the key aspects of the course. These can be related back to the agents that are designed to deal with these learning patterns and verify that the agents have either identified or responded in some way to these approaches.

CONCLUSION

Constructivist learning environments encourage flexibility and discourage attempts to prescribe actions between students. However by a judicious choice of formats, students can be encouraged to question and expand their understanding from interventions by simple intelligent agents. In particular any learning domain involves either work patterns, rules of design or simply communication patterns, which can be extracted from the data logged by a CSCL system. These patterns and rules can be developed into Implementation Patterns which provide the basis for coding agents to support their use in learning.

References

Akhras, F.N., & Self, J.A. (1997). Modelling learning as a process. *Proceedings of the 8th World Conference on Artificial Intelligence in Education (AI-ED'97)*, Kobe, Japan.

Akhras, F.N., & Self, J.A., (2000). System intelligence in constructivist learning. *International Journal of Artificial Intelligence in Education, 11*, 344-376.

Alexander, C., Ishikawa, S., & Silverstein, M. (1977). A Pattern Language, Oxford: Oxford University Press.

Barros, B., Verdejo, M. F., Read, T., & Mizoguchi, R. (2002). Applications of a collaborative learning ontology. *Proceedings of MICAI'2002 Mexican International Conference on Artificial Intelligence*, LNAI, (pp. 301-310). New York: Springer-Verlag.

Bayle, E., Bellamy, R., Casady, G., Erickson, T., Finicher, S., Grinter, B., Gross, B., Lehder, D., Marmolin, H., Moore, B., Potts, C., Skousen, G., & Thomas, J. (1998). Putting it all together: Towards a pattern language for interaction design. A CHI'97 Workshop. *SIGCHI Bulletin, 30*(1) 17-23.

Booth, S. (1992). *Learning to program: A phenomenographical perspective.* Göteborg Studies in Educational Sciences, '89, Acta Universitatis Gothoburgensis.

Constantino-González, M., & Suthers, D.D. (2000). A coached collaborative learning environment for entity-relationship modelling. In G. Gauthier, C. Frasson, & K. Van Lehn, (Eds.), *Intelligent Tutoring Systems, Proceedings of the 5th International Conference* (ITS 2000), (pp. 325-333). Berlin: Springer-Verlag.

Constantino-González, M., & Suthers, D.D. (2001). Coaching collaboration by comparing solutions and tracking participation. In P. Dillenbourg, A. Eurlings, & K. Hakkarainen (Eds.), *European Perspectives on Computer-Supported Collaborative Learning, Proceedings of the First European Conference on Computer-Supported Collaborative Learning,* (pp. 173-180). Universiteit Maastricht, Maastricht, the Netherlands.

Constantino-González, M., Suthers, D., & Escamilla de los Santos, J.G. (2003). Coaching web-based collaborative learning based on problem solution differences and participation. *International Journal of Artificial Intelligence in Education, 13*(2-4), 263-299.

Dillenbourg, P., Baker, M., Blaye, A., & O'Malley, C. (1996). The evolution of research on collaborative learning. In E. Spadia & P. Reman (Eds.), *Learning in human and machine: Towards an interdisciplinary learning science*, (pp. 189-211). Oxford, UK: Elsevier.

Kay, J., Halin, Z., Ottomann, T., & Razak, Z. (1997). Learner know thyself: Student models to give learner control and responsibility. *Proceedings of the International Conference on Computers in Education (ICCE'97)* (pp. 17-24), Malaysia. Charlottesville, VA: Association for the Advancement of Computing in Education.

Kutay, C. (2003a). Intertac. Software architecture, technical Report No. 0318. Computer Science and Engineering, University of New South Wales. Retrieved from http://www.cse.unsw.edu.au/~ckutay/Intertac/IntertacArchitecture.pdf, 2003

Kutay, C. (2003b). Implementation patterns for supporting learning and group interactions. Unpublished doctoral dissertation, The University of New South Wales, Sydney.

Kutay, C., & Ho, P. (2003). *Assisting students by analysing their interactions and learning*, submitted to AIED'03 Workshop on Learner Modelling for Reflection, Sydney University.

McManus, M.M., & Aiken, R.M. (1995).Monitoring computer-based collaborative problem solving. *International Journal of Artificial Intelligence in Education, 6*(4), 307-336.

McManus, M.M., & Aiken, R.M. (1999). *Petri nets: Using a formal model to describe synchronous rules in concurrent processes.* Technical Report TUCIS-TR-1999-001, Temple University.

Maiden, N.A.M., Cisse, M., Perez, H. & Manuel, D., (1998, June). *CREWS validation frames: Patterns for validating systems requirements.* Fourth International Workshop on Requirements Engineering: Foundation for Software Quality (RESFQ), Pisa, Italy.

Martin, D., Rodden, T., Rouncefield, M., Sommerville, I., & Viller, S. (2001). Finding patterns in the fieldwork. *Proceedings of ECSCW 2001*, (pp. 39-58), Bonn, Germany.

Mühlenbrock, M., Tewissen, F., & Hoppe, H.U. (1998). A framework system for intelligent support in open distributed environments. *International Journal of Artificial Intelligence in Education, 9*, 256-274.

Nardi, B. A. (Ed.) (1996). *Context and consciousness. Activity theory and human-computer interactions.* Cambridge, MA: MIT Press.

Papatheodorou, C., Vassiliou, A., & Simon, B. (2002). Discovery of ontologies for learning resources using word-based clustering. *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA 2002)*, (pp. 1523-1528), Denver.

Pask, G. (1975). *Conversation, cognition, and learning.* New York: Elsevier.

Savery, J.R., & Duffy, T.M. (1995). Problem based learning: An instructional model and its constructivist framework. *Educational Technology, 35*(5), 31-7.

Schümmer, T. (2002). PATORG. Retrieved from http://www.groupware-patterns.org

Simone, C. (1994). Supporting collaborative dialogues in distance learning. In M. F. Verdejo & S.A. Cerri (Eds.), *Collaborative dialogue technologies in distance learning.* NATO ASI Series, F133 (pp. 156-169). New York: Springer-Verlag.

Wood, D. (2001). Scaffolding, contingent tutoring and computer supported learning. *International Journal of Artificial Intelligence in Education, 12*, 280-292.

Funding Source

This research was generously funded by the Faculty of Engineering, The University of New South Wales, as an Educational Research Project.

Acknowledgements

The authors would like to thank: David Abdelmassih, Manoj Chandra, Sherman Lo, Zhicong Leo Liang, Zi Qi Lu and Naveed Hussain for contributing ideas and their programming skills.

Notes

[1]See Booth (1992) for similar research into learning needs for students learning programming.

[2]Similar work has been done in single-user Intelligent Learning Environment by Akhras and Self (1997).

[3]See Barros, Verdejo, Read, & Mizoguchi (2002) for extended learning ontology.

[4]Where an Implementation Pattern is described, it is written with initial capitals.

[5]Similarly, any analysis of student progress in learning by the agents can be edited to improve student learning (Kay, Halin, Ottomann, & Razak, 1997).

[6]The projects involve developing software for a realistic client, based on the requirements the client has provided.

[7]Compare to the work of McManus and Aiken (1995).

[8]Individual interaction with the group is treated as an Individual process.

[9]Rule Checkers should be applied also before the alternative designs are displayed.

A Virtual Hyperbooks Model to Support Collaborative Learning

GILLES FALQUET AND JEAN-CLAUDE ZISWILER
University of Geneva, Switzerland
Gilles.Falquet@cui.unige.ch
Jean-Claude.Ziswiler@cui.unige.ch

Learning by collaboratively writing scientific hyperbooks requires specific software tools. We present a model for creating, managing, and viewing hyperbooks. This model is comprised of a re-usable document repository (fragments repository), connected to a domain ontology. The model takes into account the notion of point of view, allowing a user to read the hyperbook according to a specific reading objective or to his or her profile. The model also includes an interface specification language for creating different hypertext views of the hyperbook contents. The hyperbook model we propose is an example of a virtual document model because the hyperdocuments the reader/writer actually sees are not stored but generated by assembling stored fragments according to an interface specification A purely declarative language allows the definition of the views that make up the interface of the hyperbook. We also present the architecture of a hyperbook management system, which is based on a database management system and a hypertext view generation system for databases.

During the last few years, we conducted several pedagogical projects that consisted of the collaborative construction of a scientific hyperbook. The principle was that the core of the hyperbook was made of lecture notes written by the teachers, and students were asked to produce new documents for the hyperbook. The teaching objectives of these projects were:

- to help the students see the relationships that exist between the different concepts presented during the course (hence the hypertextual nature of the book);
- to give student the opportunity to participate in the collaborative writing of a large electronic document; and

- to show that the same subject matter can be seen from different points of view.

Our first experiment with basic web tools (an HTML editor, a drawing application, and an HTTP server) clearly showed the need for a more sophisticated collaborative writing and publishing environment. Thus, we decided to develop a web-based, database-backed hyperbook management system. This system, and particularly its underlying models, evolved according to the needs and problems we observed.

The issues that appeared particularly important in this pedagogical context were the following:

Documents. It is necessary to make sure that the identification of a document (or document fragment) is stable over time and independent of its location (in this respect URLs are not sufficient). It is also important to have a means to categorize documents, so as to facilitate their retrieval and re-use. The information content of a document should not be cluttered with presentation or linking markers, contrary to HTML documents where linking tags must be explicitly inserted in the text.

Links. Our first experiment showed that students had difficulties creating hypertext links between pages. These difficulties were mainly caused by technical reasons (volatile URLs, access rights to HTML files, etc.). When students were provided with efficient tools to link their pages we observed a proliferation of links that were only marginally relevant. Thus, it is necessary to help the writers create meaningful and informative links.

Reading and writing interfaces. Reading a hypertext can cause cognitive overload because the reader has to manage his or her own reading path, as opposed to the linear reading of a simple text. Thus, readers should be provided with hyperdocuments that can be read sequentially without navigation effort. On the contrary, writing and linking small text chunks is generally easier than constructing large sequential texts. Thus, the reading interface must present linear texts that result from the assembly of small pieces of information. In addition, reading difficult and/or new material (sometimes called active reading) involves several auxiliary activities such as annotating, highlighting, summarizing, and so forth. Thus, an effective reading interface should provide tools to support these "writing" activities.

Terminology and concepts. In scientific writing, terminology (the definition of concepts and their relationships) plays an important role. Scientific writings either refer to well-known concepts of the studied field or they contain new concept definitions. Thus, writing and reading a scientific hyperbook entails referring to and updating a domain ontology.

Points of view. It must be possible to read (or browse) the hyperbook according to different points of view. A point of view is a specific perspective on the book's domain, it can also be a reading objective (in-depth reading, overview, etc.).

The purpose of this article is to present the hyperbook model that emerged from the combination of this pedagogical effort and our research on hypertextual interfaces for databases. In fact, this model is a virtual hyperbook model because the documents and links that the user sees are generated from the information and knowledge fragments that are stored in the hyperbook. In other words, these documents are only virtual (or potential) in the hyperbook database.

The rest of the article is organized as follows: the next section gives some background on the research area we are working in; the structural aspects section presents the structural part of the model, then the next section is on the "hyperbook ontology" which is the core of the model; the following section presents the interface and interaction part of the model, and the final section shows how this model can be implemented with a hypertext view system. Finally the conclusion proposes future research directions.

BACKGROUND AND RELATED WORKS

This hyperbook model we propose is built on ideas and concepts developed in the fields of hypertexts, document management, databases, web interfaces, knowledge bases, and collaborative working. It can be seen as an implementation of the idea of personalized virtual document.

The concept of collaborative work has prompted the development of several web-based tools, such as Basic Support for Cooperative Work (BSCW) (Appelt & Mambrey, 1999) or Learning Space (based on Lotus Notes). Most of these tools are essentially centralized document repository systems or coordination systems. Although they propose a web interface, these tools are not aimed at collaborative hypertext writing.

In the hypertext field, it is striking to note that "pre-web" systems were structurally and functionally richer than the web hypertext model, which aimed at simplicity and decentralization. Hypertext research has been headed toward different domains and objectives that are of interest to us. Systems such as Intermedia (Garret, Smith, & Meyrowitz, 1986) or Storyspace (Bernstein, 2002) were essentially developed to study and produce hypertext literature; other systems, for instance KMS (Akscyn, McCracken, & Yoder, 1988) or MacWeb (Nanard & Nanard, 1993), aimed at knowledge sharing and management; finally, some systems (HyperCard, NoteCard) were closer to highly interactive application development tools. Theoretical works have studied the fundamental notions of link, anchor, node composition, navigation, and so forth. and lead to the Dexter reference model (Halasz & Schwartz, 1994).

Several models and systems have also been proposed to integrate the notions of book and electronic publishing to create hyperbooks. The aim can be either to create hypertext versions of existing books (Rada, 1990), or to generate electronic books from paper books (Landoni, Crestani, & Melucci, 2000), or to directly write (hypertextual) electronic books (Fröhlich & Nejdl, 1997), or to integrate existing electronic documents (Brusilovsky & Rizzo, 2002).

The hypertext personalization problem has attracted many research works that lead to the definition of models and techniques for adaptable and adaptive hypertexts (De Bra & Calvi, 1997; Brusilovsky, 1998). Adaptable hypertexts can present different contents, or differently organized contents, depending on the user's profile. Adaptive systems can automatically update the user's profile by observing his or her behaviour, in this case the adaptation is dynamic. A well-known example of adaptiveness occurs in web browsers: once the user has visited a page, all the links to this page are shown in a specific colour to indicate this fact. In (Wu, de Kort, & De Bra, 2001), the authors proposed an adaptive hypertext model that included a domain model, a user's model, and adaptation rules. The domain model is a semantic network comprised of concepts and relations between the concepts. This model is essentially used to define adaptation rules that depend on what concept the user knows or masters.

Recent research works concentrate on the notion of personalizable virtual documents (Ranwez & Crampes, 1999; Crampes & Ranwez, 2000). These documents are sets of elements (generally called fragments) associated to filtering, organization, and assembly mechanisms. Given a user profile or reading objectives, theses mechanisms will produce different (real) documents that should meet the user needs. For instance, (Iksal, Garlatti, Tanguy, & Garnier, 2001) proposed a virtual document model that was based on four ontologies, namely, a domain ontology, a metadata ontology, a document ontology, and an application ontology. The model we present here belongs to this approach.

STRUCTURAL ASPECT OF THE VIRTUAL HYPERBOOK MODEL

It is generally accepted that virtual documents (Ranwez & Crampes, 1999) are made of fragments (or "pieces of information") that can be assembled to constitute directly readable real documents or hyperdocuments. We will thus consider that the basic informational contents of a virtual hyperbook consist of a collection of re-usable document fragments. A virtual hyperbook also contains an ontology that formally represents the concepts of the domain the hyperbook is about. Every fragment can be linked to one or more concepts through typed links that indicate the specific role played by this fragment with respect to this concept. The third component of a virtual hyperbook is the hyperbook ontology. Its purpose is to represent the different linking structures of the hyperbook (links between concepts and fragments and links among fragments), the fragment's organization, and the different points of view on fragments and concepts. The interface specification will be based on this ontology to generate the readable hyperdocuments.

Fragments

The basic informational contents of the hyperbook are made of reusable fragments. A fragment has a content and belongs to one or more categories. The content of a fragment is a tree of XML or XHTML elements. The category of a fragment indicates its intrinsic nature. Typical categories are: statement, question, and theorem. Categories must not be confused with roles played by fragments with respect to the domain concepts. For instance, if a fragment is an example of the concept *cyclic graph*, it is at the same time counter-example of the concept *tree*.

Fragments can be connected by structural links to form compound fragments. These typed links indicate the roles played be the different fragments in the compound fragment. For instance, an exercise could be made up of a question fragment, one or more answer fragments, and a discussion. Compound fragments can have different purposes, they can represent pedagogical units (an exercise), or argumentative units (an issue related to positions and arguments), or even hyperbook management units (group discussions or weblogs). For instance, a discussion structure could be made up of *topic* and *message* fragments connected through *about* and *reply-to* links.

The important point is that direct links between fragments are purely structural while semantic links will be inferred by referring to the domain ontology.

Since the set of fragment categories and link types depends on the subject of the hyperbook, there are no fixed, predefined categories and types. In fact, the fragment categories and fragment link types are defined in the hyperbook ontology.

Domain Ontology

It is common in virtual document architectures to distinguish between the document fragments and the semantic structure. The latter, for example an ontology or a conceptual graph, describes the domain and is used for indexing or qualifying the fragments. The domain ontology is intended to hold a formal representation of the domain’s concepts.

Concept definitions. The concept definition language is a graph-based version of a formal language that belongs to the description logic family of languages. In this formalism, a concept is either:

- a primary concept;
- a conjunction or a disjunction of a concept;

- the complement of a concept; or
- a role restriction made of a quantifier, a role name, a minimal and a maximal cardinality, and a range concept.

A role restriction is represented by an arc pointing to the range concept and labelled with the quantifier, role name, and cardinality constraints. For instance, in the graph shown in Figure 1, the arc labelled *all component* (4,4) from *Quaternion* to *Real number* means that a quaternion has at least and at most four components that are real numbers and that all the components of a quaternion are real numbers. This same graph shows that a quaternion is a number, which has exactly four real components, a multiplication operation, and an addition operation. It also shows that the addition of quaternions is commutative while the multiplication is not. The *is-a* links serve to organize the concepts in a generic/specific taxonomy.

Although the language is expressive, it is not always necessary to use all of its features, in particular when the ontology is small. However, when the ontology becomes larger it may contain concepts that are only subtly different. In this case a more precise description of each concept is crucial to show their differences and similarities. This happens in particular when one seeks to exhaustively describe a domain or a category of objects.

Concepts and points of view. It is a well-known fact that different experts would give different definitions of the same concept (or what they think is the same concept). For instance, the electron concept would be defined by a physicist as massive particle with a negative unit charge that is insensitive to strong interaction. A definition provided by a chemist would probably be different, for example: "electric corpuscle that can be dragged away, caught or shared between atoms and molecules" whereas for the electronics engineer it is "the smallest charge carrier able to move in electric circuits."

Since one of the design objectives is to present the subject matter according to different points of view, the model supports points of view dependent

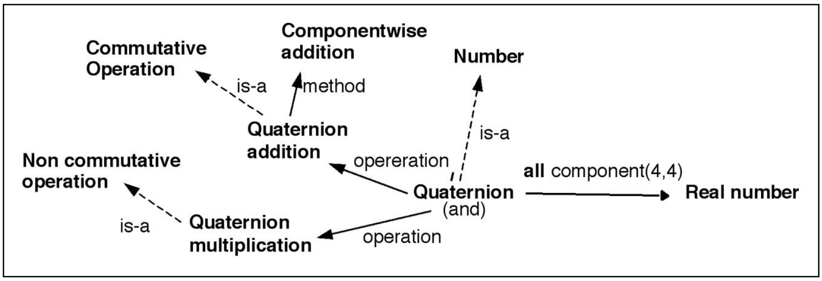

Figure 1. Concept definitions

of the definitions of concepts. To implement multiple points of view in the ontology, each arc and node can be associated to a point of view (Figure 2). Hence the definition of a concept according to a point of view is obtained by selecting those arcs that belong to the desired point of view or to a more general point of view (as we will see in the next section, points of view are hierarchically organized).

Roles of the domain ontology. Apart from precisely defining the domain's concepts, a domain ontology can play several important roles in a scientific hyperbook. Landow (1998) explained that an important hypertext design problem is how to enter the hypertext. A domain ontology provides a good entry point into the hyperbook because the number of concepts in the ontology is generally much smaller than the number of information fragments. Thus the user can browse the ontology and then go down to the fragments that are connected to the concepts he or she is interested in.

(De Bra, 2002) mentioned that the ontology is also a useful tool to personalize the reader's navigation in a hyperbook. If the system memorizes what concepts are known and not known to the user, then it can propose fragments that the user should read.

In our case, the domain ontology will play an essential role in inferring links between information fragments, as we shall see in the next section.

THE HYPERBOOK ONTOLOGY

The hyperbook ontology is the application ontology of the hyperbook. Its role is to describe the relationship between the fragment repository and the domain ontology; to describe the structures that exist in the fragment repository; and to associate concepts and links to the relevant points of view. The main classes (concepts) of the hyperbook ontology are shown on the diagram of Figure 3.

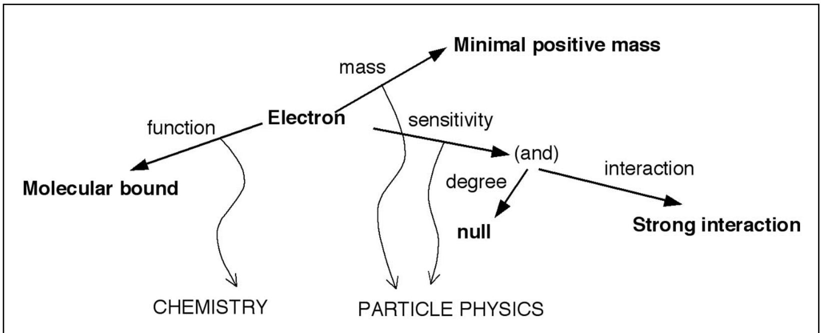

Figure 2. Different points of view on the same concept

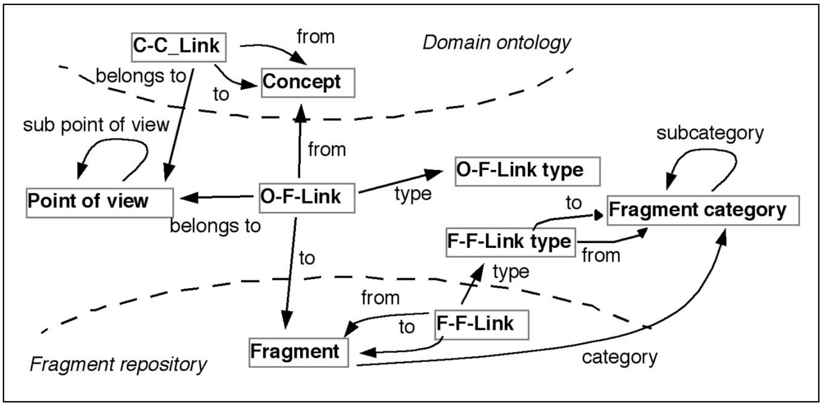

Figure 3. Classes of the hyperbook ontology

This diagram also shows how the hyperbook ontology is connected to the domain ontology and the fragment repository.

It is important to observe that if the classes are fixed, their instances can be defined specifically for each hyperbook. Hence every hyperbook will have its own fragment categories, link types, and points of view.

Fragment Structures

As previously mentioned, the fragment categories and fragment-to-fragment link types (instances of *F-F-Link type* and *Fragment category*) determine the document model. They can reflect structural relationships between fragments (composition links) as well as rhetoric, argumentative, or narrative relationships.

Links between Fragments and Concepts

The domain ontology plays two roles. On one side, it describes the concepts of the domain. On the other side, it serves as a reference to describe the information content of the fragments. By establishing typed links from fragments to concepts, one can qualify not only what the fragment is about but also what relationship it has with the domain concepts. Typical link types are:

- *instance, example, illustration*: the fragment describes a particular instance of the referred concept;
- *definition*: the fragment contains a textual (or audible, or graphical) definition of the concept;
- *property*: the fragment describes a property of the concept; and
- *reference, use*: the fragment refers to the concept (it is necessary to know the concept to understand the fragment).

These links play a crucial role to establish relevant links between fragments and to generate interface documents. The idea is to replace direct linking between fragments (often called horizontal linking) by inferred links that correspond to paths starting from a fragment, going through one or more ontology concepts, and ending on another fragment. Inferred links are preferred to direct links because users (authors) are generally able to establish correctly typed links from the fragments they write to the relevant concepts. When they are asked to link their fragments directly to other fragments, they have difficulties finding relevant fragments to link to and deciding on what type of link to establish.

The following figure shows two derived links (1) and (2) obtained by going up to the domain ontology and then down to another fragment.

Since the ontology has a graph structure, it is possible to express link inference with path expressions.

Expressing inferences by graph expressions. An interesting property of the hyperbook model is that semantically meaningful links can be obtained by simple inference rules that consist in path expressions. If we consider the global labelled graph formed by the domain ontology, the fragment collection, and the concept to fragment links, a path expression is an alternated sequence of nodes and arc specifications. A node specification is composed of a node type (concept or fragment), a category name (for fragments), or a term (for concepts). An arc specification is composed of a link type, a traversal direction, and a point of view. In addition, each node and arc can be associated to a variable. An instance of a path expression is a path in the hyperbook graph that satisfies all the specifications of the path expression. For instance, link (1) of Figure 4 is an instance of the path expression

fragment ← example – concept – example→ fragment

(start with a fragment, traverse an example link backwards to reach a concept, then traverse an example link to a fragment). Depending on the link types and fragment categories of the hyperbook, it will be possible to define link inference paths that have a precise and useful meaning for the reader.

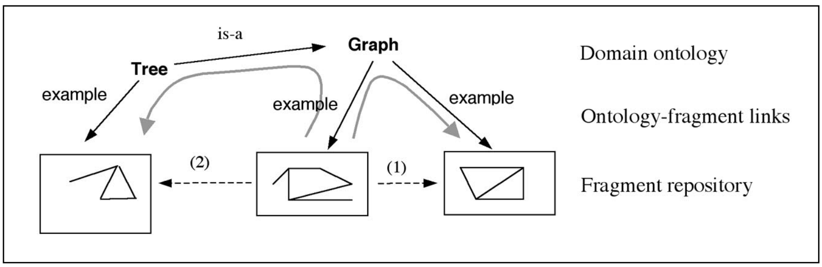

Figure 4. Link inference through the domain ontology

The following expressions show examples of link inferences that typically occur in the hyperbooks we consider.

fragment F_1 ←example – concept C_1 [←is_a – concept C_i]* – example→ fragment F_2

F_1 is linked to F_2 if F_1 is an example of a concept C_1, and C_1 has a sub concept C_2, which has an example F_2. The [←is_a – concept C_i]* notation represents the traversal of zero, one or more *is-a* taxonomic links in the generic to specific direction. Link (2) on Figure 4 corresponds to this expression.

fragment F_1 ←example/physics – concept C_1 –*→ concept C_2

If F_1 is an example of concept C_1 for the physics point of view, link it to every concept C_2 directly connected to C_1 through any kind of link.

fragment F_1–uses→ concept C – [is_a→ concept D –]*property→ fragment F_2

If fragment F_1 refers to concept C, create a link to every fragment that describe properties of any concept D that is more generic than C. If F_1 is an exercise, this will link it to all the properties of the concepts required by the exercise.

An additional property of this link inference method is its robustness with respect to the hyperbook evolution. Since the domain ontology is usually more stable than the hyperbook's fragments, a link to a concept will probably have a longer lifetime than a link to a fragment. Moreover, inferred links are, by definition, always up to date.

A remark about instances in the domain ontology. There are two ways to represent concept instances. If a fragment describes a concept instance, for example the the fragments showing a graph, it can be linked to the concept through an instance or example link. However, if an instance plays an important role in the domain, it should be represented in the domain ontology, so that it can be referred to by fragments or other concepts. In this case, we would have an atomic concept representing the instance that would be connected to the concept through an instance link and to the fragment through a definition link. For instance, if we consider that the "complete graph with 3 vertices" is a remarkable instance of graph, it must be present in the domain ontology.

Points of view

A point of view corresponds either to a category of user (*student, researcher, journalist, ...*) or to the point of view adopted by a user at a given time (corresponding to its reading/writing objectives). For instance, a student could read a hyperbook about *algorithms* and *data structures* with a software engineering mind set when he or she is developing software. The same person could also read the hyperbook with a theoretical mind set when he or she is studying complexity theory.

The notion of point of view applies both to the concepts of the domain ontology and the ontology-fragment relationships. Thus any concept (in fact any node or link of the ontology graph) an any ontology-fragment link may belong to zero, one, or more points of view.

Points of view must be "reasonably" non-contradictory. If a concept *C* belongs to points of view *P* and *Q* and if an object *O* satisfies the definition of *C* according to *P* (i.e., considering only the parts of *C*'s definition that belong to *P*), it should "in general" satisfy the definition of *C* according to *Q*. In other words, the extensions of *C* according to *P* and *Q* should be almost equal.

Some points of view can have sub-points of view that are more specialized. For instance, the *computing* point of view could be specialized into *computing theory, software engineering*, and *artificial intelligence*. This means, for instance, that an object (concept or link) may belong to the *software engineering* point of view only if it belongs to the *computing* point of view.

Interface Model

According to the virtual document approach, the user interface of an hyperbook is made of derived hyperdocuments obtained by assembling selected fragments. The specification of the views must also take into account the point of view adopted by the reader since it may influence the selection and the assemblage of the fragments. Given the richness of the static hyperbook model, it is impossible to design a single "optimal" reading and writing interface. This is why the interface model is intended to specify various views on the hyperbook content, allowing the interface designer to adapt the interface to each particular hyperbook. In addition, the interface specification language enables the designer to create simple interfaces that hide the details of the underlying hyperbook model.

INTERFACE DOCUMENTS

The hyperbook interface is a hypertext whose nodes are derived documents. The interface specification defines the building rules to apply when creating these documents. The interface specification language is an extension of the Lazy language, which was designed to specify and implement hypertext views on top of relational databases (Falquet, Nerima, & Guyot, 1999). The most important characteristic of Lazy is its declarative approach. Instead of writing procedural code to program the construction of the hypertext view, one can declare what the selection and assembly criteria are. Another important point is the hypertext model supported by Lazy. In a Lazy specification, hypertext links can be reference links (such as HTML links), or inclusion links (the target node is included in the source node at the link location), or expand-in-place links (the target node appears within the source node when the link is clicked). This rich linking model is well adapted to the construction of complex, heterogeneous interface documents. In (Falquet, Hurni, Guyot, & Nerima, 2001), we showed how to create sophisticated hypertext documents to “read” databases with Lazy.

An interface specification consists of a set of node schemas that will be instantiated on demand to produce the actual interface documents. Hence, the interface nodes (the documents the user sees) are instances of node schemas. A node schema is comprised of:

- a selection part (what fragments and concepts to select);
- a content description (how to arrange the selected objects, which attributes of the selected objects to display);
- a content structure (XML mark-up tags within the content description); and
- reference, inclusion, and expand-in-place links to other node schemas.

Example 1. The following node schema selects all the fragments connected to a given concept, its content is made of all the fragment title and contents together with the link type.

```
node examples_of [C]
      <title> "Fragments related to ", C.term </title> ,
      {       <subtitle> L.type, ": ", F.title </subtitle>
              <text> F.content </text>
      }
from Concept C –L→ Fragment F
```

The selection expression is in fact a path expression. An instance *example_of* [*x*] of this schema is obtained as follows:

1. select all the fragments *F* connected through a link *L* to concept *x*.
2. generate a <title> element containing the term that denotes concept *x*.
3. for selected *L* and *F*, generate a <subtitle> element containing *L*.type, the constant ": ", and *F*.title; a <text> element containing *F*.content (the content of *F*).

Example 2. This example illustrates the virtual document idea. It consists in generating a semantically consistent and sequentially readable document by assembling separate fragments. The node schemas shown in Figure 5 (markup tags have been omitted) specify a document that contains:

- the textual definition of a concept (found in a fragment linked through a “definition” link);
- the content of all the fragments directly linked to this concept in the “theory” point of view; and
- links to directly related concepts.

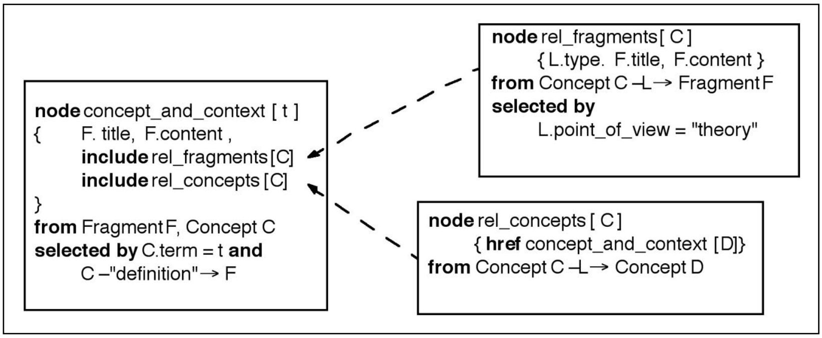

Figure 5. Definition of a composite node schema through inclusion links

These node schemas show that the interface definition language refers to the fragment repository, the domain ontology and the hyperbook ontology in a uniform way. Hence it facilitates the creation of interface documents to access the hyperbook at any level.

The following figure shows an instance of the *concept_and_content* schema generated on an actual hyperbook.

Interaction (the Writing Environment)

As mentioned in the introduction, one of our objectives was to support the hyperbook writing activity as well as the active reading of the hyper-

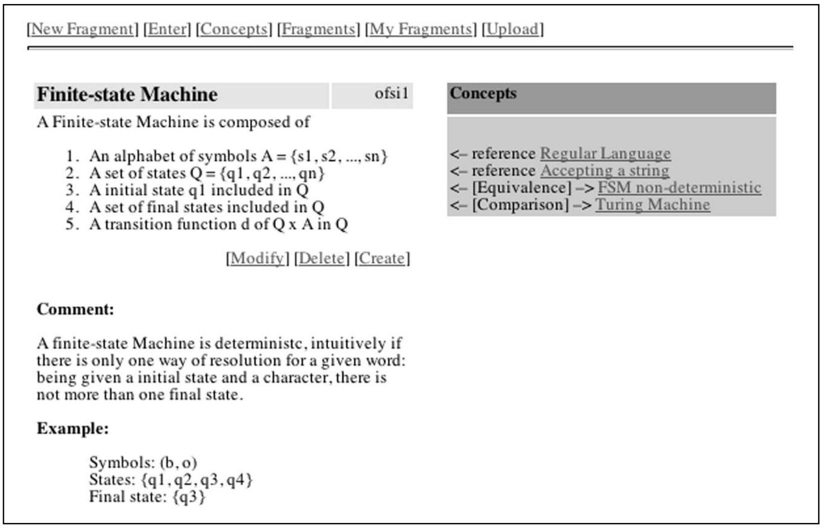

Figure 6. Instance of a composite node

book. This implies that the system must enable the users to create new fragments and concepts and to establish links among them.

The interface model supports this active part of the interface through input documents (e.g. forms) and active hypertext links. An active hypertext link is a hypertext link that triggers an action when it is followed. These actions can create, update, or delete objects in the virtual hyperbook (an example is given in the next section). The idea is to use the navigational approach as much as possible for updating the information content.

For instance, the following node schema is intended to show the content of a fragment *F*. If the user clicks the "Add Note" active link, this will insert a new fragment *G* into the fragment repository and a new link of type "note" from *F*. Then it will jump to the node *write_note*[*G, F*] that will display an input area to write the note content, update the note fragment and return to fragment *F*.

```
node show_fragment [F]
{       F.content
        active href write_note [G] (
                on "Add Note" do new Fragment G; new Link [from: F, to: G, type: "note"]
        )
}
from Fragment F

node write_note[G, F]
        active href show_fragment [F] (
        input t = textarea( ),
        on "Save Note" do update Fragment G [content: t]
        )
```

IMPLEMENTATION

During the last two years, we have implemented several virtual hyperbook systems based on different versions of this model. We took advantage of the Lazy hypertext view generation system to get straightforward implementations on top of relational databases. The hyperbook model that we have presented here is easy to translate to the relational database model. This results in a relational database schema (a set of relational tables) that represents the domain ontology, the fragment repository, and the hyperbook ontology. For instance, the concepts, concepts-to-fragment links, and fragments are represented in the following three table schemes:

Concept(id, term, operator, …)

C_F_Link(from, to, type)

Fragment(id, category, content, …)

Thus we can use the well-established relational database technology (, which handles all the concurrency, security, or query optimization issues) to store the contents of the virtual hyperbook.

Once the hyperbook model has been put in relational form, one can define the interface documents with the relational version of the Lazy language. An interface specification is thus a set of Lazy node schemas that refer to the relational tables of the hyperbook. For instance, the following node schema is equivalent of the *concept_and_content* node schema of Example 2.

node *concept_and_context* [*t*]

{ *F*. title, *F*.content ,

include *rel_fragments* [*C.id*]

include *rel_concepts* [*C.id*]

}

from Fragment *F*, Concept *C*, C_F_Link *L*

selected by *C*.term = *t* **and** *C*.type = "definition" *and L*.from = *C*.id **and** *L*.to = *F*.id

In fact, the node schemas on the hyperbook model can be automatically translated to node schemas on the relational hyperbook schema.

The current system uses the Lazy node server that dynamically generates HTML pages by querying the fragment and link database tables according to the node schemas. The Lazy system compiles the node schemas and stores their compiled form (a set of SQL statements) into a dictionary. The Lazy node server is a servlet that runs in an HTTP server. When the hyperbook user requests a node instance, the node server executes the compiled form of the corresponding schema (with the appropriate parameters) and send the resulting HTML or XML document to the user's browser. The node server also manages the inclusion and expansion links by recursively executing the appropriate nodes.

CONCLUSION

We have discussed a generic model for representing multi-point of view scientific hyperbooks in the form of virtual hyperdocuments. In this model, the hyperbook ontology plays a crucial role to interconnect the domain ontology and the information fragments, to support the multi-point of view aspect, and to generate views for accessing the hyperbook. By defining fragment categories and link types, a hyperbook designer can adapt the model to a particular domain or task. Then he or she can design a suitable interface by writing document specifications in the form of node schemas.

We have implemented several virtual hyperbook systems that are based on this model. During the last two years, students of the "formal tools for information systems" and "introduction to new information technologies" courses have collaboratively written course notes with such a hyperbook system. The course instructor was in charge of creating the domain ontology and fragments with textual definitions of the concepts. The students' task was to create examples, exercises, properties, theorems, historical notes, and so forth, to store them into fragments and to link these fragments to the corresponding concepts. Students could also create horizontal links between fragments. The reading/writing interface was comprised of about 30 node schemas. It enabled the user to read, write, and link fragments but also to navigate within the domain ontology, to compare concepts (for instance by viewing them side by side), to selectively display fragments related to a concept, and so forth. The students who used the system were able to produce good quality fragments, probably because they could concentrate on specific and limited tasks and because they did not had to take care of the hyperbook structure. In earlier experiments we remarked that when students were asked to directly link theirs fragments to fragments written by others, they produced only few relevant links. This is why we have decided to limit the use of direct horizontal links to the expression of argumentative or rhetoric relationships such as *remark, consequence, solution, contradiction, support,* and so forth. Since the students were able to correctly link their fragments to the relevant concepts, we have used these links to infer semantic links between the fragments, as previously explained. We also defined paths to infer semantic links between pairs of concepts that are directly linked to the same fragment (as can be seen in Figure 6, right). We are currently extending the system to provide a simple and efficient management of the users' points of view. We will also take advantage of the notion of point of view to dynamically adapt the interface documents to the user's reading objectives (Falquet, Nerima, & Ziswiler, 2004).

We used the same system to create a research-oriented hyperbook. In this case, we added a group discussion environment by defining suitable fragment categories (topic, message, etc.) and link types (reply, argument, etc.). The ability to manage multiple points of view is particularly useful in a research hyperbook because some concepts not yet well established and several concurrent definitions may co-exist.

In the near future, we intend to work on the interface model, with the aim to define new ways of presenting and interacting virtual documents. We will also continue to study the management of digital libraries of hyperbooks

(Falquet, Mottaz-Jiang, & Ziswiler, 2004). This will lead to an integrated view of hyperbooks corresponding to different courses.

References

Akscyn, R., McCracken, D., & Yoder, E. (1988). KMS: A distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM, 31*, 820-835.

Appelt, W., & Mambrey, P. (1999). Experiences with the BSCW shared workspace system as the backbone of a virtual learning environment for students. *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications ED-MEDIA 99*, Seattle, WA.

Bernstein, M. (2002). Storyspace. *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (Hypertext 2002)*, (pp. 172-181). College Park, MD.

Brusilovsky, P. (1998). Methods and techniques of adaptive hypermedia. In P. Brusilovsky, A. Kobsa, & J. Vassileva (Eds.), *Adaptive hypertext and hypermedia* (pp. 1-43). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Brusilovsky, P., & Rizzo, R. (2002). Map-based horizontal navigation in educational hypertext. *Proceedings of the Thirteenth ACM Conference on Hypertext and Hypermedia (Hypertext 2002)*, College Park, MD.

Crampes, M., & Ranwez, S. (2000). Ontology-supported and ontology-driven conceptual navigation on the World Wide Web. *Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia (Hypertext 2000)*, San Antonio, TX.

De Bra, P. (2002). Adaptive educational hypermedia on the web. *Communications of the ACM, 45*,(5), 60-61.

De Bra, P., & Calvi, L. (1997). Creating adaptive hyperdocuments for and on the Web. *Proceedings of the AACE WebNet Conference*, (pp. 149-155). Toronto, Canada.

Falquet, G., Hurni, J.-P., Guyot, J., & Nerima, L. (2001). Learning by creating multipoint of view scientific hyperbooks. *Proceedings of the European Conference on Computer-Supported Collaborative Learning*, (pp. 222-229). Maastricht, The Netherlands.

Falquet, G., Mottaz-Jiang, C.-L., & Ziswiler, J.-C. (2004). Ontology integration to access a library of virtual hyperbooks. *Proceedings of the European Conference in Digital Libraries (ECDL)*, Bath, England.

Falquet, G., Nerima, L., & Guyot, J. (1999). Languages and tools to specify hypertext views on databases. In P. Atzeni, A. Mendelzon, & G. Mecca (Eds.), *The World Wide Web and Databases*. LNCS (vol. 1590). Berlin, Germany: Springer.

Falquet, G., Nerima, L., & Ziswiler, J.-C. (2004). Adaptive mechanisms in a virtual hyperbook. *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT)*. Joensuu, Finland.

Fröhlich, P., & Nejdl, W. (1997). A database-oriented approach to the design of educational hyperbooks. *Proceedings of the Intelligent Educational Systems on the World Wide Web workshop at the 8th World Conference of the AIED Society*, (pp. 18-22). Kobe, Japan.

Garret, L. N., Smith, K. E., & Meyrowitz, N. (1986). Intermedia: Issues, strategies and tactics in the design of a hypermedia document system. *Proceedings of CSCW '86*. Austin, TX.

Halasz, F., & Schwartz, M. (1994). The Dexter hypertext reference model. *Communications of the ACM, 37*(2), 30-39.

Iksal, S., Garlatti, S., Tanguy, P., & Ganier, F. (2001). Semantic composition of special report on the Web: A cognitive approach. *Proceedings of the H2PTM'01 Conference*, Valenciennes, France.

Landoni, M., Crestani, F., & Melucci, M. (2000). The visual book and the hyper-textbook: Two electronic books one lesson? (pp. 247-265). *RIAO Conference Proceedings*.

Landow, G. P. (1998). *Hypertext 2.0. The convergence of contemporary critical theory and technology*. Baltimore, MD: Johns Hopkins University Press.

Nanard, J., & Nanard, M. (1993). Should anchors be typed too? An experiment with MacWeb. *Proceedings of the Fourteen ACM Conference on Hypertext and Hypermedia (Hypertext 1993)*, (pp. 51-62).

Rada, R. (1990). Hypertext writing and document reuse: The role of a semantic net. *Electronic Publishing, 3*(3), 125-140.

Ranwez, S., & Crampes, M. (1999). *Conceptual documents and hypertext documents are two different forms of virtual document.* Workshop on Virtual Documents Hypertext Functionality and the Web at the 8th International World Wide Web Conference, Toronto, Canada.

Wu, H., de Kort, E., & De Bra, P. (2001). Design issues for general-purpose adaptive hypermedia systems. *Proceedings of the Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext 2001)*, (pp. 141-150). Aarhus, Denmark.

Reusing Educational Material for Teaching and Learning: Current Approaches and Directions

ANNE-MARIE VERCOUSTRE
INRIA, France
Anne-Marie.Vercoustre@inria.fr

ALISTAIR McLEAN
CSIRO- ICT Centre, Australia
Alistair.McLean@voxsurf.com

In this article we survey some current approaches in the area of technologies for electronic documents for finding, reusing, and adapting documents for teaching or learning purposes. We describe how research in structured documents, document representation and retrieval, semantic representation of document content and relationships, and ontologies could be used to provide solutions to the problem of reusing educational material for teaching and learning.

E-learning involves different aspects of using electronic documents for learning-related activities. It ranges from managing curriculum courses on the Web (advertising, registration, scheduling, exams, etc.), to online classes, publishing course material for the students, and dedicated online tutorial systems. A lot of effort has been dedicated to creating high-quality and relevant online learning material, as well as the design and implementation of systems that support users in their learning process. Recent research has focused on adaptive learning environment that can personalise the learning experience.

However, as pointed out by Casey and McAlpine (2003), “anyone who has had to create learning materials from scratch knows just how labour intensive and time consuming the process can be, even with the existence of a detailed course descriptions and lesson plans. This creative process can be made easier by the reuse of existing teaching and learning materials.”

Preparing learning material typically involves:

- finding good document sources relevant to the topics and to the audience;

- selecting more specific parts of documents that could be reused, in particular graphics, tables, images, which have a high illustrative power, and creating new material that can be adapted for personalisation and future reuse;
- defining the sequence in which documents and fragments about some concepts should be accessed or presented (prerequests); and
- defining the curriculum planning that would fit with the pedagogic approaches, and that will hopefully adapt to the actual learner.

In this article we survey how technologies for electronic documents are being used for finding, creating, and adapting material for teaching and learning purposes. We try to identify current approaches and future directions that could support the reuse of existing curriculum material as well as instructional design.

The article is organised as follows: first a section on indexing and finding existing relevant educational material; the next section is concerned about the creation, retrieval, adaptation, and assemblage of fragments of documents; followed by a section on contrasting the navigation and access capabilities offered in tutoring systems compared to open learning environments; next we study how to integrate textual material with active components such as programs; then we offer some directions to define and implement reusable instructional design, and finally we present our conclusions.

FINDING EXISTING DOCUMENTS

Nowadays many documents can be found on the Web and used for self-learning. For example there are online tutorials, basic and advanced courses, opinions and advice, book references, and research papers.

Search engines such as Google rank documents that are pointed to by other web pages (implicit recommendation). A typical example would be asking Google for Java tutorials and getting back what look like very good answers on the first page only: you can choose between the Sun tutorials, the IBM pages, the Java Café, and so forth, or you may prefer to start with the hub assembled by Marty Hall (from the Johns Hopkins University Applied Physics Lab), or the online tutorial by Richard G. Baldwin.

However, it is very difficult to select the best document or references amongst so many answers and some extra time must be devoted to assess the quality of the documents, for example, by looking at the qualifications of the authors and cross-references using CiteSeer, or reading recommendations by other users. You may also have to carefully check for copyright statements or licence agreements before using documents and software. Furthermore, some sites that offer "distance learning courses" are effectively scams that pretend to offer real academic courses and diplomas. However these diplomas are false and often the material is scant and ill prepared.

To provide high quality learning material, many educational bodies have created Educational Libraries that index the learning material using metadata that can support a more precise selection. Examples of such libraries are the Gateway to Educational Materials (GEM)[1] and WebCT[2] in the US, Careo[3] in Canada, EdNA[4] and LRC[5] in Australia, ARIADNE[6] and SchoolNET[7] in Europe.

The advantage of these digital repositories over the Web is that, like classical libraries, they hold much more metadata on each of the resources that can help students, teachers, and systems to retrieve more relevant documents than with full text search. Some of them, such as Merlot also include annotations and peer reviews.

However, there is no universal metadata standard for learning materials and many different standards such as IMS[9] [2], UKOLIN[10], and LOM[11] are being used. For comparison between metadata standards for education see Easel (2002). The Dublin Core metadata is a first attempt to build a simple common standard for resource discovery on the Web. The Dublin Core Educational Working Group (DCMI)[12] has recognised the need for adding to the 15 core elements some elements specific to educational purposes, such as "Audience" (who would benefit the material), "conformsTo" (learning objectives), "Pedagogy" (process to achieve the learning objective), and "Quality." "Quality," sometimes replaced by "Standard," is aimed at certifying that the material has been evaluated for educational purpose by some recognised body.

The Resource Description Format (RDF) could provide a higher level description where documents and concepts can be linked together, as well as concepts between themselves. Amann, Fundulaki, and Scholl (2000) have proposed to query a digital library through an ontology and a thesaurus that have been integrated using an RDF format. This provides a rich description to the resources that can be shared by the community. Carmichael (2002) advocated the importance of the "assessment for learning" in describing reusable educational resources. He is using the Dublin Core qualified for that purpose but also RDF metadata to describe classroom activities and their relationships to broader educational strategies. We will come back to educational strategies and instructional design in a later section.

ASSEMBLING FRAGMENTS OF DOCUMENTS

In recent years, a lot of research has been dedicated to developing flexible learning material that can deliver personalised courses depending of a number of factors such as the user's learning preferences, his current knowledge based on previous assessments, or previous browsing in the material. Authoring such courses requires the authors to define reusable chunks of documents that can be retrieved, adapted, and assembled in a coherent way for a given educational purpose.

De Bra and Calvi (1998) created an adaptive hypermedia system (AHA), where the content of pages is adapted to the user by assembling fragments and fragment variants. The user model is created dynamically based on which pages the user has already read and which problems have been successfully solved.

More sophisticated approaches for dynamically generating or assembling coherent pages involve natural language generation are in Peba II (Milosavljevic, 1997) or Tiddler (Wilkinson, Lu, Paradis, Paris, Wan, & Wu, 2000). In Peba II, comparisons between animals are generated on the fly depending on the user and the animal descriptions that have already been read. In Tiddler the selection of the fragments and the coherence of their composition, including natural language text generation, is driven by a task-driven discourse model. If the task was a learning task, the discourse model could reflect the instructional steps defined by the chosen instructional design.

Virtual documents are based on declarative specifications for retrieving and dynamically assembling fragments of existing documents (Vercoustre & Paradis, 1997). Personalised virtual documents used in educational systems select fragments based on the user model and rich semantic descriptions of the fragments (Iksal, 2002). A common approach in the personalised virtual document community[13] is to describe fragments in term of concepts that are part of a domain or application ontology. Concepts are related to each other by standard ontology relationships as well as prerequisites. A concept cannot be learned before prerequested concepts are all understood. Consequently document fragments related to a concept will not be proposed by the system before fragments related to prerequested concepts have been accessed. In more intelligent learning systems, tests are proposed to the user to check whether the concepts are sufficiently understood. We will come back to this aspect in a later section.

Unfortunately, the way fragments are described and used is very much system and application dependant and cannot be reused by another system for another learning experience on the same topic but with a different objective, or a different instructional method. Most often the fragments have to be written from scratch with the particular application in mind.

Learning Object

An attempt to overcome this problem is to define and create learning objects. This is the objective of the IEEE's Learning Object Metadata (LOM) project[14] who gives this definition of learning objects: "A learning object is any resource or content object that is supplied to a learner by a provider with the intention of meeting the learner's learning objective(s)….and is used by the learner to meet that learning objective(s)."

An important aspect of the LOM model compared to library catalogues is that it incorporates metadata relevant to curriculum design and teaching methodology in addition to descriptions of content and authorship. It uses standards such as DC or IMS and extends them to describe learning objects in a similar way to the Dublin Core Educational Working Group (DCMI) for full documents, but with a stronger focus on the learning objectives.

The LOM project also recognises that "learning content has generally been developed in conjunction with some sort of learning system that keeps track of learners. As the learners interact with the content results are passed back to the system. If the system allows it, the content can also change its behaviour based on learner information stored in the system."

Although intended to be reusable, the learning objects do not carry with them the instructional structure in which they should or could be used. The instructional design is traditionally contained in the document itself. This is lost when the document is broken into small objects and must be hard coded in each learning system that reuses them. Jonassen and Churchill (2004) questioned whether there is any learning orientation in learning objects. Of course this would depend on the granularity of the learning objects. If they are large objects (documents) that contain their self argumentation then there is a need for accessing the internal learning objects, possibly with Open Learning Objects (OLOs) as proposed by Shi, Rodriguez, Chen, and Shang (2004).

Thus, it would be important to be able to reuse parts of documents that have been written as self-contained learning documents and carry with them their full argumentation model.

XML Retrieval

An alternative to independent learning objects described by external metadata is to create teaching and learning materials that contain enough information that allows them to be reused in new situations. To achieve this we need the materials to be structured in such a way that we can also retrieve their smaller constituent parts (i.e., parts of individual lessons).

Describing learning objects and documents in XML could help making them more reusable and adaptable. First, XML can make the structure of reusable chunks explicit and automatically processed. Second, it preserves the context in which a fragment has been created and can be made available to teachers and students to help understanding the value of the fragments.

Examples can be drawn from the experience with the INEX working group on XML search evaluation (Fuhr, Malik, & Lalmas, 2003). In its first year, INEX working groups took on a series of retrieval tasks (queries) on a large collection of XML documents (about 12,000 articles in the IEEE Computer Society publications since 1995). One of the proposed topics involved "finding figures about the Corba architecture and the paragraphs that refer to them."

It is well recognised that document elements such as figures or tables can be more concise than a long discourse and have high pedagogical value. However, a figure without its caption is hardly understandable and often requires complementary information. Good XML retrieval engines should be able to retrieve such elements (and rank them) while providing some context, or the full embedding document as part of the answer.

This example was taken from an XML collection for which the DTD is very much publishing oriented and does not contain many tags that are semantically significant (such as figures, tables, bibliographic references). Its tags are mostly structural, such as section, paragraphs, lists, and so forth, and, in this case, more explicit metadata may be required for fragments of documents to be directly used in a learning environment.

Another drawback in querying XML documents is the possible heterogeneity of the DTDs for different collections. It should not be expected that the users, or even a given learning system, could know the actual tags used in different collections. Amann, Fundulaki, Scholl, Beeri, and Vercoustre (2001), have proposed to query XML collection through an ontology where concepts and relations in the ontology have been mapped to fragments of XML documents.

More semantic metadata can also be attached to fragments of existing XML documents (when preparing a new course) using RDF description and URIs that refer to those fragments (e.g., using Xpaths). The RDF metadata are then seen as external annotations to the material and different authors can create their own, or possibly reuse existing ones. This is the approach taken in ELM-ART (Brusilovsky, Schwarz, & Weber, 1996) where flexible and personalised browsing is built upon existing documents.

TUTORING SYSTEM VERSUS OPEN LEARNING ENVIRONMENT

In traditional books and textual documents, the organisation of the learning material is decided by the author and the learner is expected to read the document linearly, although nothing prevents him to jump to the conclusions first or to skip a section if he is already familiar with the concepts. The flexible nature of hypertexts and online materials offers new opportunities and challenges for learning support that can guide the learner in a more personalised way. In particular, when the content is split into smaller units, the learning system is expected to provide some guidance as to which part to read next.

Eklund, Brusilovsky, and Schwarz (1997) have developed "Interbook" that provides adaptive navigation support. The system records previous user's navigation to infer what knowledge the user has already acquired and suggest links to access other pages based on the prerequisites for those pages. Eklund, Brusilovsky, and Schwarz (1998) studied the use of link annotation in educational hypermedia, while De Bra and Calvi (1998) discussed the use of colour link annotations and link hiding to provide better guidance to the learner (De Bra & Calvi, 1998). They compared learning interfaces where only a "next" button is provided with interfaces where a broader choice is offered. They concluded that, in this particular experiment, beginners may prefer a strong guidance while more experienced learners would access more material with more open choices.

Intelligent Tutorial Systems, as their name suggests, are designed to provide strong support to the learner and try to propose to the user only the best recommendation for the next step in the learning process. However, Hübscher, and Puntambekar (2001) questioned the positive learning effect of very strict guidance, arguing that "more guidance does not necessarily result in more learning." Instead of embedding the macro-structure in the text with hyperlinks, they proposed that the reader's learning process can be more successfully supported with meta-level tools such as concept maps. A concept map presents ideas in the form of nodes which are linked by a word representing a concept. Concept maps are very powerful in helping students see the numerous relationships between concepts and enforce the learning process at a higher level.

Bunt, Conati, Hugget, and Muldner (2001) suggested that Open Learning Environments can be more beneficial for learning than tutor-controlled systems because of the active role the learner plays in knowledge acquisition. They proposed to place less emphasis on explicit instruction and more on providing the learner with tools that support learning through unconstrained exploration of the target instructional domain. However, their system also monitors the users and tries to detect when they experience difficulty. The system provides more guidance only when necessary.

In Open Learning Environments, it is possible to reuse and integrate more material that have been created in other contexts since the system does not have to make strict choices on what to read next; alternatives can be offered. However, it is still very important that a good description of the underlying material is available to the system to automatically generate good concept maps or other meta-level browsing support.

What is missing at this level is a standard way of describing concept maps and, more generally, how the information is related according to instructional intention and strategies.

Problem Solving and Active Examples

So far we have only mentioned textual material (documents, fragments of documents, and their hyper textual organisation) for composing the learning material, that is, material that the learner would read and be expected to understand before going further.

However most online tutor systems also include tools to verify that the user has effectively learned what it was supposed to learn. The user can be asked to answer a few questions, to solve a problem or to write a program (De Bra & Calvi, 1998; Brusilovsky et al., 1996). This allows the system to dynamically update the user model more accurately than just based on documents that the user has previously accessed.

Learning environments therefore have to intermix documents with more active components. Although not many standards have been developed for supporting it, the idea has been presented before as *literate programming*. In 1992, Donald E. Knuth introduced literate programming, a methodology that is defined as the combination of documentation and program source together in a fashion suited for reading by human beings. He created the original literate programming tool called WEB, which he used to write TeX and MetaFont.

The idea is that the documentation used for learning a programming language should include active examples of what the language offers. By active we mean examples that the user can test and get results that are immediately included in the embedding document. In this vision, "a program is also a document that teaches programming to the reader through its own example."

A recent XML-based proposal could become a standard way to include programs and activable components into teaching material. *Active XML* is currently developed for supporting the activation of services from XML documents and returning their value under the form of XML data that can be included into the initial document (Abiteboul, Benjelloum, Manolescu, Milo, & Weber, 2002). Although Active XML is very new and not standard, a similar approach could lead to more reusable and rich learning material.

Instructional Design

So far we have discussed how information can be reused based on its content. As described in previous sections, existing approaches annotate fragments or learning objects with semantic descriptions taken from an ontology of concepts. A concept cannot be learned before prerequested concepts are all understood. If we assume for the moment that standard ontologies are accepted for specific domains then we can imagine a system and/or author that is able to coherently reuse fragments created by others.

However, such a system or author is limited to reusing the fragment within the implicit instructional intent of the original author. If, for example, we create fragments consisting of (a) a diagram illustrating the parts of an engine and (b) a photograph of an engine and describe both with concept-based content metadata such as "engine," then these fragments can only be retrieved (for reuse) with a general query. Human inspection will be required to decide on the most appropriate fragment for reuse in the new course.

Kabel, de Hoog, Wielinga, and Anjewierden (2003 ; Delestre, Pécuchet, and Gréboval (1999), noted that to make information truly reusable for teaching then information fragments need to be annotated with description-al and instructional metadata as well as content or domain metadata. If we annotate the fragments further as (a) engine: schematic representation (specific): theoretical knowledge (illustration), and (b) engine: photo (specific): factual knowledge (example) then we can make specific instructional and domain queries when constructing the course.

Going in that direction, Tutor (Czarkowski & Kay, 2001) uses an Adaptive Teaching Mark-up language to describe the course maps, the learner parameters and a set of lessons (the actual teaching material) that may be adapted.

The University of Passau in Germany has developed a didactical reference model, a teachware model and a mark-up language based on Instructional design (Süß, Freitag & Brössler, 1999). The teachware model describes the modular structure of the learning content, while the didactical model describes its didactical structure that can reflect different pedagogical model using the same material.

To use such marked-up data a rich set of instructional strategies are required along with the conditions in which they are appropriate. Curriculum authoring should be supported by good instructional designs established by Instructional Science.

Instructional Science is based on the psychology and sociology of learning and consists of theories, models, and methodologies for instruction and contains both descriptive and prescriptive components – the latter forms part of what is called instructional design. Instructional design is domain independent and theory based. The use of such knowledge will be required in writing instruction-aware learning systems and it may be that RDF (in the form of DAML+OIL – may be used to represent this knowledge in both a Human readable and machine readable form.

For these strategies to be related to the instructional intention of the authored information fragments, both the fragments and the instructional strategies need to be "ontology-aware" (Mizoguchi & Bourdeau, 2000). An instructional ontology includes concepts such as the learning goal, definitions, background, example, explanation, reminder, and so forth.

Describing instructional strategies with RDF-based ontologies will allow both authors to manually implement these strategies or adaptive systems to automatically process them. In this area the Ontology Inference Layer OIL is a proposal for a web-based representation and inference layer for ontologies (Fensel, Horrocks, Van Harmelen, Decker, Erdmann, & Klein, 2000). It combines the widely used modelling primitives from frame-based languages with the formal semantics and reasoning services provided by description logics. It is compatible with RDF Schema (RDFS) and includes precise semantics for describing term meanings (and thus also for describing implied information).

The DARPA Agent Markup Language (DAML) is an effort to develop a language and tools to facilitate the concept of the semantic web. The DAML group pooled efforts with the Ontology Inference Layer to propose

DAML+OIL, a language for expressing far more sophisticated classifications and properties of resources than RDFS. DAML+OIL is a current W3C proposal (www.w3.org/Submission/2001/12/) for a semantic markup language for web resources. Some current research is looking at building reasoning support for the language (Broekstra, Klein, Decker, Fensel, & Horrocks, 2000).

Conclusion

We have surveyed research in the area of technologies for electronic documents and shown that there are many relevant areas that the AIED community could draw on to allow educational material to be reused when creating a new course, whether that is done by an author or a system. In particular:

- Electronic document technologies can provide standard formats for describing curriculum material and associated metadata at different levels of granularity.
- While XML can provide a rich format for describing fully authored documents that support extraction of fragments, RDF provides a flexible and rich description for selecting and combining fragments to provide a more personalised learning experience.
- Active XML may provide a standard way to augment standard passive course material by embedding and activating problem solving modules into the learning material.
- To take advantage of instructional design, based on the psychology and sociology of learning, we need to represent instructional strategies in both human and machine readable form. The problem of representing instructional intention for educational material and being able to use it through appropriate application of instructional strategies may be resolved by drawing on ontology research; work in the semantic web with the DAML+OIL W3C submission appears to be particularly relevant. As tools appear that can reason with fragments of information marked up with DAML+OIL we may see the emergence of authoring environments that help the teacher compose new courses based on existing material and her teaching style. Eventually we would hope to see automated learning environments that are able to construct new curricula based on a learner's domain request and instructional preference through the reuse of existing educational material.

References

Abiteboul, S., Benjelloum, O., Manolescu, I., Milo, T., & Weber, R (2002, August). Active XML: Peer-to-peer data and web services integration. *Proceedings of the 28th International Conference on very Large Data Bases* (VLDB 2002), (pp. 1087-1090), Hong Kong, China.

Amann, B., Fundulaki, I., & Scholl, M. (2000). Integrating ontologies and thesauri for RDF schema creation and metadata querying. *International Journal of Digital Libraries (JODL), 3*(3), 221-236

Amann, B., Fundulaki, I., Scholl, M., Beeri, C., & Vercoustre, A.-M. (2001, May). Mapping XML fragments to community web ontologies. *Proceedings of the Fourth International Workshop on the Web and Databases (WebDB'2001)*. Santa Barbara, CA.

Broekstra, J., Klein, M., Decker, S., Fensel, D., & Horrocks, I. (2000, September). Adding formal semantics to the web: Building on top of RDF schema. *Proceedings of the ECDL Workshop on the Semantic Web* (SemWeb2000), Lisbon, Portugal. Retrieved from http://www.ics.forth.gr/isl/SemWeb/proceedings/session2-2/paper.pdf

Brusilovsky, P., Schwarz, E., & Weber, G. (1996). ELM-ART: An intelligent tutoring system on World Wide Web. *Proceedings of the Third International Conference on Intelligent Tutoring Systems*, ITS-96, Berlin. Retrieved from http://www.psychologie.uni-trier.de:8000/projects/ELM/elmart.html

Bunt, A., Conati, C., Hugget, M., & Muldner, K. (2001). On improving the effectiveness of open learning environments through tailored support for exploration. *Proceedings of AIED 2001, 10th World Conference of Artificial Intelligence and Education*, San Antonio, Texas. Retrieved from http://www.cs.ubc.ca/%7Econati/my-papers/aied2001.pdf

Carmichael, P. (2002) Learning how to learn: Using the Dublin core metadata element set to support teachers as researchers. *Proceedings of the International Conference on Dublin Core and Metadata for e-Communities*, (pp. 201-203).

Casey, J., & McAlpine, M. (2003). *Writing and using reusable educational materials- a beginner's guide.* Retrieved from http://www.gla.ac.uk/rcc/staff/mhairi/index.html

Czarkowski, M., & Kay, J. (2001). Tutor: Support for scrutably personalised documents. *Proceedings of the 6th Australian Document Computing Symposium (ADCS 2001)*. Retrieved from http://www.ted.cmis.csiro.au/adcs01/011_Czarkowski.pdf

De Bra, P., & Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia, 4,* 115-139.

Delestre, N., Pécuchet, J-P., & Gréboval C. (1999) How to design an adaptive dynamic hypermedia for teaching? *Proceedings of AI-ED'99*, (pp. 654-656) Le Mans, France.

Educator Access to Services in the Electronic Landscape (EASEL) (2002). Evaluation Report, EASEL - IST Project 10051 Report, March 2002.

Eklund, J., Brusilovsky, P., & Schwarz, E. (1997, July). Adaptive textbooks on the WWW. *Proceedings of the Third Australian Conference on the World Wide Web (AUSWEB97)*, Queensland, Australia.

Eklund, J., Brusilovsky, P., & Schwarz, E. (1998). A study of adaptive link annotation in educational hypermedia. *Journal of Universal Computer Science, 4*(4), 429-448.

Fensel, D., Horrocks, I., Van Harmelen, F., Decker, S., Erdmann, M., & Klein, M. (2000). OIL in a nutshell. Lecture Notes in Artificial Intelligence. In R. Dieng (Ed.), *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management (EKAW'00)*, (pp. 1-16). Berlin/Heidelberg, Germany: Springer-Verlag.

Fuhr, N., Malik, S., & Lalmas,M. (2003, December). Overview of the initiative for the evaluation of XML retrieval (INEX) 2003. In N. Fuhr, M. Lalmas & S. Malik (Eds.) *Proceedings of the 2nd Workshop of the INitiative for the Evaluation of XML Retrieval (INEX)*, Dagstuhl, Germany. Retrieved from http://inex.is.informatik.uni-duisburg.de:2003/

Hübscher, R., & Puntambekar, S. (2001). Navigation support for learners in hypertext systems: Is more indeed better? In J.D. Moore, C.L. Redfield, & W.L. Johnson (Eds.), *Artificial intelligence in education: AI-ED in the wired and wireless future* (pp. 13-20). Burke, VA: IOS Press.

Iksal S. (2002). *Spécification déclarative et composition sémantique pour des documents virtuels personnalisables*. Doctoral thesis, ENST-Bretagne, December, 2002.

Jonassen, D., & Churchill, D. (2004). Is there a learning orientation in learning objects?. *International Journal of E-Learning, 3*(2), 32-41.

Kabel, S., de Hoog, R., Wielinga, B., & Anjewierden, A. (2003). The added value of task and ontology-based instructional markup for information retrieval. *Journal of the American Society for Information Science and Technology (JASIST), 55*(4), 348-362.

Knuth, D.E. (1992). *Literate programming*. CLSI (Center for the Study of Language and Information) Lecture Notes Number 27, Leland Stanford Junior University.

Milosavljevic, M. (1997). Augmenting the user's knowledge via comparison. *Proceedings of the 6th International Conference on User Modelling*. Sardinia, Italy.

Mizoguchi, R., & Bourdeau, J. (2000). Using ontological engineering to overcome common AI-ED problems. *International Journal of Artificial Intelligence in Education, 11*(2), 107-121.

Shi, H., Rodriguez, O, Chen, S., & Shang, Y. (2004). Open learning objects as an intelligent way of organizing educational material. *International Journal on E-Learning, 3*(2), 51-63.

Süß, C., Freitag, B., & Brössler (1999, November). Metamodeling for web-based teachware management. In P.P Chen, D.W. Embley, J. Kouloumdijan, S.W. Liddle, & J.F. Roddick (Eds.), *Proceedings of the International Workshop on the World-Wide Web and Conceptual Modeling (WWWCM`99) in conjunction with ER`99*. Paris, France. LNCS 1727. Paris, France: Springer Verlag.

Vercoustre, A-M., & Paradis, P. (1997, November). A descriptive language for information object reuse through virtual documents. *Proceedings of the 4th International, Conference on Object-Oriented Information Systems (OOIS'97)*, (pp. 299-311). Brisbane, Australia.

Wilkinson, R., Lu, S., Paradis, F., Paris C., Wan, S., & Wu, M. (2000, August). Generating personal travel guides from discourse plans. *Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems.* Trento, Italy.

Notes

[1]GEM (The Gateway to Educational Materials), http://gem.syr.edu/

[2]WebCT, www.webct.com/otl

[3]Careo (Campus Alberta Repository of Educational Objects), Canada, www.careo.org

[4]EdNA (Education Network Australia), http://www.edna.edu.au/metadata/

[5]LRC (Learning Resources Catalogue), http://www.hkulrc.unsw.edu.au/

[6]ARIADNE, Alliance of Remote Instructional Authoring and Distribution Networks for Europe, http://www.ecotec.com/sharedtetriss/projects/files/ariadne.html

[7]SchoolNet(Europe), http://www.eun.org/portal/index-en.cfm

[8]MERLOT (California), http://www.merlot.org/Home.po

[9]IMS (Instructional Management System Global Learning Consortium) standards for teaching and learning materials. http://www.imsproject.org/

[10]UKOLN Metadata for Education Group, http://www.ukoln.ac.uk/metadata/education/

[11]IEEE Learning Technology Standards Committee (LTSC), http://ltsc.ieee.org/wg12/

[12]DCMI Education Working Group, http://dublincore.org/groups/education/

[13]Workshop on Documents Virtuels Personnalisables, 2002, http://iasc.enst-bretagne.fr/DVP2002/programme.htm

[14]Learnet, http://learnet.hku.hk/objects.htm

Use of Relational and Conceptual Graphs in Supporting E-Learning Tasks

ANKUSH MITTAL
Indian Institute of Technology, Roorkee, India
ankumfec@iitr.ernet.in

KRISHNAN V. PAGALTHIVARTHI
Indian Institute of Technology, Delhi, India
kvp@am.iitd.ernet.in

The learning process for a user becomes seriously restrictive in trying to discover the relationships between concepts and in searching for a part of concept (called an object) such as a solved example illustrating the concept or the application of the concept. This article presents the theory for building a relational graph that depicts how concepts are linked to each other. By selecting to zoom in on a particular concept, the user's view changes to conceptual graph where s/he can view and access all the objects related to a particular concept. Rule-based algorithms are presented to identify objects of a concept, to determine concept boundary, and to build the trees. The lecture material on Algorithms course from MIT is used for experimentation of the ideas. In addition to efficient searching for a desired topic, the system also enhances the understanding and the learning of the user.

Education through personal interaction with a teacher in a small classroom is well known to be more effective than education through texts or e-learning materials (Galusha, 1997). One of the key differences between the two paradigms is that a teacher expounds upon the relationship between various concepts of the course (and perhaps, between different courses too), which otherwise seem to be disconnected. Another advantage of learning personally from a teacher is that one could clarify a concept by asking questions and the teacher might further illustrate by giving more insight either through examples, or through analysis.

A beginner of a subject generally has to struggle with catching up with the terminology of the subject. Also, the learner does not understand or know where s/he can find the related terms and concepts mentioned in the lecture. The index provided at the back of the textbook is only elementary to serve all purposes. Moreover, for e-learning materials such as PowerPoint slides, even a table of content or an index is not provided. Searching for a particular concept or an object by giving a phrase (say "Give me an example of merge sorting" or "Give me definition of Big O") and matching for the phrase would also fail in most cases because many times the words like "example" or "definition" are not present on the slides (though they can be inferred through context of slides or any other way). Formal definition of the object is given later in this article. A few examples of the object type are "definition," "examples," "classes," and so forth.

With the rapid growth of educational support material for e-learning on the Web, technologies for enhanced learning and retrieval are desirable. The rapid success of distance education has led to extensive development of course material and its placement on the Web. The learner has access to the lectures and navigates through it trying to get what he wants. The important aspect of this style of learning is that the learner should be able to get what he is looking for without getting lost in the labyrinth of hypertexts and links.

This article is aimed at enhancing the user's learning process by providing a relational graph that can be compared to a map showing links between the cities. Users can get an overview of the subject as well as the relationship between concepts that would help them grasp the fundamentals as well as review concepts when preparing for examinations. A concept graph is also provided where a user can select an object of the concept and view the slides related to the object. In addition to depicting the relationships, a user has fast access to enhanced index to the objects. Conceptual graphs have been used in the representation of semantics of the documents (Corby, Dieng, & Hebert, 2000; Brasethvik & Gulla, 2002) to enable semantic-contents-guided search. Semantic web relies on the Resource Description Framework (RDF) (a semantic network inspired language for constructing meta-data statements) and it can not be employed unless RDF is accepted as a standard by the web community. Gelfand, Wulfekuler, and Punch (1998) presented a system for extracting concepts from unstructured text by identifying relationships between words in the text based on a lexical database. As the system does not employ domain knowledge, the relational graph that is built from a course domain (such as Computer algorithms) appears to be disconnected as a majority of the concepts are not lexically related.

E-learning lecture materials from the Singapore-MIT Alliance for the course "An Introduction to Algorithms" prepared by Professor Charles E. Leiserson were used. An overview of the system is shown in Figure 1. The objects such as "solved example" are identified using rules derived from their characteristics. The slides are clustered based on slide titles and their content. Next, concept identification is done where boundaries of a concept are determined. An initial concept tree is given as output. (Note that we use tree in broader sense as a graph rather than restricting its definition to acyclic graph.) Subsequently, the relationship between concepts can be identified from the slides related to the concept and this module refines the concept tree and outputs the relational tree.

This article is organized as follows. An account of related work is presented in the area of clustering documents and identification of concepts. Various steps in our approach are then presented. Next, the results and discussion of the system are presented. Conclusions and the scope of future work follow in the final section.

LITERATURE REVIEW

Related Work in E-Learning

Ip and Chan (1998) in their Automatic Segmentation and Index construction for the Lecture Video (Ip and Chan, 1998) used the lecture notes along with Optical Character Recognition (OCR) techniques to synchronize the video with the text. A hierarchical index is formed by analyzing the original lecture text to extract different levels of heading. But the underlying assumption made, that the slides will always represent a hierarchy, is not always the case. Many slides may have titles, which are in no way related to the previous slide.

The Content Based Retrieval Video System for Educational Purposes (Bibiloni & Galli, 1996) proposes a system using a human intermediary (teacher) as an interpreter to index the video manually. This system although indexes the video, but is highly dependent upon the vocabulary used by the teacher, which may differ from person to person. Moreover, even the same

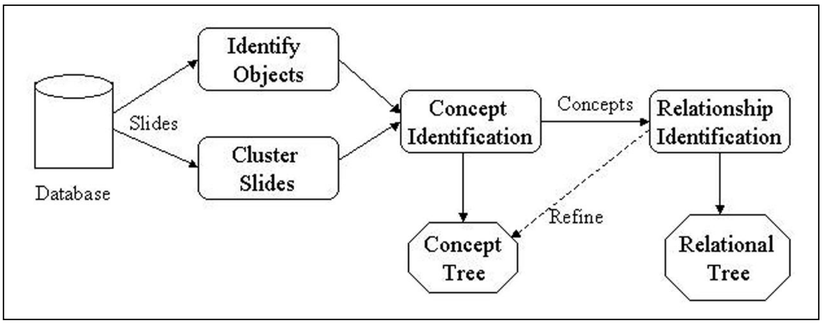

Figure 1. An overview of the system, the input is the set of slides for a course from the database and the outputs are the concept tree and the relational tree

person may have different interpretation of the same image (or video) at different times as pointed by Ip and Chan (1998) .

Hwang and Deshpande (1997) in Multimedia features for course on demand in distance learning, proposed a hyper video editor tool to allow the instructor to mark various portions of the class video and create the corresponding hyper links and multimedia features to facilitate the students access to these prerecorded sequences through a web browser. This scheme also requires a human intermediary and thus is not generalized.

Related Work in Identifying Concepts

In this article the clustering of slides is used for the task of building concept and relationship tree forming. Clustering of documents has received tremendous attention in the recent past particularly with respect to the web documents (Zamir, Etzioni, Madani & Karp, 1997). Clustering is the ability to automatically create groups of related documents. The important factor is the objective behind "related" aspect. The past work has focused on web pages clustering primarily keeping in mind the issue of efficient searching or summarizing the data (Zizi & Beaudouin, 1994). Work has also been driven by the objective of discovering hidden similarities making use of the keyword the user supplies in the query. Results of such searches have been the delivery of large clusters. Since clustering is based on low-level characteristics, the retrieved documents are sometimes unrelated to what the user had in mind.

There have been several works reported on the problem of identifying the right set of key words or phrases and automatically linking them in the domain of web pages (Green, 1998; Agosti, Melucci, & Crestani, 1995). However, application to learning material is missing, primarily because the identification of key words and phrases is significantly different in the case of learning environment. Specifically, while a web document may be related to other web documents semantically, in general there is not a temporal conceptual relationship. An example of temporal conceptual relationship is a definition of a concept, say "merge sort" followed by a couple of slides on a solved example of merge sort, which in turn is followed by time analysis of merge sort (Figure 2).

The identification of key phrases is mostly based on frequency statistics or title style words (Li & Jain, 1998). However, key phrases identified by this method cannot be accepted as concepts in learning material because, first there are too many slide titles and they would result in too many concepts. For example in the algorithm course, the number of slides is approximately 1040. Second, a slide containing the example of a concept or its definition does not contain the word "definition" or "example" in the slide. For example, Figure 2 shows the solved example of a merge sort where there is no word indicating that it is an example. Figure 3 shows an example of how

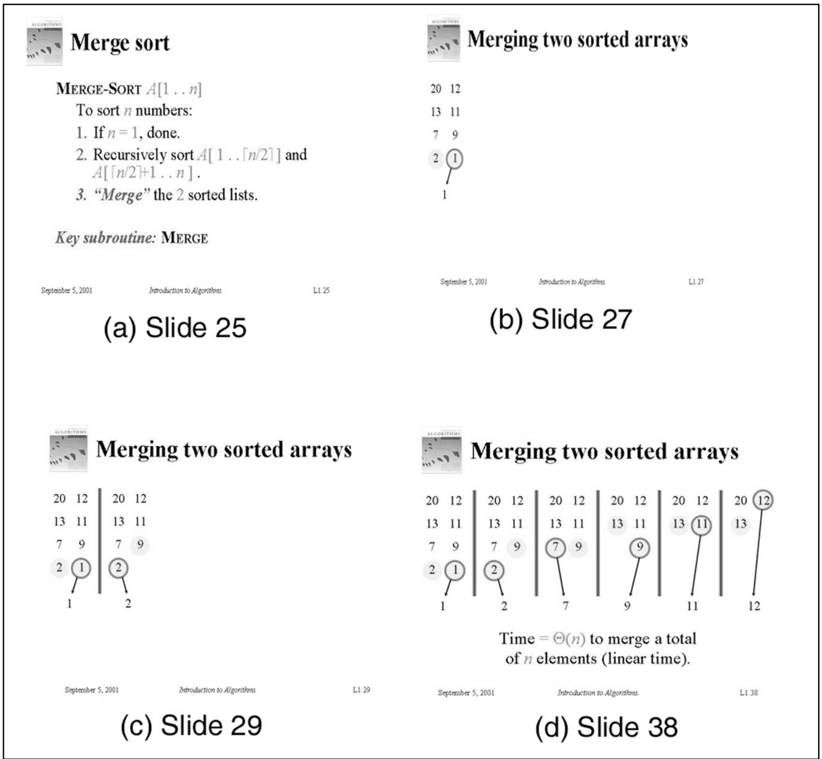

Figure 2. Slides illustrating the context in the presentation, a definition is followed by the solved example

two concepts interact with each other in a single slide title: "Recurrence for Merge Sort." Such relationships would be difficult to discover with the traditional approaches.

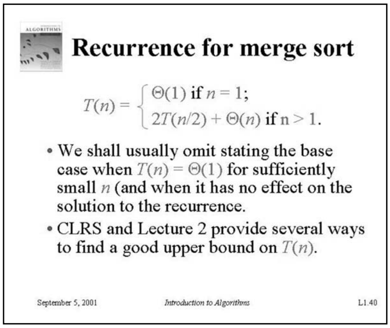

Figure 3. A recurrence concept applied to the concept of merge sort

Other Methods

Different approaches have been taken for retrieval and easy navigation through e-learning materials.

- Knowledge-based: In Knowledge-based Content Navigation in e-Learning Applications (Mendes, Martinez, & Sacks, 2002), a prototype framework for organizing and navigating online learning material with respect to the semantic context of documents was presented. They proposed the use of fuzzy clustering algorithm and *TopicMaps* to discover and represent knowledge, respectively. The usability of system depends on how accurately the algorithm could identify the knowledge domain.

- Keyword Searching: Another commonly used approach is to use traditional Information Retrieval systems based on syntactic matching of query terms with document keywords. However these systems do not take into account how words or concept are semantically related to each other. Also they do not provide the user any scope to explore the structure of the domain while formulating his query.
- Hot Spotting: Some modern systems attempts to return an actual answer by using a set of pattern matching rules (often just bag-of-words) and then augmenting this with one or more NLP techniques. Question Analysis is done to categorize the question based on a shallow parse of the question followed by *Hot Spotting* the answer region (i.e., sentence) using word overlap between question and region. Finally the answer phrase is searched in the answer region (*PinPointing*) and presented to the user. These systems have low accuracy and do not satiate a learner who needs sufficient context in the answer to understand the concept.

OUR APPROACH

Our approach is based on the identification of the relationship between concepts and the identification of objects in a concept. Figure 4 shows a concept tree and its attribute objects such as definition, class, and so forth. This

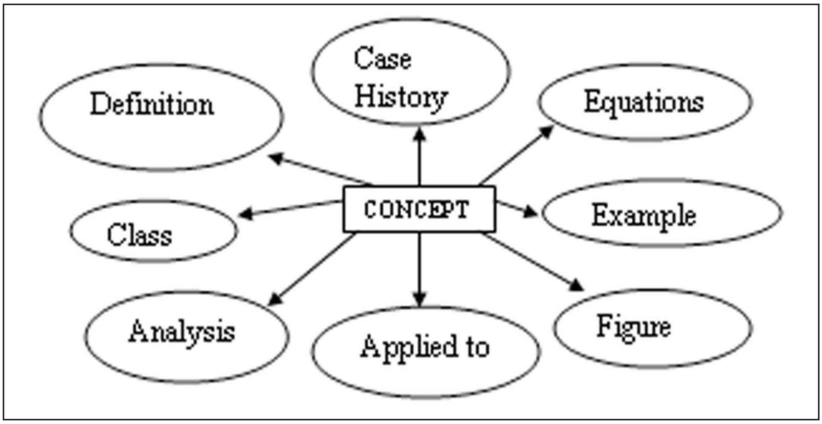

Figure 4. Illustration of generic *concept* with its constituents *objects*

section presents algorithms for determining objects automatically and subsequently identifying concepts. Most of the rules are formed keeping algorithms course and computer science lecture materials in general in mind. However, the ideas presented in this article can be suitably extended to courses such as biology, management, and so forth, by modifying the objects of the concepts and also the rules for determining the objects.

Identification of Objects

The objects are identified using various rules explained for each object.

Solved example: A general practice in teaching is to illustrate a concept through an example. There are two commonly occurring practices: one is that example follows in the same slide as the definition/introduction to the concept (for example, in concept of Big O notation) and second practice is that the example spans a few slides following the definition (e.g., Figure 2).

To detect object example, two steps are taken. First, a search for the word "example" is done. Secondly, a difference of consecutive images of slides is taken. Note that during solved example (see Figure 2), only slight changes occur from one slide to the next. Figure 5 shows the image difference plot for

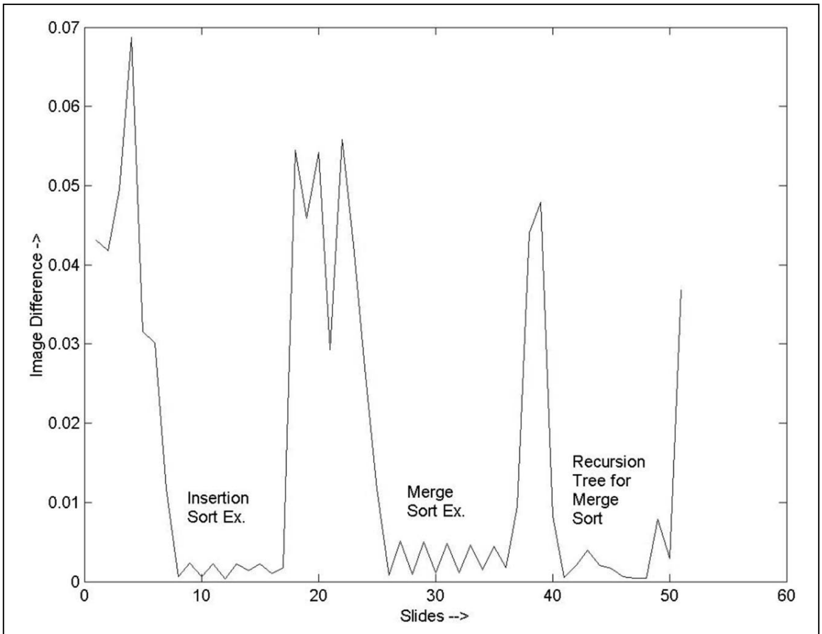

Figure 5. Image difference plot for the first lecture, there are two solved example and one illustration of recursion

the first lecture. For determining the example slides, slides $s_i, s_i+1, \ldots, s_i+_n$ which have image difference less than a threshold α and n>4 are considered. Also, for ruling out cases like recursion tree for merge sort (see Figure 5), only those slides in which the count on numerals is more than a threshold β are considered.

Definition: With definition, the concept is introduced (e.g., see Figure 2). The first occurrence of a concept term in a title slide along with words such as "input," "output," and "description" form cues for detecting definition. Since a concept is explained in consecutive slides, frequent reference to the concept would be made in the title of the slides following it. Also, it is determined that a particular set of slides belongs to "example," it can be postulated that the previous slide is a definition slide. The keywords or the words referring to definition are always red italicized, so if there is a definition in the slide it ought to have a red italicized word.

Analysis: The word "analysis" explicitly occurs in the slides and can be identified in slide title. Synonyms can be identified based on their root.

The figures and equations: Figures and equations can be identified by converting the PowerPoint to the web page. The figures and equations are stored in as separate image files (in .gif format) corresponding to each slide. The figures and equations can be distinguished by the fact that the equations have more length than width.

"Applied To" relation: Identification of this object is done as the refinement step during relationship identification phase. An example of applied to relationship is "Recurrence Analysis" applied to "Merge Sort." First concepts *A* and *B* are identified. For identifying the relationship "A applied to B" the slide titles are found where both concepts *A* and *B* occur. Moreover, concepts A and B should have more associated slides in the vicinity.

Type of / Classes: Sorting is a concept while Merge sort is a "type of" sorting and also in this case, merge sort is a concept in itself. It is found that the identification of global context can determine the "Type of" relationship. For example, first "solving recurrence" problem is described and then various methods (substitution method, master method, and the general method) are explained. The methods belong to the class of concept and have the flexibility to be concepts in themselves.

Clustering of Slides

Our basic philosophy in clustering is based on the assumption that the lecture material is structured like a movie. A movie has sequence of scenes, each conveying a part of story (Nack & Parkes, 1997). A scene consists of shots that bring forth a message. Similarly, course material is structured as series of lectures and lectures are in the form of topics. Just like an individual shot in a movie does not convey meaning, so also the slides have to be understood in context. Thus, the problem gets translated to identifying group of slides that constitutes a topic.

The basic theory behind clustering of slides into concepts follows the analogy of video shots segmentation. In a video, shots are segmented on a criterion that the frames in same shot have more "similarity" than the frames in the neighboring shots (Hampapur, Jain, & Weymouth, 1994). In a concept, collection of slides with repeatedly occurring words or same word can be used as they all describe a particular concept.

Key phrases are frequently used terms in a course that would convey some information. A set of key phrases could be a concept too. An example of key phrase in algorithm course is the word "asymptotically" which is not a concept. We build our set of key phrases by using the following steps: (a) the root of each word is considered for clarity, (b) a frequency count of the root word is done, and (c) key phrases are identified by thresholding on frequency.

After a key phrase set is build for the entire course, the slides are clustered using the similarity measure based on the number of key phrases and the slide titles.

Identification of Concepts

In Mittal, Choudary, and Sung (2003), it is shown that through the diagram for the lecture videos, a new concept starts with a definition or problem statement. Since, the identification of examples is done first and then definition is identified based on criteria mentioned in the section on definition, the beginning of a new concept can be ascertained. The slides are said to belong to this concept until the beginning of another concept is found. Although the algorithm will not generate exact concepts, either they will be missing some concepts or generating extra concepts, the cost of errors is less in this application. A user can deduce the concept details through the concept tree. The concept is named with the intersection of title slide with the key phrases of the lecture. The concept tree is then constructed mentioning the slides numbers associated with each object, so that when a user clicks an object, the appropriate slides are retrieved and shown to the user.

Identification of Relationships Between Concepts

Relational graphs are finding many applications in video representation (Ozer, Wolf, & Akansu, 2000), and text information retrieval (Ounis & Huibers, 1997). There are several types of relationship that are possible between two concepts. A concept *X* can be class of concept *Y*. A concept *X* can be prerequisite for understanding concept *Y*. A concept *X* can be applied to concept *B*. For instance, recurrence analysis can be applied to merge sort.

In general, it is known that if a concept *X* occurs in a slide related to concept *Y*, then X is related to Y. Algorithms to discover "class of" and "applied to" relationship were presented in previous sections. Note that with respect to example of "recurrence analysis of merge sort," it can be considered that both "merge sort" is applied to "recurrence analysis" and "recurrence analysis" is applied to "merge sort." For finding out prerequisite relationship, the order of presentation of concepts *X* and *Y* is considered. If *X* occurs before *Y*, and *X* is found to be related to *Y* in definition slides, then *X* is considered to be prerequisite concept required for understanding concept *Y*. The prerequisites of *X* are eliminated from being prerequisites for *Y*. (Table 1)

RESULTS AND DISCUSSION

PowerPoint material from Singapore-MIT Alliance course SMA5503[1] lecture videos were used. First four lectures were chosen to demonstrate our results. Table 2 shows the lecture slide contents for four lectures. Figure 6 shows a relational graph for these lectures.

Table 1
The Steps Involved in Building Conceptual and Relational Graphs

Algorithm (summary):
1. First find out solved examples.
2. Cluster slides with same slide titles.
3. Identify Concept. Determine start and end of concept within slides.
4. Build concept graph. Use slide numbers.
5. Identify relationships between concepts. Build Relationship graph.
6. Refine Concept graph (for example, "applied to" field is instantiated at this time)

Table 2
Lecture Content for First Four Lectures

Abstraction Type	Term Used in the Slides	Occurrence in Slides (format: lecture_no.slide_no)
Concept	Sorting	L1.6 to L1.18
Example	Example	L1.6, L2.3-2.25, L4.5 –4.16
Definition	None	L1.22, L2.2,
Equation and Figures	None	L1.7, L1.24, L1.40
Analysis	Analysis	L1.20, L1.24, L1.39
Application To	None	L1.26 – L1.38, L3.11, L3.12 L3.13, l3.25, l3.26
Type Of	None	L1.7 – L1.18

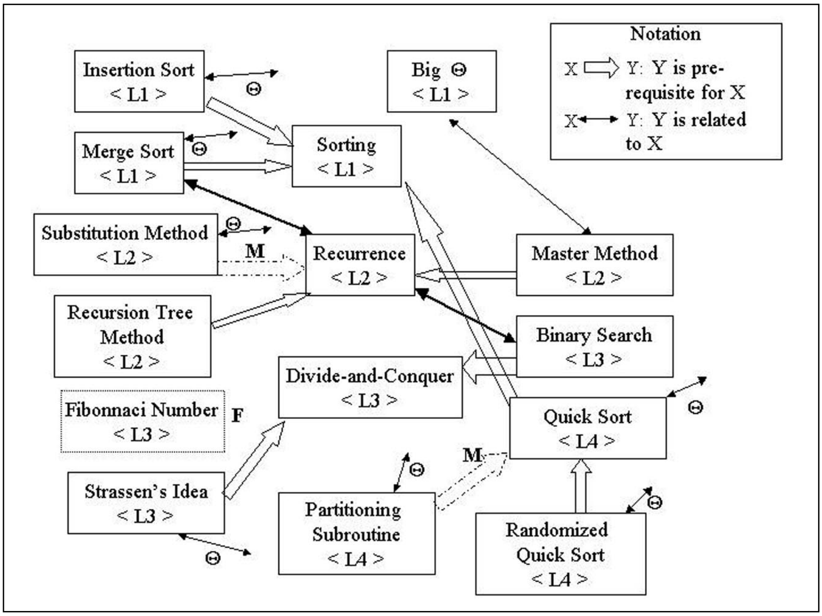

Figure 6. A relational graph for the first four lectures. The missed relationships are denoted by dotted arrow with **M** written on it. The figure shows that Fibonnaci Number is also falsely detected as a concept. In order to maintain clarity, relation between concept X and Θ Is just shown by an arrow with Θ

In four lectures, 15 concepts were found using the rules described. One of these concepts is "Fibonnaci Number" which was actually not a concept but was used as an example of a problem of recurrence relations. Two prerequisite relationships are missed. One is recurrence to be prerequisite of the substitution method. The reason was that there was no mention of recurrence in the definition slide of substitution method. Same was the case with quick sort to be prerequisite of partitioning subroutine.

Since a relational graph for 26 lectures could become large, we propose to display relationships at a distance from the desired concept given by the user. The distance referred to here is in the graph theoretic sense (an edge is counted as unit distance).

A user can select and click on a particular concept from the relational graph, let us say Merge Sort. His view is then shifted to conceptual realm where he sees the objects related to the concept. Figure 7 shows an example of conceptual graph that is displayed to the user.

At the time of query, the learner can view the relational graph and search

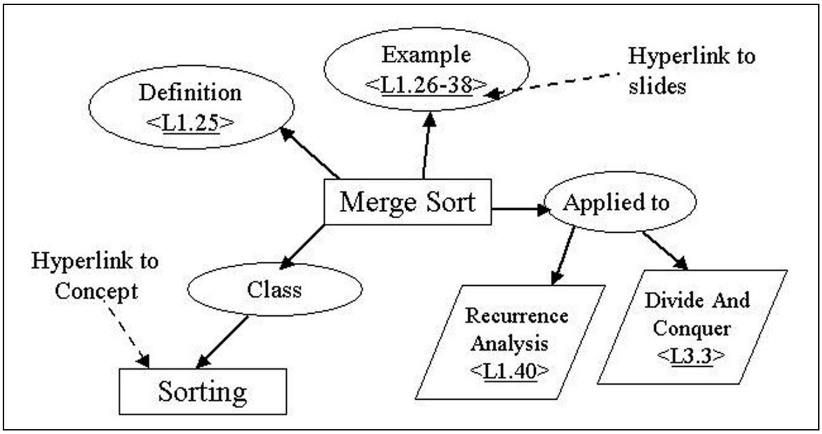

Figure 7. Conceptual graph for "Merge Sort" concept. Some of the objects such as "Applied to" and "Class" can have multiple pointers. A user can click any hyperlink to view the appropriate slides (and synchronized video).

for the appropriate concept. By clicking the concept box the learner navigates through the concept tree and the material is searched efficiently.

The Conceptual and Relational graph obtained from these lectures were used to test its usefulness for the students (studying the course). For this purpose, conceptual questions were collected from the learners to see how efficiently they could be answered by navigating through the concept tree. Table 2 contains a few examples of good questions, which were answered very easily by exploring these graphs. The table also shows some question types that can not be handled by the use of a concept graph. These questions generally involved the concept having missing relationships in the graph.

CONCLUSIONS AND FUTURE WORK

This article presented a framework based on relational and conceptual graphs to enhance learning and searching of concepts and relationships for a learner. For this purpose, the property that the lecture materials are structured to derive rules to identify objects in a concept was used. The rules though specific in one sense to computer science and mathematics like-courses are amenable for modification to other domains. The relational graph gives an overview of the entire subject, while user can get a detailed view by visualizing conceptual graph. Thus, depth-learning and breadth-learning objectives of education can be accomplished.

The rules presented in the article need to be refined further to represent

Table 3
Sample Questions for First Four Lectures

Questions that were answered easily	1. Illustrate with an example how Mergesort uses Divide-and-Conquer method to sort an array
	2. Give me the time complexity analysis of Mergesort.
	3. What are the different methods for Recurrence analysis?
	4. Give various examples of Divide-and-Conquer method
	5. How Strassen's ideas employ Divide-and-Conquer method.
	6. What are the various sorting algorithm that sorts in O(nlogn) time complexity?
	7. What is the asymptotic complexity of Binary search?
Questions that can not be handled	1. How Quicksort uses Partition subroutine
	2. How to obtain tight upper bound by subroutine method analysis
	3. What is the optimal algorithm for Matrix multiplication

semantic relationships too. For example, time analysis is related to Θ Notation. Frequently, time analysis for a concept is done without referring to Θ. Such relationships should be inferred. The relational graphs between various courses in a discipline (and perhaps inter-disciplinary) are also desired.

References

Agosti, M., Melucci, M., & Crestani, F. (1995). Automatic authoring and construction of hypermedia for information retrieval. *Multimedia Systems, 3*(1), 15-24.

Bibiloni, A., & Galli, R. (1996). Content based retrieval video system for educational purposes. *Proceedings of Eurographics Workshop on Multimedia "Multimedia on the Net,"* EGMM 96. Rostock, Germany. Retrieved from http://citeseer.nj.nec.com/53258.html

Brasethvik, T., & Gulla, J.A. (2002). A conceptual modeling approach to semantic document retrieval. *CAiSE'02 - The 14th International Conference on Advanced Information Systems Engineering.* Toronto, Canada.

Corby, O., Dieng, R., & Hebert, C. (2000). A conceptual graph model for W3C resource description framework. *Proceedings ICCS'00* Darmstadt, Germany. LNAI 1867.

Galusha, J.M. (1997).Barriers to learning in distance education. *Interpersonal Computing and Technology, 5*(3-4), 6-14.

Gelfand, B., Wulfekuler, M., & Punch, W.F. (1998). Automated concept extraction from plain text. In *Papers from the AAAI 1998 Workshop on Text Categorization,* (pp. 13-17), Madison, WI.

Green, S.J. (1998). Automated link generation: Can we do better than term repetition? In *Proceedings of the Seventh International World Wide Web Conference,* (pp. 75-84). Brisbane, Australia.

Hampapur, A., Jain, R., & Weymouth, T. (1994). Digital video segmentation. *Proceedings of ACM Multimedia,* (pp. 357-364), Edinburgh, Scotland.

Hwang, J., & Deshpande, S.G., & Sun, M.-T. (1997). Multimedia features for course-on-demand in distance learning. In *Proceedings of IEEE First Workshop on Multimedia Signal Processing,* (pp. 513-518), Princeton, NJ.

Ip, H.H.S., & Chan, S.L. (1998). Automatic segmentation and index construction for lecture video. *Journal of Educational Multimedia and Hypermedia, 7*(1), 91-104.

Li, Y.H., & Jain, A.K. (1998). Classification of text documents. *The Computer Journal, 41*(8), 537-545.

Mendes, M. E. S., Martinez, E., & Sacks, L. (2002). *Knowledge-based content navigation in e-learning applications.* Paper presented at The London Communication Symposium, London, UK.

Mittal, A., Choudary, K.R., & Sung, K. (2003). Content-based retrieval systems for personalization of educational videos. In *Proceedings of The 11th International Conference on Artificial Intelligence in Education (AIED 2003)*, (pp. 1-8), Sydney, Australia.

Nack, F., & Parkes, A. (1997). The application of video semantics and theme representation inautomated video editing. *Journal of Multimedia Tools and Applications,* (pp. 57-83).

Ounis, I., & Huibers, T. (1997). A logical relational approach for information retrieval indexing. In *Proceedings of the 19th BCS-IRSG Colloquium on IR Research*, (pp. 8-9). Aberdeen, Scotland, UK: Springer.

Ozer, B., Wolf, W., & Akansu, A.N. (2000). Relational graph matching for human detection and posture recognition. *SPIE Proceedings on International Conference on Internet Multimedia Management Systems*, Boston.

Zamir, O., Etzioni, O., Madani O., & Karp, R.M. (1997). Fast and intuitive clustering of web documents. *Knowledge Discovery and Data Mining,* (pp. 287-290).

Zizi, M., & Beaudouin, M. (1994). Accessing hyperdocuments through interactive dynamic maps. *Proceedings of the 1994 ACM European conference on Hypermedia technology,* (pp. 126-135). ISBN: 0-89791-640-9.

Acknowledgement

The authors would like to thank Dr. Edward Altman (I2R, Singapore) for introducing us to the problems of e-learning and his insightful discussions.

Note

[1]SMA 5503 details can be viewed at http://www.comp.nus.edu.sg/~sma5503/

Automatically Generating Effective Online Help

CECILE PARIS, NATHALIE COLINEAU, AND SHIJIAN LU
CSIRO[1], Australia
cecile.paris@csiro.au
nathalie.colineau@csiro.au
shijian.lu@csiro.au

KEITH VANDER LINDEN
Calvin College, USA
kvlinden@calvin.edu

Instructional text, and procedural text in particular, is a genre that users heavily rely upon when they are learning new procedures, devices or systems. It is, however, also well-known to be a genre that is difficult to produce and maintain. This article discusses Isolde[2], an environment that attempts to address this problem by supporting the semi-automated production of procedural instructions, online help in particular. This environment exploits technologies for producing documents automatically and includes an extensible set of interactive tools for acquiring, representing, and maintaining the knowledge that is required for producing the instructions. In addition, this article presents the results of an evaluation that compares the effectiveness of the instructions produced by Isolde with those produced by human authors. The results suggest that the instructions produced by Isolde are of comparable quality to similar texts found in commercial manuals.

Instructional text, and procedural text in particular, are heavily relied upon when learning to use new procedures, devices or systems. This genre is widely accepted as a crucial component of any software system or device. Figure 1 to Figure 3 show examples of instructional texts for a procedure (Figure 1), for a device (Figure 2) and for a software system (Figure 3). This sort of text is designed to help users learn to perform their tasks by answering the question, “How to?” For software systems, such text is frequently integrated directly into the running system and presented upon request as an online help document in hypertext form.

To fix the remote controller holder on the wall

1 Choose a place from where signals reach the unit .
2 Fix the holder to a wall, a pillar, etc. with the screws supplied with the holder.
3 Place the remote control in the remote control holder

Figure 1. Instructions from the operating manual of the DAIKIN room air conditioner

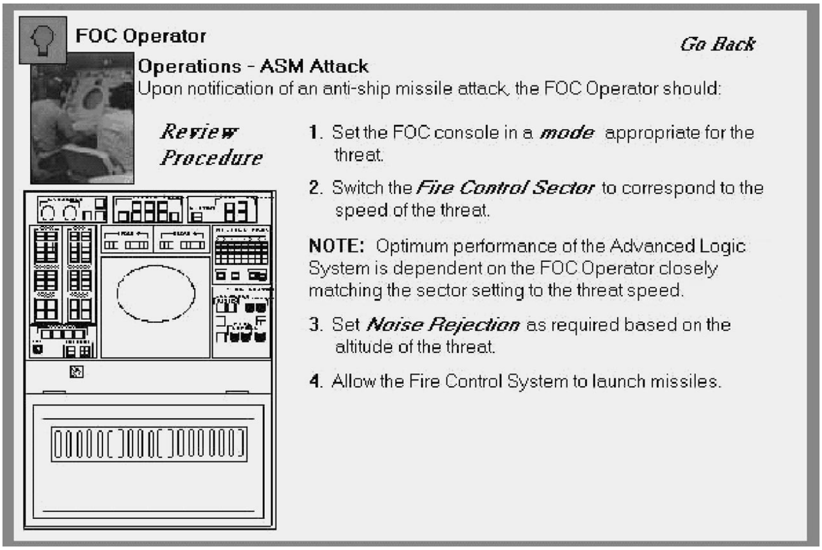

Figure 2. Instructions from a sample electronic TACMEMO, by search technology & NAWCTSD

In addition to being an important genre, however, procedural text is also widely seen as being labor-intensive and tedious to produce and maintain. This fact has led to interest in automating its production. Fortunately, automated generation systems can take advantage of two characteristics of procedural text. First, its structure is heavily based upon the behavior of the system or device, and particularly on the hierarchical structure of the users' tasks and subtasks. Second, it tends to employ simple, relatively short sentences and a small, constrained vocabulary in a consistent manner. These two characteristics, which are exemplified by the text in Figure 1 to Figure 3, tend to simplify the job of an automated generation system. They also lend themselves to the trend towards more effective, "minimalist" instructions (Carroll, 1990). While much work has been done on the automated

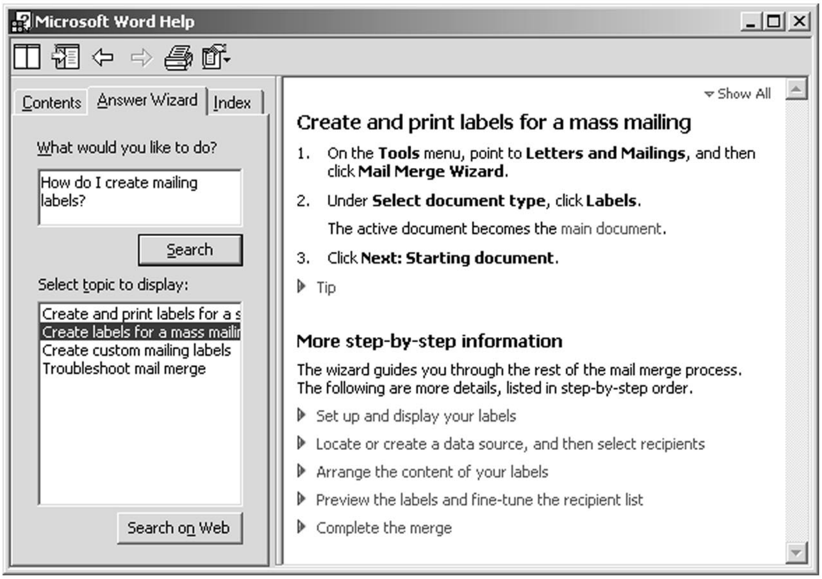

Figure 3. A sample on-line help text from Microsoft Word

generation of procedural text, there has been relatively little attention paid to the problems of acquiring the knowledgebase necessary for automated generation and of evaluating the effectiveness of the resulting texts.

This article addresses the promise and problems of the automated generation of procedural texts by discussing Isolde, an environment that supports the semi-automated production of procedural instructions as electronic documents. It starts with a review of related work, and then presents the Isolde environment, which includes an extensible set of interactive tools for acquiring, representing, and maintaining procedural knowledge, as well as a language generation tool that produces instructions based upon this knowledge. The article then discusses the results of an evaluation that compares the effectiveness of the instructions produced by Isolde with those produced by human authors. The results of this evaluation suggest that the instructions produced by Isolde are of comparable quality in terms of effectiveness to similar texts found in commercial manuals.

RELATED WORK

There is much work in the natural language generation (NLG) community on producing instructions or documentation (e.g., Mellish & Evans, 1989; McKeown, Elhadad, Fukumoto, Lim, Lombardi, Rogin, & Smadja, 1990; Hartley, Scott, Kruijff-Korbayová, Sharoff, Sokolova, Dochev, Staykova, Cˇmejrek, Hana, & Teich, 2000; Kosseim & Lapalme, 2000). These systems tend to take some form of deep process representation as input, and produce instructional texts. Primarily, this input is hand-built and contains detailed representations of the knowledge required for the generation process. This tends to be a costly process, and as a result, the application of the technology for documentation or education remains limited. As an alternative, Isolde and some other systems (Paris & Vander Linden, 1996; Power, Scott, & Evans, 1998) have developed interfaces that allow trained users to specify the input knowledge and to drive the generation process. This helps avoid the difficult task of total automation by allowing technical writers to drive the generation process, thus taking advantage of their writing skills. In addition, Isolde attempts to reduce the cost of building the deep semantic models required for generation by providing tools for extracting re-usable knowledge already contained in models built for other purposes and by using representations that can more easily be re-used by other applications thus spreading their creation cost.

Another aspect of our work is the evaluation of the automatically generated procedural text. To date, only a few of the procedural text generation projects have been formally evaluated, and in those cases where a formal evaluation has been conducted, the systems were judged on the basis of the fluency and grammaticality of the output text rather than on its effectiveness, (e.g., Hartley et al., 2000). This tends to be the case, in fact, for evaluations of NLG systems in general. People are asked to rate the acceptability of the generated texts or to compare them to human-authored texts (e.g., Lester & Porter, 1997), without measuring the actual impact of the texts on their intended users. The STOP system (Reiter, Robertson, Lennox, & Osman, 2001) is a notable exception to this trend. In that project, the researchers performed a large-scale study of how effective texts tailored to individual smokers were at convincing them to stop smoking. They compared how often readers of STOP's individually tailored texts actually stopped smoking as compared with how often readers of generic, automatically generated texts stopped smoking. In our work, we sought to perform a similar evaluation of the effectiveness of the instructional texts produced by Isolde in the context of a task and in comparison to manually-authored texts.

Another related area is adaptive web-based educational systems (cf. Brusilovsky & Peylo, 2003). These systems, however, are concerned not so much with procedural instructional text as we are, but rather with educational courseware. In general, their goal is to provide an environment which adapts itself to the need of the learners through interactions with them. Their focus is on the intelligent adaptation by knowing about the learner's goals, preferences and existing knowledge. In the work presented in this article, we were not concerned with adapting the output to the user but rather with producing effective instructions to help the user perform a task.

Finally, there has been other work in exploiting other technologies for electronic documents in educational settings (cf. Alem & McLean, 2003), in particular using techniques such as information retrieval and document reuse to develop ways to find existing material and adapt it to a learning context (Vercoustre & McLean, 2003). In this area, the most relevant work is that of Davis, Kay, Kummerfeld, Poon, Quigley, Saunders, & Yacef (2003). They exploit a workflow, which is similar to a task model, but their use of it and their concerns are significantly different from those in Isolde. The workflow is used to proactively present to the user, at every step, the documents and information that could help the user at that step. Instead, our concern in this work is to provide effective instructions on how to achieve a task (i.e., how to go through the workflow).

THE ISOLDE ENVIRONMENT

As mentioned above, previous work has shown that it is possible to automate the production of instructions using a knowledge-based approach and a specific technology for automatically producing documents: natural language generation (NLG). However, most of the prototype systems developed so far have relied on an input (i.e., the specification of a procedure, its accompanying actions and objects and their lexical information) that was specifically tailored to the NLG system and was typically manually constructed as deep semantic models – a costly process. In this work we wanted to determine whether it was possible to produce effective instructions without the use of deep semantic models and an input specifically designed for the generation process. In this section, we describe Isolde, the development environment that resulted from these efforts.

Isolde's architecture is shown in Figure 4. It is centered around Tamot, a task and domain model editor that uses an underlying XML-based language to represent: (1) task models written in the Diane+ modeling language (Tarby &

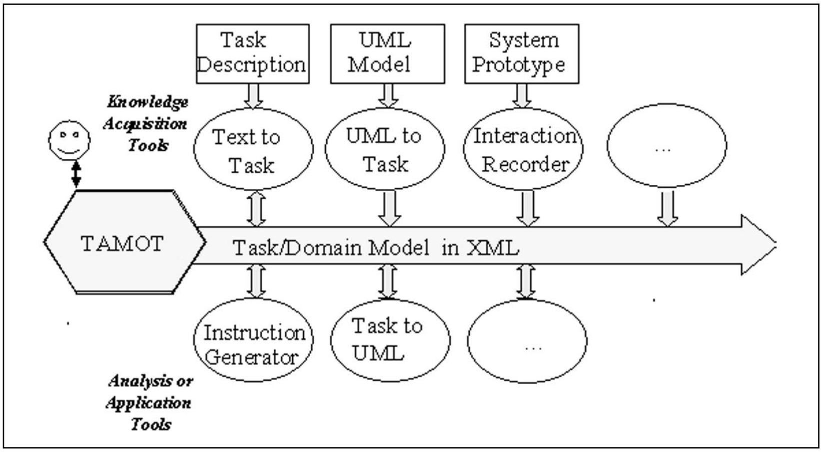

Figure 4. The Isolde environment

Barthet, 1996) and (2) their accompanying domain models[3] written in a simple KL-ONE styled knowledge base (Brachman & Schmolze, 1985). The architecture also provides sets of tools that allow users (e.g., technical writers, educators, HCI specialists, analysts) to extract task knowledge from a variety of sources (shown on top of the figure as "Knowledge Acquisition Tools") and to use the task and domain knowledge in analysis or applications (shown on the bottom of the figure as "Analysis or Application Tools"). Isolde is built as an environment to allow the addition of other tools as deemed necessary. The only requirement on such tools is that extraction tools must produce task and domain models using Isolde's open, standardised XML-based representation, and analysis/application tools must take such a representation as input.

We chose *task models* as the input to the instruction generator. Task models describe an application's functionalities from an end-user's perspective, and thus seemed an appropriate input to the instruction generator as a representation of the procedure to be documented. Task models are artifacts built by task analysts, and in Human Computer Interaction (HCI), they are advocated for their many uses throughout the software development life cycle (SDLC) – design, evaluation, etc. (Diaper & Stanton, 2004). This presents two advantages. First, HCI specialists may be producing task models as part of their own work in the design of a system, so that the input to the instruction generation process may be partially available and does not necessarily need to be built from scratch. Second, if the technical authors build a task model to produce instructions, it might be reused at other stages of the SDLC, (e.g., to help evaluate the resulting system later on). By choosing task models for the input to our generator, we therefore allow for the cost of its construction to be spread over various uses.

Having chosen task models as the input to the instruction generation process, we designed Isolde as an extensible environment that provides a heterogeneous set of specialised tools for building, maintaining and using task models, the instruction generator being one tool capable of using the models. This article now presents the various modules of this environment, concentrating on the tools that facilitate the generation of the instructional text. It starts with the language generation facility (the "Instruction Generator"), and then proceeds to the task and domain model editor (Tamot) and the extraction tools ("Text to Task," "UML to Task," and "Interaction Recorder"). One additional tool (the "Task to UML" tool), which produces UML models from task models, will not be addressed here (cf. Lu, Paris, Vander Linden, & Colineau, 2003). The tools are presented in the context of a running example, which shows the use of Microsoft Word to create mailing labels. The instructions for this task, as written by Microsoft technical writers, are shown in Figure 3.

The Instruction Generator

The instruction generator is the NLG tool that produces the procedural text. Its output for the mailing labels example is shown in Figure 5. To generate this text, the analyst sent an appropriate task and domain model (see next section) from Tamot to the Instruction Generator. Isolde's Instruction Generator is implemented as a LISP server. It includes the Moore & Paris (1993) text planner, a sentence planner implemented as extensions to the text planner, and the KPML system and its lexicogrammatical resources (Bateman, 1997). It plans the instructions using discourse and sentence plans and also decides on the appropriate hypertext links.

The generator has several attributes. Its library of discourse plans can handle a variety of task model configurations, including sequences, compositions and Boolean connectors. It does not require a deep semantic model, which alleviates the need to represent a highly detailed model that includes all the required lexicogrammatical information. In addition, it is able to integrate canned text with generated text. As an example, the "Note" at the bottom of the right-hand-side screen in Figure 5 is a sentence that was encoded as canned text in the task model. Yet it is smoothly integrated with the remainder of the text on that screen, which was automatically generated. This characteristic of the generator is a recognition that it would be hard to formally encode all the knowledge that is to be included in the instructions as a semantic model. Finally, some style parameters have been implemented, giving the writers control over various discourse or sentence level features of the output. For example, the grammatical form of the titles can be switched between the infinitive (e.g., "To Create Mailing Labels") and the "ing" form (e.g., "Creating Mailing Labels"). As another example, writers can choose to produce a concise text by suppressing levels of decomposi-

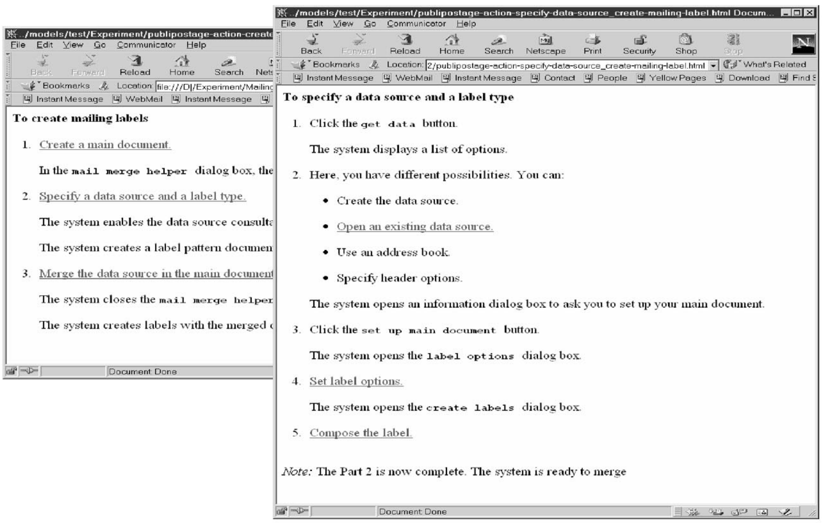

Figure 5. Text generated by Isolde: Creating mailing labels with MS Word

tion, or alternatively, they can choose to produce step-by-step instructions, in which each level of task decomposition is presented in a separate hypertext page. We now turn to the tools that allow for the construction of acquisition of the task model and its accompanying domain model.

Tamot: A graphical editor for task and domain models

Tamot is a graphical editor for task models (Lu, Paris, & Vander Linden, 2002). Although its primary role in Isolde is to allow users to consolidate, modify and extend the task models obtained through the knowledge acquisition tools, it also enables them to enter the knowledge from scratch. Figure 6 shows Tamot being used to edit the task information that lies behind the procedural text for the mailing labels example (see Figure 5). On the left, Tamot displays a hierarchical view of the tasks. On the right, it displays a set of windows showing a graphical representation of the tasks at various points in the hierarchy.

The Diane+ notation used by Tamot allows for the representation of task structure, including its decomposition, sequencing, and alternatives (through the OR/XOR Boolean connectors).[4] We conducted usability studies, which showed that technical writers found the notation to be largely readable and that they required little training to be able to produce a task model correctly (Ozkan, Paris, & Balbo, 1998).

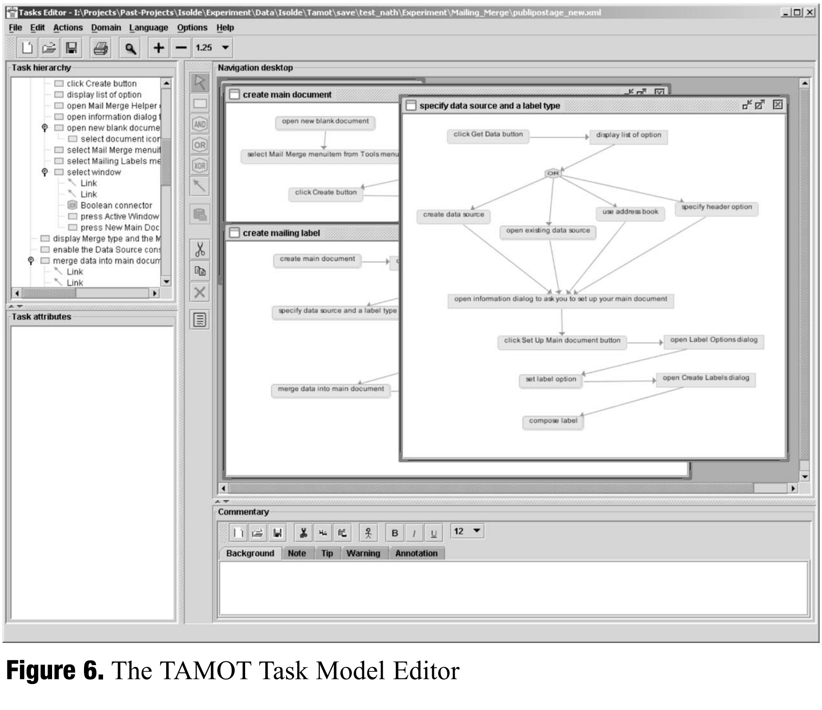

Figure 6. The TAMOT Task Model Editor

As mentioned, one advantage of using task models as a central representation language is that HCI specialists may already be producing task models as part of their own work in the design of a system, and thus these models do not need to be built only for generation purposes. This way, the task model can be produced once and then re-used in several stages of the SDLC, (e.g., in system design, instruction generation, system evaluation).

In addition to task modeling, Tamot supports the representation and manipulation of the domain knowledge associated with the tasks. While the task model indicates the structure of the text explaining the task, the domain knowledge specifies the content of each of the sentences. For example, in the mailing labels instructions, this content includes the actions being performed (e.g., "create," "specify," "merge," "click"), the actor performing the action (i.e., the user or the system), and the object being acted upon (e.g., "mailing labels," "main document," "data sources"). In addition, this content specifies how to conjugate a verb, whether a specific object requires a determiner or not,

whether it will be expressed through a plural noun, and the exact lexical items to be employed, all of which is required for automated language generation.

The domain model editor, shown in Figure 7, allows the user to build or modify the domain knowledge associated with each task. The figure shows a

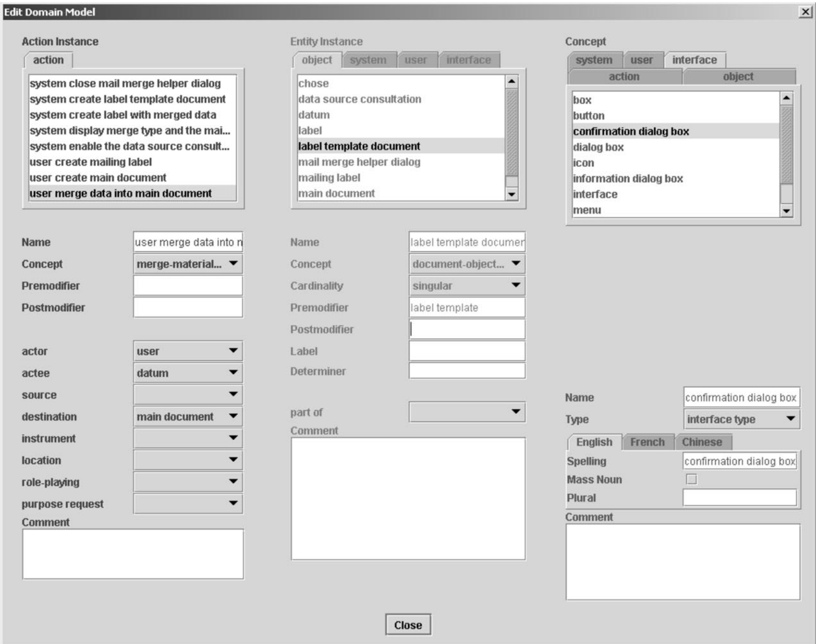

Figure 7. The Tamot Domain Model Editor

single editing window with three columns for editing action instances, entity instances, and concepts respectively. On the top left, the editor shows a list of all the action instances in the domain model. The "user merge data into main document" action is selected, and the details shown below the list box include its action concept ("merge-material-type"), its actor ("user"), and its object ("datum"). The editor is now focusing on entities and concepts referred to by this action, ignoring all the others in the model. It displays the "label template document" in the middle column and the "confirmation dialog box" interface concept in the right column. The template document is of the type "document-object type" and has a pre-modifying string of "label template."

In our system, instead of an *a priori* coding of all this information, it is acquired on the fly through both the parsing tool (briefly described below) and the domain model editor. The model required by the language generator is quite shallow, and based on a linguistic analysis of instructional text, we have embodied some defaults in the system to make it easier to construct the model. For example, the system includes defaults choices for determiners, but a technical writer can use the editor to set a specific determiner for a specific instance, thereby overriding the generator's default choice.

The Extraction Tools

Even with an interactive tool like Tamot, it can still be difficult to build task and domain models from scratch. Therefore, Isolde allows for the inclusion of tools to mine sources that may already exist in an attempt to construct a draft model automatically. Isolde currently includes three knowledge acquisition tools that extract task, domain and lexical knowledge from other sources: (1) the Text to Task extraction tool, T2T, which extracts knowledge from written text, (2) the UML to Task extraction tool, U2T, which obtains task information from Software Engineering models represented in the UML notation; and (3) the User Interaction Recorder, or UIR, which exploits a system's prototype and builds a task model by recording a user's actions while performing a task. This section gives a brief discussion of each knowledge acquisition tool. More details can be found in (Paris, Vander Linden, & Lu, 2004).

The Text to Task extraction tool: T2T

One commonly available source of task and domain knowledge is written texts, such as written task descriptions. The HCI community has pro-

To create a mailing label, create a main document. Then specify a data source. Finally, merge the data source and the main document.

Figure 8. A Possible text describing a procedure

duced tools that allow users to *manually* acquire knowledge from descriptions, (e.g., Tam, Maulsby, & Puerta, 1998). T2T, our Text to Task extraction tool, uses information extraction and parsing to *automatically* acquire task, domain and lexical knowledge from such sources (Brasser & Vander Linden, 2002).

The core of T2T is a finite-state grammar built as an ATN. It identifies the basic domain and lexical knowledge, for example, actor (subject), process (action), actee (object), instruments ("with"-clauses) and locations (indicated by such prepositions as "in," "on," etc.). T2T distinguishes between nouns and verbs based on classifications of WordNet (Fellbaum, 1998). It can work automatically, or in those cases where T2T's grammar is incomplete, allow the user to modify the constituent boundaries using an interactive parsing tool.

When there are expressions of multiple tasks, T2T extracts the procedural relationships between them. It then creates the appropriate task concepts and instances. T2T can thus obtain a draft task model from a text such as that shown in Figure 8. From this text, the parser will be able to extract a draft for the high level task for creating mailing labels (i.e., a high-level "create label" task, and sub-tasks "create document," "specify," and "merge").

The parser employed in T2T is also exploited within Tamot to extract from the names of the tasks and their steps the corresponding domain model elements (actions, subjects and objects). These can then be corrected if required through the domain model editor presented earlier. This allows the generation system to acquire some of its knowledge (e.g., lexicon) on the fly, thus further easing the construction of the resources required for generation.

The UML to Task extraction tool: U2T

Design models represented in UML are common in Software Engineering (Rumbaugh, Jacobson, & Booch, 1999) and can serve as another source of knowledge to extract task models. The UML to task tool, U2T, performs this extraction using the standard Rational Rose scripting language. It has been discussed in some detail in (Lu, Paris, & Vander Linden, 1999). U2T obtains knowledge from use-case diagrams, which identify the application users and their goals, class diagrams, which describe the basic classes of domain objects and the actions performed on them, and scenario diagrams, which specify the sequences of user and system events required to achieve selected use-cases. U2T employs heuristics to filter out system-oriented information and retain only user-oriented knowledge. As with the previous acquisition tool, the draft model can then be edited with Tamot.

The User Interaction Recorder: UIR

System prototypes, if they exist, implement the interaction an end-user is to have with an application. There are tools to extract the GUI objects and record GUI event sequences from the prototype as it runs (Hilbert & Redmiles, 2000). A prototype can thus serve as a knowledge source for the task model. UIR, our User Interaction Recorder, uses an object extraction tool and an event recorder, developed at Sun Microsystems for the Java platform, to extract task and domain knowledge. Using this tool, the user specifies a high-level goal ready for decomposition, and then starts the "recording", performing all the actions necessary to achieve the goal. UIR records all these low-level actions, and all the GIU objects involved. This process mimics what technical writers often do in their work – go through a software system step by step, documenting their moves, before writing the instructions (cf. Pemberton, 1996; Paris & Vander Linden, 1996; Paris, Ozkan, & Bonicafio, 1998). With UIR, the recording of the moves is thus automated and results in a model that can then be augmented to form the appropriate task model. Instructions can then be generated automatically from the model.

The Java tools on which UIR is based collect a wide variety of information, some of which is too low-level to be useful for our modeling (e.g., separate key-press events for every letter typed into a text field). UIR thus includes heuristics to help it translate the low-level events it extracts to the higher-level, more domain-oriented actions and objects required by the task and domain models.

EVALUATION

In order to evaluate more than just the grammaticality and style of Isolde's help text, we performed an experiment aimed at assessing how effective the text is at helping users learn their tasks. To do this, we performed the following two steps with Isolde-generated text (e.g., Figure 5) and manually authored help (e.g., Figure 3):

1. Measured the user's performance in accomplishing a specific task (i.e., task achievement, time needed and number of errors).
2. Asked the users to rate the usefulness of the texts, the adequacy of the content and the coherence of the organization.

We did not evaluate grammatical correctness as in AGILE (Hartley et al., 2000). Given the functional aims of instructional text, we wanted to ensure that the generated instructions enabled the users to learn about and achieve their task, regardless of the instructions' quality or complexity.

Experimental Design

Our methodology involved four steps: (1) choosing three tasks in Word, (2) designing the three corresponding task models, (3) producing the online help with Isolde for these models, and (4) evaluating the help on two user groups: one group received the Word help, the other received automatically generated help. Participants in the experiment were not aware of which help they were using.

Task Selection

To compare the effectiveness of Isolde help with that of manually authored help, we asked users to perform a real task using the online help as a resource. We decided to work with Word, as it provides both a task environment and online instructions. Thus, our aim was to select three tasks – two simple and one complex – that would be new for the subjects, to help prevent introducing a bias based on prior knowledge and to encourage users to read the help. Here, the differences between simple and complex tasks are in the number of elementary actions required to perform the task (18 vs. 40, in our experiment) and in the depth of the task decomposition. We also chose tasks such that their Word help text was self-contained (i.e., without extensive reference to other parts of the help), and the text generated by Isolde for the task would be of similar reading complexity. Our final constraint was that the two simple tasks had the same number of elementary actions. We had no prior assumption as to what task would be easier to model or document than any other task. With these constraints we chose:

- Task 1: Create a document template and save it in a specific directory.
- Task 2: Create index entries.
- Task 3: Create mailing labels by merging a label template with an address list, and save the label pattern.

Task Design

To generate the online help for these tasks, we first designed the task models using Tamot. As typically done by technical writers, we executed the tasks step by step, recording all the steps. Feedback expressions (e.g., display of windows, confirmation messages) were also included, either as system actions modeled within the Diane+ notation in the task model, or as notes or warnings with canned text.

Hypertext Generation

When the task model was completed, we generated the corresponding online help. We then used the Flesch (1974) score[3] to compare the readability of the generated texts with the Word help (see Table 1). This was to ensure both texts would be of similar reading complexity and would involve the same amount of time to consult. For the experiment, only one of the help texts was accessible to the user, displayed in a Netscape browser to preserve anonymity.

Formation of Subject Groups

A total of 35 subjects participated in the experiment (three tasks per subject), split into two groups. Subjects were randomly assigned to a group, but we ensured an even number of men and women in each group. The subjects were not experts in Word, but they knew how to use the application.

Table 1
Comparison of Word and Isolde readability score

	Task 1		Task 2		Task 3	
Readability scores	Word Help	Isolde Help	Word Help	Isolde Help	Word Help	Isolde Help
Average Sentence Length (ASL)	16.241	7.821	11.882	9.791	15.548	7.609
Flesch Reading Ease Score	62.821	70.644	68.712	75.671	77.668	71.951

Scenario of Experiment

The experiment consisted of asking subjects to perform the three tasks in Word described above. For each task, they were given some directions as to what was expected of them (e.g., create a template with the CSIRO logo and save it in a specific directory). The directions did not include explanations on how to achieve the task. These explanations were to be found in the online help provided. Subjects could consult the help at any time (i.e., before or while performing the task). They were told to read the directions and ask for clarification if required before starting on the task. After each task, they filled out a questionnaire asking them to rate the help they used.

The questionnaire aimed at evaluating the usefulness of the help, the quality of its content (i.e., its quantity and relevance) and the coherence of its organization. The questions and corresponding factors of acceptability that they rate are shown in Table 2. Each question was answered using a six-point scale, assigning letter grades A (high) through F (low). These letters were later converted into digits from 6 to 1 for the statistical analysis. The

Table 2
Grading factors presented to subjects

Factors	Questions
Usefulness	How would you evaluate the usefulness of the help?
Adequacy of Content	Did the help provide you with enough information to perform the task?
	Was the information provided in the help relevant for your task?
Coherence	Did the help give you a clear picture of the steps required to accomplish the task?
	How well was the help organised?

questionnaire also included questions that checked the users' previous level of familiarity with the task.

During the experiment, we recorded the time to measure the users' performance. We limited the allowable time (10 minutes for the simple tasks and 15 minutes for the complex task) to encourage the users to consult the help instead of exploring the application by themselves[5]. We observed whether subjects consulted the help or not, and the number of times they did so. Finally, to evaluate the success rate on each task, we recorded the errors made. The marking scale was set as follows:

- For Task 1, the subject lost one point if the document was saved in the wrong directory and two points if it was not saved in the template format.
- For Task 2, the subject lost one point for each index entry that they marked incorrectly.
- For Task 3, the subject lost one point if the mailing labels were not created, and another point if the label pattern was not saved.

Group 1 was assigned Word help for Tasks 1 and 3, and Isolde help for Task 2, while Group 2 was assigned Isolde help for Task 1 and 3, and Word for Task 2.

RESULTS

Because we wanted to assess the effectiveness of the help in aiding a user to accomplish a task, we first screened out users who knew the tasks before hand (based on the questionnaire)[6], and those who did not consult the help at all (based on our observations). As a result, we were left with 12 subjects out of 35 for Task 1, 34 for Task 2, and 29 for Task 3. We analyzed the data for task performance in terms of the time it took to finish the task and the number of errors made. In all cases we ran an Anova single factor, and when results are significant, we report them for a 0.05 level of confidence.

Results on Task Performance

With respect to errors, there was no evidence that either help was more effective than the other. The small differences observed were not statistically significant. This is shown in Figure 9.

With respect to time, we observed interesting differences that were contrary to our expectations. Table 3 presents the results obtained by running an Anova. The times are reported in seconds. The first column combines Tasks 1 and 2 (the two simple tasks), as Task 1 alone did not allow separate computation due to the small number of subjects. The difference in time performance was in favor of Word for the simple tasks (indicated in italics in the table). It was, however, in favor of Isolde for the complex task (indicated in bold). We hypothesized that this is due to the fact that the text generated by

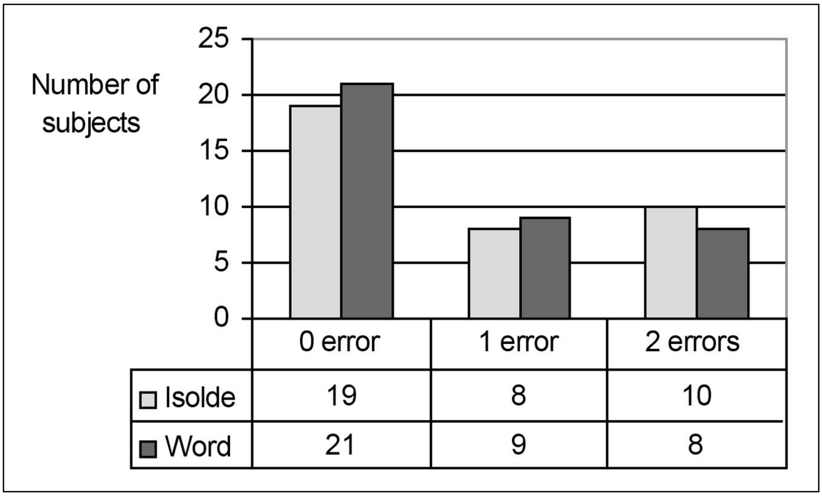

Figure 9. Task performance comparison - all tasks

Table 3
Time Performance (in seconds)

	Tasks 1 & 2	Task 2	Task 3
Isolde Help	388.86	428.58	398.83
Word Help	278.91	296.70	553.63
Difference	*109.94*	*131.88*	**154.80**
Anova (F-test)	5.77	6.33	7.82
Level of Confidence	0.02	0.01	0.01

Isolde is highly structured, strictly following the task decomposition. This is not always the case for the human authored texts, which include more aggregations. While this may result in shorter and more "elegant" text (and thus faster to read) for simple tasks, it also sometimes results in more confusing text for complex tasks.

Results on the acceptability of the Help

Figure 11 to Figure 12 show the ratings of the different help texts for the different tasks based on the responses on the questionnaire. We computed means for both Isolde and Word using the six-point scale.

For each task and for each of the help dimensions we would like to

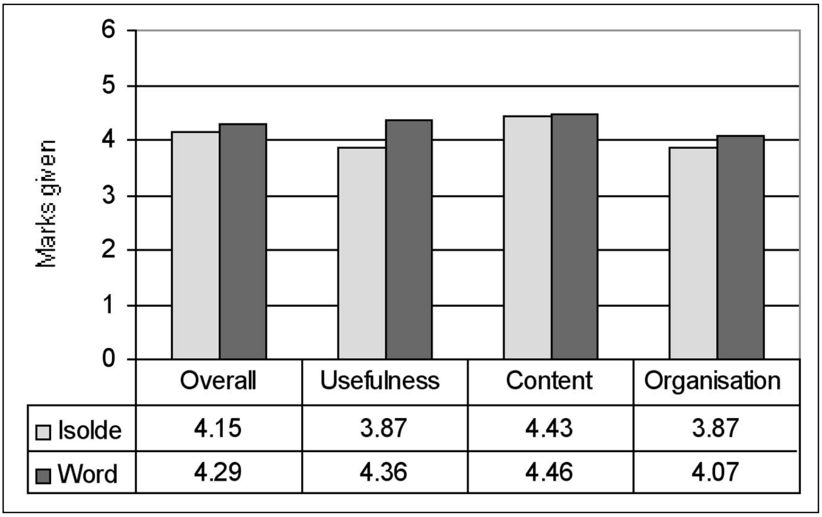

Figure 10. . Rate for the simple tasks (Tasks 1 and 2)

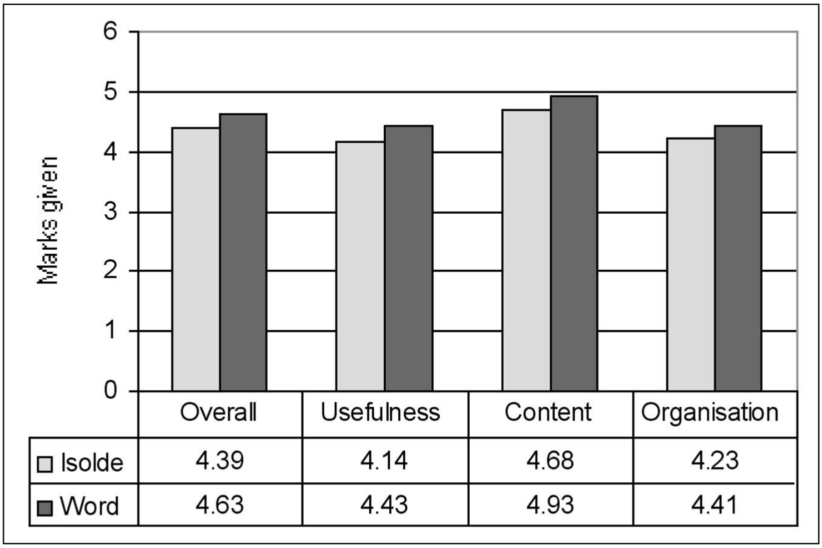

Figure 11. Rate for the complex task (Task 3)

observe: (1) the usefulness regarding the task, (2) the quality of content provided, and (3) the coherence and clarity of the help organization.

The Overall column summarizes these different values. As shown in the figures, Isolde scored closely to Word, within approximately 1/4 of a point for the content and the organization, and 1/3 of a point for the usefulness. In all cases, both Isolde and Word were positively evaluated on average. We checked whether the differences were significant or not by running an Anova on each task for each dimension. The results did not show any significant differences between the two help texts. Table 4 reports the results obtained when the scores are aggregated over all the tasks, though we also performed the test for each separate task. Our results indicate that in terms of acceptability and usefulness of the help, Isolde's performance approaches that of the manually authored texts.

DISCUSSION

Our work aims at providing help texts that enable users to quickly learn to perform tasks they need to perform, even if they have never done them before. Our challenge then was to provide enough relevant information

Table 4
Anova result combining all the tasks

	Overall	Usefulness	Content	Organisation
Anova (F-test)	2.18	1.61	0.67	0.00
Level of Confidence	0.14	0.20	0.41	0.94

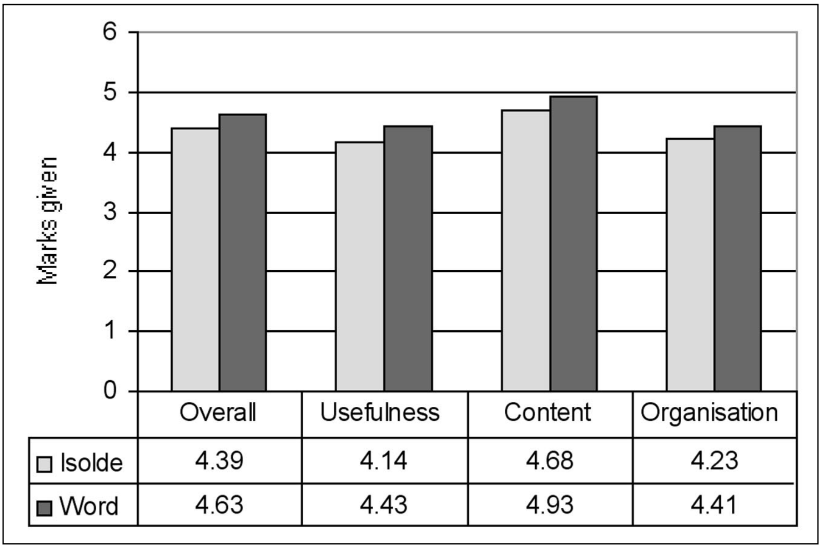

Figure 12. Rate combining all the tasks

without extraneous details within the constraints imposed by the knowledge available to generate text automatically. As shown in Table 1, the help generated by Isolde and manually-authored texts are similar in terms of their readability score. Isolde, however, generates shorter sentences with strictly the information required to achieve a task. While our generated texts often contain less information than manually-authored texts (because of the knowledge available to produce them), they constitute less text to browse (i.e., each instruction is shorter) and may thus make it easier to access important information. From the experiment it seems that providing this type of text has a significant impact on complex tasks (where the amount of consultation and the time spent understanding the tasks is greater), but no impact on simple tasks (where users seem more comfortable reading the manually-authored texts). We note that we did not evaluate our text with respect to memorability.

CONCLUSION

This article has discussed Isolde, an environment in which a variety of knowledge sources can be integrated into a knowledge base that is capable of supporting the generation of instructional text in varying domains. We showed that an instruction generator could generate appropriate instructions without a "deep" model, and that the required knowledge could, in principle, be acquired from sources that exist in practice. This article has also presented an evaluation of the procedural instructions generated by the Isolde authoring tool. The evaluation analysed the comparative effectiveness of the instructional texts generated by Isolde and those written by technical writers in the context of a real task. The results showed: (1) no significant differences with respect to the number of errors made while performing the tasks, (2) some significant differences in time performance, in favor of Isolde for complex task and of the manually-authored texts for simple tasks, and (3) no significant differences with respect to the acceptability of the help. These results are very encouraging, as they show that our automatically generated text approaches manually-authored text in its effectiveness. While the Isolde project itself has ended, the framework is now used in a new project aiming at integrating instructions and feedback into a virtual environment for surgical training (Müller-Tomfelde, Paris, & Stevenson, 2004).

References

Alem, L., & McLean, A. (Eds) (2003). *Proceedings of the AI-Ed2003 Workshop on Technologies for Electronic Documents for Supporting Learning.* In *aied2003 Supplementary Proceedings*, Alevan, Hoppe, Kay, Mizoguchi, Pain, Verdejo, & Yacef (Eds), 615-710.

Bateman, J. (1997). Enabling technology for multilingual natural language generation: the KPML development environment. *Natural language Engineering, 3*(1), 15-55.

Brachman, R. J., & Schmolze, J. (1985). An overview of the KL-ONE knowledge representation system, *Cognitive Science,* 9, 171-216.

Brasser, M., & Vander Linden K. (2002). Automatically elicitation task models from written task descriptions. In *Proceedings of CADUI'02*, France.

Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent web-based educational systems. In *International Journal of Artificial Intelligence in Education,* 13, 159 – 172.

Carroll, J. (1990). *The Nurnberg Funnel: Designing minimalist instruction for practical computer skills.* MIT Press, Cambridge, MA.

Davis, J., Kay, J., Kummerfeld, B., Poon, J., Quigley, A., Saunders, G., & Yacef, K. (2003). Using workflow, user modeling and tutoring strategies for just-in-time document delivery. In *aied2003 Supplementary Proceedings*, Alevan, Hoppe, Kay, Mizoguchi, Pain, Verdejo & Yacef (Eds), 664 – 673.

Diaper, D., & Stanton, N. (Eds) (2004). *The Handbook of Task Analysis for Human-Computer Interaction.* Lawrence Erlbaum Associates.

Fellbaum, C., (Ed) (1998). *WordNet: An electronic lexical database*, MIT Press.

Hartley, T., Scott, D., Kruijff-Korbayová, I., Sharoff, S., Sokolova, L., Dochev, D., Staykova, K., Čmejrek, M., Hana, J., & Teich, E. (2000). Evaluation of the final prototype, deliverable of the AGILE project. October, 2000.

Hilbert, D., & Redmiles, D. (2000). Extracting usability information from user interface events. *Computing Surveys, 32*(4), 384-421.

Kosseim, L., & Lapalme, G. (2000). Choosing rhetorical structures to plan instructional texts. *Computational Intelligence, 16*(3), 408-445.

Lester, J., & Porter, B. (1997). Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics, 23*(1), 65-101.

Lu, S., Paris, C., & Vander Linden, K. (1999). Towards the automatic generation of task models from object oriented diagrams. *Engineering for Human-Computer Interaction,* Chatty & Dewan (Eds), Kluwer, 169-189.

Lu, S., Paris, C., & Vander Linden, K. (2002). Tamot: Towards a flexible task modeling tool In *Proceedings of Human Factors 2002*, Melbourne, Australia.

Lu, S., Paris C., Vander Linden K., & Colineau N. (2003). Generating UML diagrams from task models. In *Proceedings of CHINZ'03: The 4th Annual Int'l Conf. of the ACM's SIGCHI, New Zealand Chapter,* Dunedin, New Zealand.

McKeown, K., Elhadad, M., Fukumoto, Y., Lim, J., Lombardi, C., Rogin, J., & Smadja, F. (1990). Natural language generation in COMET. *Current Research in Natural Language Generation,* Dale, Mellish, & Zock (Eds). Academic Press, 103-140.

Mellish, C., & Evans, R., (1989). Natural language generation from plans. *Computational Linguistics, 15*(4), 233-249.

Moore, J., & Paris, C. (1993). Planning text for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics, 19(*4), 651-694.

Müller-Tomfelde, C., Paris, C., & Stevenson, D. (2004). Interactive landmarks: Linking virtual environments with knowledge-based systems. Submitted for publication.

Ozkan, N., Paris, C., & Balbo, S. (1998). Understanding a task model: An experiment. In *People and Computers XII, Proceedings of HCI'98*, Johnson, Nigay, & Roast (Eds), Springer, 23-138.

Paris, C., & Vander Linden, K. (1996). DRAFTER: An interactive support tool for writing multilingual manuals. *IEEE Computer, 29*(7), 49-56.

Paris, C., Ozkan, N., & Bonicafio, F. (1998). The design of new technology for writing on-line help. In *People and Computers XII, Proceedings of HCI'98*, Johnson, Nigay, & Roast (Eds), Springer.

Paris, C., Vander Linden, K., & Lu, S. (2004). Environments for the construction and use of task models. *The Handbook of Task Analysis for Human-Computer Interaction.* Diaper, & Stanton (Eds). Lawrence Erlbaum Associates, 467-481.

Pemberton, L. (1996). Requirements from technical writing. *Technical report, ITRI,* November, 1996.

Power, R., Scott, D., & Evans, R. (1998). What you see is what you meant: Direct knowledge editing with natural language feedback. In *Proceedings of ECAI'98*, UK.

Reiter, E., Robertson, R., Lennox A. S., & Osman, L. (2001). Using a randomised controlled clinical trial to evaluate an NLG system. In *Proceedings of ACL-2001*, Toulouse, France, 434-441.

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). *The Unified Modeling Language Reference Manual*, Addison-Wesley, Reading, MA.

Tam, R. C., Maulsby, D., & Puerta, A. R. (1998). U-TEL: A tool for eliciting user task models from domain experts. In *Proceedings of IUI'98*, 77-80.

Tarby, J.C., & Barthet, M. F. (1996). The DIANE+ method. In *Proceedings CADUI'96*. Namur, June 5-7, 1996. J. Vanderdonckt (Ed). 95-119.

Vercoustre, A. M., & McLean, A. (2003). Reusing educational material for teaching and learning: Current approaches and directions. In *aied2003 Supplementary Proceedings,* Alevan, Hoppe, Kay, Mizoguchi, Pain, Verdejo, & Yacef (eds). 621-630.

Acknowledgements

The authors wish to thank past and present members of the Isolde team: T. Lo, S. Balbo, M. Brasser, N. Ozkan, and J-C. Tarby, as well as M. Bétrancourt for her advice on our evaluation. We also gratefully acknowledge the support of the ONR (grant N00014-99-0906), CSIRO and Calvin College.

Notes

[1]Commonwealth Scientific and Industrial Research Organisation

[2] Isolde stands for "An Integrated Software and Online Documentation Environment."

[3] A standard document has a Flesch Reading score of approximately 60 to 70. The higher the score, the easier a document is considered to be.

[4]While a task model is a representation of the procedure, there is still a need to model the various actions and objects that are utilised in the procedure. These form the domain model and are required for language generation.

[5] For more details about the Diane+ notation, see (Tarby & Barthet, 1996). Details about which features we use in Isolde can be found in (Lu et al, 2003).

[6]The subjects who knew the tasks were not filtered ahead of time to allow us to observe task performance and help usage differences between "novices" and "experts." No major differences were found (due to the lack of subjects), but it was still interesting to have qualitative input on these issues to inform future experiments.

Smart Recommendation for an Evolving E-Learning System: Architecture and Experiment

TIFFANY YA TANG
Department of Computing, Hong Kong Polytechnic University, Hong Kong
Department of Computer Science, University of Saskatchewan, Canada
cstiffany@comp.polyu.edu.hk

GORDON McCALLA
Department of Computer Science, University of Saskatchewan, Canada
yat751@mail.usask.ca; mccalla@cs.usask.ca

In this article, we proposed an evolving e-learning system which can adapt itself both to the learners and to the open Web, and we pointed out the differences of making recommendations in e-learning and other domains. We propose two pedagogy features in recommendation: learner interest and background knowledge. A description of a paper's value, similarity, and ordering are presented using formal definitions. We also study two pedagogy-oriented recommendation techniques: content-based and hybrid recommendations. We argue that while it is feasible to apply both of these techniques in our domain, a hybrid collaborative filtering technique is more efficient to make "just-in-time" recommendations. In order to assess and compare these two techniques, we carried out an experiment using artificial learners. Experiment results are encouraging, showing that hybrid collaborative filtering, which can lower the computational costs, will not compromise the overall performance of the recommendation system. In addition, as more and more learners participate in the learning process, both learner and paper models can better be enhanced and updated, which is especially desirable for web-based learning systems. We have tested the recommendation mechanisms with real learners, and the results are very encouraging.

Research on e-learning has gained more and more attention thanks to the recent explosive use of the Internet. However, the majority of current web-based learning systems are closed learning environments, where courses and materials are fixed and the only dynamic aspect is the organization of the material that can be adapted to allow a relatively individualized learning environment.

In this article, we propose an evolving web-based learning system which can adapt itself not only to its users, but also to the open Web in response to the usage of its learning materials. Our system is open in the sense that learning items related to the course could be added, adapted, or deleted either manually or automatically. Our proposed e-learning system adapts both to learners and the open Web. Figure 1 compares the traditional web-based adaptive learning system and our proposed open evolving learning system.

In a traditional adaptive e-learning system, the delivery of learning material is personalized according to the learner model. However, the materials inside the system are a priori determined by the system designer/instructor. In an open evolving e-learning system, learning materials are automatically/manually found on the Web, but automatically integrated into the system based on users' interactions with the system. Therefore, although users do not have direct interaction with the open Web, new or different learning materials in the open Web can enrich their learning experiences through personalized recommendations.

The rest of this article is arranged as follows. In the rest of this section, we will briefly discuss the overall system architecture. We will also point out the uniqueness of making recommendations for e-learning systems through a motivational example, followed by discussions on some background information on recommendation system and related work. In the second section, we will present the architecture and components of our system. In the third section, our proposed pedagogically-oriented paper recommendation techniques and concepts will be discussed in details, experiment results will also be shown in this section. We will include lessons learned from implementing our recommendation techniques and human subject studies in the fifth section.

A Brief System Introduction

Our proposed system is designed to support an advanced course for senior undergraduate or graduate students, especially when they are required to read

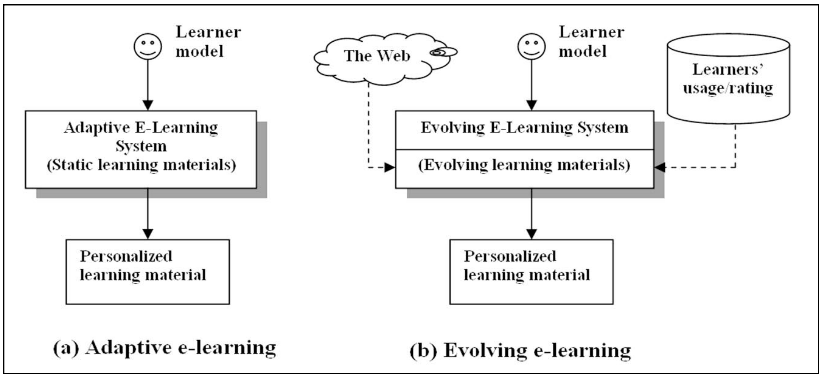

Figure 1. A comparison of evolving e-learning system vs. adaptive e-learning system

technical articles related to their course, such as journal articles, conference papers, book chapter, etc. In the rest of this article, the term "papers" is used to refer to those articles. However, this system can be generalized to a broader area, for example, for corporate learners or with richer learning materials, such as slide presentation, technical report, textbook, programming code, etc. As shown in Figure 1, there are two kinds of collaboration in the system: the collaboration between the system and the user, and the collaboration between the system and the open Web. The novelty with respect to our proposed system lies in its evolving paper repository, and its ability to make smart, adaptive recommendations based on the system's observations of learners' activities throughout their learning and the accumulated ratings given by the learners.

To achieve the system goal, each paper must be tagged based on its content and technical aspects. Moreover, learners are required to give feedback (ratings) towards the papers recommended to them. Therefore, according to both the usage and ratings of a paper, the system will adaptively change a paper's tags and determine whether or not the paper should be kept, deleted or put into a backup list. Since new papers are added into the system and useless papers are deleted from the system, the system evolves according to the usage by the learners. Thus, the most important parts in the system are the recommendation module and the paper maintenance module, which become the main focus of our research.

Figure 2 illustrates the overall architecture of the system. There is a paper repository where papers related to the course are actively maintained through the paper maintenance module, which includes a web crawler which can occasionally crawl specified digital libraries to find more papers. In addition to automatic search, authorized instructors and learners are allowed to suggest

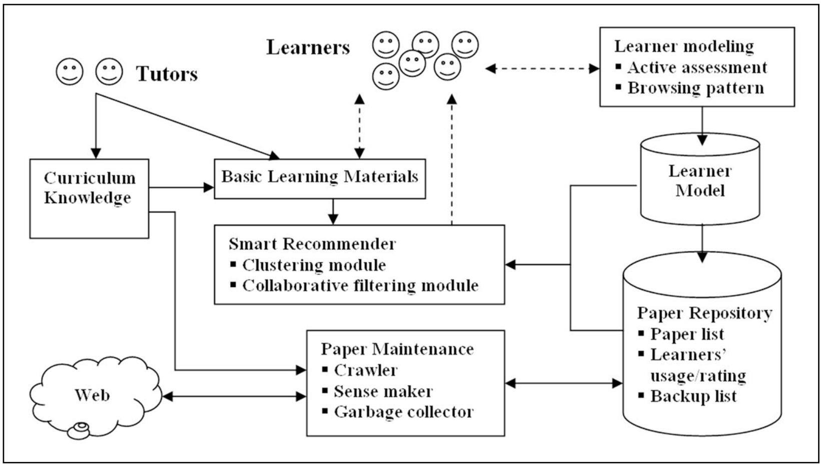

Figure 2. The architecture of the proposed system

papers to the system. The paper maintenance module will verify the suggested title by searching paper attributes from available digital libraries, and then it will refine the annotation of the paper according to the ratings by learners. It is through the recommendation module that personalized recommendations are made. The recommendation module consists of two sub-modules: the data clustering module and the focused collaborative filtering module. The data clustering module will cluster learners into a sub-class according to the purpose of the recommendation, while the focused collaborative filtering module will find the closest neighbor(s) of a target learner and recommend paper(s) to him/her according to the ratings by those closest neighbor(s). Tutors are responsible to set up the curriculum and provide basic learning material such as introduction part. Based on this information, the system can select a set of papers and find new papers if any. Learners are responsible to give ratings and other assessments at the beginning or during the middle of learning. For more detailed descriptions of each module, see (Tang, & McCalla, 2003a)

What Makes Recommendations in E-learning Different from that in Other Domains

Making recommendations in e-learning is different from that in other domains (the most studied domain of recommender system is movie recommendations, (Basu, Hirsh, & Cohen, 1998; Herlocker, Konstan, Borchers, & Riedl, 1999; Schein, Popescul, Ungar, & Pennock, 2002; Melville, Mooney, & Nagarajan, 2002). Particular issues for an e-learning recommender system include:

- Items liked by learners might not be pedagogically appropriate for them;
- Customization should not only be made about the choice of learning items, but also about their delivery (Kobsa, Koenemann, & Pohl, 2001);
- Learners are not expected to read too many papers.

For example, a learner without prior background on the techniques of web-mining may only be interested in knowing the state-of-the-art of web-mining techniques in e-commerce. Then, it should be recommended that he/she read some review papers, although there are many high quality technical papers related to his/her interest. By contrast, in other domains, recommendations are made based purely on users' interests.

For the delivery of papers, some instructors will recommend learners to read an interesting magazine article, such as a related article in Communications of ACM, before a technical paper, because they believe it will help learners understand the technical paper and make them less intimidated. However, this is not the case in e-commerce recommendations where site managers prefer to leave the list of recommended items unordered to avoid leaving an impression that a specific recommendation is the best choice (Schafer, Konstan, Riedl, 2001). In our proposed system, we will organize papers not only based on their main research categories, but also their technical levels. In addition, making recommendations in the context of intelligent tutoring systems is more tractable than in other domains since learners' interests, goals, knowledge levels etc., may be better traced in a constrained learning environment.

Finally, the amount of papers recommended to a learner in each course is limited. It is commonly accepted that each learner read no more than twenty papers in a course. In some cases, the amount may less than five. However, the amount of ratings by learner can affect the accuracy of collaborative filtering. Besides, in most collaborative filtering, cold-start problem remains one of the most important problems. Cold-start problem is the situation when there is no rating available for a new paper, or there is a new user who has not rated any paper. The common solutions are by assigning a set of pre-selected items or randomly assigned a set of items. Since in e-learning we cannot ask every learner to read too many papers and new papers are published every year, the solutions above are not appropriate in our system. (Tang, & McCalla, 2004b) describe the adoption of artificial learners to successfully solve the cold-start problem in our system.

It is commonly recognized that the sources of data on which recommendation algorithms can perform include users' demographic data. In our article, we will consider a special kind of user data different from the majority of recommender systems, (i.e., pedagogically-oriented data). The pedagogically-oriented data is different in the sense that it can directly affect as well as inform recommendation process, thus enhancing the quality of recommendations in the context of the web-based learning environments. The main pedagogical features used are the learner's goal and background knowledge, although other factors such as learning preferences are also important. To illustrate, consider the three learners A, B, and C in Table 1.

Table 1
A comparison of learner model A, B and C

	Learner A	**Learner B**	**Learner C**
Knowledge in Statistics	Strong	Weak	Weak
Knowledge in Marketing and Management science	Strong	Weak	Weak
Knowledge in network security e.g. SSL)	Strong	Weak	Weak
Interest	Network security, social network	Network security, social network	Data mining & web-mining application in e-commerce
Paper Preferences	Technical/theoretical	Application and magazine survey, technical/theoretical	Application and magazine survey

Suppose we have already made recommendations for the learners A, B and C. From Table 1, we can conclude that learner A and B have some overlapping interests, but since their knowledge background differs, especially with respect to their technical background, the papers recommended to them would be different. But for learner B and C, although they have different application interests, their technical background is similar; therefore, they might receive similar technical papers. In the next section, we will describe related work and underlying technique.

BACKGROUND KNOWLEDGE: RECOMMENDER SYSTEM

There are two basic approaches to providing personalized recommendations: *content-based* and *collaborative filtering* (Jameson, Konstan, & Riedl, 2002). Regardless of the approach used, at the core of personalization is the task of building a model of the user.

Content-based filtering approach

Content-based approaches recommend items purely based on the contents of the items a user has experienced/consumed before. Representative content-based recommender systems include News Dude (Billsus, & Pazzani, 1999) and WebWatcher (Joachims, Freitag, & Mitchell, 1997). Since user profiles in the content-based approach are built through an association with the contents of the items, this approach tends to be quite narrowly focused and with a bias towards highly scored items. Moreover, the content-based approach only considers the preferences of a single user. The Collabroative Filtering (CF) approach is an approach capable of exploiting information about other similar users.

Collaborative filtering approach

CF makes recommendations by observing like-minded groups. It works by matching a target user against his/her *neighbors* who have historically had similar preferences to him/her. GroupLens is a pioneer rating-based automatic recommendation system which successfully adopts the CF approach (Resnick, Iacouvou, Suchak, Bergstrom, & Riedl, 1994). Firefly is another rating-based CF system for music albums and artists (Shardanand, & Maes, 1995). Compared to a content-based approach, the CF approach has received more popularity and worked very successfully both in research and practice (e.g., Melville, Mooney, & Nagarajan, 2002; Sarwar, Karypis, Konstan, & Riedl, 2000).

Hybrid approach

A purely content-based approach only considers the preferences of a single user, and concerns only the significant features describing the content of an item; whereas, a purely CF approach ignores the contents of the item, and only make recommendations based on comparing the user against clusters of other similar users. By combining these two techniques, perhaps we can have both individual as well as collective experiences with respect to the items being recommended. (Balabanovic´, & Shoham, 1997) is one of the representative hybrid recommender systems.

RELATED WORK

Paper Recommendation

There are several related works concerning tracking and recommending technical papers. Basu et al.. (2001) studied this issue in the context of assigning conference paper submissions to reviewing committee members. Reviewers do not need to key in their research interests as they usually do; instead, a novel autonomous procedure is incorporated in order to collect reviewer interest information from the Web. Bollacker, Lawrence, and Giles (1999) refined CiteSeer, through an automatic personalized paper tracking module which retrieves each user's interests from well-maintained heterogeneous user profiles. Woodruff, Gossweiler, Pitkow, Chi, and Chard, (2000) discuss an enhanced digital book with a spreading-activation mechanism to make customized recommendations for readers with different types of background and knowledge. McNee, Albert, Cosley, Gopalkrishnan, Lam, Rashid, Konstan, and Riedl, (2002) investigate the adoption of collaborative filtering techniques to recommend papers for researchers. They only study how to recommend additional references for a target research paper. In the context of an e-learning system, additional readings in an area cannot be recommended purely through an analysis of the citation matrix of the target paper, because the system should not only recommend papers according to learners' interests, but also pick up those not-so-interesting-yet-pedagogically-suitable papers for them. In some cases, pedagogically valuable papers might not normally be of interest to learners, and papers with significant influence on the research community might not be pedagogically suitable for learners. Therefore, we cannot simply present all highly relevant papers to learners; instead, a significantly modified recommending mechanism is needed. Recker, Walker, and Lawless (2003) study the pedagogical characteristics of a web-based resource through Altered Vista, where teachers and learners can submit and review comments provided by learners. However, although they emphasize the importance of the pedagogical features of these educational resources, they did not consider the pedagogical features in making recommendations.

Dynamic Curriculum Sequencing and Adaptive Hypermedia

Recently, adaptive hypermedia has been studied extensively. According to Brusilovsky (2001), there are two kinds of adaptation: adaptive navigation ("link level") and adaptive presentation ("content level"). Adaptive presentation is then sub-grouped into text and multimedia adaptation, while adaptive navigation is mainly sub-grouped into link ordering (Kaplan, Fenwick, & Chen, 1993), link annotation (Pazzani, Muramatsu, & Billsus, 1996), and link hiding (including removal, hiding and disabling (De Bra, & Calvi, 1998)). Early research in adaptive hypermedia has concentrated mostly on adaptive presentation technology (Boyle, & Encarnacion, 1994), capable of adaptively presenting the content of a given page or collections of pages which have been viewed by a user. More recently, more aspects of learners are utilized in order to tailor the delivered content, (e.g., Stern, & Woolf, 2000). It is obvious that the contents of the pages are used as clues to derive important learning features of students such as their interests, knowledge state, etc. From another perspective, part of this branch of study can be viewed alternatively as content-based recommendations when users' past reading items/pages are recorded and analyzed. Over the past few years, link-orientated adaptation technologies are increasingly reported in the literature (De Bra, & Calvi, 1998; Weber, & Brusilovsky, 2001).

Document Value

The majority of scientific literature as well as other document retrieval systems (and many other systems) has been focusing on finding documents relative to users' interests, to name a few (Bollacker et al., 1999; McNee et al., 2002). Recently, there have been approaches that augment the mostly commonly adopted similarity-based retrieving. Among them, Paepcke, Garcia-Molina, Rodrigues-Mula, and Cho (2000) propose a *context-aware content-based filtering*. In particular, *context-aware* content-based filtering attempts to determine the contextual information about a document, for example, the publisher of the documents, the time when the document was published, etc. For instance, they argued that "documents from the New York Times might be valued higher than other documents that appear in an unknown publication context." This contextual information provides additional rich information for users, thus constitutes a very important aspect of the value-ness of the item.

Our proposed approach takes into account of one type of contextual information: the *pedagogical* feature of learners. In particular, we argue that users' pedagogical goal and interest should be regarded as two of the most critical considerations when we are making recommendations in e-learning systems. We believe that the value added to the paper, in term of new knowledge that the learner learned from the paper, depends on the richness of information and the learner willingness to digest it. Moreover, this willingness depends on learner interest and motivation, which is reflected as learner goal.

COMPONENTS AND TECHNIQUES

Paper Repository

Electronic versions of all papers, including magazine articles, conference papers, workshop papers, etc., will be stored in the Paper Repository. If the electronic version is not available, then the hyperlink pointed to the paper or relevant publication information will be stored in the system. In addition, the repository also stores ratings and comments by users. Generally, the repository is a database with PaperID# as the primary key.

Four major tables are used in the database: MainTable, ActivePaperList, CandidatePaperList and UserRating. MainTable contains all completely verified paper information but comments by users. ActivePaperList contains only information of active papers which are ready to be recommended to users. CandidatePaperList contains a list of "new" papers which have not completely verified. A completely verified paper will be deleted from CandidatePaperList and added into MainTable. UserRating contains all user ratings and other comments in text format. Generally, ActivePaperList is a part of MainTable. The reason to have it separately is to facilitate the recommendation module, because a higher cost of processing a larger data and a frequent access of the ActivePaperList by the recommendation module. The UserRating is used to monitor user progress and to refine paper information.

For each paper, there are three kinds of tags to describe it: content tag, technical tag, and usage tag. Content Tag is the paper's attributes available from most digital libraries. It includes PaperID#, Paper Title, Author(s), Publication Year, Publication Place (journal or conference name), Publication Type (book chapter, journal, conference, etc.), Category Contents in terms of subject and keywords and paper length, etc. Technical Tag is additional information which describes the technical level of the paper content (for novice, medium or advanced learners) and can be used in content-based filtering (model-based recommendation). The granularity of technical level is determined by the instructor. For instance, in a Software Engineering course, we can use a coarse grain such as Statistics, Discrete Math, Internet Computing, OO-programming, etc. But for a course in Artificial Intelligence we may use finer grain such as Predicate Calculus, Probability Theory, Utility Theory, etc. Technical tags are usually added manually when the paper is newly added, or inferred and adjusted based on the feedback given by learners, which will be explained in the next section. In order to keep a complete record of paper usage, we also need a usage tag which includes: userID of users who rated the paper and ratings and the time the ratings are submitted.

RECOMMENDATION MODULE

Data Clustering Module

Clustering learners based on their learning interests is handled by the data-clustering module. In our approach, we will perform a data clustering technique as a first step to coarsely cluster learners based on the recommendation goal, their interest, background knowledge, etc. Basically, for each prototypical user group, there are representative candidate papers associated with it. It is obvious that these representative papers are the centers of their respective clusters. In addition, since a learner might fall into more than one cluster, the clustering algorithm should allow overlapping clusters.

Clustering is good at finding a densely populated group of users with close similarities, but it fails to provide personalized information for these users. In order to make up for this, individualization can be achieved by further performing a collaborative filtering technique. The advantages to first applying clustering is not only to scale down the candidate sets, but also to guide collaborative filtering into a more focused area where high quality, personalized recommendations can be made (Tang, & McCalla, 2003b).

Focused Collaborative Filtering Module

After clustering is performed, learners are categorized in clusters based on their learning goal, interests, etc. However, recommendations cannot be made at this point, because even for learners with similar learning interests, their ability to consume papers can vary due to the dissimilarity of their knowledge level (as shown in our example in the previous section). Therefore, during this process, recommendations will be made not on the whole pool of users as most recommender systems do (e.g., Herlocker et al., 1999; Schein et al., 2002; Melville, Mooney, & Nagarajan, 2002), but on the clustered areas (Tang, & McCalla, 2003b).

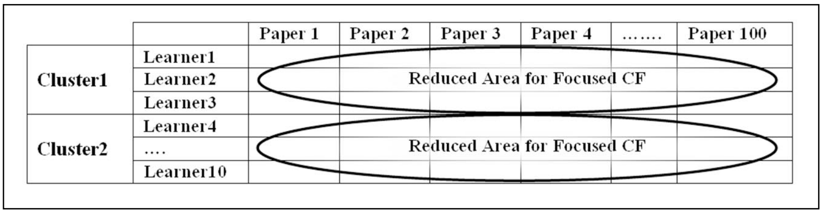

Figure 3. An illustration for focused CF

INTELLIGENT PAPER MAINTENANCE MODULE

The maintenance module is mainly responsible for updating (including adding, deleting, putting into backup list), collecting, and making sense of papers. Figure 4 shows the components of this module.

Topic-Driven Web Crawler

There is a web crawler embedded in this module, which is responsible for crawling digital libraries. To date, there has been a huge amount of research concerning web crawlers. Similar to the crawlers in other literatures, our web crawler is a topic-driven web crawler, which exploits the content-similarity between course topics and candidate papers. In the simplest form, a database is used to store the links to available e-journals. The database also stores the template (rules) for retrieving the information inside the journal.

Sense-Maker

The Sense-Maker is mainly responsible for filtering out loosely related papers and grouping them into their appropriate topical categories. Paper tagging will be accomplished during this process, where the results of this process are candidate papers with appropriate tags. The sense making here is adaptively performed based on the collective learning behaviors and interests of users instead of an individual learner.

But when there are accumulated ratings for the paper, the Sense-Maker can adaptively determine the appropriate technical tag for the paper. For instance, the majority of learners might find the paper to be highly technical which requires more extensive knowledge of both collaborative filtering and association rule mining, and their given ratings can be reflected in the paper's technical tag. Therefore, each paper's technical tag evolves according to the collective usage and ratings of its learners.

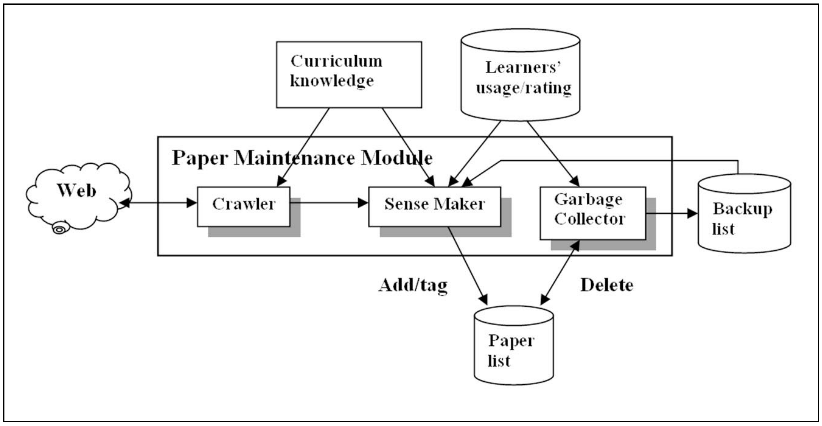

Figure 4. Intelligent Paper Maintenance Module

Garbage Collector

In order to keep the Paper Repository from growing too large, an intelligent Garbage Collector is used to decide whether or not to discard a paper completely or put it into a backup list for possible specialized needs. Although as pointed out in McCalla (2000), patterns of user behavior might be needed to perform garbage collection, in our system we will only focus on the usage of papers as a reliable form of users' paper reading patterns, which indirectly determine the 'survival time' of a target paper. In addition, compared to the survival analysis proposed in (Pitkow, & Pirolli, 1997), our module is simpler. In spite of it, we argue that in the context of our system, it is enough for us to capture both the overall usage and ratings of a target paper in order to determine whether or not to discard the paper.

There are several criteria in determining whether a paper should be deleted or not. For example, overall frequency within the specified category most recent frequency (Debevc, Meyer, & Svecko, 1997), overall cross-category frequency (since one paper might fall into more than two topical categories, its overall cross-category frequency measures its accumulative usage across these categories), average rating, and minimum acceptable rating. The first three factors concerning the usage of a paper mainly measure the frequency with which a target paper is recommended and read; the last two measure users' ratings of the target paper. If a paper consistently receives low ratings over a pre-defined period of time, it will be deleted.

MODEL AND FEASIBILITY STUDY

In this section we will provide formal concepts used in our system, followed by a simulation study in order to check the effectiveness of collaborative filtering in the context of learning.

In order to show that learner interest is not the primary factor in recommendation, we conducted a small-scale survey by asking our colleagues and friends (most of them have completed or are currently in a graduate program). Survey results substantiate our previous claims that uninteresting, yet pedagogically valuable papers should be recommended. These pedagogically useful, yet uninteresting papers (items) are not *false positives*) Sarwar et al.., 2000), because they could be helpful in one way or another to fulfill learners' learning expectations. Details of the survey can be found at (Tang, & McCalla, 2004a).

Pedagogically-Oriented Paper Recommendation

Our goal can be stated as follows. *Given a collection of papers and a learner's profile, recommend and deliver a set of pedagogically suitable materials in an appropriate sequence, so as to meet both the learner's pedagogical needs and interests.* Ideally, the system will maximize a learner's utility such that the learner gains a maximum amount of knowledge and is well motivated in the end. However, the content-based recommendation, which is achieved through a careful assessment of learner characteristics and then matches these models against papers, is very costly due to the following reasons:

- When a new paper is added into the system, a detailed identification is required (e.g., tagging it with detailed information of the background knowledge needed for understanding it), which cannot be done automatically;
- When a learner gains some new knowledge after reading a paper, a new matching process is required in order to find the next suitable paper for him/her, resulting in the updating of his/her learner model;
- The matching between learner model and paper model may not be a one-to-one mapping, which increases the complexity of the computation.

Alternatively, we can use collaborative filtering (CF) to reduce the complexity of the recommendation process. The idea of CF is to let peer learners filter out those not suitable materials, while the system does not need to know the detailed characteristics of them. Hence, the matching process is not performed from learner models to learning materials, but from one learner model to other learner models. Since the system also utilizes some characteristics of papers and considers both learner knowledge and interest, then it is not a pure CF but a hybrid-CF. The remaining question is whether or not the hybrid-CF is as effective as the content-based recommendation. To answer this question, we carried out an experiment using artificial learners for two types of pedagogical-oriented recommendation techniques: pure content-based, which makes recommendations based on the matching of learner models to papers, and hybrid CF, which is based on peer learner recommendation.

A Formal Notation of Paper Recommendations

In a formal notion, we can state our recommendation steps as follow:

1. For a learner model ***l***, find a group of similar learners, **N**(***l***). *(cluster of learners)*
2. Given content **C**, find a group of relevant papers **R**(**C**). *(cluster of papers)*
3. Find a subset of learners **N**'| **N**(***l***), who have read/rated any paper in **R**(**C**); denoted by ***f***: **N**(***l***)↔ **R**(**C**) ♦ **N**'. *(refine the cluster of learners)*
4. Based on the attributes of **R**(**C**), use content-based filtering to find a set of recommended papers **R**' | **R**(**C**) such that they match learner model ***l***.
5. Or, based on the ratings given by **N**', use collaborative filtering to find a set of recommended papers **R**' | **R**(**C**).

The first step can be achieved without complex computation – for example, grouping by learners who have taken the same courses together, or grouping according to their learning profile (e.g., average grade of previous courses). We can also ignore this step if not many learners are recorded in the database. The second step can be realized by checking the subject and keywords of the papers. The third step is easily checked from the database. Finally, the fourth and fifth steps need more attention which will be described using the following basic definitions.

Definition 1. *A paper in the domain being learned, denoted by* $\boldsymbol{r}$*, is called commonly well selected if it is pedagogically suitable for all learners* $\boldsymbol{L}$ *under common learning constraints (time, prior knowledge, availability, etc.). The same definition applies for a set of all papers, denoted by* $\boldsymbol{R}^{C}$*,(i.e,. it is commonly well selected if all papers* $\boldsymbol{r} \in \boldsymbol{R}^{C}$ *is commonly well selected).*

Definition 2. *A paper in the domain being learned is individually well selected if it is pedagogically suitable for a specific learner* $\boldsymbol{j} \in \boldsymbol{L}$*, under his/her individual learning constraints (common learning constraints plus individual learner characteristics, such as learning style, prior knowledge, preference, etc.). The same definition applies for all individually well selected papers, denoted by* $\boldsymbol{R}^{I}$*, (i.e., it is individually well selected if all papers* $\boldsymbol{r} \in \boldsymbol{R}^{I}$ *is individually well selected).*

Definition 3. *The set of all individually well selected papers is called the aggregate well selected paper, denoted by* $\boldsymbol{R}$*.*

Thus, we get $\boldsymbol{R}^{C} = {}_{j \in L}\boldsymbol{R}^{I}_{j}$ and $\boldsymbol{R} = \cup_{j \in L}\boldsymbol{R}^{I}_{j}$. Additional paper beyond $\boldsymbol{R}$ is unnecessary. However, deciding $\boldsymbol{R}^{I}$ is a non-trivial task, because in an ideal case, the tutor needs to decide proper pedagogical criteria in recommending the paper. In our proposed system, we left the learners and garbage collector to decide the set of $\boldsymbol{R}$.

Definition 4. *Similarity of two papers* $\boldsymbol{r}_1$ *and* $\boldsymbol{r}_2 \in \boldsymbol{R}$*.*

v-similarity (version-based): $\boldsymbol{r}_1$ *and* $\boldsymbol{r}_2$ *share the same topic, might be written by same authors, but one is a refined/updated version of another.*

c-similarity (comparison-based): $\boldsymbol{r}_1$ *and* $\boldsymbol{r}_2$ *discuss the same topic, with different approach.*

t-similarity (technique-based): $\boldsymbol{r}_1$ *and* $\boldsymbol{r}_2$ *use the same technique to solve two different problems.*

s-similarity (simplicity-based): $\boldsymbol{r}_1$ *and* $\boldsymbol{r}_2$ *concern the same topic and have the same level of simplicity in order to be understood.*

Ideally, in content-based recommendation, we want to include papers with *c-similarity* but exclude *v-similarity*. According to learner interest and background knowledge, the system will recommend similar papers based of *s-similarity*. At current stage, we did not consider *t-similarity*. In CF, we do not consider those similarities which make our system less complicated.

Definition 5. *Ordering of a set of papers* $\boldsymbol{R}^s$*, where* $\boldsymbol{R}^s \subseteq \boldsymbol{R}$ *and* $|\boldsymbol{R}^s| > 1$*.*

t-order: sequence of $\boldsymbol{R}^s$ *according to their technical difficulty.*

l-order: sequence of $\boldsymbol{R}^s$ *according to their length.*

p-order: sequence of $\boldsymbol{R}^s$ *according to the abstraction of their presentation.*

r-order: sequence of $\boldsymbol{R}^s$ *according to the prestige of their publications.*

c-order: sequence of $\boldsymbol{R}^s$ *according to the chronology of their publications.*

In general, in the context of both content-based and collaborative filtering, we only consider *c-order* and *l-order*. The paper maintenance module will filter out papers from less prestigious publications by crawling only prestigious publishers and adding new papers only; thus it is not considered in the recommender module. The abstraction of paper presentation is not considered as well. The ordering in CF is realized after the system gets all the higher rated papers **R'** and is ready to deliver them to the learner. The straightforward way is by using *t-order*, from the easiest one, or combining it with *l-order*, with the shorter one first.

The "Conflict of Understanding and Interest" Problem

For paper $\boldsymbol{p}$, and user $\boldsymbol{U}$, we might have the learner sequence $\boldsymbol{U}_i{}^t$, where $\boldsymbol{t}$ is the time when the user accessed/read the paper. Therefore, when a user reads a paper at different time, he/she might have different ratings toward it, in that his/her understanding towards the paper might change (either for the better or for the worse) due to his/her own increasing background knowledge on the subject. This will also lead to a so-called "conflict of understanding and interest" problem where a user might provide largely different ratings towards a paper. But from both learners' and tutors' perspectives, this phenomenon is natural given the increasing pedagogical ability of learners as time goes by. Therefore, we will not make effort to "solve" this conflict; instead, these traces of living conflicts will be explored later to make a deep understanding of both the usage of a paper, and the learning curve of a learner.

It is obvious that as we can cluster users purely based on their browsing behaviors, we can also cluster the annotated user models with respect to a specific paper, or sequences of papers. Technically, the sequences of user models along with the collections of paper will provide rich information related to both users, user patterns, papers and paper usage patterns, which, in turn, can make more refined recommendations, provide both personalized and groupalized recommendations and form dynamic and collaborative groups based on clusters of learners with different interests, pedagogical backgrounds (Tang, & Chan, 2002).

EVALUATING PEDAGOGY-ORIENTED HYBRID COLLABORATIVE FILTERING

As stated previously, we argue that two pedagogical features may be important in the recommendation system (RS): interest and knowledge. Moreover, content-based filtering may not as convenient as CF. In this section we will describe our experiment with both recommendation techniques using both pedagogical features.

Simulation Setup

For the purpose of testing, we first generate 500 artificial learners and use 50 papers related to data mining as the main learning materials. The RS then delivers recommendations of 15 papers to each learner according to each individual learner model (pure content-based). Each artificial learner rates these papers according to their properties. After that, we generate 100 additional artificial learners, who become the target learners. Then, two recommendation techniques are applied for these target learners in order to evaluate their differences as well as performances. The first technique is the same as the technique used in the first 500 learners, (i.e., content-based recommendation). The second technique uses a hybrid-recommendation technique (model based with collaborative filtering).

Learner Properties

In the simulation, we use minimal learner properties to generate artificial learners, as shown below:

- ***Learner ID*** #.
- ***Background knowledge*** as vector [(k_1, k_2) (k_3, k_4) (k_5, k_6) k_7, k_8, k_9, k_{10}], where k_i represents its strength on knowledge i-th, and $k_i \in [0, 1]$. We assume that k_1 and k_2 are two basic mathematics topics, k_3 and k_4 are two discrete mathematics topics taught in computer science or mathematics, k_5 and k_6 are two statistics topics, k_7 is algorithm analysis, and k_8, k_9 and k_{10} are topics in database, bioinformatics, and AI in education. k_1 is derived from truncated inverse standard lognormal distribution with $\sigma = 1$ and reduced by factor 1/5. And k_2 is lower than k_1 by the factor which also follows truncated standard lognormal distribution with s = 1 and reduced by factor 1/10. k3, k4 and k5 are derived from truncated inverse lognormal distribution with $\sigma = 1$ and reduced by factor 1/5. k_6 is derived from truncated standard lognormal distribution with $\sigma = 1$ and reduced by factor 1/5. k_7 is derived from uniform distribution U[0, 1]. k_8, k_9 and k_{10} are derived from truncated standard lognormal distribution with $\sigma = 1$ and reduced by factor 1/5.
- ***Interest toward specific topics*** as vector [I_1, I_2, I_3, …., I_{12}] , where I_i represents its interest on topic i-th, and $I_i \in [0, 1]$. We assume that all inter-

ests except I_1 (general topical knowledge) are generated randomly following uniform distribution. And I_1 is generated using truncated inverse standard lognormal distribution with $\sigma = 1$ and reduced by factor 1/5.

- ***Motivation*** as value $M \in [0, 1]$. Where 1 represents that the learner's willingness to spend more time to learn something not covered/understood before and 0 represents the learner's unwillingness to do so. M is generated using truncated standard lognormal distribution with $\sigma = 1$ and reduced by factor 1/5.

Paper Properties

We use the following properties for the papers:

- ***Paper ID #.***
- ***Technical knowledge*** as vector $[(k_1, k_2)\ (k_3, k_4)\ (k_5, k_6)\ k_7, k_8, k_9, k_{10}]$, where k_i denotes the extensiveness of the knowledge i-th used inside the paper. The extensiveness of a knowledge means that a learner needs a good background of the corresponding knowledge in order to be able to understand the paper thoroughly. If the learner lacks that knowledge, then he/she can gain the corresponding knowledge by reading the paper carefully and spending more time to look at the references. We assume $k_i \in [0, 1]$, and each of them represents the same topic as that described in learner properties. This feature indirectly affects the technical level of the paper.
- ***Paper topics*** as vector $[I_1, I_2, I_3, \ldots, I_{10}]$, where I_i denotes the corresponding topic in the learner's interest.
- ***Authority level***, which is used to determine whether the paper is an important paper or not, for example, a classical paper or highly cited paper, etc.

Learning Constraints

Five core papers, (i.e., papers that are pedagogically required for all learners; for example, the seminal paper by Agrawal in 1994 introduced, for the first time, the notion of association rule mining and its technique), will be recommended by the system regardless of learners' interests or knowledge. Those papers are core papers in the simulated course. They are specifically chosen as follows: either paper ID #1 or #2, either paper #5 or #6, paper #8, one of paper #26 or #27 or #35, and either paper #33 or #48. Moreover, at least two papers with high technical level should be recommended to the learner. In total, 15 papers must be read and rated by each learner.

The above requirements define the constraints of recommendation, which differentiates the recommendation in an e-learning system from that in other application areas.

MODEL-BASED RECOMMENDATIONS

Rule Generations

The content-based recommendation is based on the following rules (learner-centric):

- System starts with recommending acceptable papers in terms of learners' knowledge level (understandable) and the similarity of learners' interest towards the topic in the paper (the understandable level and the interest similarity will be described later). Up to eight authoritative papers will be selected first; if no more authoritative papers can be selected, then the system will recommend non-authoritative papers.
- Two interested papers, but with very high technical level will be recommended in order to improve the learner's knowledge.
- Some not-interested-yet-pedagogical-useful (authoritative) papers will be provided as the part of the learning requirement in the end.

After learners finish a paper, some additional knowledge may be acquired, which depends on the learner motivation. In our simulation, we assume that the increment is based on:

IF $paper.k_j > learner.k_j$ **AND** $paper.authority$ = TRUE **THEN**
$learner.k_j = (paper.k_j - learner.k_j) \times learner.M \times Interest \times w_1 + learner.k_j$

IF $paper.k_j > learner.k_j$ **AND** $paper.authority$ = FALSE **THEN**
$learner.k_j = (paper.k_j - learner.k_j) \times learner.M \times Interest \times w_2 + learner.k_j$

Where w_1 and w_2 represent factors that might affect learning speed after reading an authoritative/non-authoritative paper. They are two of the control variables in our experiment. *Interest* represents the similarity of a learner's interest to the paper's topic which will be described later. Moreover, the total gain made by the learner is defined as the value added from reading the paper, or

$Value\ added = \Sigma_i$ ((new) $learner.k_i$ – (old) $learner.k_i$)

The rule to measure a learner's understanding (*Understand*) will be based on the knowledge gap between the learner's knowledge and those required to fully understand the paper. In addition, the similarity of learners' interest to the paper's topic (*Interest*) is generated according to the following rules:

$y = 1$ if $\exists j$ such that $learner.I_j$ and $paper.I_j \geq 0.9$

$y = 0.9$ if $\exists j$ such that $learner.I_j$ and $paper.I_j \geq 0.8$

...

$y = 0.1$ if $\exists j$ such that $learner.I_j$ and $paper.I_j \geq 0.0$

$Interest = \mathbf{Max}(y)$

Rating Generation Rules

After a learner reads a paper, we need rating-generation rules to generate learner rating toward the paper. We use the following rules in our simulation:

1. If learners are interested in the topic AND understand the paper, a higher rating is generated. Or, matching based on both interests and background knowledge generates higher ratings 4 or 5 under the following formula.

$$\text{Rate} = \text{Round}\,(\mathit{Interest} \times \mathit{Understand} \times 2) + 3 \text{ if } \mathit{Interest} \geq 0.7 \text{ and } \mathit{Understand} \geq 0.7$$

2. Learners give ratings to a paper based on the amount of knowledge that could be acquired (*value added)* AND the understanding of the paper (easy to follow, or *pedagogical-readiness*), OR the importance of the paper to their interest.

Rating Implications

If the rating is high (e.g., 4 or 5), learner motivation will increase randomly following uniform distribution to upper bound value 1, with increasing rate *x*. If the rating is low (e.g. 1 or 2), learner motivation will decrease randomly also following uniform distribution to lower bound 0, also with decreasing rate *x*. If the rating falls into the medium of the scale (e.g. 3), learner motivation unchanged. x is another control variable which represents how much motivation a learner gain/loss after reading a paper.

Hybrid Recommendations

The following rules are used for the hybrid recommendation.

- Neighborhood finding. For each target learner (*tlearner*) find five neighbors (*nlearner*) based on the similarity of their interest and background knowledge. The similarity measurement is calculated based on the following:

$$\mathit{Positive\ Similarity} = \Sigma\,(\mathit{nlearner.I_i}) \qquad \text{IF } \mathit{tlearner.I_i} \geq \mathit{nlearner.I_i}$$

$$\mathit{Negative\ Similarity} = \Sigma\,(\mathit{nlearner.I_i} - \mathit{tlearner.I_i}) \qquad \text{IF } \mathit{tlearner.I_i} < \mathit{nlearner.I_i}$$

$$\mathit{Similarity} = \mathit{Positive\ Similarity} - \mathit{Negative\ Similarity}$$

The *similarity* formula is used to find similarity in learner background knowledge. The rationale to adopt this measurement is when a learner has a lower interest than a target learner. The magnitude of the learner's interest is credible for recommending a paper to the target learner, therefore the positive similarity measures the total of learners' interest in such condition. However, if the learner's interest is higher than the target learner's interest, then an error may appear regarding the learner's recommendation, and the gap between those two interests may be the cause of the error. Therefore, the negative similarity denotes the sum of the gaps. The same is used for the similarity measure of two learners' background knowledge.

- From these five nearest neighbors, we can get a set of candidate papers based on their ratings. In our simulation, each learner has rated 15 papers; therefore, at least 15 papers will be in the candidate set. Then, we order those papers in candidate set from the highest ratings from the closest neighbor to the lowest rating from the furthest neighbor.
- The system will recommend up to eight authoritative papers starting from those receiving the highest rating followed by recommending non-authoritative papers. Then, the system will choose and recommend two very interesting and highly technical papers, and recommend five pedagogically required papers, if the learner has not read them. Finally, the system recommends the rest of the papers according to the rating order, up to 15 papers in total.

Evaluation Metrics and Control Variables

Those commonly adopted metrics (Herlocker et al., 1999), (e.g., ROC) in the research community cannot be applied here due to the inherent features of recommendation for e-learning. These metrics, for example, ROC, are mainly adopted to test the users' satisfaction in terms of item interest. However, we argue that since the most critical feature of recommending learning items is to facilitate learning (not just to provide interesting items), it is not applicable in our domain. Therefore, we propose two new metrics as follows:

- **Average learner motivation** after recommendation
- **Average learner knowledge** after recommendation

And for the purpose of comparison, we compare the percentage differences between content-based recommendation and hybrid recommendation.

In our simulation, control variables w_1, w_2 and $\boldsymbol{x}$ are adjusted to differentiate artificial learners as follows: $\boldsymbol{x} = 1$ for fast motivation change (FMC), $\boldsymbol{x} = 0.3$ for moderate (MMC), $\boldsymbol{x} = 0.1$ for slow (SMC), and $\boldsymbol{x} = 0$ (NMC). Moreover, we use eight pairs of (w_1, w_2), which are (1, 0), (U[0, 1], 0), (U[0, 0.3], 0), (1, U[0, 0.3]), (U[0, 1], U[0, 0.3]), (U[0, 0.3], U[0, 0.3]), (1, U[0, 1]), (U[0, 1], U[0, 1]), (1, 1), where U[0, y] means a random value generated from a uniform distribution function. The pair value represents the effect of authoritative and non-authoritative papers in the increment of the learner's knowledge. For example, (1, 0) indicates that only authoritative papers can fully increase a learner's knowledge. And (1, 1) indicates that both authoritative and non-authoritative papers are equally weighted and can fully increase a learner's knowledge. Each group of experiments is repeated thirty times for statistical analysis.

EXPERIMENT RESULTS AND DISCUSSION

Experiment Results

Table 3 shows the results of the experimentation. The value shown in each cell is the pair value of the percentage difference between content-based recommendation and the hybrid-CF technique in terms of average learner knowledge and motivation. A negative value indicates that the content-based recommendation technique is better than hybrid-CF. And a positive value represents the reverse situation. For example, the pair value (**0.65; 2.93**) represents that using hybrid-CF is 0.65% and 2.93% better than using content-based in terms of the average learner knowledge and motivation respectively. All results are checked by t-test for equal mean hypothesis (assuming different variance). The value in italics inside the table shows that the null hypothesis is not rejected (for $\alpha = 0.05$), or the difference between content-based and hybrid-CF is not statistically significant. If we exclude zero and italic values in Table 3, then there are 14 and 6 negative values for the difference of learner knowledge and motivation respectively, with the lowest values equal to -1.05% and -5.68% respectively. And there are 8 and 12 positive values for the difference of leaner knowledge and motivation, with the highest values equal to 1.20% and 19.38%, respectively. Thus, we conclude that using hybrid-CF results in a lower performance in terms of

Table 3

The differences between content-based and hybrid recommendation (in percentage %). The first value in each cell represents the difference of final knowledge and the second value represents the difference of final motivation.

(w_1, w_2)	FMC	MMC	SMC	NMC
(1, 0)	**0.59; 2.77**	**-0.70;** *-0.06*	**-0.77;** *-0.42*	*-0.43; 0.00*
(U[0, 1], 0)	**0.98; 7.97**	*-0.28;* **3.85**	*0.21; -0.32*	**0.54;** *0.00*
(U[0, .3], 0)	*-0.47;* **15.15**	*-0.52; 0.75*	*0.33;* **-5.42**	**1.09;** *0.00*
(1, U[0, .3])	**-0.57; 1.61**	**-1.05; -1.05**	**-0.76;** *-0.90*	*-0.29; 0.00*
(U[0, 1], U[0, .3])	*0.30;* **8.09**	**-0.44; 3.41**	*0.22; -0.01*	**0.69;** *0.00*
(U[0, .3], U[0, .3])	**-0.85; 19.38**	**-0.69;** *-0.19*	*0.06;* **-5.68**	**1.20;** *0.00*
(1, U[0, 1])	**-0.52; 1.13**	**-0.96; -0.8**	**-0.82; -0.84**	*-0.27; 0.00*
(U[0, 1], U[0, 1])	**0.96; 7.36**	*-0.15;* **4.68**	*0.16; -0.06*	**0.88;** *0.00*
(1, 1)	*-0.34;* **1.47**	**-0.69; -1.31**	**-0.47;** *-0.81*	**-0.43;** *0.00*

learner average knowledge. However, since hybrid-CF usually needs lower computational cost than content-based recommendation (which is not measured here) and the performance loss is not big, hence hybrid-CF is very promising in e-learning system.

So far, it is unclear why the individual result of our simulations, especially some values which show high differences, especially when motivation changes quickly (FMC). However, we can conclude that using hybrid-CF may not always result in a lower performance. And if it happens, the difference may not higher than 5%. This conclusion is useful, since hybrid-CF needs lower cost than content-based recommendation. Thus, if the performance lost is not big, then hybrid-CF should be used instead of the traditional content-based recommendation.

Discussion

Computer simulation has long served as a tool of applying artificial intelligence on intelligent tutoring systems (Chan, & Baskin, 1990; Tang, & Chan, 2002). Although a simulation program can only model part of the real environment where real learners involve, it can afford a powerful tool for gaining insights for paper recommendations in complex settings. Therefore, the simulation discussed here can serve as a guide in our future study. In fact, we have designed a follow-up human subject study, which extends the resulting recommendations made by the artificial learners on the real human learners to solve the cold-start problem in our domain successfully (Tang, & McCalla, 2004b).

LESSONS LEARNED

One interesting approach associated with the recommendation module is that papers actually can be annotated by users themselves (so their user models), which is fundamentally different from current paper tagging techniques. Indeed, it is obvious that in such evolving e-learning systems, when more and more users interact with the system, each paper annotations will be greatly refined and automated, thus instead inform as well as improve the quality of future recommendations.

Another interesting issue concerns our experiments on both artificial learners and human subjects. More specifically, we conducted a human-subject study where each human user received papers based on the ratings given by the artificial learners (Tang, & McCalla, 2004c) and the rating generation mechanism follows what we describe in this paper. Experiment results showed that the majority of learners have struggled to reach a harmony between their interest and educational goal – they are willing to read non-interesting-yet-pedagogically-useful papers in order to acquire new knowledge for either their group project or their long-term goal. Hence, from this perspective, learners seem to be more tolerant than users in commercial recommender systems. Nevertheless, as educators, we should still maintain a balance of recommending interesting papers and pedagogically helpful ones in order to retain learners and continuously engage them throughout the learning process. This is especially true in our case, since most of the students in Hong Kong are more application-oriented.

CONCLUSION

In this article, we proposed an evolving e-learning system which can adapt itself both to the learners and the open the Web and pointed out the differences of making recommendations in e-learning and other domains. We propose two pedagogy features in recommendation: learner interest and background knowledge. A description of paper value, similarity, and ordering are presented using formal definitions. We also study two pedagogy-oriented recommendation techniques: content-based and hybrid recommendations. We argue that while it is feasible to apply both of these techniques in our domain, a hybrid collaborative filtering technique is more efficient to make "just-in-time" recommendations. In order to assess and compare these two techniques, we carried out an experiment using artificial learners. Experiment results are encouraging, showing that hybrid collaborative filtering, which can lower the computational costs, will not compromise the overall performance of the RS. In addition, as more and more learners participate in the learning process, both learner and paper models can better be enhanced and updated, which is especially desirable for web-based learning systems. We have tested the recommendation mechanisms with real learners, and the results are very encouraging. Details can be found at (Tang, & McCalla, 2004b, c).

References

Balabanovic´, M., & Shoham, Y. (1997) Fab: Content-based collaborative recommendation. *Communications of the ACM, 40*(3): 66-72.

Basu, C., Hirsh, H., Cohen, W., & Nevill-Manning, C. (2001). Technical paper recommendations: A study in combining multiple information sources. *JAIR, 1,* 231-252.

Billsus, D., & Pazzani, M. (1999). A hybrid user model for news story classification. *Proceedings of UM'99*, 99-108.

Bollacker, K., Lawrence, S., & C. Lee Giles, C. L. (1999). A system for automatic personalized tracking of scientific literature on the web. *ACM DL*, 105-113.

Boyle, C., & Encarnacion, A. O. (1994). MetaDoc: An adaptive hypertext reading system. *UMUAI,* 4, 1-19.

Brusilovsky, P. (2001). Adaptive hypermedia. *UMUAI, 11*(1/2), 87-110.

Chan, T., & Baskin, A.B. (1990). Learning companion systems. *ITS 1990*, 6-33.

Debevc, M., Meyer, B., & Svecko, R. (1997). An adaptive short list for documents on the world wide web. *IUI 1997*, 209-211. U.S.A.

De Bra, P., & Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia, 4*, 115-139.

Herlocker, J., Konstan, J., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. *SIGIR'99*, 230-237.

Jameson, A., Konstan, J., & Riedl, J. (2002). AI techniques for personalized recommendation. *Tutorial notes, AAAI-02,* Edmonton, Canada.

Joachims, T., Freitag, D., & Mitchell, T. (1997). WebWatcher: A tour guide for the World Wide Web. *Proceedings of IJCAI'97*, 770-775.

Kaplan, C., Fenwick, J., & Chen, J. (1993). Adaptive hypertext navigation based on user goals and context. *UMUAI, 3*(3), 193-220.

Kobsa, A., Koenemann, J., & Pohl, W. (2001). Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review 16*(2):111-155.

McCalla, G. (2000). The fragmentation of culture, learning, teaching and technology: Implications for the artificial Intelligence in education research agenda in 2010. *IJAIED. 11*(2): 177-196. 2000.

McNee, S, Albert, I., Cosley, D., Gopalkrishnan, P., Lam, S., Rashid, A., Konstan, J., & Riedl, J. (2002). On the recommending of citations for research papers. *ACM CSCW'02*, 116-125.

Melville, P., Mooney, R., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. *AAAI/IAAI 2002*, 187-192. Edmonton, Canada.

Paepcke, A., Garcia-Molina, H., Rodriguez-Mula, G., & Cho, J. (2000). Beyond document similarity: Understanding value-based search and browsing technologies. *SIGMOD Records, 29*(1): 80-92.

Pazzani, M., Muramatsu, J., & Billsus, D. (1996). Syskill and Webert: Identifying interesting Web sites. *AAAI'96*, 54-61.

Pitkow, J., & Pirolli, P. (1997). Life, death, and lawfulness on the electronic frontier. *ACM CHI 1997*, 383-390.

Recker, M., Walker, A., & Lawless K. (2003). What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education. *Instructional Science*, 31:299-316.

Resnick, P., Iacouvou, N., Suchak, N., Bergstrom, P., & Riedl, J. (1994). GroupLens: An open architecture for collaborative filtering of Netnews. *ACM CSCW'94*, 175-186.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. *ACM EC'2000*, 158-167. Minneapolis.

Schafer, J., Konstan, J., & Riedl, J. (2001). Electronic commerce recommender applications. *Data Mining and Knowledge Discovery, 5*, (1/2, 2001), 115-152.

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating 'word of mouth'. *ACM CHI'1995*, 210-217 Denver.

Schein, A., Popescul, A., Ungar, L.H., & Pennock, D. (2002). *Proceedings of SIGIR'02.* 253-260.

Stern, M. K., & Woolf, B.P. (2000). Adaptive content in an online lecture system. *AH*, 227-238

Tang, T.Y., & Chan, K.C.C. (2002). Feature construction for student group forming based on their browsing behaviors in an e-learning system. *PRICAI 2002*, LNCS 2417, 512-521.

Tang, T. Y., & McCalla, G. (2003a). Smart recommendations for an evolving e-learning system. *Workshop on Technologies for Electronic Documents for Supporting Learning, AIED'2003*.

Tang, T. Y. and McCalla, G. (2003b) Mining implicit ratings for focused collaborative filtering for paper recommendations. *UM 2003, Workshop on User and Group Models for Web-based Adaptive Collaborative Environments*, Johnstown, U.S.A.

Tang, T. Y. & McCalla, G. (2004a). On the pedagogically guided paper recommendation for an evolving web-based learning system. *FLAIRS Conference, 2004*. AAAI Press. 86-91.

Tang, T. Y. & McCalla, G. (2004b). Utilizing artificial learners to help overcome the cold-start problem in a pedagogically-oriented paper recommendation system. *In Proceedings of AH 2004: International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems. August 23-26*, Eindhoven, Netherlands, 245-254.

Tang, T Y. & McCalla, G (2004c) Laws of Attraction: In Search of Document Value-ness for Recommendation. In Proceedings of ECDL 2004: the 8th European Conference on Digital Library, Sept. 12-17 2004, Bath, UK. 269-280.

Weber, G., & Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for web-based instruction. *International Journal of AI in Education*, 12: 1-35.

Woodruff, A., Gossweiler, R., Pitkow, J., Chi, E., & Card, S. (2000). Enhancing a digital book with a reading recommender. *ACM CHI 2000*, 153-160.

Workflow-Based Personalised Document Delivery

S. HOLDEN, JUDY KAY, JOSIAH POON, AND KALINA YACEF
University of Sydney, Australia
sholden@it.usyd.edu.au
judy@it.usyd.edu.au
josiah@it.usyd.edu.au
kalina@it.usyd.edu.au

New employees in an organisation typically undergo a period of relatively intense training when they commence their employment. Often the quantity of information imparted is too large for the newcomers to assimilate during the short training period. Moreover, much of the information may not be relevant until months or even years after the initial training period, by which time it has long been forgotten. This article describes JITT, a smart personal assistant which delivers training documents to its user in a just-in-time manner. JITT uses workflow technology to drive the delivery of documents in a timely manner, based on organisational processes through which the user is working. User-modelling is incorporated into the system to ensure that redundant or previously known information is not delivered, thereby reducing the problem of information overload. Finally, the teaching agent, which is the engine of JITT, identifies the concepts to be learned, retrieves the best documents for teaching these concepts to the current user and customises the presentation of the documents' links in the user interface.

There are many learning challenges within organisations. Assimilation of organisational knowledge is a problem encountered in almost any organisation (Dieng, Corby, Giboin, Golebiowska, Matta, & Ribière, 2000). A particularly important group of problems relates to helping people learn things that are already well documented within the organisation. The challenge in supporting such learning is that the people within the organisation often have difficulty in gaining access to that information at the right time. In cases where the person is aware of their lack of knowledge, the primary challenge is to help them find the documents which can answer their needs. Even more challenging is the common case where the person is unaware that they have a learning need. This problem is compounded when people are feeling overloaded, either by the demands of their work or information overload.

One important aspect of this workplace learning relates to the learning load. The timing of teaching is critical to the success of learning. Hoyt pointed out that the Future Search Committee held during the ASTD International Conference and Exposition reported that “E-learning is dominated by an increasing demand for just-in-time learning” (Hoyt, 2002). In an organisational context, it is important to take account of the problems of information overload, especially in the case of people who are new to the organisation or who have recently changed their roles and responsibilities. Such newcomers to roles or organisations have a great deal to learn and are only able to absorb a limited amount of information at one time. In general, teaching should be paced so that a person is given information in manageable chunks, so as not to overload their short-term memory (Sweller, 1993).

It is particularly appealing to improve the delivery of documents at just the time that the learner is ready for them, when they need to learn. So, for example, suppose an employee has been in a job for six months at an organisation which requires him/her to take annual leave within the next six months. This employee would do well to begin planning that leave. So this would be a good time to ensure that they are aware of this leave policy. A simple way to achieve this would be to deliver a short prompt, perhaps via email, pointing them to the particular paragraph that they need know about in the leave policy document. This is just one simple example of a learning need that the employee may not have been aware of.

Another crucial aspect of workplace learning is the appropriateness of the information content given to the user – it should be adapted to the current context of their work, and to their pre-existing knowledge. There is a need to understand the user’s needs in order to adapt the delivery of information. Some researchers have adopted a task-driven approach to model the context of the users’ needs and build heuristics to tailor the information to them and present it in a just-in-time manner (Budzik, Hammond, & Birnbaum, 2001; Paradis, Crimmins, & Ozkan, 2003). In our project, we use a workflow approach to determine the context of the user’s needs.

In organisations where there are automated processes for managing workflow, these processes can provide a framework for supporting learning. For example, our university has a process for managing examination scripts. This process could well be supported by a conventional workflow system. A new academic who is responsible for running a teaching unit in our organisation would need to learn about several aspects of the unit and its related processes. Since newcomers need to learn so many other things when they first join the university, it is not surprising that they generally have problems learning what they need to know to get through this process. It is typical of many learning processes which run over a period of several months and have several stages. The process requires completion of several forms and there are several relevant policy documents. There are many support learning documents, including the policy documents of the institution, and of the school and there are examples of ways to complete sub-tasks. It is very appealing to tap into a workflow for a process like this so that it supports not only the normal management of the documents and approval processes, but also the learning about those processes. This should be in good time as well as just-in-time.

Both the examples mentioned, leave and examination script management, are also typical of many aspects of workplace planning in that there is considerable tacit knowledge involved. It is important for an organisation to have a system in place for employees to acquire this tacit knowledge (Nonaka, & Takeuchi, 1995). It would be valuable to facilitate learning of these aspects by helping newcomers and those transitioning to new roles by helping them make contact with people who have recent relevant experience and willingness to help.

This article describes the current version of JITT, a Just-In-Time Training system which tackles these problems by building upon workflow models and technology (Davis, Kay, Lin, Poon, Quigley, Saunders, & Yacef, 2002). The next section of this article describes the overall architecture of JITT, and the subsequent sections describe the interface, the details on each of the major architectural elements, and related work.

ARCHITECTURE

The overall architecture of JITT is illustrated in Figure 1. At the top is the Human Resources (HR) staff team. They need to initialise the system when a new employee starts. This process can be triggered by the typical administrative tasks associated with this starting stage.

For example, the new employee will have a role within the organisation, and the role will be reflected in an obvious way in their user model. This will show their position and level. Normal duty statements within the organisation would naturally feed into the definition of default tasks for each user. For example, a new member of academic staff would typically have responsibility for teaching and research. Each of these roles can be used to create a default user model for these aspects of a new member of staff. This would include the elements that are a normal part of these roles. The teaching role would, for instance, generally include the need to set examination papers and submit these and associated paperwork. The initial model would include a teaching context and within that, there would be a context for handling examination scripts. For a new staff member, these would initially be modelled with a stereotype (Rich, 1979) to reflect that the user does not know the procedures.

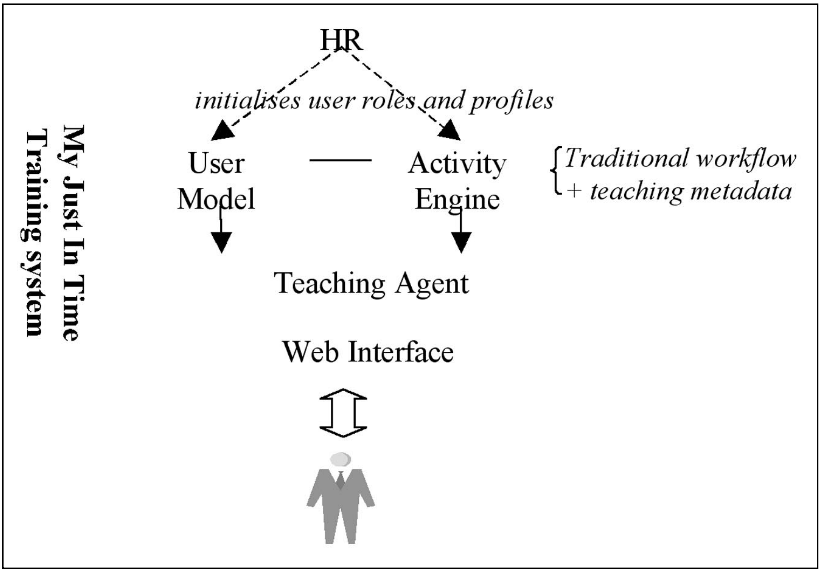

Figure 1. Architecture of JITT

At the bottom of Figure 1, we show the user. S/he can interact with JITT and update the user model. Although the HR staff creates the default model, the user can alter it. Users interact with JITT via the interface layer shown at the bottom of the main block in the figure.

When the Human Resources staff member establishes a new user in JITT, s/he also initiates the activity engine. This activates a set of workflows, each associated with a set of knowledge components in the JITT user model structure. So, for example, JITT creates user model components for the user's knowledge of leave processes within the organisation. This starts one workflow for that user for each of the forms of relevant leave. Similarly, the user model components for handling examination scripts are associated with a workflow for that process.

The workflow is managed by the activity engine. One important class of these is based on the typical process workflows that are currently used in organisations. One of these workflows could be used for the management of examination scripts. Unlike the case of typical business workflows, the JITT representation captures processes that affect an individual and their learning about the appropriate processes and policies within the workplace. This means that we identify stages in the individual's work even when the stages may not involve other people. The critical factor defining the stages we model in the workflow is that we identify the subtasks or sub-activities that constitute a larger activity. It is this breakdown of a task that makes it possible to order the teaching of relevant aspects with the user being taught how to do the current sub-process just as they need to tackle it.

The final part of the architecture is the teaching agent. This takes the information from the Activity Engine and the user model in order to formulate the presentation of information to the user. Since JITT has been designed to be highly modular, we designed it to allow for several teaching agents, each potentially taking a different strategy for its teaching. We use one of these agents as the default and allow the user to switch to the others.

We now discuss each of these elements in detail.

INTERFACE

We now describe the interface element of the architecture. This also provides an overview of JITT from the user's perspective. We have designed JITT so that it could potentially use a range of mechanisms to communicate with the user. At this stage, as indicated in Figure 1, our prototype implementation uses just one interface element – a web site. We will illustrate the operation of the interface in terms of the following pacing: a user, who we call Jane, starts employment and interacts with JITT on her first day on the job. We then show her the JITT screen at later stages.

JITT is intended to work in either push or pull mode. We first describe JITT's push mode of operation. As indicated in Figure 1, an individual user's JITT agent is initiated by the actions of the Human Resources staff. This means that the new employee's JITT agent can establish an initial user profile and student model and it can use these to formulate an initial set of teaching goals. As soon as JITT is initialised for the user Jane, she can log onto the JITT web site and see the activities that are already activated for her. These activities represent her set of initial recommended learning goals. Figure 2 shows an example of a JITT screen with this set of learning goals, which are organised into categories. From the user's point of view, these goal categories are intended to structure the presentation in terms of major activities in the workplace. In terms of our architecture, each main category maps to one workflow within JITT.

The top left box shows the list of current activities for the user Jane. As we can see in Figure 2, these are composed of "Course planning," "Postgraduate Supervision" and "Administrative Procedures for New Staff."

Directly below, the user can find one box for each current activity, and each box notes the stage she has reached for that activity (not all boxes are visible on the screenshot). In particular, the links of the documents relevant to the user for the current stage of the activity are pushed to the user. Sometimes, depending on the teaching agent, links to documents that the system judges to be already known by the user can also be shown in the same box under "Refresher documents." That way the user can consult them again if needed. At any time, the user can view the workflow that corresponds to the

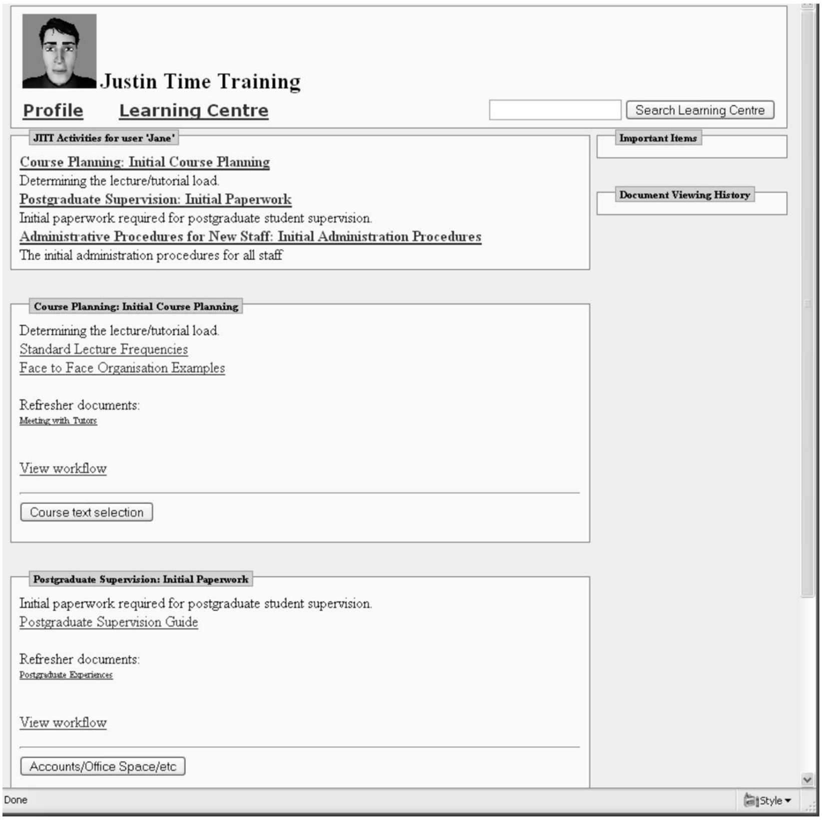

Figure 2. JITT, day one for user Jane

activity and see her current stage. This allows the user not only to visualise which stage she is at in the process, but it also provides the context in which a particular document is relevant now. An example is shown in Figure 3.

The right hand part of the screen in Figure 2 is composed of two areas. The “Important Items” shows the documents that the user has put aside for later. The “Document Viewing History” lists all the documents recently accessed with the most recently accessed document listed first. If a document is accessed more than once, it appears according to its last access date. These two areas are naturally empty when the user logs in for the first time, but later on in the training they start filling up (see Figure 5).

The results are sorted by activity in JITT. Essentially, the user should see learning recommendations based on a combination of the request and the user's current learning context.

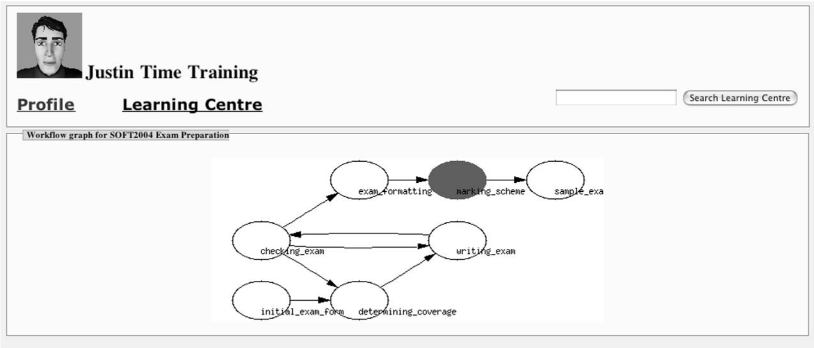

Figure 3. User visualising a workflow

JITT also operates in a pull mode when the user initiates a search request (in the "Search Learning Centre" text area in the top right corner of the window). The system looks up both documents that are linked to the currently active workflows for this user and other collections of organisational documents. It then returns these documents, as well as the corresponding workflows that teach this concept. For instance in Figure 4, user Jane has just performed a query on "leave." She is provided with a set of available documents related to

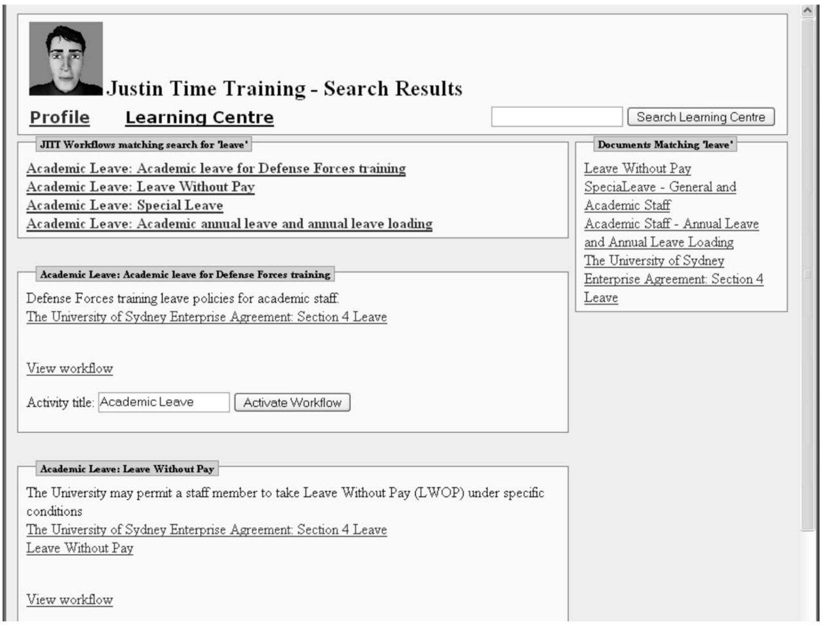

Figure 4. JITT, after a search request

the concept “leave” (shown in the right frame) and a set of workflows that match the concept “leave” (shown in the top left frame). The user can then either simply consult the documents or choose to activate one of the workflows and decide to follow the system’s advice on learning about this concept. In the latter case, the workflow will be added to the list of learning activities.

Figure 5 indicates the types of changes in the interface in the later visits to the JITT site. Let us suppose that Jane, after her search request on “leave,” decided to activate the workflow “Academic Leave: Academic annual leave and annual leave loading.” As shown in Figure 5, the workflow is added to the list and she now sees four active workflows. The overall learning categories, as active workflows, will tend be stable over long periods of time. However, at each stage, different learning objectives will be suggested for some of them. For instance, Jane has progressed in the first learning activity (Course Planning). She is at a stage called “Exam Preparation,” which corresponds to a node in the workflow. Four document links are pushed to her and she also has a list of refresher documents under that activity. Once she is ready she will click on the button “Preparing the Marking Scheme” and the

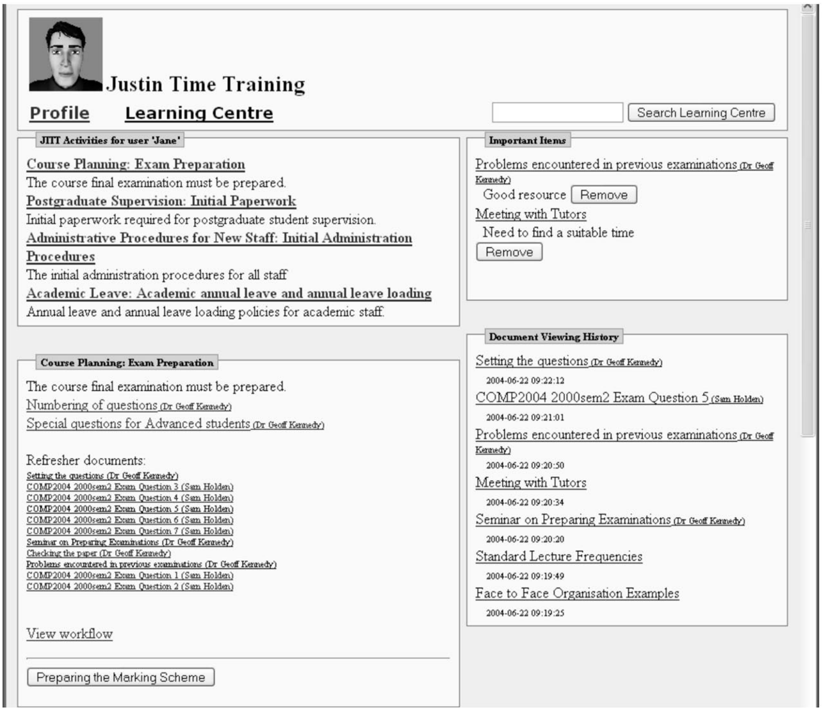

Figure 5. JITT, at a later date

workflow will progress to the next node, updating the set of links in the box for that activity. We can also see in this figure that the two frames on the right now have information in them. Jane has put two documents aside for later, with a short description as to why (i.e., "Need to find a suitable time"). Her document viewing history has also grown, showing that the most recent document she accessed was "Setting the questions." Documents appear in reverse chronological order, and if a document is accessed a second time it is moved to the top of the list, hence only showing the latest access date.

ACTIVITY ENGINE

The workflow technology lies in the activity engine. In the literature, when workflow technology is applied to teaching, it is used to handle the administration side of teaching or to let students pace their learning activities by relaxing the time constraints. The Flex-eL project (Lin, Ho, Sadiq, & Orlowska, 2001) is an example of such a system. JITT uses workflows differently. Rather than modeling the teaching workflows, JITT uses the existing business workflows and deduces the pacing of the training from them.

The workflows in JITT are simplified versions of standard business workflows that may exist for the organization, such as a purchasing process or examination script process, except that their nodes may link to metadata about the concepts being dealt with at this point of the workflow. Figure 6 shows a workflow for managing examination scripts in our university. Many nodes have concepts attached to them (we have only shown some of them so as not to clutter the diagram). These concepts are depicted with a dashed box, and can be associated with several nodes across different workflows. For instance, the concept "Large scale copying procedures" (present for the node "Large scale copy" in this workflow) is actually also relevant to two other workflows: *Stationery* and *General teaching*. If the user has already seen these procedures in the context of either of the two other workflows (as indicated in the user model), then the procedures will not be suggested again to the user. However, if the user has not seen them, they will be pushed to the user.

The organisational documents are tagged and indexed with metadata representing the concepts explained in the documents. For instance, a document called "Seminar on preparing examinations" contains the concepts "exam aligned with scope and objectives," "setting exam questions," "write exam question answers," "develop marking scheme" and "common exam problems."

The same metadata is also used to tag the various stages of each workflow. So, for example, the node in the previous figure called "Write exam" contains the concepts "Confidential labelling of exam papers," "Setting exam questions" and so on (this node actually corresponds to a smaller grained workflow, so it contains many concepts). When the user reaches the node "Write exam," the teaching agent may decide that the document "Sem-

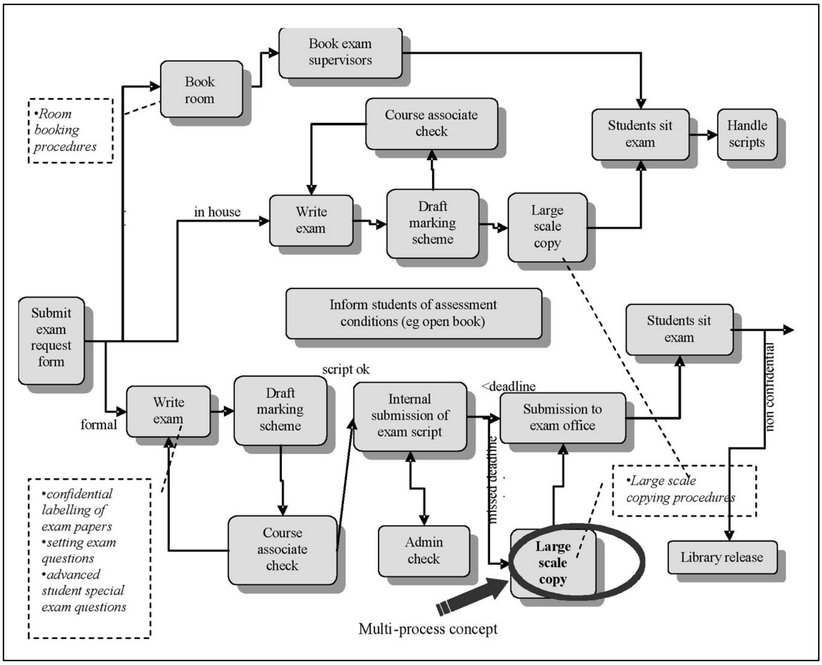

Figure 6. Workflow for exam preparation

inar on preparing examinations" should be given to the user.

The role of the activity engine is to encode the workflows and establish the correspondence between stages of the workflows and the documents that need to be known for each stage. However, the decision about exactly what documents should be shown to the current user is made by the teaching agent.

Workflows are represented using XML files defining the states, the transitions between states and the concepts that need to be known in order to do the work of each state.

USER MODEL

The user model holds the user profile data as well as the student model. For example, it holds profile details such as the user's name and position in the organization. The student model holds the learning concepts. For the workflow described above, these include knowledge of how to organise an in-house examination, a formal examination, as well as the associated concepts such as how to do large-scale copying, how to write an examination, and how to draft a marking scheme.

The user model is implemented using the *accretion-resolution* representation (Kay, 1995; Kay, 2000). This is based on the same approach as the

Personis user model server (Kay, Kummerfeld, & Lauder, 2002). Essentially, this *accretes* all evidence about the user's knowledge of a concept and then uses a *resolver* to determine the value of each component as it is needed. This resolution process is very flexible and the design of the system allows for multiple resolvers with the user being able to alter the choice of resolver. This means that the user can maintain some level of control over the way that their student model is interpreted by JITT.

The student model holds the details of the currently active workflows for this user. Each workflow has a set of associated concepts. The student model represents the user's knowledge of each of the concepts for that workflow. When a workflow is initiated for a user, the concepts associated with it in the student model can be initialised to a stereotypical set of values. As in typical uses of stereotypes (Rich, 1979; Rich 1983), these operate as default assumptions which are intended to be overridden once more reliable information becomes available.

The current JITT prototype has two sources of evidence about the user's knowledge. The first applies each time the user accesses a document from the JITT page. This access causes a *tell* operation to the user model. This form of evidence is distinguished in the student model. Our resolvers treat this as low grade evidence for the user knowing the concept. The second form of evidence is provided directly by the user. At the end of each document, the user is invited to select a popup option indicating how well they have learned each of the concepts in that document. Recall that JITT has been designed to operate with existing documents. This means that one document may have several concepts. The user may feel that they understand some better than others. This will happen most often when there are multiple concepts explained in a document. In this case, it may be the best document available about a concept that is relevant to the user at a particular time. It may also include other content that is not relevant to the current learning goal. The user may skim the irrelevant parts. In such cases, the user can indicate their increased knowledge level for the concepts that they have learned from the document. For others, they can just do nothing (indicating no change in their knowledge) or rate their knowledge. At any time, the user can access their user model and see all the concepts that are relevant to the currently active workflows. The user can also see the system's estimate of how well they know each of these concepts. They can also check the exact evidence used to determine that value and the way its value was resolved.

TEACHING AGENT

One of the advantages of the way we use workflow technology in this just-in-time training system is that the scheduling of documents within a specific workflow process is given precisely by a workflow. That way the workflow provides the basis of the curriculum of the training process. This means that as the learner reaches a particular task in the workflow, that state defines the concepts that they need to know and, in turn, this defines the documents that the learner should read. The heart of JITT is the teaching agent.

The tasks of the teaching agent are to:

- maintain the user's state in relation to each of the workflows that are active for them;
- identify the concepts the learner needs to know for the current state in the workflow;
- find the documents which can teach the concepts relevant to the current step in the workflow;
- select the best of these for the user, based on user model details of preferences;
- customise the presentation of the links to the documents at the user interface.

Let us return to the "large scale copy" example given with the context of the exam preparation workflow (Figure 6). We suppose that the user has reached this stage in the workflow. This means that the teaching agent models this as the user's current state in that workflow. The teaching agent then identifies all the concepts that the user must understand in order to complete this task. In general, for workplace training we would expect that there would only be a small number of such concepts.

Clearly it is critical to model these concepts at the appropriate level of granularity. If for example, we take an excessively fine-grained concept model, we will introduce needless complexity with many concepts always occurring together. In practice, we believe that this problem is naturally addressed when the teacher who creates the metadata chooses the concepts to be as coarse grained as possible. At the same time, it is important to avoid modeling in terms of concepts at too coarse a grain-size. If we did this, it would be impossible to reuse concepts across workflows. This might appear to be a difficult problem. Indeed, in the general case of teaching systems, it can be (McCalla, & Greer, 1994). However, in JITT, the nature of the learning and the use of workflows to define the granularity of learning tasks makes it easier for the teacher to define concepts (and hence metadata) at the right grain size.

Once the teaching agent has identified the concepts to be learned for the "large scale copy," it searches its document collection for those that teach the "large scale copy" concept. In theory, there might be several documents, perhaps a detailed one for new staff, a shorter one for people who partially know this concept, or a document that actually directs the user to a suitable person in the organization. The design of JITT and its underlying architecture is intended to exploit existing document bases within organizations. Certainly there is a significant problem in helping employees to access these at the right time.

Given our approach, the choice of documents is limited to those we are able to find in the organisation. We can expect that this will generally constitute a small set that teaches any one concept. This means that the teacher has only to invest modest amounts of time in marking up documents and it also means that, in practice, JITT tends to have few documents to choose from. If there is a choice of documents, the teaching agent needs to select from these on the basis of the user model combined with the metadata available for each document.

At the push interface of JITT, the teaching agent presents links to the documents it recommends for the current stage of each active workflow. In our current implementation, we customise the presentation by using the architectural feature that allows multiple teaching agents.

For example, one of these only recommends documents that the user is modelled as not knowing. If the JITT model for the user indicates that the user knows all the concepts for the current stage in the workflow, it continues to show the user that stage but does not display any documents. This should be a preferred presentation strategy for users who like JITT as reminder of the overall processes they need to follow, but do not want to see document references where the user knows the concepts well. We designed this presentation for the case of the long term employee who knows the workings of the organisation well. This person still needs assistance in remembering processes that they do not often perform. The important thing is that if the organization does alter the procedures, this must be handled by defining a new concept. In our example, suppose that the "large scale copy" task requires understanding of the concept "authorization to do large scale copying." Suppose further, that after some period of time this concept is changed, perhaps because the earlier mechanism is unworkable. The organization would document this change in a new document (or a new version of an old document). The teacher in the JITT system would create a new concept to represent this new procedure. This concept would be added to the "large scale copy" task in each workflow and when a long term employee next came to this stage of the workflow, JITT would represent them as not knowing this new concept. They would be provided with the new document. Importantly, this long-term employee would see that the JITT was recommending a document – something it normally did not do.

Another teaching agent employs a different presentation strategy. It continues to show the user all documents associated with the current task. However, in the case where the user is deemed to know the concepts associated with a concept, it uses a very small font for the link to the document teaching it. We designed this as the interface that we expected most people to prefer.

Importantly, we have designed JITT so that the user can always change the teaching agent to select the presentation strategy they prefer. This functionality is available at the user profile part of the interface. As described in the interface overview, JITT also maintains two lists of documents the user has accessed. One contains those documents the user has explicitly put aside and the other has the documents the user has accessed in reverse chronological order of access. The teaching agent constructs these lists and controls the associated interface elements. The JITT architecture makes it straightforward to create new teaching agents that provide variants on this functionality. For example, suppose there were users who preferred that neither of these lists were displayed because they preferred more screen area for the lists of active workflows and associated recommended documents. It is easy to add a new teaching agent with this variant. In organisations, this type of flexibility may be very important since even small deviations from a user's preferred interface may significantly detract from the acceptability of the whole JITT system.

The JITT teaching agents are based upon the Scrutable Adaptive Teaching System approach (Holden, & Kay, 1999; Holden, & Kay, 2001; Lum, Holden, & Kay, 2002).

RELATED WORK

Our research lies at the intersection of workflow systems, user modelling, information delivery and teaching agents. In this section, we briefly relate our work to other projects using these four dimensions.

Workflow

We use workflows to model the tasks within the whole process the user has to learn to do. Another project that has applied workflow as its driving mechanism is Flex-EL (Lin, Ho, Sadiq, & Orlowska, 2001). Their workflow was deployed in a university course environment, whereas the goal of the users in JITT was to accomplish a certain business activity. Although workflow also played an important role in the investigation in the project of Zhao, Kumar and Stohr (Zhao, Kumar, & Stohr, 2001), the goal of the workflow was to deliver information, whereas information delivery is not the goal of our workflow but only a supporting activity to empower a user to carry out the tasks. Many other projects that have not used workflow explicitly, have employed alternative ways to model tasks. For example, Paris, Linden, Colineau and Lu (2004) generated online help from task models. Their task model is coded in XML and basically contains how-to instructions. JITT is somewhat similar in that the information is associated with the intermediate steps in a process. However, their tasks are comparatively simpler and short-lived while the workflow in JITT typically operates over a more lengthy time span. To support opportunistic learning in an organization, Alem and McLean (2004) had explicit representation of the business processes. They associated the documents, electronic discussions between people, their roles and responsibilities in a group/project memory.

User modelling

One of the crucial components in our system is to model the user's knowledge of the concepts needed to do the tasks in the workflow. A lot of the other projects use concept hierarchies or a domain ontology to constrain the operation of their systems. The hyperbook introduced by Falquet and Ziswiler (2004) is comprised of a re-usable document repository that is related to a domain ontology. The concepts in the ontology are defined through a graph-based description logic language. The definition of a concept, according to a point of view, is obtained by selecting those arcs that belong to the desired point of view or to a more general point of view. Their approach is based upon points of view that correspond to a category of users or point of view adopted by a user. A document in our system is linked to the concept(s) that are associated with different activities in a business process; hence, our approach is workflow-driven instead of knowledge-driven. Our system uses a separate user model to adapt to individual needs, rather than using the same point of view representation as in the hyperbook. The know-how concepts in the online help generation system (Paris, Linden, Colineau, & Lu, 2004) were specified in task models. Their system *creates* new instructions by combining existing knowledge whereas we explicitly make use of existing document. Mittal, Dixit, Maheshwari and Sung (2004) organised the inter-relationship among concepts in a relational graph. They built a rule-based system to identify and relate the learning objects to these concepts. They claimed that providing the conceptual graph to the users would enhance their understanding and their learning, and it was also more efficient to search for a desired topic.

Interface

In our system we provided push and pull interfaces to existing document bases. This contrasts with, for example, systems which operate purely from information requests initiated by the user in (Alem, & McLean, 2004), (i.e., it is a pull-type of delivery only). A similar approach to work, such as Zhao, Kumar and Stohr (2001), delivered information by email, though the users can also “pull” interesting news from an associated bulletin board. In other words, although that work involves different interface modalities, it too has forms of both the *push* and *pull* mode of information delivery available in JITT. The hyperbook by Falquet and Ziswiler (2004) provided an interface with the specified points of view so that a user can select (pull) the one that suits his/her needs.

Teaching agents

We provided multiple teaching agents, each able to support different teaching strategies and information presentation strategies. The point of view in Falquet and Ziswiler's hyperbook (Falquet, & Ziswiler, 2004) provides a similar effect. A view corresponds to a category of users or point of view adopted by a user, but it is not necessarily a teaching strategy. JITT has a different representation to store the models for different users, rather than using the same representation (point of view) as in hyperbook.

CONCLUSION

This paper has described the architecture of JITT, a Just-In-Time Training system which builds upon workflow technology to support organisational learning. In particular, the aims of JITT are to reduce the information overload faced by newcomers by scheduling the delivery of the right documents at the right time for them.

JITT can be implemented as an extra layer on an existing workflow system if one exists. However, whilst workflow management systems can be quite complex and strict in their use, JITT can make use of an arbitrary abstract workflow, defined to distinguish the sub-activities a user needs to be able to complete in order to progress through a complete activity.

We have built a prototype of the system and carried out a careful qualitative evaluation based upon think-aloud protocols and a questionnaire administered to collect affective data. A field trial that ran over several months would be needed for a meaningful assessment of the workplace utility of JITT. The evaluation was designed to be completed in a single session, making modest demands on the time of the participants. The evaluation involved ten participants (staff and students in our university) for the task of preparing a first exam (Holden, Kay, Poon, & Yacef, 2004). This demonstrated the usability of the system and generally positive user response to its value. Users commented that the system was useful, though some added that this obviously depended on the quality of the documents available. They found that having the workflow was useful as it provides an outline of the steps required and the order in which they should be done. Overall, the users thought that the system would also be useful to jog their memory as they worked through the process for later exams as well as for learning how to write the first one. JITT represents a novel combination of personalisation and tutoring controlled by a user-selectable teaching agent, with the sequencing of the teaching being defined by an organisational workflow. JITT has a novel level of user control over the personalisation, enabling the user to alter the teaching processes. It takes workflow approaches beyond the business processes, linking them to the learning processes that an employee must undertake in order to work through the various activities in their workplace. Moreover, it operates within the existing organizational framework in that it makes use of existing documents.

References

Alem, L., & McLean, A. (2004, submitted). Supporting organisational learning with technologies for electronic documents. *IJEL* (Special issue on Technologies for Electronic Document for Learning).

Budzik, J., Hammond, K. J., & Birnbaum, L. (2001). Information access in context. *Knowledge Based Systems, 14*: 37-53.

Davis, J., Kay, J., Lin, K., Poon, J., Quigley, A., Saunders, G., & Yacef, K. (2002). Workflow based just-in-time training. In J. T. Judy Kay (Eds) *Seventh Australasian Document Computing Symposium (ADCS 2002)*, Sydney, Australia.

Dieng, R., Corby, O., Giboin, A., Golebiowska, J., Matta, N., & Ribière, M. (2000). *Méthodes et outils pour la gestion des connaissances.* Paris, Dunod.

Falquet, G., & Ziswiler, J.-C. (2004, submitted). A virtual hyperbooks model to support collaborative learning. *IJEL* (Special issue on Technologies for Electronic Document for Learning).

Holden, S., & Kay, J. (1999). The scrutable user model and beyond. In R. Morales (Eds), *Proceedings of Open, interactive, and other overt approaches to learner modelling, held in conjunction with International Conference for Artificial Intelligence in Education,* Le Mans, France, 51 - 62.

Holden, S. & Kay, J. (2001). Document selection in SITS. In A.-M. Vercoustre & J. Kay (Eds) ADCS'2001, *Australian Document Computing Symposium,* 79 - 82.

Holden, S., Kay, J., Poon, J., & Yacef, K. (2004). JITT, Just-In-Time Training system technical report. Sydney, University of Sydney: in progress.

Hoyt, B. R. (2002). Facilitating agreement for effective knowledge management systems: The abilene paradox is thriving in the new millenium. *International Journal on E-Learning,* July-September: 15-20.

Kay, J. (1995). The UM toolkit for cooperative user modelling. *User Modeling and User-Adapted Interaction, Kluwer 4*(3): 149-196.

Kay, J. (2000). Accretion representation for scrutable student modelling. In G. Gauthier, C. Frasson, K. VanLehn (Eds) *Intelligent Tutoring Systems (ITS'2000)*, Springer-Verlag, 514-523.

Kay, J., Kummerfeld, R. J., & Lauder, P. (2002). Personis: A server for user models. In P. De Bra, P. Brusilovsky, & R. Conejo (Eds) *Adaptive Hypertext (AH'2002)*, Malaga, Springer Verlag, 203 -212.

Lin, J., Ho, C., Sadiq, W. & Orlowska, M. (2001). On work enabled e-learning services. In (Eds) *IEEE International Conference on Advanced Learning Technologies (ICELT'2001)*, Madison, USA.

Lum, A., Holden, S., & Kay, J. (2002). Visualisation of document and concept spaces. In J. Kay & J. Thom (Eds) *ADCS2002, Australian Document Computing Symposium*, Sydney, Australia.

McCalla, G. I., & Greer, J. E. (1994). Granularity-based reasoning and belief revision in student models. In J. E. Greer & G. I. McCalla (Eds) *Student Modelling: The Key to Individualized Knowledge-Based Instruction*. Springer-Verlag: 39-62.

Mittal, A., Dixit, S., Maheshwari, L. K., & Sung, W.-K. (2004, submitted). Enhanced understanding and retrieval of e-learning documents through relational and conceptual graphs. *IJEL* (Special issue on Technologies for Electronic Document for Learning).

Nonaka, I., & Takeuchi, H. (1995). *The knowledge creating company: How Japanese companies create the dynamic of innovation.* New York, NY, Oxford University Press.

Paradis, F., Crimmins, F., & Ozkan, N. (2003). A task oriented approach to delivery in mobile environments. In (Eds) *4th International Conference on Mobile Data Management.*

Paris, C., Linden, K. V., Colineau, N., & Lu, S. (2004, submitted). Producing Instructions. *IJEL* (Special issue on Technologies for Electronic Document for Learning).

Rich, E. (1979). User modeling via stereotypes. *Cognitive Science, 3*(4): 329-354.

Rich, E. (1983). Users are individuals: Individualizing user models. *International Journal of Man-Machine Studies*, 18: 199-214.

Sweller, J. (1993). Some cognitive processes and their consequences for the organisation and presentation of information. *Australian Journal of Psychology, 45*(1): 1-8.

Zhao, J. L., Kumar, A., & Stohr, E. (2001). Workflow-centric information distribution through email. *Journal of Management Information Systems, 17*(3): 45 - 73.

Acknowledgments

This one-year foundation project was funded in part by the Cooperative Research Centre Smart Internet Program through the Department of the Prime Minister and Cabinet of the Commonwealth Government of Australia. We are grateful to Bob Kummerfeld for designing the interface of JITT and implementing its earlier prototype.

A Fully Automatic Question-Answering System for Intelligent Search in E-Learning Documents

PRAVEEN KUMAR, SHRIKANT KASHYAP,
ANKUSH MITTAL, AND SUMIT GUPTA
Indian Institute of Technology, India
pkmaxuec@iitr.ernet.in
shrikuec@iitr.ernet.in
ankumfec@iitr.ernet.in
sumitfec@iitr.ernet.in

E-learning is a novel method for presenting information to students for the purpose of education. Currently a sea of information is available in the form of PowerPoint slides, FAQs and e-books. However the potential of this large body of information remains unrealized due to lack of an effective information retrieval system. Current search engines are used only for the web and return ranked lists of documents. Such engines would not be effective searching tools for e-learning documents and it would be difficult for a user to find the intended answer. This article introduces a fully automatic Question-Answering (QA) System that allows students to ask a question in common language and receive an answer quickly and succinctly, with sufficient context to validate the answer. The system uses Natural Language Processing (NLP) techniques to identify the semantic and syntactic structure of the question. It configures itself to a particular domain by automatically recognizing the entities from the course material. The information retrieval engine is used to extract answer passages using contextual information. A closed loop dialogue with the user leads to effective answer extraction through extensive passage analysis. Experimental results of the system are shown over the course material of Computer Networks.

E-learning has underlined the importance of quick access to relevant study material for effective education with the major advantage of enabling people to access learning facilities regardless of their location and at the time that is most convenient to them. Business enterprises are widely using this online learning for employee training and education because of its cost saving advantages, especially with respect to time and travel parameters (Dorai, Kermani, & Stewart, 2001). Currently a sea of information is available in the form of PowerPoint slides, digital text and FAQs. However, a lot of time is spent on e-learning by users searching for a desired concept or answer from this huge repository of information.

To fill in this gap, an effective Question-Answering (QA) system is required that can retrieve answers to students' questions from the course material, suggest alternatives in case of any ambiguity in the question and thus help them to search for the intended answer. Examples of such interactive closed-loop QA systems developed are HITIQA (Small, Liu, Shimizu, & Strzalkowski, 2003) and SPIQA (Hori, Hori, Isozaki, Maeda, Katagiri, & Furui, 2003). The rapid success of distance education has led to extensive development of course material and its placement on web. A learner does not understand and knows where he can find the related terms and concepts mentioned in the lecture. Searching for topics through table-of-contents or index pages can be tedious and impractical due to a large volume of information present in these domains. For instance, the user wants to know which algorithms sort an array in a particular time complexity (i.e., O(nlogn)). Since such algorithms are distributed throughout the book (like BinarySearch, Mergesort, Binsort, Radixsort, MinHeap sort, etc.), table-of-contents or index pages cannot provide the user much information and he has to search through the entire book.

Modern search engines (such as Google) are able to cope with the amount of text available. They are most useful when a user presents a query to the search engine which only returns a couple of documents of which the user can then manually search to find the relevant information. Such engines would not be effective searching tools for e-learning documents and it would be difficult for a learner to find the intended answer from the list of retrieved documents. Searching for a particular concept by keyword or phrase matching is insufficient because in many cases (i.e., for the question, "What is the difference between RIP and BGP protocol?") words like "difference" may not be present; instead, words like "compare" or "contrast" can be there. In other cases like, "Give the time complexity of Mergesort," some semantically related terms like "asymptotic" or "Big O notation" have to be identified.

The approach taken here is to implement a QA system based on searching in context and *entities* of a domain for effective extraction of answers to even domain specific questions. The system recognizes the entities by searching from the course material. It is fully automatic as it does not require any manual intervention for configuring it to any particular domain. The focus is on context based retrieval of information. For this purpose a *retrieval engine* that works on locality-based similarity heuristics is used to retrieve relevant passages from the collection, (i.e., passages that can potentially answer the question). During *query formulation* and *expansion*, the system tries to make judicious interpretation in order to tap the semantics of question. The system utilizes natural-language parsers and heuristics in order to return high-quality answers. This system can be used to serve as a first step towards automatic FAQs. It has good utility for a novice in a subject who does not know where to find related terms and concepts. It can also be quite helpful to students just before their exams for getting answers to review questions.

Contribution of the Article

The following are the contributions of the article:

- Automatic Entity Recognition: The system is not restricted to only one domain. It is fully automatic as it learns about the domain by recognizing the entities from the course material. Manual development of structured data or annotations (as commonly used in other systems) is not required.
- Integration of Alternative Resources: Different e-learning documents like scanned books and PowerPoint slides have different information and presentation methods. Books are illustrative and give detailed analysis of concepts. Slides are condensed, highlighting the key points. Moreover the collection of material may be comprised of books and slides of different authors and teachers (who present the subject in different styles and concepts). The system tries to integrate information from different types of documents and present the summarised answer to the user.
- The system's ability to recognize the context of the problem by using locality based similarity heuristics and *query expansion* (with the help of WorldNet).
- Closed loop Q&A: The user is provided with a feedback of related keywords which can help the user to reframe a relevant question (within the limits of e-learning materials) and extract the answer from the system.

Organization of the Article

The rest of the article is organized as follows. The Literature Review and Background section gives an account of related work in e-learning and provides background on Question-Answering. The QA System section describes the different components of this QA system in detail. The Results section provides the results of the experiments and the method adopted to test the system's utility. Conclusions and future work follow these sections.

LITERATURE REVIEW AND BACKGROUND

Related work in E-learning

Efforts have been made in the direction of providing ease to the student in extracting information from e-learning documents with respect to effective retrieval and presentation of knowledge. A similar system COVA (on content-based retrieval) enables remote users to access specific parts of interest from a large lecture database by contents (Cha, 2002). However, manual development of XML schemas or annotating the vast amount of information can be laborious and impractical. Another approach introduces Genetic Algorithms into a traditional QA system which uses the concept of Case-Based Reasoning (CBR) (Fu, & Shen, 2004). The huge number of cases that would be generated with large repository (with continual growth) and failure in case of complex queries put limitation in its practical use.

A different approach taken in knowledge-based content navigation in e-learning applications presents a prototype implementation of the framework for semantic browsing of a test collection of RFC documents (Mendes, Martinez & Sacks, 2002). They propose the use of fuzzy clustering algorithms to discover knowledge domains and represent those knowledge domains using *TopicMaps*. However, success largely depends on how accurately the clusters are identified and the representation still suffers from the drawback attributed to table-of-contents page.

E-learning Media Navigator (ELM-N) from IBM Research is a system with which a user can access and interact with online heterogeneous course materials (Dorai, Kermani, & Stewart, 2001). Their efforts are aimed to reduce human effort and manual annotation work in order to make the system viable for voluminous information. Furthermore, challenges remain in the area of easy-to-use content delivery, access and augmented interaction.

Background on Question Answering

A QA system provides direct answers to user questions by consulting its knowledge base. It attempts to allow the user to ask questions in natural language and receive an answer quickly and succinctly, with sufficient context to validate answer (Hirschman, & Gaizauskas, 2001). Some QA systems that cater to a specific domain have been developed at very early stage. LUNAR (Woods, 1973) was such a closed domain QA system that it only answered questions related to moon rocks and soil gathered by the Apollo 11 mission. However, it relied on having the data to be available in a highly structured form and not as completely unstructured text.

The availability of huge document collections (for example, the web itself), combined with improvements in information retrieval (IR) and Natural Language Processing (NLP) techniques, has attracted the development of a special class of QA systems that answers natural language questions by consulting a repository of documents (Cody, Oren, & Daniel, 2001). Most of the QA systems that have been developed treat the web as a collection of documents and thus cater to huge variety of questions. One of the commercial search engines known as AskJeeves responds to natural language questions, but its recall is very limited because the search engine uses its knowledge base (which is at least partially hand constructed) to answer questions and to update the knowledge base when asked a question which it has not encountered before.

Another QA system, MULDER (Kwok et al., 2001) is claimed to be the first general-purpose, fully-automated question-answering system available on the web. MULDER's architecture, relies on multiple search-engine queries, natural-language parsing, and a novel voting procedure to yield reliable answers (with a recall of the same level as that of Google). However, the difficulty of NLP has limited their ability to give accurate answers to questions that are quite specific to a domain. In addition to the traditional difficulties associated with syntactic analysis, there remains many other problems to be solved, (e.g., semantic interpretation, ambiguity resolution, discourse modelling, inference, common sense, etc.).

QA systems on the web try to answer questions that require a fact or one word answer. This is difficult for questions that are specific to a domain because the targeted domain is unrestricted and no assumption can be judiciously made. E-learning questions are more complex than TREC-type questions as they require domain knowledge and long answers need to be extracted from multiple documents. Moreover these questions have inherent ambiguity. The objective is to allow the user to submit exploratory, analytical, non-factual questions such as, "*How does Mergesort sort an array*?" The distinguishing property of such questions is that one cannot generally anticipate what might constitute the answer. While certain types of things may be expected, the answer is heavily conditioned by what information is available on the topic. Users generally prefer answers embedded in context, regardless of the perceived reliability of the source documents (Lin, Quan, Sinha, Bakshi, Huynh, Katz, & Karger, 2003). When users search for a topic, an increased amount of text returned significantly decreases the number of queries that they pose to the system.

The QA System

Figure 1 shows the architecture of our QA system. The user begins by configuring the system to the particular course domain by triggering the *Automatic Entity Generator* module which recognizes domain specific *entities* from that particular course's documents. The question submitted by the user is classified in *Question Classification* to identify its case. The question is parsed using the *Link Parser* which constructs the linkage structure of the question. This information is used for extracting relevant information (like part of speech) during *Question Parsing*. Subsequently, *Query Formulation* translates the question into a set of queries that are given as keyword input to the *Retrieval Engine*. *Query Expansion* is needed to tap the semantic of the question and improve the answer extraction. The engine returns top passages after weighting and ranking them on basis of locality. Finally, *Answer*

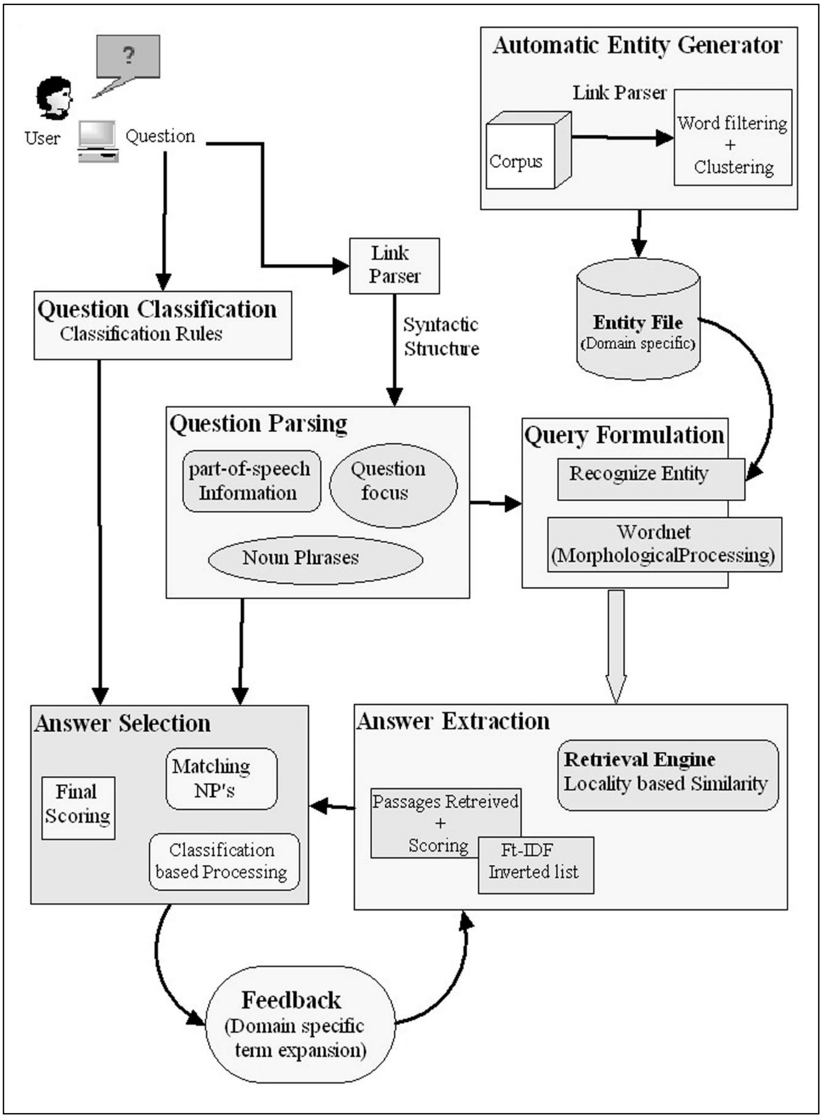

Figure 1. Architecture of the system

Selection is done by further extensive passage analysis, and is then presented to the user. To improve answers (if the user is not satisfied) the system takes user feedback which is again followed by answer extraction and selection. Each part is described in detail in the next section.

Question Classification

The *Question Classifier* used pattern matching based on *wh*-words and simple information to determine question types. The questions were broadly classified into the following categories:

- Questions containing the keywords such as 'various,' 'ways,' 'difference,' 'types,' and 'compare.' These keywords require answers to be extracted from more than one passage. For example, "What are the various algorithms for sorting an array in O (nlogn) time complexity?" or, "What is the difference between RIP and BGP?" Normally, answers to such questions need to be extracted from several passages.
- Questions that ask for numerical data or date. Such questions were identified by a *wh-phrase* (*"How many?", "How tall?", "When?"*). The answer passages must focus on numerical data.
- Questions that can be answered from one passage. The *Question Focus* (object of the verb) is used to find the relevant answer.

Question Parsing

Usually search engines use keywords from the question to construct queries neglecting unimportant words like 'of,' 'for,' 'at,' etc. No importance is given to the syntactic structure of the question while picking up keywords. In such cases the meaning of the question is lost. For example, no difference exists among the questions 'how,' 'why,' or 'what.' This QA system uses Link Grammar Parser to parse the question in order to determine its syntactic structure. This structure is then used to extract part-of-speech information. The *Question Focus* is identified by finding the object of the verb. Also, *Noun Phrases* are identified to tap the semantic structure of the question. This information is used to select plausible answers from the e-learning materials.

The *Link Parser* is a syntactic parser of English, based on link grammar, an original theory of English syntax. Given a sentence, the system assigns to it a syntactic structure, which consists of a set of labelled links connecting pairs of words. (Temperley, Sleator, & Lafferty, 1993). The parser has a dictionary of about 60,000 word forms. It has coverage of a wide variety of syntactic constructions, including many rare and idiomatic ones. The parser is robust; it is able to skip over portions of the sentence that it cannot understand, and assign some structure to the rest of the sentence. It is able to handle unknown vocabulary, and make intelligent guesses from context and spelling about the syntactic categories of unknown words. It has knowledge of capitalization, numerical expressions, and a variety of punctuation symbols.

The *Link Parser* works as follows. The dictionary of nouns, verbs, adverbs, prepositions and adjectives is used to parse a sentence. The parser starts at the right end and searches linkages throughout the sentence. It considers each entry for the word as a different word and generates all linkages found for all entries. This parser considers relationships between pairs of words. For example, in the sentence shown in Figure 2 there is an S (subject) relation between "Internet" and "is," and a D (determiner) relation between "a" and "network."

The requirements, like parts of speech, syntactic functions and constituents, can be recovered from the link structure rather easily. For example, whatever word is on the left end of an S-link is the subject of a clause (or the head word of the subject phrase); whatever is on the right end is the finite verb; whatever is on the left-end of a D-link is a determiner, etc. The system finds the question focus by using the S or O linkage to get the object of the verb. Importance is given to question focus by assigning it more weightage during retrieval of answers. Moreover, all nouns, verbs, and adjectives in the dictionary are subscripted (as ".n," ".v," or ".a"), so in these cases the syntactic category of the word is made explicit.

The constituent structure of sentences, while not absolutely explicit, is also quite close to the surface in linkage structures. Constituents can be defined as sets of words which can be reached from certain links, tracing in a certain direction. For example, a verb phrase is everything reachable from an S-link, tracing to the right – that is, not tracing through the left end of the S-link itself. For noun phrases there are several possibilities. Anything that can be reached from an O-link by tracing right is an NP (noun phrase). The system tries to find all possible NP in the question. For example, the following NPs were found in the question, "Why are buffers needed at the output port of a switch?" – [buffers] , [the output port of a switch], [the output port], [a switch].

Automatic Entity Recognition

This module tries to recognize the *entities* in a particular course (domain specific entities) to which the user wants to pose questions. This configures the system automatically to any type of course domain. The system admin-

```
         +---------------Ost-------------------+
         | +-------------Ds-------------------+
         | |        +---------A---------------+
+--Ss--+ |        |          +------AN------+
|      | |        |          |              |
Internet is.v a worldwide.a computer.n network.n
```

Figure 2. The Linkage structure formed by Link Parser

istrator on the server providing distance learning (or the user who wants to search answer from documents present in his local system) gives an index or table of contents file as input. The module runs *Link Parser* on every line giving its syntactic structure. It takes nouns, adjectives and verbs (ending with *ing*) as entities (as they carry the focus of the sentence). In the absence of table of contents or index pages, the system searches through the main heading and sub headings of slides or digital text for recognizing the entities. If no linkage is formed it tokenizes the string and *word filtering* is done to remove any elementary words (as shown in Table 1). If no elementary words are found in the string then the whole string is also taken as an entity (for example, *Binary Search Tree*). The output is stored in the *Entity File* for subsequent use. This file contains domain specific entities.

Query Formulation

The query formulation module converts the user's question into a set of keywords (query) which is then sent to the retrieval engine for answer extraction. The system uses the entity file to recognize the domain specific entities in the question. During initialization, the system reads from default file (which can be set to a particular course by the user) and constructs a hash table of these entities. Individual words in the question are compared from this table to identify the entities. These keywords are considered most important and are given the maximum weightage of 2.

The question focus (object of the verb) identified during question parsing is also given the same weightage of 2. Elementary words (as shown in Table 1) are given the weightage 0. The rest of the words in the question are given the weightage 1.

Query Expansion: Extending the query through query expansion enhances the search process by including semantically related terms and thus retrieves texts in which the query terms do not specifically appear (Gonzalo, Verdejo,

Table 1
Examples of removed words

Words Removed			
By	Is	So	As
To	Otherwise	The	Will
An	In	For	Of
Does	At	Are	Did
Be	Over	We	Our

Chugur, & Cigarran, 1998). For example, in questions like, "Compare and contrast link state and distance vector routing algorithm," the answers may occur in sentences such as "The difference between..." The system uses a popular thesaurus called WordNet to identify semantically related concepts. WordNet is a semantic network containing words grouped into sets called *synsets.* Synsets are linked to each other by different relations such as synonyms, hypernyms and meronyms. For nouns, the most common and useful relation is the *is–a* relation. This exists between two concepts when one concept *is–a–kind–of* another concept. Such a concept is also known as a *hypernym*. For example, a *computer* is a hypernym of *machine*. This creates a network where related concepts can be identified (to some extent) by their relative distance from each other.

Only those query terms were expanded which do not occur as domain entities. Gaining from this knowledge, query evaluation is no longer restrained to query terms submitted by users but may also embody synonymous or semantically related terms. However, caution is taken as these newly found terms are not as reliable as the initial terms obtained from users. Only closely related terms are taken that have direct relation with either the query term itself or with the words that are directly related to the query term. An appropriate weighting (0.5) scheme allows a smooth integration of these related terms by reducing their influence over the query.

Answer Extraction

To extract passages from the collection of documents an Information Retrieval engine is needed to analyse the keywords and passages in detail. The answers to a query are locations in the text where there is local similarity to the query, and similarity is assessed by a mechanism that employs as one of its parameters the distance between words (Kretser, & Moffat, 1999). For this purpose it was found that the locality-based similarity heuristic (in which every word location in each document is scored) provides retrieval effectiveness as good as the document- based technique, and has the additional advantage of presenting focussed answer passages (instead of the whole document) with sufficient context to validate the answer. Therefore, the engine used is based on this concept and has been customized for this application.

The important features of Locality-Based Retrieval (with Similarity) in this context are:

- The focus is on local context by considering top n ranked passages, instead of the top n documents.
- Each term has a certain scope, where its importance decreases with respect to the distance from that term.
- Similarity is computed as the sum of weighted overlaps between terms. It is based on intuitive notion that the distance between terms is indicative of some semantics of the sentence.

The entire retrieval process is carried out using a word-level inverted index using all of the terms in the automatically generated query. An example of a construction of word-level inverted page list is shown in Figure 3. The drawback of the seamless approach is that more index information must be manipulated and that querying requires more resources, but with the use of appropriate techniques these costs are manageable. Using this fully automatic mechanism, results as good as or better than comparable document-

Example Text

Document	Text
1	TCP (the Transmission Control Protocol) and IP (the Internet Protocol) are two of the most important protocols in the Internet
2	TCP provides reliable data transfer, flow control and congestion control.
3	IP is said to be an unreliable service
4	TCP congestion control prevents any one TCP connection from swamping the links

A word Level Inverted Page List

Number	Words	Position (Document : Word)
1	TCP	(1:1), (2:1), (4:1), (4:7)
2	IP	(1:7), (3:1)
3	Transmission	(1:3)
4	Control	(1:4), (2:7), (3:10)
5	Protocol	(1:5), (1:10), (1:17),
6	Internet	(1:9), (1:20)
7	data	(2:4)
8	Flow	(2:6)
9	Congestion	(2:9), (4:2)
10	Connection	(4:8)
11	Unreliable	(3:7)
12	Prevents	(4:4)
13	Links	(4:12)
14	Swamping	(4:10)
15	Reliable	(2:3)
16	transfer	(2:5)

Figure 3. An example of the construction of a word-level inverted page list. The Example Text Table shows some sample texts from three different documents and the Word-Level Inverted Page list shows the corresponding page list.

based retrieval techniques, and are obtained within relatively modest resource requirements.

Rather than considering the text collection to be a sequence of documents, it is considered to be a sequence of words, and query term occurrences within the collection are presumed to exert an influence over a neighbourhood of nearby words. Then, supposing that the influence from separate query terms is additive, the contribution of each occurrence of each query term is summed to arrive at a similarity score for any particular location in any document in the collection. This concept is illustrated in Figure 4.

The contribution function $\mathbf{c_t}$ is then defined in terms of **l**, the location of the query term (as an integral word number); **x**, the word location at which we seek to calculate a contribution; $\mathbf{h_t}$, the peak height assigned to the term, assumed to occur at the word position occupied by the term in question; and $\mathbf{s_t}$, the one-sided spread of the term. The parameters that are used for scoring the passages are:

- **N**: Total number of terms in the collection
- Term frequency ($\mathbf{f_t}$): How often the term t appears
- $\mathbf{F_{q,t}}$: Within query frequency of the term
- Inverse document frequency (*idf*): log (N/ f_t)
- Height (h_t): The height assigned to a term t is a monotonic function of the term's scarcity in the collection.

$$h_t = F_{q,t} * \log (N/ f_t)$$

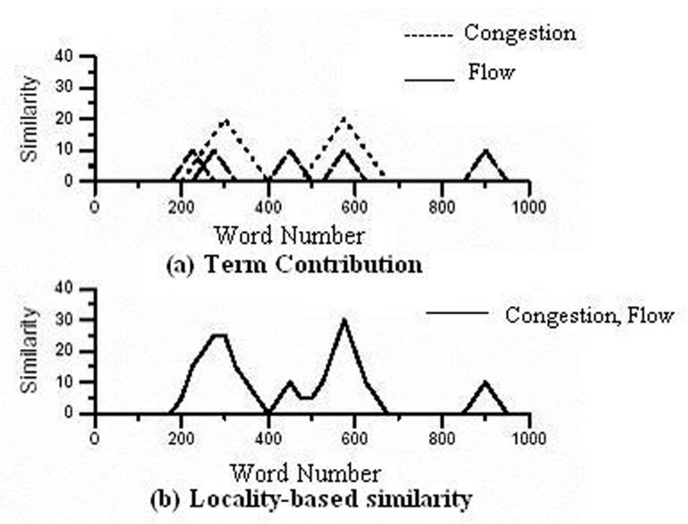

Figure 4. Sample example of locality-based querying

- d = |x – l| is the distance in words between the term in question and the location at which its influence is being evaluated. In each case the value of ct(x; l) is defined to be zero when |x - l| > s_t

$$C_t\,(x\,,\,l) = h_t * \sqrt{(1 - (d / s_t)^2)}$$

The top N (value set by the user) ranked passages (window surrounding the location) is returned after scoring all the locations of the query term according to the weightage assigned to them.

The implementation also handles *case folding* and *stemming* (to match up a keyword with any of its other grammatical forms) of words while searching the words and indexing them into the inverted page list. For repeated use, the system can be configured to reduce the retrieval time manifold. This is done by searching all the domain specific entities (as already identified) from the documents and indexing them into the table beforehand. This increases the speed of the system since each time the question is asked most of the query terms location are already available and the system does not need to search again (except in the case when additional documents have been added).

Answer Selection and Presentation

The top-ranked passages which are now returned (after weighting and ranking on basis of locality and context) are answer candidates. These are further processed to select those answer passages that will be presented to the user. Some passages may be ranked higher just because of frequent occurrence of one of the principal terms in the query without actually illustrating the intended relation for which the user has asked. For example, the user gives following question: “What is the difference between Bus and Star network topologies?” It is probable that a passage from the introduction of network topology (where occurrence of “network” and “topology” is more frequent with just a reference to Bus and Star topology). To avoid these situations the system searches the occurrence of *Noun Phrases* (identified in the Question Parsing section) in the passages. Those passages in which matches are found are ranked higher amongst the top ones.

After phrase matching, the system processes the passages according to the classification done in *question classification*. If the question was classified in the second category requiring any date or numerical expression then the system searches for these terms in the passages to match the answer type. For questions in the first category, the system extracts information from more than one passage (those which are scored higher than a threshold value) and presents all of them to the user along with the links to their respective locations in the documents (as shown in Figure 5). This helps the learner to quickly find the relevant information from many documents and to understand the concept. Furthermore, if the top passages are coming from different resources (slides or books of different authors) then they are ranked separately (amongst the same type of resource) and best answer passage from each is presented to the user.

Feedback

Feedback is the one of the important parts of the QA system that distinguishes it from other QA systems being used today. It provides interactivity between the user and the system. When the question is ambiguous, proper feedback can guide the user to improve the query or reformulate the question and get the intended answer. This mechanism prevents the system from failing in case of questions where focus was not clear and proper context was not used. It provides feedback to the user by suggesting extra keywords to be included in the query (as shown in Figure 6). This is done through a closed loop dialogue.

Closed Loop Dialogue

The user inputs a question at the specified place in natural language. After the user has entered the question he observes a sequence of passages as

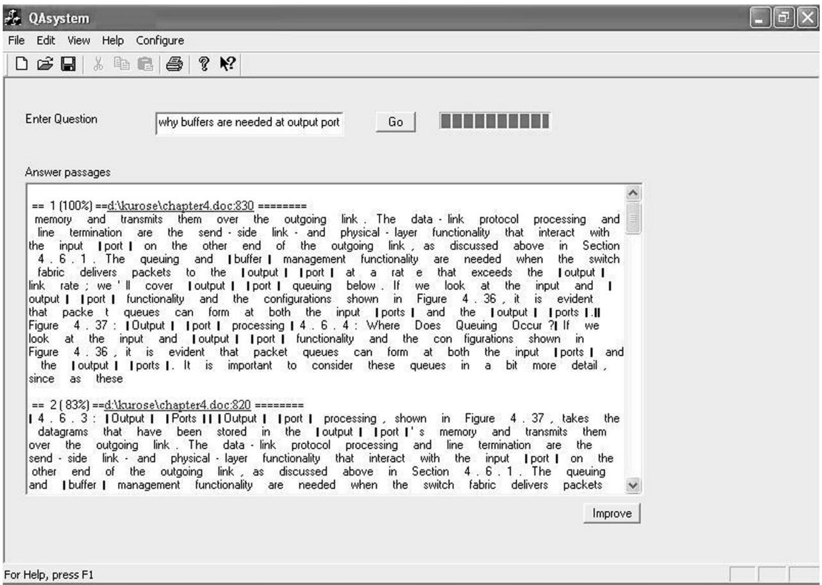

Figure 5. Output of the system. The answer to the question was obtained in the first passage with full confidence (100%) giving also the information and link to the specific location in the relevant document.

probable answers to his question. With the passages hyperlinks are provided so that user can access the documents concerned. If the user is not satisfied with the answers provided he can opt for improving the query. This is done through domain-specific query expansion. In such cases the system goes for extensive passage analysis where domain specific entities are searched from lower ranked passages. These entities are then suggested as extra keywords to be included in the query. This guides the user on how to improve the query or reformulate the question in such a way that can extract relevant answers from the system. The user can choose any number of entities (amongst the suggested ones) which he thinks can improve his question. He may also opt for reformulating the whole question.

RESULTS

The main goal of our experiments was to determine the efficiency of our system to locate the exact answers or give an indication of having the exact answer just near to the retrieved passages. For experiment purpose, a course on Computer Networks was selected. Text books (scanned) of *"Computer Networking: A Top-down Approach Featuring the Internet"* by James F. Kurose and Keith W.Ross and *"Computer Networks 4th Edition"* by Andrew

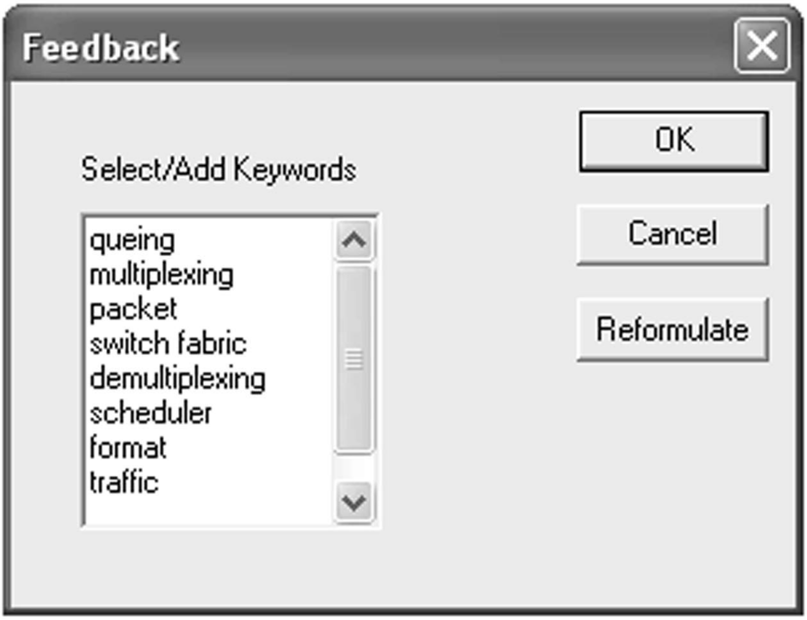

Figure 6. Feedback given to user

S. Tanenbaum were used along with their PowerPoint slides. The questions used for testing were picked from review questions at the back of the chapters and FAQs available on Internet. Also a separate collection of questions was drawn out by a survey among students with their knowledge of computer networks varying from beginners (not familiar with the subject) to students who performed well in the subject.

The questions covered a wide range of topics on computer networks. They were of varying type, complexity and difficulty. Questions were non-factual, explanatory and required extracting passages from different places. Three results per query were extracted. The results are shown in Table 2 and Table 3. The time for information retrieval was quite negligible and we aim to make it faster in the near future. The percentages of confidence (on average) the system had that the answer was present in the first, second and third passage was on average 100%, 85%, and 65%. In Table 2, the questions that were answered in these passages are given in the second, third and fourth columns. Under the column *Directs*, those questions were included which were not answered directly, but gave the indication of the exact answer to be contained in the same document (near the retrieved passages). Those questions which were answered, only after taking feedback from the user, were included in the next column. Those questions which could not be answered by the system were included under the column *Failed*.

In nearly 11% of the questions, our system failed to get the right answer. Amongst these, nearly half of the questions were not within the purview of the material. The rest of the cases were because the frequency of occurrence of keywords factor failed, giving undue importance to certain keywords. In 7.5% of the questions, the answer improved from failure to exact (because of our query expansion technique). It successfully answered questions like,

Table 2
Our questions (mostly Review questions and FAQs)

#Questions	ANSWER 1	ANSWER 2	ANSWER 3	DIRECTS	FEEDBACK	FAILED
150	72	15	3	32	12	16

Table 3
Questions collected from survey

Questions	ANSWER 1	ANSWER 2	ANSWER 3	DIRECTS	FEEDBACK	FAILED
25 (experts)	10	5	2	3	1	4
25 (naives)	14	4	1	2	2	2

"What is the difference between source-based tree and centre-based trees in…" by extracting passages from two different documents. The results are quite pleasing and the importance of feedback is made apparent because it improved the system in the case of failure by giving the right answer.

CONCLUSIONS

In this article, a QA system is proposed which can solve a learner's problems to a great extent with minimal human-computer dialogue. Using the concept of entities the system is fully automated to work in any subject domain with some input from human expertise. The system is based on searching in context and utilizes syntactic and partial semantic information. This achieves good accuracy in results. While additional work is required to enhance the speed and prediction accuracy of the system and to enable it to withstand a very high workload, our initial experiments are promising. The system can handle multiple resources as is frequently available in e-learning domain.

The current implementation utilizes only partial semantic information during answer extraction and selection. It is believed that recall would be much higher if these factors were taken more into consideration. Improvement upon the search facility can be done by storing previous queries and links of their respective answers which were accepted by users in full confidence. Fundamental approach used by (Kutay, & Ho, 2003) for the analysis of students' interaction and learning could be helpful in such a design. Such a facility could be used to help future users and will facilitate group learning, although this will be a burden on the memory of the system. In addition, a learner model similar to building a user model as done by (Davis, Kay, Kummerfeld, Poon, Quigley, Suanders, Yacef, 2003) could be used for enhancing accuracy for repeated use by a learner.

References

Cody, C. T. K., Oren, E., & Daniel, S. W. (2001). Scaling question answering to the Web. *Proceedings of the Tenth International Conference on World Wide Web*, 150-161.

Cha, G. (2002). COVA: A System for content-based distance learning. *Proceedings International WWW Conference* (11), Honolulu, Hawaii, USA.

Davis, J., Kay, J., Kummerfeld, B., Poon, J., Quigley, A., Saunders, G., & Yacef, K. (2003). Using workflow, user modeling and tutoring strategies for just-in-time document delivery. *AIEd2003. Workshop on Technologies for Electronic Documents for Supporting Learning* (Volume X).

Dorai, C., Kermani, P., & Stewart, A. (2001). ELM-N : E-learning media navigator. *International Multimedia Conference Proceedings of the Ninth ACM International Conference on Multimedia,* 634 – 635.

Fu, Y., & Shen, R. (2004). GA based CBR approach in Q&A system. *Expert Systems with Applications, 26(*2), pp. 167-170(4).

Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP.Montreal,* Canada, 38-44.

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: The view from here. *Natural Language Engineering*.

Hori, C., Hori, T., Isozaki, H., Maeda, E., Katagiri, S., & Furui, S. (2003). Study on spoken interactive open domain question answering. *SSPR2003*, pp.111 – 113.

Kretser, O.D., & Moffat, A. (1999). Effective document presentation with a locality-based similarity heuristic. *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, San Francisco, pp 113-120.

Kutay, C., & Ho, P. (2003). Using intelligent agents for the analysis of students interaction and learning. *AIEd2003. Workshop on Technologies for Electronic Documents for Supporting Learning*, (Volume X).

Lin, J., Quan, D., Sinha, V., Bakshi, K., Huynh, D., Katz, B., & Karger, D.R. (2003). The role of context in question answering systems. *Proceedings of the 2003 Conference on Human Factors in Computing Systems.*

Mendes, M. E. S., Martinez, E., & Sacks, L. (2002). knowledge-based content navigation in e-learning applications. *The London Communication Symposium.*

Small, S., Liu, T., Shimizu, N., & Strzalkowski, T. (2003). HITIQA: An interactive question answering system, a preliminary report. *Proceedings of ACL '03, Workshop on QA*. Sapporo, Japan.

Temperley, D., Sleator, D., & Lafferty, J. (1993) Parsing English with a link grammar. *Third Annual Workshop on Parsing Technologies.*

Woods, W. (1973). Progress in natural language understanding – An application to lunar geology. *AFIPS Conference Proceedings, 42,* 441-450.

Note

Source code for the implementation work can be requested at the following email address: ankumfec@iitr.ernet.in