Novel Approaches to Information Systems Design

Naveen Prakash
Indraprastha Institute of Information Technology, Delhi, India

Deepika Prakash
NIIT University, India

A volume in the Advances in Computer and Electrical Engineering (ACEE) Book Series

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Prakash, N. (Naveen), editor. | Prakash, Deepika, 1983- editor.
Title: Novel approaches to information systems design / Naveen Prakash and Deepika Prakash, editors.
Description: Hershey, PA : Engineering Science Reference, an imprint of IGI Global, [2020] | Includes bibliographical references and index. | Summary: “This book examines recent and on-going research in the area of information systems”-- Provided by publisher.
Identifiers: LCCN 2019046103 (print) | LCCN 2019046104 (ebook) | ISBN 9781799829751 (hardcover) | ISBN 9781799829768 (paperback) | ISBN 9781799829775 (ebook)
Subjects: LCSH: System design. | Computer systems. | Information storage and retrieval systems.
Classification: LCC QA76.9.S88 N685 2020 (print) | LCC QA76.9.S88 (ebook) | DDC 003--dc23
LC record available at https://lccn.loc.gov/2019046103
LC ebook record available at https://lccn.loc.gov/2019046104

This book is published in the IGI Global book series Advances in Computer and Electrical Engineering (ACEE) (ISSN: 2327-039X; eISSN: 2327-0403)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Advances in Computer and Electrical Engineering (ACEE) Book Series

ISSN:2327-039X
EISSN:2327-0403

Editor-in-Chief: Srikanta Patnaik, SOA University, India

MISSION

The fields of computer engineering and electrical engineering encompass a broad range of interdisciplinary topics allowing for expansive research developments across multiple fields. Research in these areas continues to develop and become increasingly important as computer and electrical systems have become an integral part of everyday life.

The **Advances in Computer and Electrical Engineering (ACEE) Book Series** aims to publish research on diverse topics pertaining to computer engineering and electrical engineering. **ACEE** encourages scholarly discourse on the latest applications, tools, and methodologies being implemented in the field for the design and development of computer and electrical systems.

COVERAGE

- Electrical Power Conversion
- Qualitative Methods
- Computer Hardware
- Chip Design
- Microprocessor Design
- Applied Electromagnetics
- Optical Electronics
- Sensor Technologies
- Power Electronics
- VLSI Fabrication

IGI Global is currently accepting manuscripts for publication within this series. To submit a proposal for a volume in this series, please contact our Acquisition Editors at Acquisitions@igi-global.com or visit: http://www.igi-global.com/publish/.

The Advances in Computer and Electrical Engineering (ACEE) Book Series (ISSN 2327-039X) is published by IGI Global, 701 E. Chocolate Avenue, Hershey, PA 17033-1240, USA, www.igi-global.com. This series is composed of titles available for purchase individually; each title is edited to be contextually exclusive from any other title within the series. For pricing and ordering information please visit http://www.igi-global.com/book-series/advances-computer-electrical-engineering/73675. Postmaster: Send all address changes to above address.

Titles in this Series

For a list of additional titles in this series, please visit:
https://www.igi-global.com/book-series/advances-computer-electrical-engineering/73675

Handbook of Research on New Solutions and Technologies in Electrical Distribution Networks
Baseem Khan (Hawassa University, Hawassa, Ethiopia) Hassan Haes Alhelou (Tishreen University, Syria) and Ghassan Hayek (Tishreen University, Syria)
Engineering Science Reference • © 2020 • 439pp • H/C (ISBN: 9781799812302) • US $270.00

Major Applications of Carbon Nanotube Field-Effect Transistors (CNTFET)
Balwinder Raj (National Institute of Technical Teachers Training and Research, Chandigarh, India) Mamta Khosla (Dr. B. R. Ambedkar National Institute of Technology, Jalandhar, India) and Amandeep Singh (National Institute of Technology, Srinagar, India)
Engineering Science Reference • © 2020 • 300pp • H/C (ISBN: 9781799813934) • US $185.00

Neural Networks for Natural Language Processing
Sumathi S. (St. Joseph's College of Engineering, India) and Janani M. (St. Joseph's College of Engineering, India)
Engineering Science Reference • © 2020 • 227pp • H/C (ISBN: 9781799811596) • US $235.00

Emerging Applications of Differential Equations and Game Theory
Sırma Zeynep Alparslan Gök (Süleyman Demirel University, Turkey) and Duygu Aruğaslan Çinçin (Süleyman Demirel University, Turkey)
Engineering Science Reference • © 2020 • 284pp • H/C (ISBN: 9781799801344) • US $225.00

For an entire list of titles in this series, please visit:
https://www.igi-global.com/book-series/advances-computer-electrical-engineering/73675

IGI Global
DISSEMINATOR OF KNOWLEDGE

701 East Chocolate Avenue, Hershey, PA 17033, USA
Tel: 717-533-8845 x100 • Fax: 717-533-8661
E-Mail: cust@igi-global.com • www.igi-global.com

Table of Contents

Detailed Table of Contents

Agent-based models have emerged as a promising paradigm for addressing ever increasing complexity of information systems. In its initial days in the 1990s when object-oriented modeling was at its peak, an agent was treated as a special kind of "object" that had a persistent state and its own thread of execution. But since then, agent-based models have diversified enormously to even open new conceptual insights about the nature of systems in general. This chapter presents a perspective on the disparate ways in which our understanding of agency, as well as computational models of agency, have evolved. Advances in hardware like GPUs that brought neural networks back to life may also similarly infuse new life into agent-based models and pave the way for significant advancements in research on artificial general intelligence (AGI).

A data stream is a real-time continuous sequence that may be comprised of data or events. Data stream processing is different from static data processing which resides in a database. The data stream data is seen only once. It is too voluminous to store statically. A small portion of data called a window is considered at a time for querying, computing aggregates, etc. In this chapter, the authors explain the different types of window movement over incoming data. A query on a stream is repeatedly executed on the new data created by the movement of the window. SQL extensions to handle continuous queries is addressed in this chapter. Streams that contain transactional data as well as those that contain events are considered.

The declarative approach specifies what is to be done rather than how to do it. When adopted in information systems development, this implies that the system should be seen as a collection of business rules that can be enacted using a business rules engine. Business rules should be expressed in a form that is as close to the one in which business people perceive the rules. A business rules management system is needed to acquire, store, and allow modification of a business rules database. The rules are then handed over to a rules engine for enactment. The BRMS considered in this chapter uses an antecedent-consequent form for representing rules. These are based in a first order logic. Rules are formed with courses of actions and conditions in rules antecedents and courses of actions in rule consequents. It also introduces notions of state change in the business rule and temporal relation within rule and between different rules. Business rules are structured into atomic, complex, and abstract rules. The business rules are translated into enactment rules and converted to Java.

Traditionally, data warehouse requirements engineering is oriented towards determining the information contents of the warehouse to be. This has resulted in a de-emphasis of the functional perspective of data warehouses. Consequently, it is difficult to specify functions needed for computing business indicators. The authors' approach aims to elicit needed business indicators from organizational decision makers. Thereafter, indicator hierarchies are built. Then they associate functions with business indicators of the hierarchy. These functions are visualized as use case diagrams. To do this, they extend these diagrams to allow for actor aggregation in addition to actor specialization. Further, they introduce the 'estimated from' relationship between use cases, in addition to the 'extend' and 'include' relationships of UML. They illustrate their proposals with an example.

Optimal selection of materialized views is crucial for enhancing the performance and efficiency of data warehouse to render decisions effectively. Numerous evolutionary optimization algorithms like particle swarm optimization (PSO), genetic algorithm (GA), bee colony optimization (BCO), backtracking search optimization algorithm (BSA), etc. have been used by researchers for the selection of views optimally. Various frameworks like multiple view processing plan (MVPP), lattice, and AND-OR view graphs have been used for representing the problem space of MVS problem. In this chapter, the authors have implemented random walk grey wolf optimizer (RWGWO) algorithm for materialized view selection (i.e., RWGWOMVS) on lattice framework to find an optimal set of views within the space constraint. RWGWOMVS gives superior results in terms of minimum total query processing cost when compared with GA, BSA, and PSO algorithm. The proposed method scales well on increasing the lattice dimensions and on increasing the number of queries triggered by users.

With the increasing application of web services in our lives, selecting the right web service is becoming unprecedentedly difficult. Indeed, before paying the price of a web service, the customer always tries to make sure of his choice. One of the mechanisms used to put the customer in trust is to make available the opinions of other customers who have already used this web service. In the literature, many solutions for measuring the reputation of web services have been proposed. Unfortunately, they ignore certain aspects that we find important to ensure a more meaningful assessment of the reputation. Firstly, consumers do not always have the same satisfaction criteria, and as a result, they can judge the same web service differently. Thus, without knowing the consumer's preferences, it is almost impossible to give meaning to his opinion. Secondly, the qualities of a web service can be changed over time, and hence, the old ratings are no longer representative. In this chapter, the authors propose a novel reputation computation approach to deal with the problems mentioned above.

The supply demand gap in energy sector in any country is a major challenge. Demand side management (DSM) and energy efficiency (EE) are the well-known solutions in the short term, and capacity addition is the long-term solution. However, both DSM and EE initiatives require significant investment and logistics if implemented in the traditional approach. The contemporary Industry 4.0 principles can be effectively applied to resolve several issues. This chapter proposes a novel enterprise information system (EIS) by treating the modern power systems as cyber physical system and to manage the processes of DSM and EE. A prototype system is suggested to pave the path for EIS, and the functional characteristics are illustrated with a few data visualizations.

The world is moving towards a new generation of internet, the internet of things (IoT). This technological jump is improving the way we live by creating a bridge between the physical and the virtual world. Researchers are curious about the field from different perspectives especially managing of the complexity and dynamicity of these systems. The main problem being targeted in this chapter is that IoT systems are exposed to many structural and behavioral changes due to internal and external factors. This study is about the necessity of having a mechanism that enables IoT systems to perform without breaks or shutdowns regardless of context changes. The solution consists of a contextual dynamic reconfiguration process implemented by a reflexive multilayered architecture. This process is based on the autonomic computing loop. The authors also integrated evolution styles to make reusable the reconfigurations applied on the architecture of the system. Validation of the proposed approach was made on an e-health scenario, which was simulated using Cisco Packet Tracer before performing real development.

The manufacturing industry is rapidly changing due to widespread adoption of information and communication technologies. This new landscape, described as the fourth industrial revolution, will be characterized by highly complex and interdependent systems. One particular aspect of this shift is horizontal integration, or the tight coupling of firms within a value chain. Highly interconnected and interdependent manufacturing systems will encounter new challenges associated with coordination and collaboration, specifically with regards to trust. This purpose of this chapter is to explore the potential of blockchain to address these challenges. Survey data collected from manufacturing professionals suggests that the perceived nature of trust and resource value can be bounded and controlled. Concepts from game theory, systems theory, and organizational economics are used to augment this research data and inform a collaborative manufacturing blockchain model and architecture.

Access control models (ACM) offers the guarantee that only the qualified users can gain access to the artifacts contained in business processes. Business processes are designed, implemented, and operated using many industrial standards that challenge the interoperation with access control standards. Enterprise engineering (EE) introduces rigorous capabilities to design and implement the essential concepts related with the dynamic of business processes. ACM deals with the systematic design and implementation of dynamic and static access control concepts to qualify the access of the users to the artifacts. This chapter proposes an ontological integration between EE and ACM concepts in order to enable the discussion of access control in the deep structure of the business processes. ACM integrated with EE allow the run-time qualification of the actors while they perform all the business process steps and not only at invocation time. The proposal encompasses business process designed with DEMO ontology and role-based access control concepts using a mathematical model logic description.

Preface

The area of Information Systems Engineering, ISE, emerged in the 1980s as a response to the need for relating computer based products to the larger business environment in which these are embedded. The idea was to develop products that were better fitting and therefore better accepted in the business. In its initial areas, ISE was predominantly concerned with models, tools, and techniques to design systems. Thereafter, in the 1990's attention shifted to the requirements engineering stage. Thus, the entire life cycles from requirements, design to implement was addressed. Simultaneously, attention shifted to the enterprise itself and enterprise architecture and modeling evolved.

The foregoing also reflected the complexity of information systems that could be handled. Starting from single function systems (payroll) through multi-function systems (hotel reservation), we developed ISE techniques to develop systems for entire organizations and inter-organization systems.

The ISE community dealt with two aspects of business, the operational aspect as reflected in the transactions and the decisional aspect that requires decision support systems. The forte of early ISE was transactional systems even though support for decisions was provided for what-if analysis and the like. It was only with the rise of data warehouse technology and business Intelligence that decisional systems acquired full stature. Again, the entire life cycle from requirements to implementation was covered.

In recent years we have seen the rise of completely new kinds of technologies and products that exploit these. New technologies around big data, machine learning, the Internet of Things, and the ubiquitous use of computers has led to newer concerns. Thus we see the emergence of cyber physical systems, the high emphasis on security, social networks and the like. This evolution impacts the traditional view of ISE: its models need change, development processes need re-positioning and tools need sharpening.

This book is an attempt to bring together in one place the diversity of the new information systems. Not only does this volume expose the reader to a range of new information system products, it also deals with the approach to their design and development. What ISE can bring to the new products is its insistence that they fit better in their environment through a practice of obtaining requirements that form the basis for detailed design and implementation.

The chapters by Srinath and Parimala provide overviews of the role of agents and data streams respectively. The former treats agents agnostically, without considering their use in any particular branch of computer science. However, agent orientation has been used in software engineering as well as in data warehousing. Parimala considers data streams from the point of view of their management. The important issue of querying data stream data for relational and non-relational systems is addressed in her chapter.

Sharma considers adopting inference engines as a way of implementing information systems and, in doing so, departs from current practice. He gives central position to business rules and their formulation to provide a comprehensive view of the business. These rules are then converted into rules of an inference engine.

Bhardwaj builds a requirements engineering proposal for data warehouses. This is based on the notion of targets set in companies for different managers. These are interpreted as a form of business indicators. The approach is to elicit indicators needed and sent to various stakeholders in an organization and thus obtain the set of needed information. This is augmented with the arguments of business indicators.

Gosain looks into the problem of selection of materialized views in a Data Warehouse system. They adapt the evolutionary optimization algorithm Random walk Grey wolf optimizer (RWGWO) algorithm for materialized views. They show that their algorithm scales as the number of user queries increase.

The chapter by Benaboud considers the issue of selecting web services. The assumption is that customers would not pay before s/he trusts the web service. Thus, establishing the reputation of a web service is important. The chapter proposes a reputation computation approach to also take into account evolving qualities of a web service.

The chapter by Sastry proposes a solution to the supply demand gap in the energy sector. They develop a prototype Enterprise Information System (EIS) by applying Industry 4.0 principles. This work has been done in the context of smart grids using diverse energy generation sources.

Hakim looks at the evolution of IoT systems in the face of structural and behavioral changes. These changes may occur due to internal and/or external factors in an organization. The authors propose a multi-layered IoT architecture to cater to such contextual changes in software systems with connected components.

The chapter by Kelley considers the requirements of horizontal integration, between production nodes in manufacturing systems, in the context of Industry 4.0 and the effect horizontal integration has on decisions regarding system design.

The authors explore the use of a distributed ledger or a blockchain technology to address these requirements.

The security issue in business process modeling is addressed by Guerreiro. The authors integrate access control models with enterprise engineering in an Ontology. The idea is to carry out access control at run time rather than merely at the time of invocation of a process.

The book ends with a chapter by Deepika that discusses the state of convergence of three Information Systems technologies used in system design namely, Business Intelligence, Big Data and Machine Learning. The chapter goes on to propose a set of rules for implementing the Business Intelligence system, a Data Warehouse, using a Big Data system, a NoSQL database.

Acknowledgment

The editors of this book would like to thank all the authors who have proposed chapters to this book. Despite the rather tight deadlines with which we were working, authors have made strenuous efforts to meet these deadlines.

Special thanks go out to those authors who have acted as reviewers. Their insightful comments have helped to improve the quality of the material and made the chapters more presentable.

Finally, our thanks go to IGI Global for providing guidance and supporting us at every stage.

Chapter 1
The Evolution of Computational Agency

Srinath Srinivasa
International Institute of Information Technology, Bangalore, India

Jayati Deshmukh
International Institute of Information Technology, Bangalore, India

ABSTRACT

Agent-based models have emerged as a promising paradigm for addressing ever increasing complexity of information systems. In its initial days in the 1990s when object-oriented modeling was at its peak, an agent was treated as a special kind of "object" that had a persistent state and its own thread of execution. But since then, agent-based models have diversified enormously to even open new conceptual insights about the nature of systems in general. This chapter presents a perspective on the disparate ways in which our understanding of agency, as well as computational models of agency, have evolved. Advances in hardware like GPUs that brought neural networks back to life may also similarly infuse new life into agent-based models and pave the way for significant advancements in research on artificial general intelligence (AGI).

1 INTRODUCTION

Today's information systems are complex, distributed, and need to scale over millions of users and a variety of devices, with guaranteed uptimes. As a result, top-down approaches for systems design and engineering are becoming increasingly infeasible.

DOI: 10.4018/978-1-7998-2975-1.ch001

Starting sometime in the 1990s, a branch of systems engineering, has approached the problem of systemic complexity in a bottom-up fashion, by designing "autonomous" or "intelligent" agents that can proactively and autonomously act and decide on their own– to address specific, local issues pertaining to their immediate requirements. They also can communicate and coordinate with one another to jointly solve larger problems. The autonomous nature of agents require some form of a rationale that justifies their actions. Given that, object oriented modeling had attracted mainstream attention at that time, the distinction between mechanistic "objects" and autonomous "agents" were often summarized with this slogan (Jennings et al., 1998): *Objects do it for free, agents do it for money*.

Early research in agent-based systems focused on designing architectures, communication primitives, and knowledge structures for agents' reasoning. Several such independent research pursuits, also resulted in the emergence of standards organizations like FIPA[1], which is now an IEEE standards organization for promoting agent-based modeling and interoperability of its standards with other technologies (Poslad, 2007).

But research interest soon moved from communication and coordination, to address the concept of agency itself. Agents are meant to take decisions "autonomously"– and the term "autonomy" needed sound conceptual and computational foundations. An autonomous agent needs to operate "on its own" and definitions for what this entails, distinguished different models of autonomy. Broadly, approaches to computational modeling of autonomy can be divided into the following research areas: *normative*, *adaptive*, *quantitative*, and *autonomic* models of agency.

Normative models of agency, interpret agency as a combination of imperatives and discretionary entitlements. They also implement logical frameworks that encode different forms of individual and collective goals (Castelfranchi et al., 1999; Van der Hoek & Wooldridge, 2003; L´opez et al., 2006). Some normative elements for agents include: encoding of their goals, that in turn lead to encoding of their intentions or deliberative plans to achieve their goals, their belief about their environment, their obligations, their prohibitions, and so on. Interacting pairs of normative agents create *contracts* that regulates their independent actions with respect to the others' actions. Systems of multiple, normative agents adopt collective *deontics* or constitutions, that regulate overall behaviour (Andrighetto et al., 2013).

Adaptive frameworks for modeling agency, have emerged from problems where agents have to interact with complex and dynamic environments, like in autonomous driving and robotic navigation. These frameworks can either be model-driven where an underlying model of the environment is learned through interactions; or model-agnostic, where adaptations happen purely from positive or negative reinforcement signals from the environment (Macal & North, 2005; Shoham et al., 2003).

The third paradigm of agency is based on quantitative methods based on decision theory and rational choice theory (Ferber & Weiss, 1999; Parsons & Wooldridge, 2002; Semsar-Kazerooni & Khorasani, 2009). These represent agents with a self-interest function, which then interact with their environment to obtain different kinds of payoffs resulting in a corresponding *utility*. Rational agents then strive to make decisions in a way that results in *utility maximization*. Rational choice is represented as pair-wise preference functions between choices, or as numerical payoffs. Interactions between agents are modeled as games representing *confounded rationality*– where rational choices of one agent may (positively or adversely) affect the prospects for others.

A related stream of development, which we treat as the fourth paradigm of agency– started somewhat independently from agent-based modeling, approaches agency by building a model of "self." The field of autonomic computing (Ganek & Corbi, 2003; Kephart & Chess, 2003) first introduced by IBM, aimed to provide computational entities with self-management properties (also called "self-*" properties) like self-healing, self-tuning, self-recovery, etc. The field of Autopoiesis started by Maturana and Varela (Maturana & Varela, 1991), developed computational models of self-referential entities based on biological models of cognition. Yet another related field are those of Cybernetics and Artificial Life (Johnston et al., 2008; Komosinski & Adamatzky, 2009), that addressed self-regulatory mechanisms that characterize natural life, into computational elements. Models developed here, were also used in the study of natural systems in evolutionary biology.

In this chapter, we will organize our study of computational modeling of agency, along the four paradigms detailed above. We will look at the disparate viewpoints towards agency and the primary challenges addressed by them.

2 NORMATIVE MODELS OF AGENCY

In this approach, autonomy is defined in terms of rule-based specifications of discretionary and deliberative elements.

Rule-based systems for autonomous decision-making may seem like a contradiction. That is, if an agent is dictated by rules, can it still be called autonomous?

In the early days of agents research, rule-based approaches were adopted for agents to respond to various kinds of stimuli. These were called reflex agents (Franklin & Graesser, 1996), whose rules were in the form of ECA (Event-Condition-Action) statements. One or more ECA rules would trigger in response to an external stimuli (Event), and based on the conditions that hold, the appropriate action would be performed. While these agents could display rich, adaptive behaviour, the rules and actions themselves had to be specified a priori.

Subsequent research in normative models addressed a broader problem, to use rules to specify the boundaries within which autonomous decision-making takes place. Any system architecture, would need to have two forms of imperatives, which are mandatory. These are its *liveness* and *safety* criteria. These mandatory elements are specified by way of rules.

Liveness criteria, also called the set of "obligations," represent properties that need to hold for the system to be considered functional. A property p is said to be *obligated*, represented as the modality Op in deontic logic, if the system becomes inconsistent whenever p does not hold. The term Op is read as "p ought to be true," rather than "p is asserted to be true" as with predicate logic statements. Similarly, safety criteria called "prohibitions" or "forbidden" properties are of the form Fp, which make the system inconsistent whenever they hold.

It is important to note that, liveness and safety are not negations of one another. A property that is not obligated to hold, is not forbidden to hold; similarly, a property that is not forbidden to hold, is not obligated to hold. An agent that is not obligated to make a particular choice, is not forbidden from choosing it. Similarly, an agent that is not forbidden from choosing something, is not obligated to choose it.

Hence, the logic of norms has at least three modalities of truth. In between the obligated and forbidden regions, is the "permitted" or "viable" region (Egbert & Barandiaran, 2011; Dignum et al., 2000; Mukherjee et al., 2008) in which the agent can operate at its own discretion. Sometimes, the liveness criteria are also considered as part of the viable region. But here, we distinguish between the two because, upholding liveness properties are not subject to the agent's discretion. They are mandatory conditions, which the agents should necessarily uphold.

The viable region is characterized by "deliberative" logic that guides autonomous decision-making by agents. Deliberative logic can in turn be broadly classified into two kinds: *goal oriented* logic, and *truth maintenance* logic.

Goal oriented deliberative logic, represents autonomous decision-making in pursuit of a goal from a set of possible goals. One of the widely popular models for goal-oriented deliberative agents is the Belifs-Desires-Intention (BDI) model and its several variants (Dignum et al., 2000; Kinny et al., 1996; Meneguzzi & Luck, 2009; Rao & Georgeff, 1991; Rao et al., 1995). This model comprises of three elements:

- **Beliefs:** These represent the informational state of the agent, encoding its supposed knowledge about the environment and of other agents. Elements of an agent's belief may or may not be true, and can be revised on interaction with the environment.
- **Desires:** These represent a set of goals that the agent wishes to perform.

- **Intentions:** These represent a (or a set of) goal(s) that the agent has committed to. An intention involves committing a goal to a set of actions, by choosing a plan from a set of available plans.

The creation of plans itself is not part of the BDI model, and is relegated to either human planners or a planning application. The choice of intention may be driven by several factors which involve one-shot, rational decision-making, learning, etc. Some BDI models also incorporate events that trigger activity in the agents– like pursuing a goal or updating its beliefs.

In contrast to goal oriented deliberative logic, truth maintenance systems (TMS) (Doyle, 1977; Huhns & Bridgeland, 1991; McAllester, 1990) require agents to act autonomously to maintain one or more properties in the viable region, while interacting with external, uncertain environments. The generic model of a TMS comprises of the following: a property (or a set of properties) Φ which need to be maintained, a set of *constraints*– typically the liveness and safety constraints, and a set of *premises* Σ, which represent a set of knowledge or belief elements about the state of the world, based on external interactions. The objective of the TMS is to compute entailments to establish whether $\Sigma \rightarrow \Phi$ holds.

If the assertion $\Sigma \rightarrow \Phi$ can be proven to hold, then the TMS can maintain the required properties Φ as well as provide a justification for its maintenance related actions. If Σ ® Φ can be proven to be false, then the TMS would need to perform corresponding actions such that Σ no longer holds, and is replaced by another set of premises Σ^0, which can entail Φ.

For example, consider an aircraft where Φ represents the altitude that needs to be maintained. Σ represents the premise derived from the set of all input data like air speed, attitude, bank angle, drag, etc. If the premises can entail Φ it means that the current state of the aircraft can support maintenance of the altitude. If on the other hand, Σ can be shown to not entail Φ it means that corrective action needs to be taken to adjust the aircraft state itself, so that the altitude can be maintained.

There is a third case where the premises can neither entail the property that needs to be maintained, nor entail the negation of the property. In such cases, it is unknown to the agent whether Φ can be maintained. Such cases require TMS to employ non-monotonic and/or auto-epistemic elements to update its system of entailment rules.

Unlike goal-oriented logics, truth maintenance logics need to run continuously, to check current premises, entail required properties, and/or perform belief revision or non-monotonic updates to deal with uncertainty. Truth maintenance does not end with a single entailment computation.

3 ADAPTIVE LEARNING BASED MODELS

Adaptive learning based models are used in applications where agents have to interact with complex, dynamic environments; and need to continuously respond to changes. Examples include autonomous driving, robotic navigation, stochastic scheduling, swarm robotics, ant colony optimization, etc.

Adaptive learning agents are modeled using reinforcement learning (Sutton et al., 1998), which are in turn typically modeled as a Markov Decision Process (MDP). An MDP is characterized by a set of states *S*, and a set of actions *A*, and associated probability of state transitions for any given action. A term of the form $P_a(s,s^0)$ denotes the probability of the MDP state to transition from *s* to s^0 on performing action *a*. Any action by the agent may change the state of the interaction, and may also provide a feedback (sometimes called the "reward") from the environment, that could be either positive or negative. This is denoted as $R_a(s,s^0)$ indicating the reward on reaching s^0 from *s*, by performing action *a*.

Reinforcement learning addresses two forms of underlying challenges: the "exploration vs exploitation" dilemma, and the "lookahead" dilemma.

The *exploration vs exploitation* dilemma involves deciding between choosing the action with the best expected payoff at any given interaction state; or choosing a new action to explore more of the interaction state space. The *look ahead* dilemma involves deciding whether to choose an action based on its immediate expected reward, or consider longer term prospects of having chosen such an action. Different reinforcement learning heuristics exist for addressing both the dilemmas.

For finite MDPs, a well known algorithm called Q-learning Watkins & Dayan (1992), based on power iterations, is widely used to compute strategic payoffs based on unbounded look ahead.

Reinforcement learning is conventionally used for single agent interactions with its environment. Other agents are considered to be part of the complex, dynamic environment that the given agent interacts with.

However, the extension from a single-agent RL to a multi-agent RL problem is not straightforward. A shared space with several agents can thus be thought of as independent RL runs by each agent separately. This however, is known to be ineffective due to agents overfitting their best responses to each other's behaviours (Lanctot et al., 2017). Multi-Agent Reinforcement Learning (MARL) models were hence developed based on concepts of joint policy correlation between agents, where policies generated using deep reinforcement learning, are evaluated using game theoretic principle (Bu¸soniu et al., 2010; Shoham et al., 2003). With finite state spaces, Q-learning approaches were also extended to multi-agent systems (Claus & Boutilier, 1998). It is also seen that it is harder to design systems of joint learners as compared to independent learners, and while independent learners overfit to each other's behaviours, joint learning often don't perform significantly better as they become entrenched in local minima.

A related area of research is adaptive social learning agents. These are adaptive agents operating in a shared state space, that not only respond to feedback from the environment, but also interact with other agents either competitively or cooperatively, and may also share instantaneous information, and episodic and general knowledge (Littman, 1994; Tan, 1993).

Swarm intelligence (Kennedy, 2006; Bonabeau et al., 1999; Beni, 2004) is another direction of adaptive learning based models which is motivated by nature. It models a system of agents which act as a group without any centralized control. A variant of Swarm intelligence, called Ant Colony Optimization (Dorigo & Di Caro, 1999) has been useful in context of multi-agent systems. It has been used in a variety of use-cases like resource-constrained project scheduling (Merkle et al., 2002) and optimization in continuous domains (Socha & Dorigo, 2008).

4 RATIONAL CHOICE BASED MODELS

In this model of agency, autonomous agents are modeled as rational maximizers, driven by a self-interest function, and operating towards utility maximization. Mathematical underpinnings of such models derive from rational choice theory, decision theory, and game theory.

Given a set of elements or actions A, classical rational choice theory going back to the works of von Neumann and Morgenstern (Von Neumann et al., 2007), defines the following preference functions between pairs of elements of A: $\prec$ (strong preference), $\leq$ (weak preference), and $\|$ (indifference).

Mechanisms for converting pairwise preference functions into quantitative payoffs are also provided, which are based on the concept of expected utility. Any given choice L is represented as a set of pairs of the form (a_i, p_i), where a_i Î A represents one of the elements, and p_i is the probability of receiving a_i. For any pair of choices L and M, a quantitative payoff function $u()$ can be formulated such that $L \prec M$ iff $E(u(L)) < E(u(M))$, where $E()$ is the expected value of the payoff function based on the elements of the corresponding choices.

Rational choice and game theoretic formalisms have been widely employed in designing agent-based systems (Boella & Lesmo, 2002; Hogg & Jennings, 1997; Kraus, 1997; Panait & Luke, 2005; Parsons & Wooldridge, 2002). This paradigm of agent-based modeling has been found to be particularly attractive for applications involving simulation and gamification for policy design, where the agents represent human stakeholders (Parker et al., 2003; Pan et al., 2007; Schreinemachers & Berger, 2006). Rational choice theory and game theory have a long history of being used as mathematical underpinnings for human decision making and behavioural economics– and agent-based simulation models offer an attractive opportunity to model and simulate the repercussions of policy changes.

Human rationality is known to deviate considerably from the classical model of rational choice. In addition to rational maximization, human rationality is characterized by factors like consideration for empathy and fairness, risk aversion, bounded rationality, and a variety of cognitive biases. Agent-based modeling have addressed these in various ways in order to simulate human behaviour and its emergent consequences more accurately (Deshmukh & Srinivasa, 2015; Kant & Thiriot, 2006; Manson, 2006; Santos et al., 2016; Vidal & Durfee, 1995). These extensions to classical rational choice models are important in simulating probable outcomes in emergency situations (like fire evacuations, for example), involving humans Pan et al. (2005); Tang & Ren (2008).

While rational choice theory is used to direct the behaviour of individual agents, this is insufficient when multiple agents have to operate in a shared state space. Interactions between disparate agents in a shared state space, can be broadly seen as either non-cooperative or cooperative, in nature. Correspondingly, these kinds of interactions derive theories from non-cooperative game theory and negotiation theory for the former (Chakraborti et al., 2015; Gotts et al., 2003; Tennenholtz, 1999), and from cooperative game theory and allocation theory for the latter forms of interaction (Adler & Blue, 2002; Albiero et al., 2007).

A related application area that have used the rational choice paradigm for modeling agents, is multi-agent networks. These applications study complex networks, by combining network science, rational choice theory and other related areas like evolutionary algorithms, to study different kinds of emergent properties arising from agents acting rationally in a networked environment (Mei et al., 2015; Patil, Srinivas et al., 2009; Patil, Srinivasa, & Venkatasubramanian, 2009; Villez et al., 2011). Some indicative problems addressed by multi-agent networks include: constrained negotiation and agreement protocols (Nedic et al., 2010; Meng & Chen, 2013), modeling diffusion and synchrony (Faber et al., 2010; Kiesling et al., 2012; Kim et al., 2011), etc.

5 MODELS OF SELF AND AGENCY

Lastly, we review literature from related fields that developed independently of agent-based modeling. All of these fields have tried to model the concept of “self” in computational entities, which is becoming increasingly relevant in agent-based systems as well.

The field of *autonomic computing* became a major area of research, after IBM coined this term in 2001, to denote self managing systems Computing et al. (2006). These include database backed information systems that could configure, protect, tune, heal and recover from failures on their own. Subsequently, autonomic computing has been pursued in various forms (Kephart & Chess, 2003; Kephart, 2005; Huebscher & McCann, 2008). The main motivation (from natural self-governing systems) of building autonomic systems was to have systems which can manage themselves instead of having a team of skilled workforce to manage the system. The four main principles of self management (Kephart & Chess, 2003) are *self-configuration*: systems which can configure its components, *self-optimization*: systems which keep improving over time, *self-healing*: systems which can diagnose and rectify its problems and *self-protection*: systems which can defend itself from attacks.

Autonomic architecture is used to design autonomic computing systems. It creates a network of autonomous elements, each managing its own internal state and interacting with other elements as well as the external environment.

On similar lines, taking inspiration from cell biology, Maturana and Varela (Varela et al., 1974; Maturana & Varela, 1991) coined the term autopoiesis, representing systems which can sustain itself without any external interaction. These systems determine its own structure in order to sustain in an environment. Autopoeisis has since then been extended by designing computational models (McMullin, 2004; Di Paolo, 2005) in context of social autopoietic systems (Seidl, 2004).

Advancements in biology and computational science has lead to the development of a new area at its intersection called *Artificial Life* or *ALife* (Langton, 1997; Bedau, 2003; Aguilar et al., 2014; Bedau et al., 2000). It is used to model natural life and its associated processes using computational models. It can be used to analyze things like evolution and dynamics in natural systems. ALife models are classified as *soft*: involving simulations of systems, *hard*: involving hardware implementations (using robots) or *wet*: involving biochemical synthesis of elements.

All these autonomic models like autopoesis, artificial life, autonomic computing etc can be considered as the foundational models for designing autonomous systems. It can provide an initial framework to build systems of agents having agency, autonomy as well as self-interest. It is going to be even more relevant with the recent advancements in areas like self-driving cars or autonomous drones.

6 AGENTS AND AGI

The ultimate dream of artificial intelligence (AI) research is to create computational models of “general” intelligence, or AGI. AGI, also called “strong” or “full” AI, is contrasted with “weak” or “narrow” AI that is built for specific applications. A basic ingredient of general AI is the need for “common sense” form of intelligence, that is adaptive and applicable in different contexts, and each contextual experience enhances its overall intelligence.

Architectures for AGI have explored different pardigms. The Novamente AGI engine (Goertzel et al., 2004; Goertzel & Pennachin, 2007), incorporates several paradigms of narrow AI including reinforcement learning, evolutionary algorithms, neural networks and symbolic computing into an underlying model of mental processes based on complex systems theory. Approaches like Universal AI (Hutter, 2001) and G¨odel Machines (Schmidhuber, 2007) develop self-rewriting systems that can completely reprogram themselves subject only to computability constraints. Architectures like SOAR (Laird, 2008, 2012; Young & Lewis, 1999) and ACT-R (Anderson, 1996) incorporates several elements of human cognition like semantic and episodic memory, working memory, emotion, etc. from elementary building blocks, to create an architecture for generic problem-solving. The ARS architecture Schaat et al. (2014) aims to develop an agent-based model of the human mind simulated as an artificial life engine, to explore general intelligence.

The recent resurgence of interest in AI has been brought about by advances in parallel computing architectures like Graphics Programming Units (GPUs) that enabled implementation of large artificial neural networks (ANNs). This led to the field of deep learning where ANNs could even detect features automatically, and perform several forms of perception, recognition and linguistic functions.

However, it is widely recognized that an artificial neuron does not represent how a natural neuron works. An artificial neuron is modeled as a gate, where an activation function is triggered by values on its several input lines. The gate metaphor of the foundations of computation has its roots in electrical engineering. However, natural neurons and other building blocks of life (muscles, cells, tissues, etc.) are known to be autonomous decision-makers Moreno & Etxeberria (2005) rather than passive gates. Agency in nature seems to be a balancing act between autonomous entities striving to sustain themselves, and explore or interact with their environment.

General intelligence hence needs to be more of a truth maintenance system, rather than as a goal-oriented system. Preferences defining self interest, as well as declarative elements of one’s knowledge are in turn rooted in considerations of sustainability of one’s self and interaction with the environment.

7 CONCLUSION

In this chapter, we looked at how models of computational agency have evolved over time. Initially, agents were designed as normative elements comprising of discretionary and imperative regions. Although agents could incorporate different forms of logic in the discretionary viable space, this freedom to take actions granted only a basic level of autonomy to the agents. Such models of agency were also restrictive as there was no learning involved. We next discussed about adaptive agents, based on learning based models. These models have been designed so that agents can learn about taking actions using specific strategies by interacting with the environment and other agents. However, in this case there is no motivation for the agents to choose specific strategies or actions apart from greedily increasing their rewards. We then looked into rational choice and game theoretic models where agents have well defined self-interest functions, and they choose to take actions which maximizes their utility. However, is agency just about self-interest and utility maximization? Is an agent just about its preference function over the action space? Are preference relations arbitrarily defined, or are there underlying foundations that guide an agent's preferences? We addressed this question by involving the concept of self into models of agency. These models posit that actions taken by agents are such that it can maintain a stable state of being (using various self-* properties). The preference functions or its action space are not just about greedy maximization of immediate utility but about prolonging the system to persist in its stable state.

In our opinion, the concept of self would need to receive increased research attention in order to address deeper elements of intelligence, like general intelligence. There needs to be an intricate model of self for agents which can link their preference and action space to its self. The model of self, need not just be the model of self of an individual agent, it can also represent the collective self.

REFERENCES

Adler, J. L., & Blue, V. J. (2002). A cooperative multi-agent transportation management and route guidance system. *Transportation Research Part C, Emerging Technologies*, *10*(5-6), 433–454. doi:10.1016/S0968-090X(02)00030-X

Aguilar, W., Santamar'ıa-Bonfil, G., Froese, T., & Gershenson, C. (2014). The past, present, and future of artificial life. *Frontiers in Robotics and AI*, *1*, 8. doi:10.3389/frobt.2014.00008

Albiero, F., Fitzek, F. H., & Katz, M. (2007). Cooperative power saving strategies in wireless networks: an agent-based model. In *2007 4th international symposium on wireless communication systems* (pp. 287–291). 10.1109/ISWCS.2007.4392347

Anderson, J. R. (1996). Act: A simple theory of complex cognition. *The American Psychologist*, *51*(4), 355–365. doi:10.1037/0003-066X.51.4.355

Andrighetto, G., Governatori, G., Noriega, P., & van der Torre, L. W. (2013). *Normative multi-agent systems* (Vol. 4). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Bedau, M. A. (2003). Artificial life: Organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*, *7*(11), 505–512. doi:10.1016/j.tics.2003.09.012 PMID:14585448

Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., ... Ray, T. S. (2000). Open problems in artificial life. *Artificial Life*, *6*(4), 363–376. doi:10.1162/106454600300103683 PMID:11348587

Beni, G. (2004). From swarm intelligence to swarm robotics. In *International workshop on swarm robotics* (pp. 1–9). Academic Press.

Boella, G., & Lesmo, L. (2002). *A game theoretic approach to norms and agents*. Academic Press.

Bonabeau, E., Marco, D. R. D. F., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems (No. 1)*. Oxford University Press.

Bu¸soniu, L. Babuˇska, R., & De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. In Innovations in multi-agent systems and applications-1 (pp. 183–221). Springer.

Castelfranchi, C., Dignum, F., Jonker, C. M., & Treur, J. (1999). Deliberative normative agents: Principles and architecture. In *International workshop on agent theories, architectures, and languages* (pp. 364–378). Academic Press.

Chakraborti, A., Challet, D., Chatterjee, A., Marsili, M., Zhang, Y.-C., & Chakrabarti, B. K. (2015). Statistical mechanics of competitive resource allocation using agent-based models. *Physics Reports*, *552*, 1–25. doi:10.1016/j.physrep.2014.09.006

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI, 1998*(746-752), 2.

Computing, A. (2006). An architectural blueprint for autonomic computing. *IBM White Paper, 31*(2006), 1–6.

Deshmukh, J., & Srinivasa, S. (2015). Evolution of cooperation under entrenchment effects. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems* (pp. 1717–1718). Academic Press.

Di Paolo, E. A. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences*, *4*(4), 429–452. doi:10.100711097-005-9002-y

Dignum, F., Morley, D., Sonenberg, E. A., & Cavedon, L. (2000). Towards socially sophisticated bdi agents. In *Proceedings fourth international conference on multiagent systems* (pp. 111–118). 10.1109/ICMAS.2000.858442

Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new metaheuristic. In *Proceedings of the 1999 congress on evolutionary computationcec99 (cat. no. 99th8406)* (Vol. 2, pp. 1470–1477). 10.1109/CEC.1999.782657

Doyle, J. (1977). *Truth maintenance systems for problem solving* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Egbert, M. D., & Barandiaran, X. E. (2011). *Quantifying normative behaviour and precariousness in adaptive agency*. Ecal.

Faber, A., Valente, M., & Janssen, P. (2010). Exploring domestic microcogeneration in the netherlands: An agent-based demand model for technology diffusion. *Energy Policy*, *38*(6), 2763–2775. doi:10.1016/j.enpol.2010.01.008

Ferber, J., & Weiss, G. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence* (Vol. 1). Addison-Wesley Reading.

Franklin, S., & Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International workshop on agent theories, architectures, and languages* (pp. 21–35). Academic Press.

Ganek, A. G., & Corbi, T. A. (2003). The dawning of the autonomic computing era. *IBM Systems Journal*, *42*(1), 5–18. doi:10.1147j.421.0005

Goertzel, B., Looks, M., & Pennachin, C. (2004). Novamente: an integrative architecture for artificial general intelligence. *Proceedings of aaai symposium on achieving human-level intelligence through integrated systems and research*.

Goertzel, B., & Pennachin, C. (2007). The novamente artificial intelligence engine. In *Artificial general intelligence* (pp. 63–129). Springer. doi:10.1007/978-3-540-68677-4_3

Gotts, N. M., Polhill, J. G., & Law, A. N. R. (2003). Agent-based simulation in the study of social dilemmas. *Artificial Intelligence Review*, *19*(1), 3–92. doi:10.1023/A:1022120928602

Hogg, L. M., & Jennings, N. R. (1997). Socially rational agents. In *Proc. aaai fall symposium on socially intelligent agents* (pp. 8–10). Academic Press.

Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—Degrees, models, and applications. *ACM Computing Surveys*, *40*(3), 7. doi:10.1145/1380584.1380585

Huhns, M. N., & Bridgeland, D. M. (1991). Multiagent truth maintenance. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(6), 1437–1445. doi:10.1109/21.135687

Hutter, M. (2001). Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions. In *European conference on machine learning* (pp. 226–238). 10.1007/3-540-44795-4_20

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, *1*(1), 7–38. doi:10.1023/A:1010090405266

Johnston, J., Moscovitch, M., & Umilt'a, C. (2008). *The allure of machinic life: Cybernetics, artificial life, and the new ai*. MIT Press. doi:10.7551/mitpress/9780262101264.001.0001

Kant, J.-D., & Thiriot, S. (2006). Modeling one human decision maker with a multi-agent system: the codage approach. In *Proceedings of the fifth international joint conference on autonomous agents and multiagent systems* (pp. 50–57). 10.1145/1160633.1160639

Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187–219). Springer. doi:10.1007/0-387-27705-6_6

Kephart, J. O. (2005). Research challenges of autonomic computing. In *Proceedings of the 27th international conference on software engineering* (pp. 15–22). Academic Press.

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41–50. doi:10.1109/MC.2003.1160055

Kiesling, E., Gu¨nther, M., Stummer, C., & Wakolbinger, L. M. (2012). Agentbased simulation of innovation diffusion: A review. *Central European Journal of Operations Research*, *20*(2), 183–230. doi:10.100710100-011-0210-y

Kim, S., Lee, K., Cho, J. K., & Kim, C. O. (2011). Agent-based diffusion model for an automobile market with fuzzy topsis-based product adoption process. *Expert Systems with Applications*, *38*(6), 7270–7276. doi:10.1016/j.eswa.2010.12.024

Kinny, D., Georgeff, M., & Rao, A. (1996). A methodology and modelling technique for systems of bdi agents. In *European workshop on modelling autonomous agents in a multi-agent world* (pp. 56–71). 10.1007/BFb0031846

Komosinski, M., & Adamatzky, A. (2009). *Artificial life models in software*. Springer Science & Business Media. doi:10.1007/978-1-84882-285-6

Kraus, S. (1997). Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, *94*(1-2), 79–97. doi:10.1016/S0004-3702(97)00025-8

L'opez, F. L., Luck, M., & d'Inverno, M. (2006). A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, *12*(2-3), 227–250. doi:10.100710588-006-9545-7

Laird, J. E. (2008). Extending the soar cognitive architecture. *Frontiers in Artificial Intelligence and Applications*, *171*, 224.

Laird, J. E. (2012). *The soar cognitive architecture*. MIT Press. doi:10.7551/mitpress/7688.001.0001

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., P'erolat, J., . . . Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. In Advances in neural information processing systems (pp. 4190–4203). Academic Press.

Langton, C. G. (1997). *Artificial life: An overview*. MIT Press.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In Machine learning proceedings 1994 (pp. 157–163). doi:10.1016/B978-1-55860-335-6.50027-1

Macal, C. M., & North, M. J. (2005). Tutorial on agent-based modeling and simulation. In *Proceedings of the winter simulation conference*, *2005* (pp. 14–pp). Academic Press.

Manson, S. M. (2006). Bounded rationality in agent-based models: Experiments with evolutionary programs. *International Journal of Geographical Information Science*, *20*(9), 991–1012. doi:10.1080/13658810600830566

Maturana, H. R., & Varela, F. J. (1991). *Autopoiesis and cognition: The realization of the living* (Vol. 42). Springer Science & Business Media.

McAllester, D. A. (1990). *Truth maintenance* (Vol. 90). AAAI.

McMullin, B. (2004). Thirty years of computational autopoiesis: A review. *Artificial Life*, *10*(3), 277–295. doi:10.1162/1064546041255548 PMID:15245628

Mei, S., Zarrabi, N., Lees, M., & Sloot, P. M. (2015). Complex agent networks: An emerging approach for modeling complex systems. *Applied Soft Computing*, *37*, 311–321. doi:10.1016/j.asoc.2015.08.010

Meneguzzi, F., & Luck, M. (2009). Norm-based behaviour modification in bdi agents. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems-volume 1* (pp. 177–184). Academic Press.

Meng, X., & Chen, T. (2013). Event based agreement protocols for multi-agent networks. *Automatica*, *49*(7), 2125–2132. doi:10.1016/j.automatica.2013.03.002

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, *6*(4), 333–346. doi:10.1109/TEVC.2002.802450

Moreno, A., & Etxeberria, A. (2005). Agency in natural and artificial systems. *Artificial Life*, *11*(1-2), 161–175. doi:10.1162/1064546053278919 PMID:15811225

Mukherjee, S., Srinivasa, S., & D, S. C. (2008). Validating for liveness in hidden adversary systems. *Electronic Notes in Theoretical Computer Science*, *203*(3), 53–67. doi:10.1016/j.entcs.2008.04.086

Nedic, A., Ozdaglar, A., & Parrilo, P. A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, *55*(4), 922–938. doi:10.1109/TAC.2010.2041686

Pan, X., Han, C. S., Dauber, K., & Law, K. H. (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI & Society*, *22*(2), 113–132. doi:10.100700146-007-0126-1

Pan, X., Han, C. S., & Law, K. H. (2005). A multi-agent based simulation framework for the study of human and social behavior in egress analysis. In Computing in civil engineering (2005) (pp. 1–12). doi:10.1061/40794(179)92

Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, *11*(3), 387–434. doi:10.100710458-005-2631-2

Parker, D. C., Manson, S. M., Janssen, M. A., Hoffmann, M. J., & Deadman, P. (2003). Multi-agent systems for the simulation of land-use and land-cover change: A review. *Annals of the Association of American Geographers*, *93*(2), 314–337. doi:10.1111/1467-8306.9302004

Parsons, S., & Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *5*(3), 243–254. doi:10.1023/A:1015575522401

Patil, S., Srinivas, S., Mukherjee, S., Rachakonda, A. R., & Venkatasubramanian, V. (2009). Breeding diameter-optimal topologies for distributed indexes. *Complex Systems*, *18*(2), 175.

Patil, S., Srinivasa, S., & Venkatasubramanian, V. (2009). *Classes of optimal network topologies under multiple efficiency and robustness constraints. In 2009 ieee international conference on systems, man and cybernetics* (pp. 4940–4945). doi:10.1109/ICSMC.2009.5346272

Poslad, S. (2007, November). Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems*, *2*(4), 15. doi:10.1145/1293731.1293735

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a bdi architecture. *KR*, *91*, 473–484.

Rao, A. S., & Georgeff, M. P. (1995). *Bdi agents: from theory to practice* (Vol. 95). ICMAS.

Santos, F. P., Santos, F. C., Melo, F. S., Paiva, A., & Pacheco, J. M. (2016). Dynamics of fairness in groups of autonomous learning agents. In *International conference on autonomous agents and multiagent systems* (pp. 107–126). 10.1007/978-3-319-46882-2_7

Schaat, S., Wendt, A., Jakubec, M., Gelbard, F., Herret, L., & Dietrich, D. (2014). Ars: An agi agent architecture. AGI.

Schmidhuber, J. (2007). G¨odel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence* (pp. 199–226). Springer. doi:10.1007/978-3-540-68677-4_7

Schreinemachers, P., & Berger, T. (2006). Land use decisions in developing countries and their representation in multi-agent systems. *Journal of Land Use Science*, *1*(1), 29–44. doi:10.1080/17474230600605202

Seidl, D. (2004). Luhmann's theory of autopoietic social systems. *LudwigMaximilians-Universit¨at Mu¨nchen-Munich School of Management*, 36–37.

Semsar-Kazerooni, E., & Khorasani, K. (2009). Multi-agent team cooperation: A game theory approach. *Automatica*, *45*(10), 2205–2213. doi:10.1016/j.automatica.2009.06.006

Shoham, Y., Powers, R., & Grenager, T. (2003). Multi-agent reinforcement learning: a critical survey. *Web Manuscript*.

Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, *185*(3), 1155–1173. doi:10.1016/j.ejor.2006.06.046

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 2). MIT Press.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning* (pp. 330–337). 10.1016/B978-1-55860-307-3.50049-6

Tang, F., & Ren, A. (2008). Agent-based evacuation model incorporating fire scene and building geometry. *Tsinghua Science and Technology*, *13*(5), 708–714. doi:10.1016/S1007-0214(08)70115-9

Tennenholtz, M. (1999). *Electronic commerce: from economic and gametheoretic models to working protocols*. IJCAI.

Van der Hoek, W., & Wooldridge, M. (2003). Towards a logic of rational agency. *Logic Journal of the IGPL*, *11*(2), 135–159. doi:10.1093/jigpal/11.2.135

Varela, F. G., Maturana, H. R., & Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *Bio Systems*, *5*(4), 187–196. doi:10.1016/0303-2647(74)90031-8 PMID:4407425

Vidal, J. M., & Durfee, E. H. (1995). *Recursive agent modeling using limited rationality*. ICMAS.

Villez, K., Gupta, A., Venkatasubramanian, V., & Rieger, C. (2011). Resilient design of recharging station networks for electric transportation vehicles. In *2011 4th international symposium on resilient control systems* (pp. 55–60). 10.1109/ISRCS.2011.6016089

Von Neumann, J., Morgenstern, O., & Kuhn, H. W. (2007). *Theory of games and economic behavior (commemorative edition)*. Princeton University Press. doi:10.1515/9781400829460

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–292. doi:10.1007/BF00992698

Young, R. M., & Lewis, R. L. (1999). The soar cognitive architecture and human working memory. *Models of working memory: Mechanisms of active maintenance and executive control*, 224–256.

ENDNOTE

[1] http://www.fipa.org/

Chapter 2
Processing Data Streams

Parimala N.
Jawaharlal Nehru University, India

ABSTRACT

A data stream is a real-time continuous sequence that may be comprised of data or events. Data stream processing is different from static data processing which resides in a database. The data stream data is seen only once. It is too voluminous to store statically. A small portion of data called a window is considered at a time for querying, computing aggregates, etc. In this chapter, the authors explain the different types of window movement over incoming data. A query on a stream is repeatedly executed on the new data created by the movement of the window. SQL extensions to handle continuous queries is addressed in this chapter. Streams that contain transactional data as well as those that contain events are considered.

INTRODUCTION

Traditional data bases are flat structures that store static data. Insertions, deletions and updates to the data base do take place but the main interaction with the data base is that of querying. Queries are executed and the answer reflects the current state of the data base. The data bases support applications that are business data processing oriented. However, there is an emergence of a new kind of data which naturally occurs as a sequence of data values, referred to as a data stream. Examples include sensor data, financial ticker data etc. Data stream querying is substantially different from traditional data base querying. Data arrives continuously as a stream and queries have to be executed continuously. In this chapter, we consider extensions to SQL for processing stream data.

DOI: 10.4018/978-1-7998-2975-1.ch002

An entire stream cannot be stored on hard disk for computation as it is voluminous. Therefore, a part of the data referred to as a window of data is considered at a time for querying/analysis. That is, the query or analysis, as the case may be, is performed on the data in a window. The window moves over new data that has arrived. The movement itself can be captured in many different ways. It can be forward, backward, only end moving or both ends moving. The movement decides what data remains in the existing window and what data moves out of the window. The query/analysis is performed on the new window. In this manner, a query is issued once and is executed continuously over the new data that arrives in the data stream.

Traditional Data Base Management System (DBMS) was designed to handle static data that resides in a data base. Data Stream Management System (DSMS) was built to handle data streams. In some systems, it is possible to combine static data in a relation and the dynamic stream data for querying. In this chapter we explain a few of DSMS systems.

SQL is a de-facto standard for querying in a DBMS. Extensions to SQL were natural for querying data streams. DSMS incorporated extensions to SQL by retaining SELECT-FROM-WHERE format. The FROM clause was now essentially a stream or a combination of a stream and a relation. "window" feature was added appropriately.

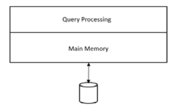
Stream processors handled events in a stream and performed event analytics like computing aggregate functions, raising an alarm etc. Streaming SQL became popular in 2016. Stream processors provided streaming SQL language as part of the interface. Window construct was naturally part of Streaming SQL. In this chapter, we consider stream processors from the point of view of streaming SQL.

DATA STREAMS

A data stream is a continuous, real-time, ordered sequence of data values produced by a data source as defined in (Golab & Özsu, 2003a). The data stream data is considered as a relational tuple or sequence of events ordered by time. The data values produced by a data source is seen only once as the old data moves out and new data takes its place. Further, the data is constantly changing as new elements are produced. It does not take the form of a persistent relation (Raghavan & Henzinger, 1999).

The typical applications which generate stream data are temperature sensor applications, financial tickers, stock market, call detail records in telecommunications etc. The data is generated in real-time and is continuous. The applications generate new data at every tick making the data large in volume and never ending. In other words, the data is unbounded and voluminous. It is too large to store entirely. Only a portion of the data is visible at any given point in time.

Figure 1. Traditional query processing



The data is considered ordered. The ordering may be explicit or implicit. Explicitly, a time stamp attribute can be associated with the data which can be used to order the data. Implicitly, the arrival time can be used for ordering the data, (implicitly by arrival time or explicitly by timestamp) sequence of items, too large to store entirely. The first property of a data stream is that it is continuously automatically generated by various data sources. The data that is used to evaluate a query changes constantly since newly generated data is part of the new evaluation.

Due to the continuous unbounded nature, it is not possible to store the entire stream in a data base. The traditional DBMS cannot be used to answer queries as data is arriving continuously. Therefore, a strategy known as *continuous queries* is adopted. Here, a query is repeatedly evaluated as and when new data arrives, producing new results continuously. The main goals of DSMS is to output result even while data is continuously arriving and all the data cannot be stored in memory (Panigati, Schreiber, & Zaniolo, 2015). At best, synapses are stored for future use.

TRADITIONAL VS DATA STREAM QUERY PROCESSING

In a traditional DBMS, the data is persistent and is stored on the disk. The storage is unbounded. As shown in Figure 1, the queries are processed using this persistent data. In traditional system, time is not associated with the data. No historical data is maintained. Therefore, it is not important to record the time at which data was relevant and only the data that is currently valid is maintained. A given query is executed only once using this data. There are no real-time services.

On the other hand, in a DSMS, query processors have to handle transient streams. As shown in Figure 1, data streams enter main memory from one end and drop off at the other end. The unbounded data is too large to be held in main memory. The query is processed by the query processor not once but continuously for the newly arrived data.

Figure 2. Data stream processing

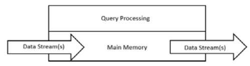


Table 1. A comparison

DBMS	DSMS
Persistent data	Transient data
Random access	Sequential access
Unlimited disk store	Limited main memory
Query once	Continuous queries
Little or no real time requirements	Real time
Low update rate	Continuously updated

The following differences are considered while developing a DSMS:

1. Data stream is infinite and hence cannot be stored on the disk as opposed to the data stored in a data base.
2. In a data stream, data is arriving continuously which implies rapid insertions or append whereas in a DBMS modifications to the data base are less frequent.
3. Data stream is unbounded and hence, cannot be queried in one go. A small portion of the stream is used for analytics at a time. Whereas in a DBMS, the entire data in a data base is used to answer a query.
4. A relation is defined as a set and hence in unordered. The ordering of data in a stream is of significance while answering a query.
5. A query is executed only once using the persistent data in a DBMS. In a DSMS, a query is continuously executed over newly arriving data.

Table 1 summarizes the differences between DBMS and DSMS.

Window

Streaming data is not stored on hard disk but retained in memory. It is practically impossible to store completely rapidly arriving data in memory. Further, in most applications old data may not have much significance. In view of these difficulties, a window of data is considered at a time in query formulation. A part of recently arrived data is considered every time. A bounded portion of the stream which is of interest at any given point in time is referred to as a *window*. A window limits the scope of processing. A window is mainly defined in two ways namely:

1. Physical window.
2. Logical window.

A physical window, referred to as a time-based window, assumes the existence of a time attribute. A time interval determines the data that forms part of the window. For example, a window of a temperature sensor can be defined to hold the temperature reading of the last 5 minutes. On the other hand, a tuple count based window, also referred to as a logical window, is defined in terms of the number of tuples that form a window. For example, a window may consist of 100 tuples. Window *size* is the value which is expressed as time unit or tuple count.

In addition, sometimes punctuations are inserted in a data stream to identify the end of a sub-stream. Punctuations are nothing but explicit markers that signify the end of a window. The end-of-processing markers are inserted by the applications. The sub-stream is of finite length and constitutes a window. The windows are of variable length depending on the data items. An application of punctuation based windows would be auctions. The end of each auction can be signaled by punctuation.

Immaterial of how a window is defined, there are two end points of a window (Golab & Özsu, 2003b). Each of these can be either fixed or moving. As a result we get three types of windows namely:

1. **Fixed Windows**: Here, both the end points are fixed.
2. **Sliding Window**: Here, both the end points move, either forward or backward. Moving end points imply that that new data is replacing the old data. The new data replaces the old data as the end points slide over the incoming data.
3. **Landmark Window**: If one of the end points is fixed and the other is moving, either forward or backward, then the window is referred to as landmark window.

Movement of a Window

The window of data is used for a single computation. Subsequently, the window moves over the input stream. The movement gives rise to different window movement for processing the streams (Abadi et al., 2003). Some of these are considered below.

Figure 3. Tumbling and hopping windows

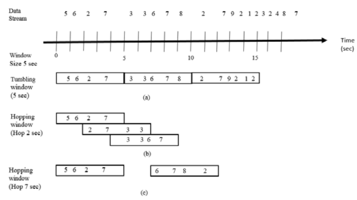
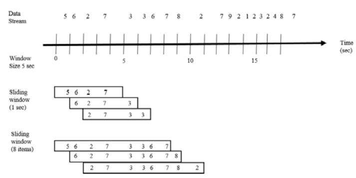


Figure 4. Sliding Window



Tumbling Window

Tumbling windows are fixed sized windows. The size may be determined by time or tuple count. The next window is placed right next to the current window. This placement continues as long as input stream produces data items. The windows do not overlap. That implies that each input is assigned to exactly one window. Tumbling window with a size of 5 sec is shown in Figure 3(a).

Hopping Window

Hopping window is also a fixed sized window but introduces a new parameter – hop size. The window jumps by hop size instead of moving by window size. The hop size can be defined by time unit or by a count of tuples. If the hop size is same as the window size, then we get tumbling window. If the hop size is less than the window size, then the windows overlap. In this case, a data item can belong to more than one window. Figure 3(b) depicts a hop size of 2 seconds which is less than the window size of 5 seconds. Therefore, the windows overlap. If the hop size is larger than the window size, then there is a gap between the windows. Some data items which are in the gap do not belong to any window. In Figure 3(c), the items 3 and 3 do not belong to any window and are not processed.

Sliding Window

As brought out earlier, in a sliding window both ends move. It can be viewed as a hopping window with hop size tending to zero. This can give rise to infinite number of windows. In practice, the quantum of movement is discretized either using time unit or a count of data items. Both of these cases is shown in Figure 4.

A given query is evaluated for the data in a window. As new data appears in the window, the query is re-evaluated. In the case of a sliding window, as new data appears in the window, two issues need to be addressed – re-evaluation strategies and tuple invalidation procedures (Golab & Özsu, 2003b). These are *eager re-evaluation* vs *lazy re-evaluation* and *eager expiration* vs *lazy expiration*. In the case of an *eager re-evaluation* of a query, the query is re-executed every time a new tuple arrives in the window. This may be not practically possible if the input rate is very high. In the case of *lazy re-evaluation*, the re-evaluation is delayed till a certain time period has elapsed. When the time period elapses, the window is updated with the new tuples that have arrived. The new tuples, beyond the data in the current window, have to be maintained in memory which puts additional constraints on memory requirements. Similarly, in *eager expiration*, an old tuple is removed the moment a new tuple arrives. On the other hand, in *lazy expiration* the old tuples are removed from the window after a time period.

MONOTONIC VS NON-MONOTONIC QUERIES

Persistent queries over data streams are generally classified into two categories- *monotonic* and *non-monotonic*. Let $Q(\tau)$ be the answer of a continuous persistent query Q at time τ. Q is monotonic (Golab & Ozsu, 2005) if $Q(\tau)$ Í $Q(\tau + \varepsilon)$ for all τ and all ε ³ 0 . In the case of monotonic queries the new result is appended and data is never deleted from the result. For example, when the operator is projection, whenever a new tuple arrives the unwanted columns are deleted and the tuple with the remaining attributes is appended.

A non-monotonic query is defined as the continuous persistent query for a window such that the continuous or periodic results which are produced may be appended to the output stream but are not related to the previous ones.

Consider, now, blocking operators. According to (Babcock, Babu, Datar, Motwani, & Widom, 2002) "A blocking query operator is a query operator that is unable to produce the first tuple of the output until it has seen the entire input." Selection returns an input tuple if it satisfies the condition. So it is non-blocking operator for a stream. So is the conventional projection operator. For each input tuple, there is an output tuple with only the required column. However, the traditional aggregate operators (MAX, AVG, etc.) are blocking.

It has been shown in (Law, Wang, & Zaniolo, 2011) that non-monotonic query operators are blocking whereas monotonic operators are non-blocking.

In data stream processing, a window of data is processed at a time. In the case of a sliding window, tumbling or hopping window, the queries are non-monotonic. As the window moves, the persistent query is reevaluated and produces a new answer. Even though the stream tuples are part of many successive windows when the window slides, the current output set may not necessarily subsume its preceding one, hence it is non-monotonic.

In the landmark window one point is fixed and the new tuples are appended at the other end. The operators like selection, projection, merge-union, difference, and duplicate elimination produce append-only output tuples. So, only the landmark window produces monotonic results.

As brought out earlier, punctuated streams are the streams with punctuations and punctuations are nothing but the explicit markers that signify the end of sub-streams. The punctuation mark guarantees that no tuple after it will be part of the subset which is to be evaluated. As a result, the blocking operator gets unblocked and starts processing the data and outputs the result.

Timestamp

A stream consists of data which is normally ordered by time. The ordering is achieved by the timestamp attribute. There are two types of timestamps namely, Explicit timestamp and Implicit timestamp. These are explained below.

- **Explicit Timestamp**: The timestamp is assigned by the source producing the stream. For example, when the temperature sensor outputs the temperature value, an additional attribute which show the time the temperature was recorded is an explicit timestamp.
- **Implicit Timestamp**: Implicit timestamp an integer value assigned to the timestamp field by the destination.

Generally, explicit timestamp is the time marked by the physical clock and the implicit timestamp is the logical clock which is the order of arrival of every incoming tuple. There are some pros and cons of using explicit timestamp. The system does not have to be bothered about whether the tuples have arrived in the right order. But there is a possibility that a tuple with a timestamp t-1 arrives after a tuple with timestamp t. In this case, the processor has to make provisions for this unordered arrival of ordered data items. This increases processing and memory requirement time. If there are multiple sources, then the problem gets compounded. For these reasons, implicit timestamp is sometimes preferred over explicit timestamp.

PROCESSING CONTINUOUS QUERIES

SQL is a de-facto standard for querying data bases. There are many properties of SQL which has led to adopting SQL for continuous querying. SQL is a declarative language which does not specify the manner in which queries are to be computed. SQL query optimization techniques are well known and can be extended to SQL queries on data streams. SQL is known by most users, and therefore, it is easier for users to adapt to SQL queries over data stream. In this chapter, SQL extensions to querying data streams are addressed.

Relation Based Systems

The first attempt to support continuous query of data streams within the context of a DBMS was made in Tapestry project (Terry, 1992). The database was append-only database and streams append-only tables. The application that was considered was e-documents like e-mails and news messages. The concept of long-running continuous queries was introduced in this system. Tapestry Query language (TQL) was proposed to write ad-hoc query and then convert it to a continuous query. New updates to the data base implied re-evaluation of the query. Instead of answering the query with the old and the new tuples (where most of the result would be part of an earlier execution of the same query), the system used incremental evaluation scheme. A TQL query was transformed to a monotone query, incrementalized and then converted to a SQL query. The continuous evaluation was expressed via a “FOREVER DO “ “ENDLOOP” with a “SLEEP” clause where the query was re-executed after every SLEEP period, that was defined, got over. The relationship between monotonic and non-blocking queries was defined in this system.

STREAM, (Arasu, et al., 2003a) for STanford stREam datA Manager, was a general-purpose DSMS which was developed at Stanford. Streams consisted of data and a time stamp. A partial ordering on the timestamp value was assumed. STREAM proposed CQL, the Continuous Query Language which is an extension of SQL for querying data streams (Arasu, Babu, & Widom, 2003b). CQL converts streams to relations and then the query is executed in the normal way, using relations. The result which is a relation is converted to a stream. Thus, there are three operators, stream-to-relation, relation-to-relation, and relation-to-stream. Window operators belong to the stream-to-relation class and pick up data from the stream as per the window definition. The data is transformed to a relation. Standard query processing is applied to produce a result which is a relation. Output relations are then transformed to streams via one of three operators: RStream, IStream, or Dstream. The RStream is obtained from the relation R at time t consisting of the content of R present at time t. The IStream of a relation R contains all the tuples that were added to R between time t – 1 and t. The DStream of a relation R at time t is defined as it contains the tuples that were recently removed from R and is the inverse on IStream. A modified example from (Arasu, Babu, & Widom, 2003b) is given below:

```
Select vehicleId, speed
From PosSpeedStr [Range 20 Seconds]
```

Here, PosSpeedStr is an input stream and Range 20 Seconds specifies a sliding window of 20 seconds. The data is converted to a relation and the SELECT part of the query is executed. This is essentially the application of projection operator on a relation. The result is converted to a stream.

GigaScope (Cranor, 2003) was a stream data base for network applications. It proposed a query language, GSQL, a pure stream query language with SQL-like syntax. The input as well as the output was a data stream. It was assumed that the data stream has an ordering attribute such as time stamp. The ordering attribute was used for querying. GSQL does not implement continuous queries. Therefore, there is no notion of a sliding window. Instead, the queries are executed directly on data streams. This approach is acceptable for non-blocking operators. Since, blocking operators need a fixed set of stream data for evaluation, GSQL creates a window by using the timestamp and ordering attribute of incoming data. The blocking operators now become essentially stream operators. The attributes of the data which are to be used for ordering is defined in the data definition language. GSQL queries are translated into C or C++ code. A modified example query from (Cranor, 2003) is given below:

```
Select count(*)
FROM Dest
Group by time/60
```

Here, the bucket is a minute long since the attribute time is a 1-second timer. GSQL had some of the SQL operators like selection, aggregation with group-by, and join with predicates including the ordering attributes. The new merge operator defined by the GSQL was an order preserving operator used for the merging of two streams using ordering attributes of the stream.

The Telegraph project (Chandrasekaran, 2003) was also developed as a general-purpose DSMS. The aim was to support network applications which have large amounts of data. StreaQuel was the query language of Telegraph DSMS. A StreaQuel query was expressed using SQL syntax. The system supports windows beyond sliding and landmark windows. In order to do so a for-loop construct is introduced in the SQL syntax. An example of for-loop is given below:

```
Select ….
From…..
Where
for (t = 101; t <= 1000; t++)
{ WindowIs(StockPrices, 101, t); }
```

Here, if the unit of time is one day, the for-loop is executed every day starting after the 100th day upto 1000th day. The WindowIs construct identifies the stream, its left end and its right end. This is a landmark query where the first point in the timeline is fixed and the last point moves with time. It is also possible to express a sliding window query. Even though the for-loop is a low level construct, it is possible to express queries over different kinds of windows. The output of a StreaQuel query is a set. When the query is repeatedly executed, each set is time stamped with the time at which it was executed. However, the output stream cannot be used as an input stream to a subsequent query.

AQuery was a SQL-based language defined in (Lerner & Shasha, 2003) which was defined for the relational model of data but introduced an order on the data. The creation of a table included an ORDERED BY clause by which the data was ordered using the attribute specified in this clause. The tables are called arrables (for array tables) as they are treated as arrays. AQuery was a semantic extension of SQL-92. The order was introduced in the query language as a declarative and the extension was introduced in the FROM clause. The extension specified the order of arrable as ASSUMING ORDER. Order dependent functions (like first) in contrast to order independent functions (like count) of SQL were also introduced. Essentially, the data stream tuples were the rows in the table whose columns were treated as arrays, on which order-dependent operators such as next, previous, first and last could be applied. A modified example from (Lerner & Shasha, 2003) is given below:

```
SELECT first(10, client)
FROM Connections
ASSUMING ORDER Timestamp
```

Table 2. Windows

Tumbling window	Group By
Hopping window	Multi Group By
Sliding window	Window functions
Cascading window	Window functions

The query requests the first ten clients who were present in Connections.

The Aurora project (Abadi et al., 2003) was built as a DSMS for monitoring streams. The applications considered were financial trading applications, telecommunications systems monitoring, and government and military surveillance tasks. This system monitored the continuous data streams and automatically triggered to an abnormal event in the stream thereby made a database active and human passive system. SQuAl is the query language of Aurora DSMS. A query in SQuAl could be expressed via a graphical interface by arranging query operators and joining them with directed arcs to specify data flow. The system had a network of triggers where each trigger was a data-flow graph with each node being one among the eight built-in operators like slide, filter, drop, join, latch, etc. The user could add operators depending on the type of application and further the operators at the later stage of design could be removed if found unnecessary. The Aurora Storage Manager (ASM) of the Aurora DSMS managed the storage of tuples that were required for an Aurora network. The system could handle both compile-time optimization and run-time optimization of the trigger network. For compile-time optimization the operators could be re-ordered and for the run-time optimization load shedding could be performed when the system was about to run out of memory.

Apache Calcite (Begoli, Camacho-Rodríguez, Hyde, Mior & Lemire, 2018), is a framework using which databases and data base management systems can be built. In Apache Calcite SQL has been extended to handle streams. Querying streams is seamlessly integrated into the system. There is no difference in the syntax between SQL used for relations and streams. If the target is a stream, the query operates on a stream rather than a table. In the example below:

```
Select Stream *
From Order
```

selects all the rows and columns from Order. Of course, Order must be defined as a stream in the schema. Let the definition be Order (timestamp, OId, PId, qty). The query on a stream never terminates. As new data arrives on the input stream, it forms part of the output stream in the above query. A WHERE clause can be used as in SQL. The interesting part is the manner in which windows are handled. The system supports four types of windows. These are expressed using the GROUP BY clause or window functions. The types of windows and the SQL stream clause that supports them are given in Table 2.

In this chapter, an example of tumbling window is considered. Recall that tumbling windows are non-overlapping and of fixed size. The windows are contiguous. Consider the Order schema. Suppose it is desired to get the subtotal of the units of a product ordered every hour. Here, the subtotals are desired for a tumbling window of one hour. It can be expressed as:

```
SELECT STREAM CEIL(timestamp TO HOUR) as perhour
PId, sum(units)
FROM Order
GROUP BY CEIL(timestamp TO HOUR), PId
```

Here, the output stream consists of total units ordered per hour. The timestamp field is used to group orders per product but for each hour as separate groups.

Many stream processors like Apache Flink (Carbone et al., 2015) and Samza (Noghabi et al., 2017) have integrated with Calcite so that they can provide streaming SQL. Stream Processors are dealt with next.

Non-relational Systems

COUGAR (Bonnet, Gehrke, & Seshadri, 2001) was an object-based system to support continuous queries on data streams. The abstract data types (ADT) were defined for the stream data and SQL was used to query the object-based language. In this object-based model the sources and item types were modeled as (hierarchical) data types with associated methods. The COUGAR system managed the sensor data using ADTs one for each sensor, whose interface consisted of signal-processing methods supported for a particular type of sensor. The query language of the system had SQL-like syntax. In order to support continuous query execution a $every() clause indicated the frequency of query re-execution. A simple example would be, a query that runs every thirty seconds and returns temperature readings from all sensors on the fifth floor of a building could be specified as follows:

```
SELECT R.s.getTemperature()
FROM R
WHERE R.floor = 5 AND $every(30).
```

Tribeca (Sullivan, 1996) supported querying over network packet streams. The stream was modeled as structured records and supported the stream contents as per the network protocol hierarchy. The query language used in Tribeca accepted a single stream as input, and returned one or more output streams. The query language was data-flow oriented. Queries could be compiled and executed on a stream of data. Also, a set of stream-specific operators like qualification, multiplex, de-multiplex were defined that were different from the standard SQL relational operators.

QUERYING RDF DATA

DSMS handle data streams where the stream data type is an extension of a relation. Attempts have been made to query streams which have a format different from a relational tuple. Streams of RDF data is considered in (Barbieri D. F., 2010). Resource Description Framework (Lassila & Swick, 1998) model is used to express metadata about properties and property values. SPARQL (Prud'hommeaux & Seaborne, 2017) has emerged as a standard language for querying RDF data.

Continuous SPARQL (C-SPARQL) is a language designed for continuous queries over streams of RDF data (Barbieri, Braga, Ceri, Valle, & Grossniklaus, 2010). It extends SPARQL to handle continuous queries. The RDF stream data consists of a pair of RDF triple and a timestamp. The stream is identified using an IRI locator which has an IP address and a port for accessing data. The query language supports both, logical and physical window. For example, the query below selects objects in the last 30 minutes with a sliding window of 3 minute.

```
SELECT DISTINCT ?object
FROM STREAM [RANGE 30m STEP 3m]
```

A new RDF stream can be the result of a C-SPARQL query. A new stream must be registered using REGISTER statement. An example for generating RDF stream for the above query is given below.

```
REGISTER STREAM New
CONSTRUCT ?object
FROM STREAM [RANGE 30m STEP 3m]
```

STREAM PROCESSORS

Stream processors are software platforms to enable users to process streams, analyze them and sometimes, respond to the data (Dayarathna & Perera, 2018). Imagine that a trajectory of person's movement has to created. The co-ordinates of the person are continuously arriving. A moving average has to be computed. As another example, consider temperature sensors in a room which are continuously sending the temperature in a room. If the temperature exceeds a certain threshold, then an action may have to be taken to create an alert. Stream processors get the events from the input stream, processes it and produces a result. In the first example, the result of processing may be used to update a web page.

Early stream processors like Apache Storm (Iqbal & Soomro, 2015) and Apache Spark (Zaharia et al., 2016) expected users to write code and place them inside agents. Writing code is not really a high level interface. Even DBMS has moved forward from writing DML programs by providing a query language like SQL. Stream processors quickly adopted SQL for streams. The language is called stream SQL or streaming SQL.

Streaming SQL as the name implies is SQL for streams as against the standard SQL which operates on relation. Unfortunately, there is no standard defined yet for streaming SQL. Streaming SQL has been incorporated in many stream processors. Some of them are considered below.

WSO2 stream processor (Jayasekara, Perera, Dayarathna & Suhothayan, 2015) is a streaming SQL based platform for processing events. Siddhi Streaming SQL (Suhothayan et al., 2011) is an SQL-like language using which stream processing and event processing logic can be expressed. The input to a query can be one or more event streams and zero or one table. The output is another stream or table which may be processed further. A stream can be defined as follows:

```
define stream Tstream (id int, floor int, temp double);
```

A query contains an input section which specifies the input stream and an output section which specifies the output stream. Nestled between these two is the select clause which is called the projection section. For example, in the query given below Tstream is the input stream and Ostrem is the output stream. The query is a SQL query which operates on Tstream and produces Ostream:

```
from Tstream
select floor, avg(temp)
group by floor
insert into Ostream;
```

In this system, a window can be either a sliding or a tumbling window. If a window is to be used with a stream then the stream must be suffixed with #window. The length of hte window is given as length(number). The number specifies the number of events. For example, in the query given below, the maximum temperature of the last 5 events is selected which is inserted into Ostream:

```
from Tstream#window.length(5)
select max(temp)
insert into Ostream;
```

The window above is a sliding window. Tumbling window can be specified using lengthBatch() instead of length().

Azure Stream Analytics (Gunda, 2019) which is developed by Microsoft is an event-processing engine that analyzes event streams from more than one source at the same time. The input streams can be click streams, sensors, social media feeds, geospatial data, data from IoT devices. Patterns and relationships are identified which may trigger actions or initiate workflows. Stream analytics query language is a SQL-like language which operates on streams of events. All events have a well defined timestamp. The timestamp can be chosen to be one of the attributes of the incoming data stream by using the keyword TIMESTAMP BY. For example, the query given below selects creation_time as the timestamp for the twitter stream data:

```
SELECT topic
FROM twitter_stream TIMESTAMP BY creation_time
```

Azure query language (Kinsman, McCready & Krishnamurthy, 2016) supports SQL like query constructs. Of interest, in this chapter, is the specification of windows for continuous queries. It supports four types of windows namely hopping window, session window sliding window and tumbling window. The window option is specified in the Group By clause. The data in a window is grouped on an attribute. A hopping window is an overlapping window where the window hops by some unit each time there is to be a movement. A hopping window is specified using the keyword HoppingWindow. It has three parameters. The first is the unit of time to be considered, the second is the size of the window and the third the hop size. An example is given below:

```
SELECT topic, count(*)
FROM twitter_stream TIMESTAMP BY creation_time
GROUP BY Topic, HoppingWindow (second, 20, 5)
```

This query defines a window of 20 seconds hopping each time by 5 seconds. For each window, the query is executed to display the topic and its count.

A sliding window is similarly specified but by using the keyword SlidingWindow. It has two parameters – the unit of time and the size of the window. For example, the query below specifies a window size of 5 seconds:

```
SELECT topic, count(*)
FROM twitter_stream TIMESTAMP BY creation_time
GROUP BY Topic, SlidingWindow (second, 5)
```

When a sliding window is specified, there are potentially infinite answers. Azure outputs data only when the content of the window changes:

Recall that tumbling windows are non-overlapping, fixed size, contiguous windows. The parameters to a tumbling window are unit of time and window size. Below is a query using tumbling window:

```
SELECT topic, count(*)
FROM twitter_stream TIMESTAMP BY creation_time
GROUP BY Topic, TumblingWindow (second, 5)
```

The last of the window options is SessionsWindow. It has three mandatory parameters – unit of time, timeout and maximum duration. A session starts when an event arrives. If no additional event arrives within timeout then the window is closed. On the other hand, if new events arrive with timeout period, then the window is extended till maximum duration is reached. The query given below outputs twitter counts the twitter for a maximum duration of 10 minutes filtering out periods up to 3 minutes where no twitter arrives:

```
SELECT topic, count(*)
FROM twitter_stream TIMESTAMP BY creation_time
GROUP BY Topic, SessionWindow (minute, 10, 3)
```

CONCLUSION

In this chapter we have considered queries over data streams. The queries are continuously evaluated over the incoming data. In order to have a finite set of data for query computation, a window was defined. Different kinds of window movement has been explained.

Data stream management in the early years was applied to transactional data. DSMS, as different from DBMS, were built. In this chapter, for DSMSs under consideration, the emphasis has been to explain the manner in which SQL has been extended and windows have been implemented.

In recent times, stream processing has gained popularity. The stream in this case is a stream of events. The emphasis of this chapter on stream processing systems has also been to explain the manner in which streaming SQL is implemented and the forms of windows that are supported by these systems.

REFERENCES

Abadi, D. J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., & Zdonik, S. (2003). Aurora: A new model and architecture for data stream management. *The VLDB Journal*, *12*(2), 120–139. doi:10.100700778-003-0095-z

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., & Motwani, R. (2003a). Stream: The stanford stream data manager. *IEEE Data Eng. Bull.*, *26*(1), 19–26.

Arasu, A., Babu, S., & Widom, J. (2003b). CQL: A language for continuous queries over streams and relations. In *International Workshop on Database Programming Languages* (pp. 1-19). Springer.

Arasu, A. B., Babu, S., & Widom, J. (2006). The CQL continuous query language: Semantic foundations and query execution. *The VLDB Journal*, *15*(2), 121–142. doi:10.100700778-004-0147-z

Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002). Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 1-16). ACM. 10.1145/543613.543615

Barbieri, D. F., Braga, D., Ceri, S., Valle, E. D., & Grossniklaus, M. (2010). Querying rdf streams with c-sparql. *SIGMOD Record*, *39*(1), 20–26. doi:10.1145/1860702.1860705

Begoli, E., Camacho-Rodríguez, J., Hyde, J., Mior, M. J., & Lemire, D. (2018). Apache calcite: A foundational framework for optimized query processing over heterogeneous data sources. *Proceedings of the 2018 International Conference on Management*. 10.1145/3183713.3190662

Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache flink: Stream and batch processing in a single engine. *A Quarterly Bulletin of the Computer Society of the IEEE Technical Committee on Data Engineering*, *36*(4).

Chandrasekaran, S. C. (2003). Telegraphcq: Continuous dataflow processing for an Uncertain world. In Cidr (Vol. 2, p. 4). Academic Press.

Cranor, C. J. (2003). *Gigascope: a stream database for network applications. In 2003 ACM SIGMOD international conference on Management of data* (pp. 647–651). ACM. doi:10.1145/872757.872838

Dayarathna, M., & Perera, S. (2018). *How to Choose a Stream Processor for your App.* Retrieved from www.infoq.com

Golab, L., & Ozsu, M. (2005). Update-pattern-aware modeling and processing of continuous queries. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 658-669). ACM. 10.1145/1066157.1066232

Golab, L., & Özsu, M. T. (2003a). Issues in data stream management. *SIGMOD Record*, *32*(2), 5–14. doi:10.1145/776985.776986

Golab, L., & Özsu, M. T. (2003b). Processing sliding window multi-joins in continuous queries over data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29* (pp. 500-511). VLDB Endowment. 10.1016/B978-012722442-8/50051-3

Gunda, S. e. (2019). *What is Azure Stream Analytics?* Retrieved from https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction

Iqbal, M. H., & Soomro, T. R. (2015). Big data analysis: Apache storm perspective. *International Journal of Computer Trends and Technology*, *19*(1), 9–14. doi:10.14445/22312803/IJCTT-V19P103

Jayasekara, S., Perera, S., Dayarathna, M., & Suhothayan, S. (2015). Continuous analytics on geospatial data streams with WSO2 complex event processor. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems* (pp. 277-284). ACM. 10.1145/2675743.2772585

Kinsman, M., McCready, M., & Krishnamurthy, V. (2016). *Stream Analytics Query Language Reference.* Retrieved from https://docs.microsoft.com/en-us/stream-analytics-query/stream-analytics-query-language-reference

Lassila, O., & Swick, R. R. (1998). *Resource description framework (RDF) model and syntax specification.* Retrieved September 11, 2019, from https://www.w3.org/TR/REC-rdf-syntax/

Law, Y. N., Wang, H., & Zaniolo, C. (2011). Relational languages and data models for continuous queries on sequences and data streams. *ACM Transactions on Database Systems*, *36*(2), 8. doi:10.1145/1966385.1966386

Lerner, A., & Shasha, D. (2003). AQuery: query language for ordered data, optimization techniques, and experiments. In *Proceedings 2003 VLDB Conference* (pp. 345-356). Morgan Kaufmann. 10.1016/B978-012722442-8/50038-0

Noghabi, S. A., Paramasivam, K., Pan, Y., & Ramesh, N., B. J., Gupta, I., & C. R. (2017). Samza: stateful scalable stream processing at LinkedIn. *Proceedings of the VLDB Endowment*, *10(12)*, 1634-1645. 10.14778/3137765.3137770

Panigati, E., Schreiber, F. A., & Zaniolo, C. (2015). Data streams and data stream management systems and languages. Data Management in Pervasive Systems, 93-111.

Prud'hommeaux, E., & Seaborne, A. (2017). *SPARQL query language for RDF.* W3C Recommendation.

Raghavan, P., & Henzinger, M. (1999). Computing on data streams. *DIMACS Workshop External Memory and Visualization*, 50, 107.

Suhothayan, S., Gajasinghe, K., Loku Narangoda, I., Chaturanga, S., Perera, S., & Nanayakkara, V. (2011). Siddhi: A second look at complex event processing architectures. *Proceedings of the 2011 ACM workshop on Gateway computing environments*. 10.1145/2110486.2110493

Terry, D. G. (1992). *Continuous queries over append-only databases*. Academic Press.

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... Ghodsi, A. (2016). Apache spark: A unified engine for big data processing. *Communications of the ACM*, *59*(11), 56–65. doi:10.1145/2934664

Chapter 3
A Declarative Approach to Systems Development

Deepak Kumar Sharma
I. K. Gujral Punjab Technical University, India

ABSTRACT

The declarative approach specifies what is to be done rather than how to do it. When adopted in information systems development, this implies that the system should be seen as a collection of business rules that can be enacted using a business rules engine. Business rules should be expressed in a form that is as close to the one in which business people perceive the rules. A business rules management system is needed to acquire, store, and allow modification of a business rules database. The rules are then handed over to a rules engine for enactment. The BRMS considered in this chapter uses an antecedent-consequent form for representing rules. These are based in a first order logic. Rules are formed with courses of actions and conditions in rules antecedents and courses of actions in rule consequents. It also introduces notions of state change in the business rule and temporal relation within rule and between different rules. Business rules are structured into atomic, complex, and abstract rules. The business rules are translated into enactment rules and converted to Java.

INTRODUCTION

Business rules have been extensively investigated in Artificial intelligence & Expert systems, Databases and Software engineering/Information system development domain. In the Expert systems, Rule has been used as a medium of knowledge representation and knowledge inference in rule-based systems (Kingston, 1987).

DOI: 10.4018/978-1-7998-2975-1.ch003

Languages like PROLOG (Colmerauer & Roussel, 1996) and LISP (Norvig, 1991) have been developed to infer knowledge form using rules. Forgy (1982) developed a Patten matching algorithm, namely the Rete Algorithm, to implement production rule system/expert systems. Various expert systems shells also referred as rule engines have been developed. Droople, ILOG JRule, JEES, BizTalk are some of the popular rule engines implementing Rete. In the software systems, Rules Engines are used to execute business rules at runtime (Ernest, 2008).

In software systems development, business rules are recognized as important because such ruler describe how to do business (Ioana & Ileana, 2007). A business rule can be either used to express business requirements, or define the constraints under which a business operates. As the business grows and its operations expand, the rules change/evolve and this may happen at varying frequencies, sometimes very frequently. In order to capture this dynamic nature of business rules and adopt rule changes, the software system should be flexible enough to adapt such changes gracefully. Traditional procedural approaches are known to be relatively inflexible as compared to declarative approaches to systems development.

Declarative approaches aim to specify needs, perhaps, as rules, and rely on the existence of specific platforms that "execute" these declarations. The idea is to manage and store rules and enact these as and when required. Such systems have been referred to as Business Rule Management Systems, BRMS. These systems use a rule engine as a core component to perform inferencing (Husemann & Schafer, 2006). BRMS (Oubelli & Oussalah, 2016; Zoet, 2014).

A BRMS hides the complexity of rule management and provides:

- A repository to store rules.
- A user interface to write, manage and evolve business rules.
- A rule engine to invoke and execute rules.

A BRMS allows business people to define and maintain its operations and reduces the intervention of IT people in rule management. It is crucially dependent on the formulation of the right business rules that faithfully reflect the business. It may also get tied to the specific vendors and moving away from legacy platforms may be difficult.

To be sure, there is a body of opinion that is not favorably disposed towards the BRMS approach. According to Smith (2017) rule engines provides a complicated solution to simple problems. Further, the challenges any organization face during software development are about how to represent their core function like processes and logic instead of how to execute logic. According to (Mahmoud, 2005; Pau, 2010) the BRMS approach just makes thing worse due to lack of standards and ends up "deterring businesses from using rule-based applications".

On the other hand, a number of attempts have been made around the BRMS approach. One such attempt was made by Rai (2003). Rai proposes to move from business rule to UML diagrams. This provides a way to support the software engineering view with business rules as the starting point. This approach does not find favor with agile development. According to Balte (2013), instead of creating a formal diagram in UML notation, developers prefers to create their own scratch diagram in agile development (Balte, 2014; Balte, 2016). Yet, UML diagrams are not to be discarded (Bhandari, 2017). Indeed, UML is a powerful tool to graphically represent a software system. Therefore, in agile development, the UML diagrams are created using reverse engineering approaches: generate UML diagram from code so as to have a design document of the system for future enhancement and understating.

The crucial step in the declarative approach to system development is to obtain the set of business rules. This requires a business rules model that can be instantiated by business people to yield the set of business rules of interest.

Thereafter, the business rules must be enacted so as to “implement” the system represented as rules.

We consider a technique to automatic generate Java class template from rules. The generated Java class template consists of the following information:

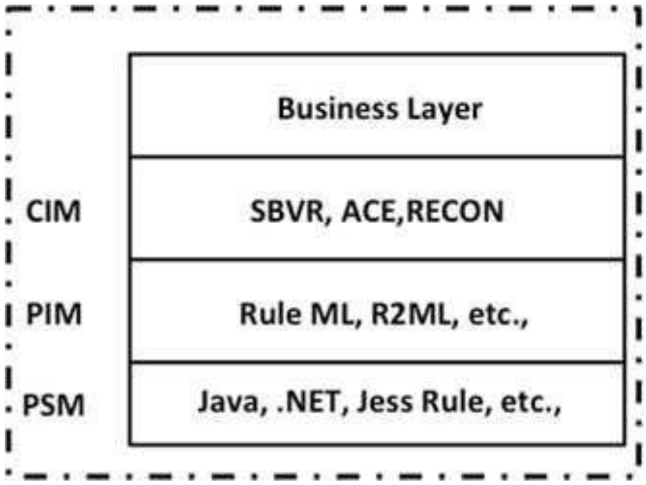
- The classes need to be created.
- Identify the core attributes and functions.
- Identify attributes needing to be changed while function executes.
- Pre-conditions for the execution of a function.
- The order of function execution.

Further, the developer can easily extend the generated Java class template to full functional class.

THE BUSINESS RULE MODEL

The Business Rules Manifesto (Ross, 2003), BRM, the Business Motivation Model (OMG, 2011), BMM, and SBVR (OMG, 2008) give high importance to the notion of a business rule. The BRM issues a call to give primacy to business rules. Thus it says, “Rules are first-class citizens of the requirements world” and “Rules are essential for, and a discrete part of, business models and technology models” respectively. The Business Motivation Model treats business rules as directives and leaves it to SBVR to define them. SBVR expresses rules in the extended first order logic and is positioned at the CIM layer of OMG MDA.

Figure 1. Business rules in MDA



However, the question remains as to whether a CIM level proposal is the one that business people would be comfortable with. To address this, we looked at business rules as organized in the three-layered architecture of MDA (OMG, 2011), (Gaweł & Skalna, 2012) shown in Fig.1. Business rules at the CIM layer are an abstraction of business rules, RuleML (Horrocks, Peter, Boley, Tabet, Grosof & Dean, 2004), R2ML (Giurca, 2007), etc. of the PIM layer. However, the three proposals of the CIM layer of Fig. 1 do not perform an ab-initio examination of business rules and do not ask questions like:

- What is the essential nature of business rules in a business?
- What is their structure?
- What are their properties?
- What are the inter-relationships between business rules?

We need a model that should answer these question thereby reflecting business rules as they really are. The aim of this chapter is to discuss a model for business rules from the business perspective and develop a technique to arrive at system requirements from business rules expressed in this model. We refer to this model as Business Rules Oriented Business Model (BROBM).

Evidently, we need to investigate the world of business to provide us our model. It is for this reason that we propose a fourth layer, the business layer, on top of the CIM layer (see Fig. 1) where we address these questions. Further, we develop a BRMS, a business rule management system, for business rules expressed in this model.

DESIDERATA OF BROBM

Now, as mentioned earlier, we need to look into the world of business to obtain the concepts of our model. For this, we revisit the Business Motivation Model to extract the main concepts around business rules. This provides to us a list of desired features that business rules should have from the business perspective.

The BMM was developed from the business perspective and provides a "scheme or structure for developing, communicating, and managing business plans in an organized manner". BMM is organized as a set of business concepts and inter-relationships between them.

BMM says that a business rule is a directive that guides/governs a course of action. A course of action may enable another course of action and a business rule governs this 'enabling' as well (Prakash, Sharma, & Singh, 2013), (Prakash, Sharma, & Singh, 2014). We draw two conclusions from this:

1. The structure of courses of action has a strong bearing on the structure of business rules. If the former is 'flat', non-hierarchical in nature then the business rule that governs it is also flat. On the other hand, if a course of action is complex, hierarchical, then its business rule is complex and must have business rule components that govern the component courses of action of the complex course of action.
2. Enabling of a course of action by another is a source of relationships between business rules. That is, the business rule for enabling establishes a relationship between the business rule of the enabled course of action and the business rule of the course of action that enables it.

It follows that BROBM must have features for structuring business rules: complex rules would be defined from simpler component rules until atomic rules are reached. Additionally, we need to represent the relationship, enables, between business rules.

We can infer the third requirement of the business layer from the BMM view that an End can be expressed in terms of states. Thus, BMM says that a Vision is about the 'future state' of a business. Similarly, an intended result (goal or objective) is a state that is to be brought about or sustained. Changes in state are the result of courses of action. Thus, state changes are governed/guided by business rules and it is therefore possible to relate business rules to state changes. Evidently, we need to model states, state changes and their relationship with business rules.

The fourth and last requirement of the business layer is that business layers have temporal properties. We obtain this from the notion of the objective of BMM. An objective is SMART, Specific, Measurable, Achievable, Relevant and Time bound. If the intended result is time bound, then the course of action that produces it must be time bound and therefore, the business rule that governs the course of action must govern this punctuality. Evidently, we need to specify the temporal properties of business rules.

BUSINESS RULE MODEL

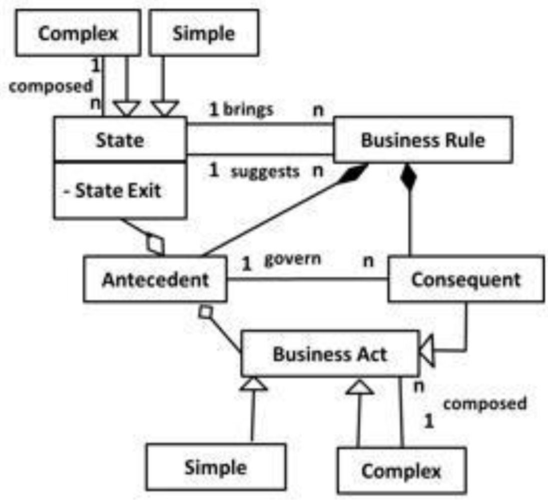
The business rule model (Sharma et al., 2017) presented in Fig. 2 captures the four requirements of BROBM discussed earlier. Accordingly, this section is organized in four sub-sections, (i) Business Rule Structure, (ii) Relationships across Rules, (iii) the notion of state and its relationship to a business rule and (iv) Temporal properties of business rules.

Business Rule Structure

The basic structure of a business rule is in two parts a) what is to be governed and b) how it is governed. Consequently, we define a business rule (see Fig. 2) as an aggregate of antecedent and consequent, in which the 'how' aspect is represented in the antecedent and the 'what' aspect in the consequent of the rule. The figure shows that a consequent is a business act. A business act is an active component that is executable. As the figure shows, there are two kinds of business acts, atomic and complex. An atomic business act cannot be decomposed further. The figure shows that a complex business act is composed of one or more business acts. For example, Late_return (book) is a business act built over the business acts *Return book* and Levy fine respectively.

An antecedent may be a situation that is a state of the enterprise, a business act or any combination of these formed using the Boolean operators AND, OR, NOT. When the situation is a state, we will say that the antecedent is a condition upon the satisfaction of which the consequent is enacted. When it is a business act then its enactment enables another business function. An example of a condition is the rule in which the consequent, Issue book is governed by the antecedent "the requestor is a registered borrower" giving rise to the rule <the requestor is a registered borrower, Issue book>. An example of enablement is <Register requestor, Deregister requestor>, that is the enactment of Register enables the enactment of Deregister.

Figure 2. The business rule model



The Notion of State

Now, consider the notion of a state. A state can be simple, that cannot be decomposed into simpler states, and complex, one that is composed of other simpler states. In other words, a state is complex if it uses conjunction or disjunction; it is simple otherwise.

When enacted, a business rule changes the state of the enterprise as shown by the relationship, changes, in the figure. The cardinality of this relationship says that the state of the enterprise can be changed by more than one business rule but a business rule changes only one (simple or complex) state. For example, the state, book availability, can be changed by two business rules for issuing a book and returning a book respectively.

When the enterprise is in a state, s, then it can be forced out of s by enacting one or more business rules. This 'forcing out' may be

- **Constrained**: Here the forcing out is subject to an exit condition that specifies a limit of occupancy of the state. This is captured in the attribute, State Exit of State of Fig. 2. For example, let it be that a book can be issued for up to 10 days only. That is, when the issue rule changes the state of book from available to issued, then the State Exit should be 10 days. This says that the book must be forced move out of state, not available, within 10 days. We capture constrained State Exit using UNTIL (Goranko, Valentin, Galton, &

Table 1. The SUGGESTS relationship

State	Suggested Rule
Available	Issue book
Issued	Normal return, overdue rule
Overdue	Late return rule

Antony, 2015) for example UNTIL 10 days. The formal representation of UNTIL is presented later in this chapter.

- **Unconstrained**: This is the case when State Exit is unspecified. It is possible for the state never to be 'forced out'. For example, given that a book is in the state, available, it is possible that the book may never be issued and remains available.

The business rules that can take the enterprise out of a state is modeled by the relationship, suggests, shown in Fig. 2. It is so named because the state, s, suggests the several business rules that can apply to it. Again, there is a 1: N relationship between state and business rule. Suggests is the inverse of the relationship, brings, which says that a business rules brings the business into a state.

We illustrate the foregoing by considering a small example from a library system as follows. The set of business rules, B is B = {Issue, Normal return, Late return, Overdue rule}Issue is for issuing books; normal return is for those cases in which the book is returned before the due date has expired; the Overdue rule deals with books not returned by the due date; and Late return is for processing books returned after the due date.

The set of state S for books is

S = {available, issued, overdue}

The relationship suggests is represented in Table 1.

That is, if a book is available then the only applicable rules is Issue; if 'issued' then the applicable rules are normal return rule and overdue rule; if it is in overdue state then the applicable rule is the late return rule. One interpretation of Table 1 is as a state transition diagram as given in Fig. 3.

We can use relationship, *suggests,* in a number of ways as follows:

Figure 3. State diagram of book of library

1) **Discovering New Rules and States**: Analysis of the states of the relationship can reveal new rules. For example, in Table 1, when the book is available then it can be sent for binding. Thus, we get a book binding rule and the state "being bound". Similarly, if a book is issued, then it can also be reserved by some borrower. This gives us the rule reserve book and the state, reserved.
2) **Inverse Rules**: With each new state-business rule relationship that is discovered, we ask the question, can the state be reversed? This gives us new rules, for example Reserve yields Free that undoes the reservation.
3) **Consistency Checking**: We formulate two consistency checks as follows:
    - **Rule Consistency**: Given the set of business rules B, the set of suggested rules must be equal to B. Our example shows consistency because the set of suggested rules from Table 1 is {Issue book, normal return, overdue rule, late return} is equal to the set of business rules that we postulated in section 2.2.
    - **State Consistency**: The set of states participating in state changes must be equal to the set of states participating in the relationship, suggests. Table 1 shows the latter set of states to be {available, issued, overdue} and this set is equal to the set of states we started out with. Thus, our example is state consistent.

Notice that a consistent definition does not mean that the enterprise is completely defined. In our example, the three business rules define a consistent enterprise but rules for indenting, stock taking etc. are missing.

TYPOLOGY OF BUSINESS RULES

Fig. 2 did not, for graphical reasons, present the different types of business rules. This typology is shown in Fig. 4. As seen there are three types of rules, atomic, complex, and abstract (Prakash, Sharma, & Singh, 2014). Complex and abstract business rules are constructed from simpler ones until atomic business rules are reached. Complex business rules are themselves of three kinds, aggregates, transitive rules, and bunches. We consider each of these below:

1) **Atomic Business Rules**: An atomic business rule is one whose consequent is an atomic business act and whose antecedent is simple. Examples of atomic business rule are as follows:

< *Borrower. Type= 'Student', Issue Book*>
< *Register Borrower, Issue Book*>

In the first rule, the consequent, Issue Book, is an atomic business act and its antecedent is a simple state. In the second rule, the antecedent is simple; it is an atomic business act. Further, its consequent, Issue Book, is also simple.

Complex Business Rules

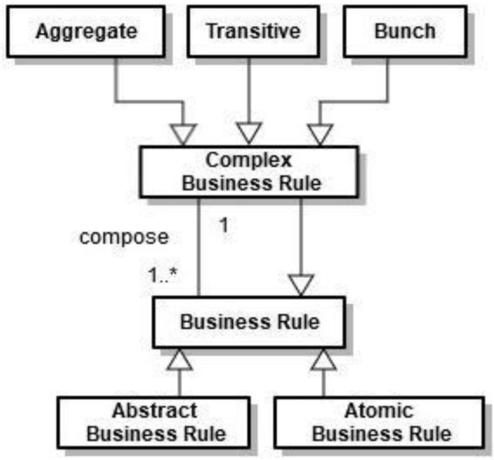
A complex business rule is a meaningful collection of simpler business rules. A complex rule implies that the antecedent is complex or the consequent is complex. There are three kinds (see Fig. 4) of complex business rules, namely 1) Bunch, 2) Aggregate, and 3) Transitive. The first two of these use a complex antecedent to construct business rules having the same common consequent. This consequent may be complex or atomic. The third kind, transitive, must have a complex business act as its consequent. Its antecedent may or may not be complex. We consider each of these in turn.

Bunch

A bunch is a named collection of business rules having an antecedent on the same common state variable and a common consequent. For example, consider the collection of atomic business rules as follows:

a) <Borrower.Type = 'Student', Register Borrower>
b) <Borrower.Type = 'Teacher', Register Borrower>
c) <Borrower.Type='Administrator',RegisterBorrower>

Figure 4. The business rule typology



The Bunch formed is as follows:

Bunch
< Borrower.Type = 'Student' OR Borrower.Type = 'Faculty' OR Borrower.Type = 'Administrator', Register Borrower>

Of

<Borrower.Type = 'Student', Register Borrower>
<Borrower.Type = 'Teacher', Register Borrower>
<Borrower.Type = 'Administrator', Register Borrower>

Aggregate

An Aggregate is a named collection of business rules having an antecedent on different state variables but with a common consequent. Notice the difference with bunch where business rules having antecedents on a common state variable. As an example of aggregate, consider the two of atomic business rule as follows:

a) <Borrower.Type= 'Student', Issue Material>
*b) <Borrower.NoIssued<=10, Issue Material>*These rules may be aggregated as
Aggregate
< Borrower.Type= 'Student' AND Borrower.No Issued <=10, Issue Material >

Of

<Borrower.Type= ‘Student’, Issue Material>
<Borrower.No Issued<=10, Issue Material >

Transitive

It is possible to construct rules using the notion of transitivity. There are two ways in which transitivity arises, directly or indirectly. Direct transitivity occurs between business acts. Let A1, A2, and A3 be business acts then the following holds by transitivity:

<A1, A2><A2, A3> implies <A1, A3>

The rule *<A1, A3>* is a complex business rule built over two simpler ones. As an example, from our library, consider the rules,

<Borrower.RegistrationRequest, Register Borrower >
<Register Borrower, Provide Services>

We obtain the transitive business rule as follows.

<Borrower.RegistrationRequest, Provide Services> transitivity on
<Borrower.RegistrationRequest, Register Borrower >
<Register Borrower, Provide Services>

Indirect transitivity occurs when the consequent of one rule affects the antecedent of another rule. Let us be given the rule

a) < A1, A2>

Let A2 change the value of S, that we express as, *Affects (A2, S)*
Let there be another rule as follows:

b) <S, A3>

Then we get, by indirect transitivity the rule *<A1,A3>*. Taking an example from our library, let us be given

a) <Material.Status= ’Damaged’, Reorder Material>

On

b) <Material.Status= 'Damage', Withdraw Material> Affects (Withdraw Material, Material.Q_O_H)
c) <Material.Q_O_H<= threshold, Reorder Material>

Rule (a) calls for damaged material to be withdrawn. This withdrawal changes the values of quantity on hand, Q_O_H (rule b). Rule (c) reorders material if it falls below the threshold level.

Abstract

An abstract business rule is a generalization of other business rules. This generalization can occur when the antecedent and/or consequent of business rules enter into generalization/specialization relationship. Let us given two business rules:

a) <Student Borrower.Status='valid' AND Student Borrower.No issued < 4, Issue Book>
b) <Faculty Borrower.Status='valid' AND Faculty Borrower.No issued < 10, Issue Book>

The antecedent of rule contains state variables Student Borrower and Faculty Borrower. These can be generalized into the state variable Borrower. As result, we obtain the abstract rule:

Abstract

<Borrower.Status='valid' AND Borrower.No issued < maximum, Issue Book>generalization of

<Student Borrower.Status='valid' AND Student Borrower.No issued < 4, Issue Book>
<Faculty Borrower.Status='valid' AND Faculty Borrower.No issued < 10, Issue Book>

Similarly, there can be abstraction based on the consequent. As an example, consider the abstract rule constructed above.

a) <Borrower.Status='valid' AND Borrower.Type= 'Faculty', Issue Text Book>
b) <Borrower.Status='valid' AND Borrower.Type= 'Faculty', Issue Ref. Book>

Table 2. Temporal nature of business rule

Antecedent(A)	Consequent (C)	Condition	Execution
Time point	Time point	Ct > At	Long running
Time point	Time point	Ct = At	Instantaneous
Time point	Time interval		Long duration
Time interval	Time point		Long duration
Time interval	Time interval		Long duration

The consequents Issue Text Book and Issue Reference Book respectively may be generalized as a business act Issue Book giving rise to the abstract business rule.

Abstract

<Borrower.Status='valid' AND Borrower.Type
= 'Faculty', Issue Book> generalization of
<Borrower.Status='valid' AND Borrower.Type ='Faculty', Issue Text Book>
<Borrower.Status='valid' AND Borrower.Type ='Faculty', Issue Ref. Book>

D. Temporal Properties of Business Rules

A business oriented business rule may be time-bound. This implies that it needs to be understood whether the rule must finish within a period of time or it is instantaneous. In the former case, we need to represent the fact that the business rule is long running. Given that a business rule consists of an antecedent and consequent part, there are two factors that we need to examine, namely, (a) the temporal properties of antecedent and consequent respectively, and (b) time interval between antecedent and consequent. We consider each of these in turn.

First notice that time can be viewed as point time or as time interval. Let both the antecedent *A* and the consequent *C* be time points. Since it is not possible for the consequent to precede the antecedent, we get two cases (i) the consequent succeeds the antecedent (row 1 of Table 2) and (ii) both occur at the same time (row 2 of Table 2).

In the former case, the business rule is long running whereas in the latter case it is instantaneous. On the other hand, it is possible that either *A* or *C* or both are spread over a time interval. Evidently, in these cases (see rows 3, 4, and 5 of Table 2) the business rule is long running.

Now consider the factor (b), that is, there is a time interval between *A* and *C*. Even if it were the case that both *A* and *C* are time points, the mere presence of this time interval says that the business rule is long running.

Table 3. Temporal relations with conditions

Temporal Relation	Condition
A before C	Aet<Cst
A meets C	Aet = Cst
A equal C	Ast=Cst and Aet=Cet
A starts C	Ast=Cst and Aet<Cet
A during C	Ast>Cst and Aet<Cet
A overlaps C	Ast<Cst and Aet<Cet and Aet>Cst
A finishes C	Ast>Cst and Aet=Cet

Overall then, if we are to handle both our situations, we need to do an analysis based on the start and end times of antecedents and consequents. For example, if the start time of a consequent is greater than the end time of its antecedent then there is a time interval between the two that results in a long running business rules.

Using the notion of ‘start time’ and ‘end time’ of time intervals, Allen (1983), proposed seven temporal relations, namely, *BEFORE, MEETS, EQUALS, STARTS, DURING, OVERLAPS* and *FINISHES* between intervals. These, along with their conditions are shown in Table 3 where *A* and *C* are two-time intervals. Note that the subscript, *st*, refers to start time and the subscript, *et*, refers to the end time.

In the rest of this section, we consider these relations from the point of view of business rules. As we will see, we need to introduce a new relation to handle instantaneous business rules and use only two out of these seven for long running business rules.

1) **Instantaneous Business Rule**: First, notice that due to the assumption that A and C are time intervals, the relations of Table 3 cannot be used to express time point. This is because for a time point $A_{st} = C_{st} = A_{et} = C_{et}$ and these relations are not defined to handle this condition.

Therefore, we need to explicitly introduce a relation, INSTANT, for specifying an instantaneous business rule. We define INSTANT as having two arguments, INSTANT (A, C) which says that both A and C occur at the same moment, or that, $A_{st} = C_{st} = A_{et} = C_{et}$.

2) **Long Running Business Rule**: Now, let us consider long running business rules. There are two basic constraints that such business rules should comply with. The first is that the start time of a consequent cannot be earlier than the

Table 4. Long running rules

Temporal Relation	Condition	Business Rule
A before C	Aet<Cst	Long running
A meets C	Aet = Cst	Long running
A equal C	Ast=Cst and Aet=Cet	Violates condition 1
A starts C	Ast=Cst and Aet<Cet	Equivalent to MEETS
A during C	Ast>Cst and Aet<Cet	Violates condition 2
A overlaps C	Ast<Cst and Aet<Cet and Aet>Cst	Violates condition 1
A finishes C	Ast>Cst and Aet=Cet	Violates condition 1

end time of its antecedent. This is because the truth value of the antecedent is known only when it completes.

Constraint 1: $C_{st}>=A_{et}$

Following from this constraint is the second constraint that the start time of the consequent cannot be earlier than the start time of its antecedent. We need this to reason about the conditions of Table 2.

Constraint 2: $A_{st}<= C_{st}$

Now, let us determine which of temporal relations of Table 2 comply with these conditions. Let us look at the third column of Table 4 that shows the relevance of temporal relations to long running business rules. The first row of the table says that the end time of the antecedent is less than the start time of the consequent. This satisfies constraints 1 and 2 above and results in a long running business rule as shown in the third column of the table. Similarly, the second row of the table says that the end time of the antecedent is the same as the start time of the consequent. This satisfies constraints 1 and 2. As before, this results in a long running business rule. The third row of the table satisfies the second constraint. It violates the first constraint: since A_{et} is greater than A_{st} which is equal to C_{st},A_{et} is greater than C_{st}. Thus, the relation *EQUALS* of the third row is not relevant to us.

The fourth row meets the second constraint but not the first. The same argument as for *EQUALS* applies. Thus, *STARTS* is not relevant to business rules. The fifth row violates condition (2); the sixth row violates condition (1) and the seventh row violates condition (2). Therefore, *DURING, OVERLAPS,* and *FINISHES* are also not relevant to our analysis.

Table 5. Temporal relationships

Antecedent	Consequent	Nature of Business Rule
Payment is instantaneous at time t	Service provision is instantaneous at t	Instantaneous, INSTANT(A, C)
Payment is instantaneous at time t	Service provision is instantaneous but after a delay at t' > t	Long running, BEFORE(A, C)
Payment is over an interval	Service provision is instantaneous but starts at end time of payment	Long running, MEETS(A, C)
Payment is over an interval	Service provision is over an interval but with a delay after end time of payment	Long running, BEFORE (A, C)

It can thus be seen that only two, *BEFORE* and *MEETS* yield a long running business rule. To illustrate, let us apply these to a business rule of a library:

If a borrower pays the library fee then the borrower is provided library service.

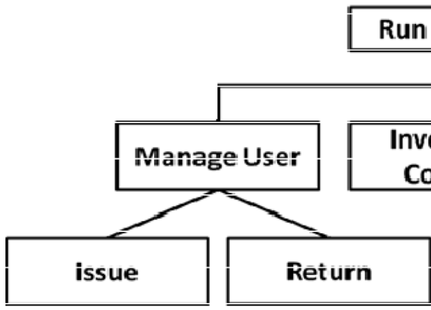
The antecedent of this rule is Payment and the consequent is service provision. We consider a few cases of the temporal inter-relationships between these as shown in Table 5.

3) **Representing UNTIL**: The last temporal issue is that of representing the UNTIL condition applicable the notion of a state. Thus, UNTIL can be represented using the corresponding notion of UNTIL found in temporal logic (Goranko, Valentin, Galton, & Antony, 2015). This logic introduces operators like NEXT, UNTIL, RELEASE, FINALLY etc. The difference between non-temporal and temporal logic is that the latter allows the truth value of its predicate to change whereas the former treats the truth value as never changing. Therefore, temporal logic is capable of dealing with dynamic situations. Temporal logic defines UNTIL as a binary operator, for example, (x UNTIL y). This says that x holds during the entire period when y does not hold. The moment y holds, x ceases to be true. Examples of this situation are: talk UNTIL lecture end, alive UNTIL dead.

Notice that we use UNTIL as a way to express precisely such a situation. Thus, we say, in our example of a library, "issued UNTIL end semester"; "issued UNTIL returned"; "issued UNTIL 10 days".

4) **Packaging Rules:** While constructing business rules, we need to group business rules to make them intellectually manageable. Thus, the set of business rules related to a business area may be packaged together in one package. We refer to such a package as a Rule package. For example, the collection of business

Figure 5. Rule package of library



rule comprising Procurement may be packaged together in the rule package Procurement Material. We allow the possibility of rule packages being contained in other rule packages. This constructs a hierarchy to rule packages. Thus, two rule packages Procure Material and Maintain Material may comprise rule package Manage Material.

In Fig. 5, Run Library is a rule package that consists of three sub rule packages, Manage User, Inventory Control and Staff Management. Manage User rule itself has two sub rule packages Issue and Return. Issue and Return contain the collection of business rules for issuing and returning material from the library.

Representing Business Rules

Instead, we specifically consider extensions proposed by us as follows:

- A new binary operator that captures the enablement of one business act by another. This operator called enables, is =>. It has two operands, both of which are functions, for example, F1 => F2, which says that F1 enables F2. Enables is similar to implication can also be represented as (NOT F1 OR F2). However, we propose to use the => operator because of its clarity in representing enablement.
- Five temporal predicates:
  - MEETS.
  - BEFORE and its inverse AFTER.
  - INSTANT.
  - UNTIL.

As already explained all of these 2-place predicates.

For the relatively more formal minded, the logic is defined as follows:

- A constant is an individual object in world.
- A variable denotes an object in the world.
- A term is a constant or a variable.
- There are n-argument functions of the form G(x1, x2, …. xn) where xi is a term. Functions reflect the mapping of an individual object to another object.
- There are n-place predicates of the form P F(y1, y2, …. yn) where yi is a term. A predicate reflects the mapping of an individual to a truth value.
- An atom is a term, function, or predicate.
- There are standard 2-place predicates EQ, NE, LT, GT, GTE, and LTE corresponding to the six relational operators.
- There are standard 2-place predicates namely BEFORE, AFTER, MEET, INSTANT and UNTIL.
- If P and Q are atoms, then ~P, P V Q, P ^ Q, and P => Q are atoms.
- Every atom is a formula. If F is a formula, then Q(F) is a formula where Q can be the universal quantifier or the existential quantifier.
- Brackets may be put as required.
- Nothing else is a formula.

THE BUSINESS RULES MANAGAMENT SYSTEM

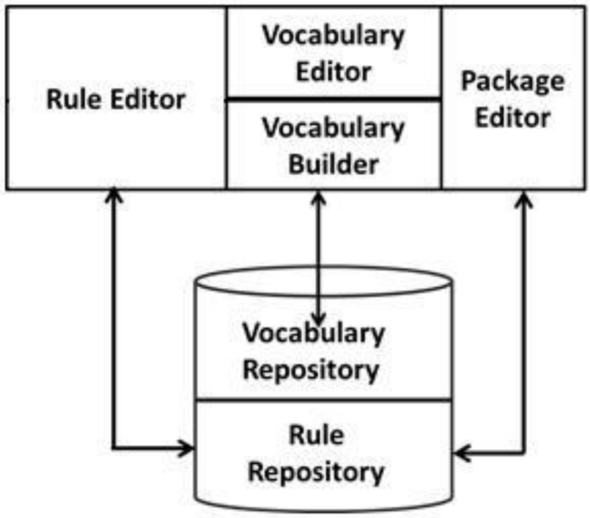
The architecture of the BRMS is shown in Fig. 6. The tool has three interfaces, namely, Rule Editor, Vocabulary Editor and Package Editor. The repository is organized in two parts, Vocabulary Repository and Rule repository.

1) **Rule Editor**

The Rule Editor allows us to create, edit and delete business rules that are available in the rule repository. The basis of the Rule Editor is an atomic rule from which it constructs other business rules, bunch, aggregate etc. For this construction purpose, the Rule Editor uses axioms specific to the rule being constructed.

That is, for a bunch, it uses the bunch axiom, for an aggregate it uses the aggregate axiom, and so on.

Figure 6. BRMS architecture



2) **Package Editor**

Notice that a package of rules, since it is a collection of logically related business rules, does not have any temporal property associated with it nor does it have a State or a State Exit. A package is just a convenient grouping/classification device for easy partitioning of related business rules.

The architecture of Fig. 6 shows a package editor. This editor is responsible for creation and deletion packages as well as modification of package content by inserting rule in a package, deleting a rule from a package. Package construction does not allow the modification of rules. Rule modification is allowed in the rule editor as discussed above. Any modification of a rule has implication on rule construction.

The package editor operates in two modes: in the first mode, business rules from the rule repository are displayed and the user makes an appropriate selection of these to form the package. In the second mode, the user can search the rule repository to retrieve rules heaving specified values in the antecedent or consequent. Selection from the retrieved rules can then be made to form the package.

3) **Vocabulary Editor and Builder (VEB)**

This part of the architecture is used for the construction of the vocabulary repository (shown in Fig. 6.). The vocabulary consists of business objects used in the business rules along with the respective states and business acts of business objects.

VEB operates in two modes. In the first mode, the business vocabulary can be managed through the vocabulary editor for direct input by add, update and delete capacity. In the second mode, VEB picks vocabulary from the rule editor. When a new business rule is written in the rule editor, the vocabulary that is not available in the vocabulary

repository is picked up by VEB and stored in this repository. As a result, all new vocabulary is made available in the vocabulary for future rule writing.

THE LIBRARY EXAMPLE

The foregoing business rule model was applied to the business rules as given to us by the Library Committee of the library of our Institute. We were given a total of 34 business rules covering issue of material (books, journals etc.) by different types of borrowers; return and reservation of library material; stock verification, handling of misplaced material; and requisition to procure new material.

In order to provide a flavor, we consider 4 of these rules related to the task of issuing library material to borrowers. We now illustrate the rules by first given the English statement that was provided to us.

Thereafter, we convert the rules into our form. We include the state changes as well as the temporal conditions of each rule.

Rule 1: Books can be issued to Faculty members and teaching assistants for a maximum of one semester.

{

<((Borrower.BType= 'Faculty Member' OR Borrower.BType= 'Teaching Assistant') AND Material.Name='Book'), Issue Material >

State brings: Borrower. No issued +1Material.Status='issued' UNTIL 1 semester on Material.Name.

Temporal Relation: MEET

}

Notice that the consequent is a business act called Issue Material. The state change includes raising the number of books issued by one and changing the state of book from available to not available. The temporal condition imposed is that the material can be kept for one semester. This condition is imposed on *Material.Name* and rule itself is long running as issue is valid for 1 semester.

Rule 2: Print Journals/Magazines/Project report will be issued for 9 days to faculty members and teaching assistants.

{

<(Material.Name = 'Print Journal' OR Material.Name ='Magazine' OR Material. Name = 'Project Report') AND (Borrower.Btype = 'Faculty' OR Borrower. BType = 'Teaching assistant'), Issue Material>

State brings:

Borrower. No issued +1

Material.Status=' issued' UNTIL 9 days on Material.Name.

Temporal Relation: MEET

}

Rule 3: Students can be issued books from library for a maximum period of 14 days during a semester.

{

<(Borrower.Btype='Student' AND Material.Name ='Book'), Issue Material>

State brings:

Borrower. No issued +1

Material. Status= 'issued' UNTIL 14 days on Material.Name

Temporal Relation: MEET

}

Rule 4: No material in the reference section of the Library will be issued.

{

<(Material.type != 'Reference'), Issue Material>

State change:

Material. Status= 'issued'

Temporal Relation: MEET

}

CONVERSION RULES

The conversion of rules to Java is semantically derived from FORL form to Java. We have identified four types of rule structures in rule typology, Atomic, Bunch, Aggregate and Transitive. We have to map these rules to the IF-Else construct of Java. As per the guideline of the Business rule Manifesto, we modeled any exception as a separate rule. Consequently, we of required the IF construct of Java and omit the Else part. Now, it is choice of the developers whether wants to have a separate rule in case of any exception or a single rule with Else construct. Beside this logical operator AND, OR and NOT are also used to create a complex condition in Java.

FORL to Java conversion is shown below in Table 6.

Table 6. Conversion of BROBRM artifacts to java

HRF	FORL	Java Artifact
Business Entity	Variable	Class
State	Variable	Class data member/instance variable
Value assign to State	Constant	Constant value assign to data member
Condition	The condition, a Well-formed formula	A condition in if statement
Business act	Function	Class method
UNTIL	UNTIL Predicate	An attribute named" Duration"
BEFORE	BEFORE Predicate	Thread.sleep()
MEET	MEET Predicate	Default Java behavior to execute IF construct
INSTANT	INST Predicate	Execute class using multithread
Relational operators	Relational Operators in FOPL form	Relational Operator in Java

So, the business entity specified in the rule is mapped to Variables in FORL. For example, "Borrower. Type" entity in Library is defined in FORL as a predicate which takes "Type" as a term which is represented as Class in Java. Variable and Constant become the data member of the class and the value assigned to the class data members respectively. The condition defined as the well-formed formula is converted as condition part of the IF statement of Java. The logical operators are also used in condition, as and when is required. The function defined in FORL becomes the method of a class in Java. The temporal predicate BEFORE is implemented in Java using Thread.Sleep (time delay) whereas the time delay is the time difference between the antecedent and consequent of a rule. The MEET predicated is the default behavior of execution of IF statement in Java i.e.

```
IF(Condition)
{
-do action-
}
```

So soon as condition become True the action part executes. The UNTIL predicate is mapped with a class data variable namely, Duration, which is used to specify the time.

The only remaining issue is related to the state changes specified in the rule. This conversion will directly move to Java in the form of the variable changes in the functions of the class. Let's us now illustrate the conversion of Rule Typology in Java.

CONVERSION OF RULE TYPOLOGY IN JAVA

Now, let us explain the conversion of each type of our rule (atomic, bunch etc.) into Java code in turn.

Atomic Rule

As discussed in above section, the antecedent of an Atomic rule can be either a condition on simple state variable or simple business act and in both of the case, successful execution of antecedent trigger the execution of business act specified in consequent of the rule. Below is the conversion both form in turn.

Here is an atomic business rule,

```
Business rule Name: Issue Journal to Faculty
{
<Borrower.Type=”Faculty”,Issue(Journal,Borrower)>
State Exist:
                 Journal.AvalCopy-1
                 Borrower.NoIssued+1
                 Journal.Status=’Unavailable’ UNTIL 3days
Temporal Relation: MEET
}
The Java code:
Class Issue
{
        void function()
        {
        IF(Borrower.Type==’Faculty’)
        {
                 Issue(Journal, Borrower);
        }
        }
        Issue(Journal, Borrower)
        {
                 Journal.avalCopy-1;
                 Borrower.NoIssued+1;
                 Journal.Status= ‘Unavailable’;
                 Duration=1days;
        }
};
```

Another example of atomic rule consists of Business act as antecedent:

```
Business Rule Name: Issue Library Card
{
<validBorrower(Borrower),IssueLibraryCard(Borrower)>
StateExit:
        Borrower.id=id  UNTIL 1years;
Temporal Relation: MEET
}
```

The respective java code template is as follows:

```
Class Issue
{
        void function()
        {
        IF(Borrower.Type=='Faculty')
        {
                Issue(Journal, Borrower);
        }
        }
        Issue(Journal, Borrower)
        {
                Journal.avalCopy-1;
                Borrower.NoIssued+1;
                Duration=1days;
        }
};
```

In subsequent rule conversion, only rule body section conversion is shown as the remaining component will be similar to atomic rule.

Bunch Rule

Business rules have the same state variable in an antecedent and a common consequent. For example, consider the collection of atomic business rules as follows:

```
<Borrower.Type = 'Student', Register Borrower>
<Borrower.Type = 'Teacher', Register Borrower>
<Borrower.Type = 'Administrator', Register Borrower>
```

From these, we get the bunch

```
<Borrower.Type = 'Student' OR Borrower.Type = 'Faculty' OR
Borrower.Type = 'Administrator', Register Borrower>
```

The bunch rule is converted to Java rule by using OR relational operator and the rule is written as:

```
IF(Borrower.Type = 'Student' OR Borrower.Type = 'Faculty' OR
Borrower.Type='Administrator')
{
        Register(Borrower)
}
```

Aggregate

An aggregate business rules has different state variables in an antecedent but share a common consequent. As an example of aggregate, consider the two atomic business rules as follows:

```
<Borrower.Type= 'Faculty, Issue Journal>
<Borrower.No Issued Journal<=2, Issue Journal>
```

The aggregate rule is

```
<Borrower.Type= 'Faculty' AND Borrower.No Issued Journal<=10,
issue Material >
```

The AND relational operator is used to write the aggregate antecedent

```
IF(Borrower.Type= 'Faculty' AND Borrower.No Issued Journal<=10)
{
        Issue(Journal,Borrower)
        }
```

Transitive

In BROBM two forms of transitive rules are defined, direct or indirect. Direct transitivity occurs between business acts. Let's consider the following business rules:

```
<Borrower.Registration Request, Provide Services>
transitivity on
<Borrower.Registration Request, Register Borrower >
<Register Borrower, Provide Services>
```

In Java, the component rule is executed in sequential order. First the truth value of RegistrationRequest () is checked and after successful execution of RegistrationRequest(), RegisterBorrower() is executed and on its successful execution, ProvideServices() function will be executed. All three functions are implemented separately in code.

```
void EnableService(Borrower){
        If(Borrower.RegistrationRequest()
                {
                If(Borrower.RegisterBorrower())
                        {
                                ProvideServices()
                        }
                }}
```

Indirect transitivity occurs when the consequent of one rule affects the antecedent of another rule, for example:

```
<Material.Status= 'Damaged', Reorder Material>
```

Transitivity on

```
<Material.Status= 'Damaged', Withdraw Material>
Affects(Withdraw Material, Material.Q_O_H)
<Material.Q_O_H<= threshold, Reorder Material>
```

When material is damaged then it is to be withdrawn as in the first rule above. Material withdrawal affects/changes the values of quantity on hand, Q_O_H in the second rule. The last rule orders the material if Q_O_H falls below the threshold level.

Implementation in Java is as follows:

```
void Function()
{
        if (Material.Status= 'Damage')
        {
                Withdraw Material();
        }
}
Withdraw Material()
{
        Material.Q_O_H-1;
        if(Material.Q_O_H<= threshold)
        {
                ReorderMaterial();
        }
}
Reorder Material()
{
        -do-
}
```

A Java class file is created in the current working directory consisting of the Java code. If more than one rule belongs to the same Java class, then these rules are added to the same Java class in their generated order.

CONCLUSION

We have developed a business-oriented business rule model by introducing the notion state, state changes, and temporal property and set up the Business Layer sitting on top of the CIM layer. We proposed an extended first order-based rule language (FORL) and symmetricallyequivalent Human readable rule form HRL. In this paper, we have converted the HRL form to system rules defined in Java. The conversion is aligned with FORL form as well. Further, a tool is developed to showcase the practicality of Tool. The tool takes rules as input and generates Java classes along with the following information:

- Class in which rule will be implemented.
- The class variables associated with the rule.
- The class method that will be executed once the condition specified in the rule is satisfied.

- The variable updated by the class method after the execution of a rule.
- Any time delay in execution of the class method after execution of the rule.

REFERENCES

Allen, J. F. (1983). Maintaining Knowledge About Temporal Intervals. *ACM-Communiction*, *26*(11), 832–843. doi:10.1145/182.358434

Baltes, S. (2013). *Study on usage of sketches and diagrams in software engineering practice*. Retrieved from http://empirical-software.engineering/projects/sketches/

Baltes, S., & Diehl, S. (2014). Sketches and Diagrams in Practice. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*,(pp. 530-541). Academic Press.

Baltes, S., & Wagner, S. (2016). Effects of Sketching on Program Comprehension. *Proceedings of the 17th International Conference on Agile Processes in Software Engineering and Extreme Programming.* 10.1007/978-3-319-33515-5_26

Bauer, E. (2009). *The Business Rule Approach.* Retrieved from https://pdfs.semanticscholar.org/dbc3/25febb3fa29bbf2bc0b8f7fad08895e46d00.pdf

Bhandari, G., & Gupta, R. (2017). UML Class Diagram from Object Oriented Program for Reverse Engineering Purpose. In *International Conference on Innovative Entrepreneurship and Startup* (pp.147-150). Academic Press.

Colmerauer, A., & Roussel, P. (1996). *History of programming languages II: The birth of Prolog*. New York: ACM.

Ernest, J. F. H. (2008). JESS Rule Engine. Retrieved from http://www.jessrules.com/jess/docs/71

Forgy, C. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, *19*(1), 17-37.

Gaweł, B., & Skalna, I. (2012). Model driven architecture and classification of Business rules modelling languages. In *Federated Conference on Conference on Computer Science and Information Systems* (pp. 949-952). Academic Press.

Giurca, A. (2007). *Using URML and R2ML in a Financial Services Use Case (implemented with JBoss Rules and JenaRules).* Retrieved from http://www.macs.hw.ac.uk/bisel/rewerse/I1/oxygen.informatik.tu-cottbus.de/rewerse-i1/%40q%3Dnode_2f33.html#head-6

Goranko, V., & Galton, A. (2015). Temporal Logic. In *The Stanford Encyclopedia of Philosophy.* Retrieved from http://plato.stanford.edu/archives/ sum2015/entries/logic-temporal accessed on 08/012/2016

Grzegorz, J. (2012). UML Representation for Rule-Based Application Models with XTT2-based Business Rules. *International Journal of Software Engineering and Knowledge Engineering*, *22*(04), 485–524. doi:10.1142/S021819401250012X

Horrocks, I., Peter, F., Boley, H., Tabet, S., Grosof, S., & Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved from http://www.w3.org/Submission/SWRL/

Husemann, S., & Schafer, M. (2006). Building Flexible eHealth Processes using Business Rules. In *European Conference on eHealth* (pp. 25-36). Academic Press.

Ioana, A., & Ileana, A. (2007). Business Rules – Important Artefacts of Software Development. *Ecological Informatics*, *1-4*, 11–14.

Kardasis, P. (2001). *Business Rule Modeling* (Ph.D. Thesis). Department of Computation, Manchester, UK.

Kingston, J. (1987). Rule-Based Expert Systems and Beyond: An Overview. Project Report, Artificial Intelligence Applications Institute-TR-25.

Lloyd, J. (1987). Foundations of Logic Programming (2nd ed.). Springer-Verlag. doi:10.1007/978-3-642-83189-8

Mahmoud, H. Q. (2005). *Getting Started With the Java Rule Engine API (JSR 94): Toward Rule-Based Applications*. Retrieved from https://www.oracle.com/technetwork/articles/java/javarule-139829.html

Norvig, P. (1991). *Paradigms of AI Programming: Case Studies in Common Lisp*. San Francisco: Morgan Kaufmann Publishers.

OMG. (2003). *Model Driven Architecture.* Retrieved from http://www.omg.org/mda/specs.htm

OMG. (2008). *Semantics of Business Vocabulary and Business Rules (SBVR) vol. 1.0*. Retrieved from http:// www.omg.org /spec/SBVR/1.0/PDF

OMG. (2011). *Business Motivation Model.* Retrieved from http://www.omg.org/spec/BMM/1.1/

Oubelli, L.A., & Oussalah, M. (2016). Design and development of business rules management system (BRMS) using AtlandEclipse SIRIUS. *International Journal of Computer Science & Information Technology*, *8*(5), 93-105.

Pau, R. (2010). *Business Rules Approach.* Retrieved from, https://sites.google.com/site/raquelpau/Home/businessrules

Prakash, N., Sharma, D. K., Deepika, & Singh, D. (2013). A Framework for Business Rules. *Proceeding in Springer LNCS*.

Prakash, N., Sharma, D. K., & Singh, D. (2014). Business rules for Business Governance. In *16th International conference on Enterprise Information system* (*vol. 3*,pp. 360-367). Proceeding in LNBIP Springer.

Rackov, M., Kaplar, S., Filipovic, M., & Milosavljevic, M. (2016). Java code generation based on OCL rules. In *6th International Conference on Information Society and Technology* (pp. 191-196). Academic Press.

Rai, V., & Anantaram, C. (2003). Business Rules Approach to Information System Development. *The Pacific Asia Conference on Information Systems*, 123.

Ross, R. G. (Ed.). (2003). *Business Rules Manifesto: The Principles of Rule Independence, Version 2.0*. Retrieved from www.BusinessRulesGroup.org

Sharma, D.K., Prakash, N., Mahajan, M., & Singh, D. (2017). Design and Implementation of a Software System for High Level Business Rules. *International Journal on Advanced Science, Engineering and Information Technology, 7*(6), 2203-2212.

Smith, P. (2017). *Stop your (business rules) engines!* Retrieved from https://adhocteam.us/2017/10/09/stop-your-business-rules-engines/

Zoet, M. (2014). *Methods and Concepts for Business Rules Management* (Ph.D. Thesis). University of Applied Sciences Utrecht.

Chapter 4
Business Indicators for Data Warehouse Requirement Engineering

Hanu Bhardwaj
Manav Rachna University, India

ABSTRACT

Traditionally, data warehouse requirements engineering is oriented towards determining the information contents of the warehouse to be. This has resulted in a de-emphasis of the functional perspective of data warehouses. Consequently, it is difficult to specify functions needed for computing business indicators. The authors' approach aims to elicit needed business indicators from organizational decision makers. Thereafter, indicator hierarchies are built. Then they associate functions with business indicators of the hierarchy. These functions are visualized as use case diagrams. To do this, they extend these diagrams to allow for actor aggregation in addition to actor specialization. Further, they introduce the 'estimated from' relationship between use cases, in addition to the 'extend' and 'include' relationships of UML. They illustrate their proposals with an example.

INTRODUCTION

Data warehouse requirements engineering is concerned with determining the information contents of the warehouse to-be. This is in contrast to transaction oriented information systems that obtain system functionality. It is due to this reason that Data Warehouse Requirements Engineering techniques cannot directly use requirements engineering techniques developed for transactional information

DOI: 10.4018/978-1-7998-2975-1.ch004

systems. Consider the three life cycles described in (Prakash, Singh & Gosain, 2008). The main task in the data base driven (Golfarelli & Rizzi, 1999) and ER driven (Husemann, Lechtenborger & Vossen, 2000) life cycles is to restructure data bases and ER diagrams respectively to determine the required facts and dimensions. Goal oriented approaches (Boehnlein & Ende, 1999; Bonifati & Cattaneo et al., 2001; Golfarelli & Rizzi, 1999; Prakash, Singh & Gosain, 2004; Prakash & Gosain, 2008) explore system/organizational goals and associate information with goals either directly or through some intermediary like the notion of decision or process. The proposal (Prakash & Gupta, 2014) for building a data warehouse for formulation of rules for policy enforcement also focuses on the information aspect.

Obtaining the information is itself a detailed task. In (Prakash & Prakash, 2018) it is proposed that information arises from diverse concerns of managers. Four techniques are proposed and it is suggested that information discovery requires the application of all these techniques. This approach gives centre position to the decisions that managers take and the four techniques provide relevant information for taking these decisions.

In this chapter we deal with yet another way of obtaining information for a data warehouse. We observe that companies set targets for themselves, for example, sales must increase by 10%; market share must go up from 10% to 12%, production must be ramped up to 10,000 units, and so on. Such targets are then used to establish specific targets for the different units/managers of the company. It is then expected that all decision making will get oriented towards target achievement.

Target achievement is monitored periodically, and this implies the definition of appropriate business indicators that are to be estimated. These indicators may take on many forms, key performance indicators, key process indicators, key success indicators, etc. Such indicators are generated at different places in the company and are communicated to diverse stakeholders. Receiving stakeholders may themselves use the received indicators to produce other indicators and communicate these to other stakeholders. The principal source of indicators is the business process and associated business activities/events that are carried out in the organization.

The foregoing suggests that information for the data warehouse to-be can be obtained by (a) knowledge of the parameters of business indicators and (b) determining the information needed for carrying out the various activities of the organization. For this reason we emphasize, in this chapter the crucial role played by the notion of targets in data warehouse requirements engineering.

TARGETS AND THEIR DIMENSIONS

We define a target as a pair <A, I> where A is an aspect of an organization and I is the set of indicators for A. An aspect is a work area, a work unit, a service, or a quality to be preserved. Consider an animal husbandry department. Examples of aspects in this department are as follows:

1. Genetics and Breeding are two work areas.
2. Breed Conversion and Reproduction are two work units.
3. Veterinary Facility and Extension Programme are services provided by the department.
4. Distributed Services Across Regions is a quality to be reserved by the department.

A business indicator is a metric whose value identifies 'what is to be achieved'. This value may be absolute, for example, the value 27 or it may be relative, for example, current value +5%. A business indicator may be simple or composite. A composite business indicator is one whose value is obtained from its component business indicators.

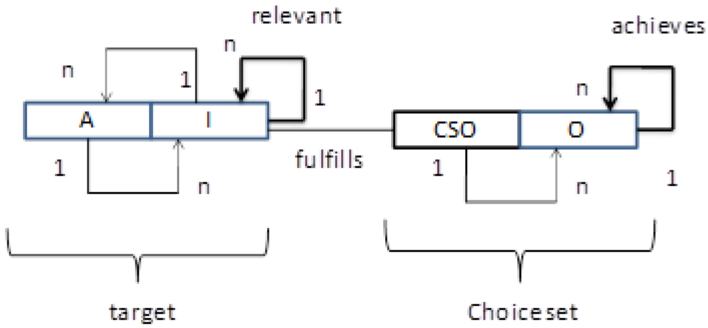
In forming a target, we create an association between aspects and business indicators. A target is thus able to specify 'what is to be achieved' by the aspect. Notice that a target may have an aspect that is associated with more than one business indicator. For example, the aspect Breed conversion may have two business indicators, (current insemination + 10%) and (hybrid animals + 5%).

The notion of a target is depicted in Figure 1. The pair <A, I> is shown by the two conjoined boxes. The arrow from A to I shows the indicators for A. The self loop on I shown by the solid arrow shows the relationship between a composite indicator and its component indicators. The arrow from I to A shows that aspects can be associated with indicators. Thus there is a bi-directional relationship between aspects and indicators.

The bidirectional A-I relationship allows us to form a target hierarchy. Let us be given a top level target. There are two approaches to target reduction:

1. Aspect driven reduction. Here, sub aspects of the top level aspect are determined. For example, if the top level aspect is a work area, then the work units comprising the work area are determined. Similarly, for a work unit, the services offered and the qualities to be preserved form its sub units. Appropriate business indicators are associated with the sub aspects to form sub targets.
2. Indicator driven reduction. If the top level business indicator is composite, then the components that go into computing its values are determined. The aspects responsible for these are associated with the sub indicators and values to be

Figure 1. A conceptual overview



achieved are identified through brain storming sessions. Thus, we formulate sub targets of top level targets.

The notion of a target identifies what is to be achieved by different aspects of an organization. It is a prescription and provides purpose for the decision making to be carried out in the organization. Decision making requires a choice set for selection of an alternative that is most helpful in achieving the target. Thus, we need to associate choice sets with targets, and formulate alternatives. This is modeled in Figure 1 through the fulfils relationship between targets and choice sets.

The right side of the figure shows that a choice set is a pair <CSO, O>. CSO is a Choice Set Objective that can be met through different options, O. As shown, the relationship, fulfils, identifies which target is affected by which choice set. This 'affectation' occurs because of the capacity of options to cause changes in the values of indicators. The self loop on O, labeled 'achieves', shows that options can have their own component options. This gives rise to the achievement hierarchy.

As for target hierarchies, it is possible to build hierarchies of choice sets. Again there can be two approaches, CSO driven and options driven that work in a manner analogous to the aspect and indicator driven approaches considered above.

We view Figure 1 as giving us a two dimensional framework for targets. The first dimension is called the relevance dimension and corresponds to the left part of the figure. In this dimension a hierarchy is set up using the self loop whose nodes are targets and edges specify a 'relevance' relationship between nodes. The second dimension is called the fulfillment dimension and corresponds to the right hand side of the figure. It models the different ways in which an indicator can be affected and comprises of the hierarchy corresponding to the self loop on O.

The Relevance Dimension

The starting point of the determination of the information to be kept in a data warehouse is found in the relevance dimension. Targets are organized in a 'relevance' hierarchy, defined as follows:

Relevant(t,T) iff di_t is a factor in estimating di_T.

Here, di_T refers to a desired indicator of T and di_t refers to a desired indicator of t. We say that t is a sub target of T. If there is no t such that Relevant(t,T) then T is an atomic target and is at the leaf of the relevance hierarchy.

Consider the meaning of estimation. Estimation can range from completely well defined estimation to purely judgmental. We illustrate this by considering a target T and its relevant target t. Consider Di_T as expressed in the set of indicators $\{A_1, A_2, ….. A_n\}$ and Di_t in $\{b_1, b_2, …., b_m\}$. Then estimation can take on any of the forms as follows:

1. **Identity Estimation**: In this case $A_i = b_j$. For example let $Di_{\text{Control Population}}$ be defined as {number, birth rate, death rate, female_male_ratio}. Let $Di_{\text{Birth Rate Control}}$ = {birth rate, calf mortality rate}. Here, 'Birth rate' of 'Control Population' is equal to 'Birth rate' of 'Birth Rate Control'.
2. **Derived Estimation**: Ai = F(bj). Again taking 'Birth rate' as example, it is possible that interest in 'Control Population' is in the Net birth rate taken as the difference between 'Birth rate' and 'Calf mortality rate' of 'Birth Rate Control', i.e., Birth rate $_{\text{Control Population}}$ = Birth Rate $_{\text{Birth Rate Control}}$ – Calf Mortality Rate $_{\text{Birth rate Control}}$. Functions like max, min etc. may also be applied.
3. **Ill-structured Estimation**: In this case, one may use techniques like trend analysis, forecasting, extrapolation etc. to estimate Ai from bj. Perhaps, there is a trend captured in the factor, 'Population elasticity' that helps us to estimate 'New population' based on 'New birth rate' and 'Old population'. Then using this, estimation of 'Control Population' can be made. The proposals of (Barone, Jiang & Amyot et al., 2011) are of interest here.
4. **Judgmental Estimation**: This is the case where estimation is completely ad-hoc and undefined. It is a pure judgment on the part of the decision maker. A rich ground for judgmental estimation is ncc targets.

The Fulfillment Dimension

Whereas the relevance dimension is concerned with targets and their relevance to one another, the fulfillment dimension deals with the alternative ways for causing the change in indicator value.

Choice Set

We view a choice set as an abstraction, a class of elements that have the same purpose, that is, a choice set must be coherent. An element is a member of a choice set **if and only if** it meets the coherence property. **Coherence** says that all elements of a choice set must achieve the same purpose. This purpose is captured in the notion of a choice set objective, CSO. For example, consider the choice set,

CSET = {Improve cattle health, Ensure cattle nutrition, Improve animal husbandry}

having the CSO, 'Improve milk production'. All elements of this set have the same objective to 'Improve milk production'. CSET is coherent.

As an example of an incoherent choice set consider

$CSET_1$= {Improve cattle health, Ensure cattle nutrition, Ensure budgetary compliance}.

The element, 'Ensure budgetary compliance' does not contribute to the stated CSO. Therefore, $CSET_1$ is not coherent.

For a coherent choice set, we may have members that are CSOs and/or functions. We illustrate this by considering the choice set, 'Improve milk production', in its different forms as follows:

- The set {Improve cattle health, Ensure cattle nutrition, Improve animal husbandry} has all its elements that specify choice set objectives.
- Consider the choice set for 'Improve cattle health', {Perform de-worming, Perform brucella vaccination, Perform mastitis vaccination}. All options of this choice set are functions to be carried out.
- For the CSO 'Ensure cattle nutrition', let us be given the choice set {Launch nutrition education programme, Maintain quality feed stock, Improve raw material procurement}. Here 'Launch nutrition education programme' is a CSO whereas 'Maintain quality feed stock' is a function.

Thus, it is possible to build a choice set hierarchy for every CSO. This is done by determining if a CSO/function provides an alternative way of meeting another CSO. We capture this in the 'achieves' relationship defined as follows:

Achieves(CSO_i, CSO) iff CSO_i is an alternative way of meeting CSO.

OR

Achieves(F_i,CSO) iff F_i is a function and is an alternative way of meeting CSO.

The Fulfils Relationship

The fulfils relationship associates appropriate choice sets with targets of the relevance hierarchy. As mentioned earlier, a target may have more than one indicator, for example, $Di_{Control\ Population}$= {number, birth rate, death rate, female_male_ratio}. Each indicator has its own choice set.

We define appropriateness as the alignment of the choice set to its indicator. That is, a choice set is **appropriate** to a target if it identifies ways to change an indicator of the target. For example, the coherent choice set CSET of section 2.2.1 above is appropriate to the indicator 'cattle milk production cmp, cmp + 4%' of the target 'Milk Production'.

We can now define the fulfillment criterion as follows:

Fulfils(CSO, I_m) iff CSO is appropriate to $I_{m,}$ where I_m refers to a indicator of T.

Fulfils says that the elements of CS individually or jointly cause the indicator of T to be reached.

APPLYING THE TECHNIQUE

The foregoing forms the basis for determining the needed information. Pertinent information is obtained as:

1. **History to be Maintained**: For example, information of registered members must be available for the last 5 years.
2. **Category-wise Information**: For example, month-wise, age wise, designation-wise etc.
3. A report or document.
4. A computation obtained by applying a function like Count of, Average etc.
5. **Comparison Information**: For example, comparison of our performance against others.

Pertinent information is elicited for all members of all choice sets.

Consider the case of a Dairy Development Board, DDB. In the current analytic environment at DDB, there is lack of access to consistent and integrated decision-

Table 1. The three targets of dairy development board

Sr. No.	Target		Sub Target	
	Aspect	**Indicator**	**Aspect**	**Indicator**
1	Finance support	Budget Compliance	Fund Management	<[current recovery cr, cr+10%], [current disbursement cd, cd +3%]>
			Loan Management	<current backlog cb, cb – 20%>
2	Technical support	Efficiency	Milk Production	<[cattle milk production cmp, cmp + 4%],[cattle in milk cam, cam + 60%]>
		Effectiveness	Milk Procurement& Distribution	<[dcs satisfaction percent dcssp, dcssp + 10%]>
3	Legal matters	DCS coverage	Legal framework developed	<current DCS cdcs, cdcs+3>
			Adequately staffed legal cell	<[dcs to legal officer ration dlo, dlo= 20:1]>

making data. The current reporting system and decision support environment is not addressing the needs of decision makers across the DDB, which causes deficiencies in data ownership and data access policies. Thus requirement of DDB is to anticipate the future and to identify data for decision making, so that it prepares itself for the emerging competitive environment and provides developments in quality of milk and genetics of animals. The information required by DDB is for strategic purposes such as trend identification, forecasting, competitive analysis and targeted market research, so as to ensure a better future for farmers and milk producers.

The activities of DDB are linked to Dairy Cooperative Societies DCS at village level, Milk Producer Union at district level, Milk Marketing federations at state level, Cattle feed plants, and research institutions, and can be broadly divided into three aspects, namely, financial, technical, and legal.

Let us develop the relevance and fulfillment dimensions, choice sets and pertinent information. For reasons of space, we shall consider a part of the total case.

The top level targets are shown in Table 1. The indicators and aspects of the targets were determined by the Board through a process of discussion and brainstorming.

In the relevance dimension of Technical support shown in the table, the indicators of sub aspect, ‘Milk production’, can be quantified: new cattle milk production must be current cattle milk production, cmp, plus 4%. Similarly, cattle in milk must rise by 60% i.e. cam must become cam+60%.

Table 2. Sub target hierarchy for target 'milk production'

Target	Sub Aspect	Scope of Sub Aspect
Milk Production, <[cattle milk production cmp, cmp + 4%],[cattle in milk cam, cam + 60%]>'	Genetics and breeding service	Genetic Improvement, Breed conversion
	Animal reproduction	Production of quality semen and creation of artificial insemination facilities
	Fodder and feed service	Production and distribution of fodder and feed
	Animal health	Disease diagnostic and control

Figure 2. The relevant sub aspects of target about 'milk production'

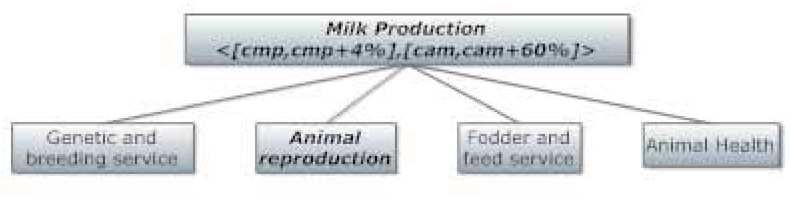
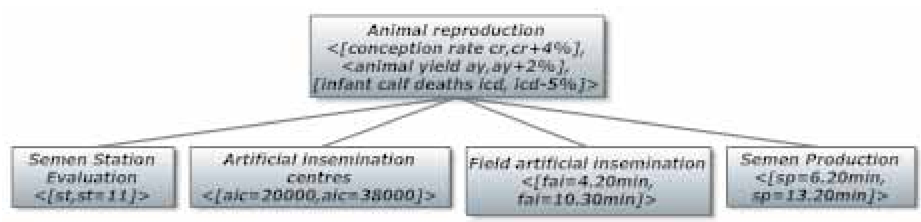


Figure 3. The relevant sub targets of target 'animal reproduction'



Further, the sub-aspect 'Milk procurement and distribution' has the indicator quantified as <[dcs satisfaction percent dcssp, dcssp + 10%]>

Now consider the target about 'Milk production'. Its relevant sub target hierarchy is shown in Figure 2. An explanation of the sub aspects is provided in the table below:

Now consider **aspect** 'Animal reproduction'; its sub targets are shown in Figure 3. The indicators of its sub aspects are 'increasing the semen stations', st, to 11 from the current st; 'raising artificial insemination centres', aic, from 20,000 to 38,000; 'raising field artificial insemination', fai, from 4.20 min to 10.30 min, and 'increasing semen production', sp, from 6.20 min to 13.20 min.

Figure 4. The fulfillment dimension for target 'milk production'

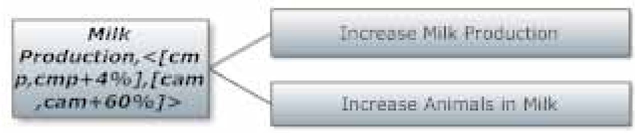


Figure 5. The fulfillment dimension for target 'animal reproduction'

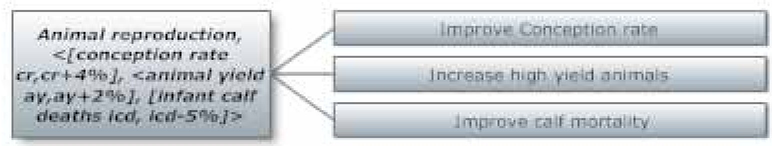
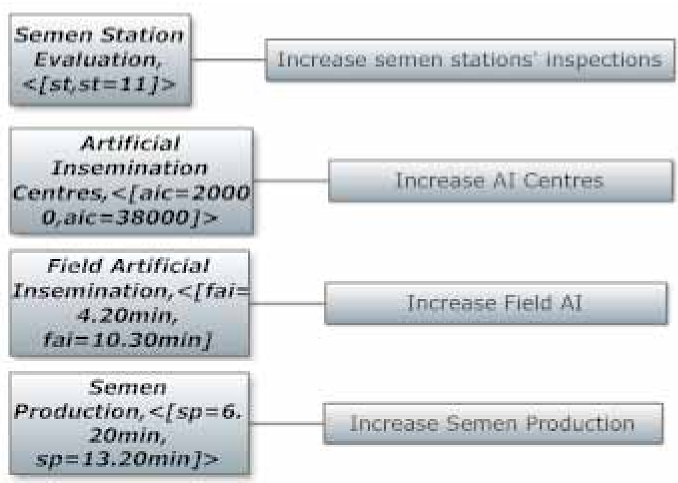


Figure 6. CSOs of the four leaf nodes



There are no sub targets of any of the leaf targets of Figure 3. It can be seen that Figures 1 to 3 can be put together to form a three level relevance hierarchy.

Now, **consider the fulfillment dimension** for the target 'Milk production, <[cattle milk production cmp, cmp + 4%],[cattle in milk cam, cam + 60%]>'' (Table 1). There are two indicators associated with it, and we get two CSOs in the fulfillment dimension. The resulting fulfillment dimension is shown in Figure 4.

Figure 7. Achieves hierarchy of 'increase ai centres'

Finally, consider the target for Animal reproduction of Figure 3. For its three indicators, we get three CSOs as shown in Figure 5.

Each of the leaf nodes of Figure 3 has one indicator associated with it; therefore each gives rise to one CSO, as shown in Figure 6.

We can now illustrate the Achieves hierarchy. Consider the CSO, 'Increase AI Centres' of Figure 6. The Achieves relationships, as shown in Figure 7 are as follows:

a. Achieves(Select location, Increase AI Centres).
b. Achieves(Create AI Franchise Centres, Increase AI Centres).
c. Achieves(Establish proprietary AI centres, Increase AI Centres).

In the foregoing, only 'Select Location' is a CSO whereas the rest are functions. We can further decompose this CSO as follows:

a) Achieves(Select region, Select location).
b) Achieves(Select District, Select Location).

As the last step in the requirements engineering process we need to discover early information for each alternative of each choice set. The result is a high level, unstructured view of data warehouse contents. We present in Table 3 the pertinent information for each of the CSO corresponding to the Figure 6.

THE REQUIREMENTS ENGINEERING PROCESS

Let us now identify the process followed in producing the pertinent information and illustrate its application in the foregoing example. The various steps comprising the process are as follows:

Table 3. Pertinent information for each CSO

Choice Set Objective	Pertinent Information
Increase semen stations' inspections	Bull population under station area. Average collection monthly, quarterly annually. Collection wise ranking of stations. inventory
Increase artificial insemination centres	Total number and number of functional centres. Comparison of centre type to be taken monthly, quarterly and yearly as also by milk union, state and region-wise. Total coverage, comparative average.
Increase field artificial insemination	Top 20 unions performing insemination. Top 20 cluster centres. Average number of inseminations carried out by centre type, All these are monthly, quarterly, yearly and by milk union, state and region wise.
Increase semen production	Semen production state and station wise. Female calves born out of insemination, cross bred animals, milk union, village and district wise. Number of fresh insemination, coverage, percentage of total and number of inseminations annually for the last 3 years

1. **Determine Top Level Aspects**: For the Dairy Board, this was accomplished by identifying the main work areas and services to be provided.
2. **Determine Business Indicators**: For each top level aspect, the requirements engineer laid down the main indicators. For our top level aspects, the indicators were qualitative. The determination of effectiveness, for example, (see Technical Support in Table 1) would be done by questionnaires and feedback sessions. Provision in the data warehouse for this would be necessary, if it is decided to keep detailed information.
3. **Formulate Top Level Targets**: This is done by associating aspects and their indicators. Thus, our top level target is <Finance support, Budget compliance>.
4. **Decompose Targets**: The relevance hierarchy is built by decomposing indicators/ aspects. The indicator 'Budget compliance' of Table 1 is decomposed and the sub indicators obtained were recovery and disbursement. This is the indicator driven reduction approach. In contrast the target for animal reproduction of Figure 3 was decomposed using the aspect driven reduction approach. The sub aspects of animal reproduction were found and then sub targets were set up.
5. **Identify Choice Set Objective Could Occur at Varying Times**: It could happen that as soon as a target/subtarget was identified, the choice set objective was identified due to the association between them. At other times, some alternatives of the choice set were first identified and then the CSO was laid down.

6. **Build Achieves Hierarchy**: This was rather more structured in the case study. Here, it was possible to focus on the decomposition task and identify the components of a choice set.
7. **Identify Information**: This part was completely guided by the tool reported in (Prakash, Prakash & Sharma, 2009)

Our observations from the process are as follows. The first is that discovery of top level aspects can be related to the functional areas and units of the DDB. Further, identification of services provided and main tasks performed could be done through brainstorming sessions. Internal management reports can be a good source for determining aspects.

Once the top level aspects were discovered, the analysis of indicators took on a major role. Indicators are found in planning meetings where targets for different functional units are laid down. Often this is on an annual basis though in India it can be done in a five-year plan perspective as well. The targets presented in this paper were obtained in a five-year plan perspective.

ELICITING BUSINESS INDICATORS FROM STAKEHOLDERS

In the foregoing, we have not addressed the issue of how business indicators are obtained. As mentioned, business indicators are dispersed in an organization and involve many people. Thus, a method for discovering stakeholders and their indicators is crucial for elicitation. We do this in two steps:

a) Identifying stakeholders and eliciting business indicators from them,
b) Determining indicator- sub-indicator hierarchy, thereby identifying functional requirements for computing indicators.

In the first step, a set of <S, BInd> pairs is identified where S is a stakeholder and BInd is the business indicator (computed by S) together with its arguments. In the second step, we construct use case diagrams so as to visualize the needed functionality for obtaining the business indicator. Our use case diagrams include two features in addition to those found in UML use case diagrams. These are (a) actor aggregation and (b) the *estimated from* relationship between use cases.

Definition of Business Indicators

We define a business indicator as a function with arguments x_1, x_2,…x_n, where x_i may be any of the following:

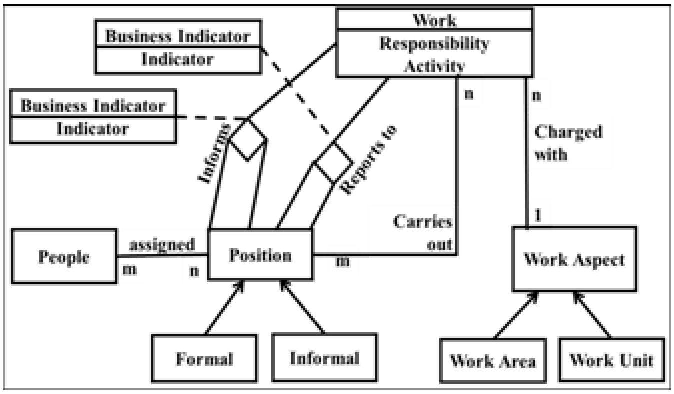
- Business indicator BInd, for example, BInd_College_Success Index = f(Engg. Success index, Science success index) – this consists of two arguments which are business indicators themselves.
- Non- functional argument or information, other than business indicator. For example, Top5=f(marks_all_students) – this consists of a list of marks obtained by all students.
- Function that is not a business indicator itself. For example, DeptSuccess_Index = f(count (passed_students), count (total_students)) – this consists of count function applied on a list of passed students and a list of total students respectively.

The Indicator Exchange Model

The Indicator Exchange Model aims to specify the indicators exchanged, sent and received by the stakeholders of an organization. The basis of this Model is the structure of an organization that reveals the different stakeholders and their inter-relationships. Our indicator exchange model is based on the Organization Structure Model, OSM. According to OSM (2006), an organization consists of organization units that may be internal or external. The OSM guidelines for defining an organizational unit are that it must be persistent and be a formal association of persons. Thus, departments, divisions, committees, can all be units. A position is a role in an organization unit and there may be more than one person holding it. Assignments link people to positions. A person may have multiple assignments, linking them to multiple positions. Business Functions are OSM meta concepts for defining what an organizational does. Figure 1 shows our Indicator exchange model.

This model borrows the notions of *people* and *position* as well as the m: n relationship between these, from OSM. This provides to us a basic set of stakeholders, from whom we will elicit indicators. *People* are assigned *positions*. As shown in the figure, there are two kinds of *positions*, *formal* and *informal*. A formal position is an 'official' position in an organization, one that lies in what is commonly known as the 'reporting structure' of an organization, for example, a departmental head reports to the vice president, production. Formal positions are found in **organization charts** of organizations. In contrast, an informal position is created for special purposes, for example committees, task forces etc. These may or may not exist permanently and have advisory, recommendatory and other such authority in the organization.

Figure 8. The Indicator exchange model



Positions carry out pieces of work (see Figure 8). A *work* is defined by its two attributes, *responsibility* and *activity*. For example, consider an instance of work, *Plan Production* where Plan is a responsibility and Production is one of the activities of planning. The relationship, *carries out*, identifies the *position* that conducts the *responsibility* and *activity* of a *work*. As shown in the figure, *carries out* is an m: n relationship.

A non-empty set of *work* is associated to a *work aspect*. This *charged with* association is 1: n as shown. There are two kinds of work aspect, work unit and work area. The *work unit* of Figure 8 is a recognized association of *positions*, reflecting the organizational structure. Thus *work unit* ISA *work aspect*. It can be seen that *work unit* corresponds to the OSM notion of organization unit. *Work area* is a recognized association of *positions* charged with a certain organization task. The Accounts Department of an organization is an example of a **work unit**: it is a set of positions and defines a structural organization unit. An example of **work area** is Tax Reform: it is a set of positions defined to reform tax computation and deduction.

We define two ternary relationships, *informs* and *reports to*, both of which are between a pair of *positions* and the *work* these positions carry out. The relationship *reports to* says that *position* P_i reports to *position* P_j, the *indicator* for a *work* responsibility and activity. Likewise, the relationship *informs* says that *position* P_i informs *position* P_j, the *indicator* for a *work* responsibility and activity.

As shown in Figure 8, *business indicator* is an association class with attribute *indicator*. Since, more than one indicator may be reported/informed, it is a multi-valued attribute.

The difference between *reports to* and *informs* is that whereas the former captures indicator exchange between formal positions, the latter captures indicator exchange between positions that are not bound by the reporting structure of the organization chart.

As we will see, these two relationships help us in two ways. They:

- Provide us a good starting point for indicator exchange elicitation. Specifically, through these relationships we can focus on *positions* as sources of indicators.
- Form a basis for capturing the dispersed business indicators of an organization referred to in the Introduction.

Let us represent an instance of *informs* as a triple, $<P_i, I_{ij}, P_j>$, where position P_i *informs* P_j about the set of indicators I_{ij}. Since P_j receives indicators from several sources, we can represent its information by the pair $<P_j, \{I_{1j}, I_{2j, \ldots} I_{nj}\}>$. Now, if we traverse *informs* in the reverse direction, then we get the receiver's view of indicators: what indicators the receiving position is expecting from its informers. The position P_j may expect indicators $EI_{1j}, EI_{2j, \ldots} EI_{nj}$ from its informers, $P_1, P_2, \ldots P_n$ respectively. Now, we can formulate a basic consistency check: the received information must be equal to the expected information. In other words, for P_j

$$\{I_{1j}, I_{2j, \ldots} I_{nj}\} = \{EI_{1j}, EI_{2j, \ldots} EI_{nj}\}$$

This equality does not guarantee completeness of informed indicators. This is because it is possible to miss an indicator from both the sender and the receiver points of view. However, it does guarantee that there is an agreement between both parties on the needed indicators. Further, it helps in elicitation when one party recognizes a missing indicator that the other party has omitted.

The Elicitation Process

We start elicitation from the *reports to* relationship as in an organization chart. **We define a position as a stakeholder if and only if it a) receives a business indicator from another position, b) sends a business indicator, or c) self generates a business indicator.**

Elicitation consists of a number of steps as follows:

1. Obtain the positions and the *reports to* relationship between them from the organization chart. We propose to do it in a top-down left to right manner here.
2. For each position, elicit its *informs* relationships.

3. For each *reports to* and *informs*, identify the indicators that a position sends to another position or expects from another position.

We elaborate the algorithms for these steps in the rest of this section.

Step 1: Obtaining positions & *reports to* relationship
Given: Organization chart
Let P_i be a position in the organization chart obtained by traversing the chart in top down left to right manner.
Let RPS be the set of $<P_i, P_j>$ pairs participating in reports to relationships
RPS= ∅
For each P_i
For each P_j that reports to P_i
Ask P_i if at least one BInd is received from P_j

If yes then

RPS = RPS U $<P_i, P_j>$

This algorithm determines the stakeholders that enter into the relationship, *reports to*.

In the second step, a similar strategy is followed to determine those stakeholders who enter into *informs* relationship with one another. The assumption is that *informs* is between positions that may be as follows:

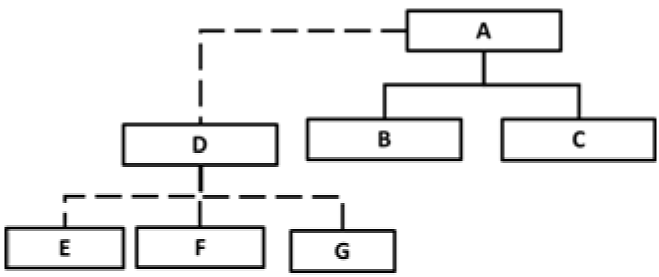
a) P_i and P_j are in the organization chart but do **not** *report to* one another.
b) P_i is in the organization chart but P_j is an informal position of the organization.
c) P_i and P_j are both informal positions.

Thus, the starting point of determining informal positions is the formal positions in the organization chart. These first informal positions are obtained by exploiting (b) above. Our assumption is that the set of informal positions thus discovered shall lead us on to obtain the entire network of informal positions through (c).

In step 2 given below, we handle case (a). We ask a *position* in the organization chart if it is informed about business indicators by those *positions* that are not a part of reporting structure.

Step 2- Eliciting *informs* relationship
Given: Organization chart

Figure 9. Example of organization chart



Let P_i be a position in the organization chart obtained in top down left to right manner.

Let IPS be the set of <P_i, P_j> pairs participating in informs relationships

IPS = ∅

For each P_i

For each P_j that informs P_i

Ask P_i if at least one BInd is received from P_j

If yes then

IPS = IPS U <P_i, P_j>

We also need to determine remaining <P_i,P_j> pairsfrom elicited informal positions as in cases (b) and (c). Therefore step 2 is recursively applied to each P_j of the discovered <P_i, P_j> pair. As a result of steps 1 and 2, the set of *reports to* and *informs* positions are determined in RPS and IPS respectively.

We illustrate the foregoing by considering a small organization chart shown in Fig. 9. Solid lines represent *reports to* relationship and dashed lines indicate *informs* relationships.

Following step 1, we start with A. Let A obtain a business indicator from B. Therefore RPS= {<A, B>}. C does not send any business indicator to A, thus RPS remains same. The next position is B. Let B self generate business indicators. Thus we obtain <B, B> and RPS = {<A, B>, <B, B>}.Moving next to C, we observe that C doesn't receive any business indicator, so RPS is unchanged.

Now let us determine the positions that enter into *informs* by applying step 2. Let it be that A receives business indicators under *informs* relationship from position D. Thus IPS = {<A, D>}. Now pickup B and let it not make a contribution to IPS. Similarly C also doesn't contribute to IPS.

Applying the foregoing recursively, let D receive business indicators under the *informs* relationship from positions E, F and G. Thus IPS = {<A, D>, <D, E>, <D, F>, <D, G>}. Further, when we examine E, F and G, we elicit that they self generate business indicators, thus IPS now becomes {<A, D>, <D, E>, <D, F>, <D, G>, <E, E>, <F, F>, <G, G>}.

Now let us consider the third step that exploits the *<responsibility, activity>* attributes of *work* of Fig.8. For a position looking after a responsibility type, the activities and indicators reported for them are obtained as follows:

Step 3: Identifying Indicators
Given: RPS
Let $ExpectBInd_{RPS}$ be a set of triplets <P_i, P_j, {BInd}> where P_i expects a set of BInds from P_j.
$ExpectBInd_{RPS} = \varnothing$
For each P_i in <P_i, P_j> of RPS
Identify the responsibility type
For each responsibility type
Identify activities
For each activity
Elicit business indicator $BInd_i$ that P_j reports to P_i
Add $BInd_i$ to the BInd part of <P_i, P_j, {Bind}> in $ExpectBInd_{RPS}$.

The above algorithm is also applied to each P_i in <P_i, $P_{j>}$ of **IPS** and **ExpectBInd**$_{\mathbf{IPS}}$ is constructed. This completes the elicitation of expected business indicators from each position.

The complete set of expected indicators **ExpectBInd** is as follows:

$ExpectBInd = ExpectBInd_{RPS}$ U $ExpectBInd_{IPS}$

We illustrate step 3 by continuing with our example. We start with <A, B> of RPS. Let A be expecting business indicators BI_1 and BI_2 from B. Thus,

$ExpectBInd_{RPS}$ = {<A, B, {BI_1, BI_2}>}.

Then we move to <B, B>. Let B self generate BI_1 and BI_2 resulting in

$ExpectBInd_{RPS}$ = {<A, B, {BI_1, BI_2}>,<B, B, {BI_1, BI}>}.

Similarly we elicit the expected business indicators from the elements of **IPS**. Let ExpectBInd constructed finally be

ExpectBInd = $\{<A, B, \{BI_1, BI_2\}>, <B, B, \{BI_1, BI_2\}>, <A, D, \{BI_5, BI_6, BI_7\}>, <D, E, \{BI_8, BI_9\}>, <D, F, \{BI_{10}\}>, <D, G, \{BI_{11}\}>, <E, E, \{BI_8, BI_9\}>, <F, F, \{BI_{10}\}>, <G, G, \{BI_{11}, BI_{12}\}>\}$.

Whereas the foregoing elicits those indicators that a position ***receives/expects from*** others, our consistency check of needs indicators that a position ***sends to*** others. For this we construct a set SentBInd similar to ExpectBInd but for keeping indicators that a position sends to another. The algorithm of step 3 is modified to elicit 'sent' indicators rather than 'expected' indicators. As a result of applying this, we get

SentBInd= $\{<B, A, \{BI_1, B_2\}>, <D, A, \{BI_5, BI_6\}>, <E, D, \{BI_8, BI_9\}>, <F, D, \{BI_{10}\}>, <G, D, \{BI_{11}, BI_{12}\}>\}$.

Consistency Check

Now, we can apply the consistency check, as explained above, on SentBInd and ExpectBInd using the algorithm as follows:

Given: RPS, IPS, ExpectBInd, SentBInd
For each $<P_i, P_j>$ in RPS
a. Find an element $<P_i, P_j, BIndE>$ in ExpectBInd
b. Find an element $<P_j, P_i, BIndS>$ in SentBInd
c. If BIndS is not equal to BIndE
Then
Signal inconsistency.

We also need to repeat steps a, b and c for each $<P_i, P_j>$ in IPS so as to look after sent and received indicators.

On applying the above algorithm on our ongoing example, we find out that there is a discrepancy in the expected business indicator claimed by A. Now, A claims that BI_7 is expected from D, whereas D does not send it. This inconsistency is signaled and we expect that A and D would either decide to add or remove BI_7. Similarly, the discrepancy in exchange of BI_{12} between G and D is also handled.

HIERARCHY CONSTRUCTION

Recall that a business indicator has arguments that may be:

- A business indicator.
- A function.
- Nonfunctional argument.

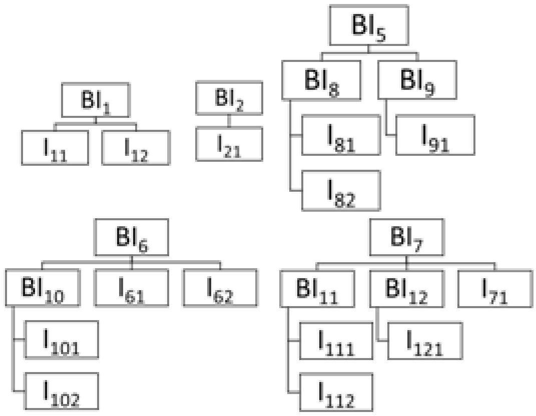
This definition is the basis of construction of indicator hierarchy.

For every stakeholder, we display the indicators expected and the indicators sent by it. This stakeholder knows what is to be computed from what, and builds the indicator sub-hierarchy. The total hierarchy is built by considering all positions. The arguments needed for each business indicator, are identified and these go into the computation of BInd.

The algorithm for building the indicator hierarchy is as follows:

Given: RPS, IPS, ExpectBInd, SentBInd
For each P_i in <P_i, P_j> in RPS
For each element in SentBInd
If there exists <P_i, P_j, BIndS> in SentBInd
Then
For each element $BIndS_i$ in <BIndS>
For each element in ExpectBInd
If for P_i there exists <P_i, P_j, BIndE> in ExpectBInd
Then
For each $BIndE_i$ of BIndE
If $BIndS_i$ is computed from $BIndE_i$
Then
$BIndE_i$ is an argument of $BIndS_i$
End if
End for
End if
End for
Elicit any other argument of $BIndS_i$
End for
End

Figure 10. Five indicator hierarchies



The above algorithm is applied to each element of IPS also.

In our example we start with A from RPS. We do not find any such element in SentBInd, which means that A does not send business indicators to other positions. We move to the next element, B. B sends $\{BI_1, BI_2\}$ to A. We now look at ExpectsBInd for B and find that it contains $\{BI_1, BI_2\}$. Since both BI_1 and BI_2 are self generated by B, information required to compute these must be available to B itself. Further, these could only be ii) or iii) above. This information is elicited from B. Let the computation information elicited be $BI_1=f(I_{11}, I_{12})$ and $BI_2=f(I_{21})$.

Now, consider the next position D. D sends $\{BI_5, BI_6, BI_7\}$ to A and receives $\{BI_8, BI_9\}$ from E, $\{BI_{10}\}$ from F and $\{BI_{11}, BI_{12}\}$ from G. The computation information elicited is $BI_5=f(BI_8, BI_9)$; $BI_6=f(BI_{10}, I_{61}, I_{62})$ and $BI_7=f(BI_{11}, BI_{12}, I_{71})$. Now the business indicators for the remaining positions E, F and G are further examined and information is elicited. The resulting functions are $BI_8=f(I_{81}, I_{82})$; $BI_9=f(I_{91})$; $BI_{10}=f(I_{101}, I_{102)}$; $BI_{11}=f(I_{111}, I_{112})$; $BI_{12}=f(I_{121})$.

The hierarchies for our business indicators are shown in Figure 3. As seen there are five hierarchies, for BI_1, BI_2, BI_5, BI_6 and BI_7 respectively.

BIND USE CASE DIAGRAM

Let us now consider building use case diagrams for business indicator functionality. First notice that stakeholders become actors of our use case diagrams, and each stakeholder has its own use case corresponding to each business indicator it computes.

Table 4. Indicators for director-VP(PPC)

Responsibility	Activity	Indicator
Plan	Finance	Expenditure
		Receipts
Monitor	Admissions	Percentage of filled seats
	Results	Success Index
		Academic Performance Index

Since we introduce some additions in UML use case diagrams, we refer to our diagrams as Business Indicator (BInd) Use Case diagrams.

The two additional concepts are as follows:

a. **Actor Aggregation**: If an indicator goes into estimating the value of another, then the stakeholder of the former is in the hierarchy of the latter. Thus, a stakeholder 'is part of' another stakeholder and we introduce the notion of aggregate actors in BInd Use Case diagrams. Notice that UML provides for actor specialization/generalization and not for actor aggregation.

b. ***Estimated from* relationship between use cases**: Since an indicator is estimated from another sub-indicator, the BInd use case diagram must contain one use case for the composite indicator and another for the sub-indicator. We now need to introduce a new relationship, *estimated from*, between use cases.

To understand the need for (b) above, consider the two UML relationships, extend and include:

a) **Extend**: Specifies that a use case extends the base use case. The base use case may stand alone, but under certain conditions, its behaviour may be extended by the other.

b) **Include**: Specifies that the base use case explicitly incorporates the behaviour of another use case at a location specified by the base. The included use case never stands alone, but is only instantiated as part of some larger base that includes it.

The *estimated from* is a relationship between a base use case and an *estimated from* use case. Both use cases are capable of independent existence and have actors associated with them. The actor of the base use case interacts with the system and

Table 5. Indicators of dean Engg.- director

Responsibility	Activity	Indicator
Plan	Faculty Load allocation	Adequacy of faculty members
Monitor	Results	Engg. Success index
		Engg. Academic Performance Index
Monitor	Admissions	Engg. Filled seats

Table 6. Indicators of head mechanical – director

Responsibility	Activity	Indicator
Plan	Finance	Mechanical Dept. Student Training & Welfare Expenditure, Mechanical Dept. Library Expenditure
		Mechanical Dept. Fees Receipts, Mechanical Dept. Sponsorship receipts

Table 7. Indicators of head mechanical – dean Engg

Responsibility	Activity	Indicator
Monitor	Results	Mechanical Department Success index
		Mechanical Department Academic Performance Index

so does the actor of the *estimated from* use case. In this sense, the *estimated from* is different from UML's *include* relationship. It is also different from *extend* in that there is no extension of any behaviour. The indicator produced by the *estimated from* use case is used by the base use case.

EXAMPLE: INDICATORS OF STAKEHOLDERS

Consider applying the foregoing technique to obtaining the pertinent information for an Institute. This warehouse was needed to make decisions about changes in the academic structure and teaching-learning processes. An organization chart of the Institute is available. It is in four levels and shows the Director, the Deans and various executive offices, the Heads of Departments, and the faculty members.

Since the number of positions is rather large we show a small part of the work. We started with the root of the hierarchy, namely, the Director. The Director was

Table 8. One level indicator hierarchy

Indicator	Sub-indicator
Expenditure	Mechanical Dept. Student Training & Welfare Expenditure, Mechanical Dept. Library Expenditure Computer Sc. Dept. Student Training & Welfare Expenditure, Computer Sc. Dept. Library Expenditure
Receipts	Mechanical Dept. Fees Receipts, Mechanical Dept. Sponsorship receipts Computer Sc. Dept. Fees Receipts, Computer Sc. Dept. Sponsorship receipts,
Percentage of filled seats	Engg. Filled seats, Science filled seats
Success Index	Engg. Success index, Science success index
Academic Performance Index	Engg. Academic Performance Index, Science Academic performance Index

reporting to the Chairman, Board of Governors and *informing* Vice President (Production, Planning, and Control) i.e. VP(PPC) and Senior Vice President i.e. SVP. For the first '*informs*' authority, namely, VP(PPC), the Director identified eight activities for Monitoring, five activities for Planning, as well as four Coordination activities. For reasons of brevity, the indicators identified for one Planning activity and two Monitoring activities are shown in Table 4.

Thereafter, we moved in a left to right manner to the next level in the organization chart. We present below the sample of indicators that Dean of Engineering, Dean Engg., *reports to* Director in Table 5.

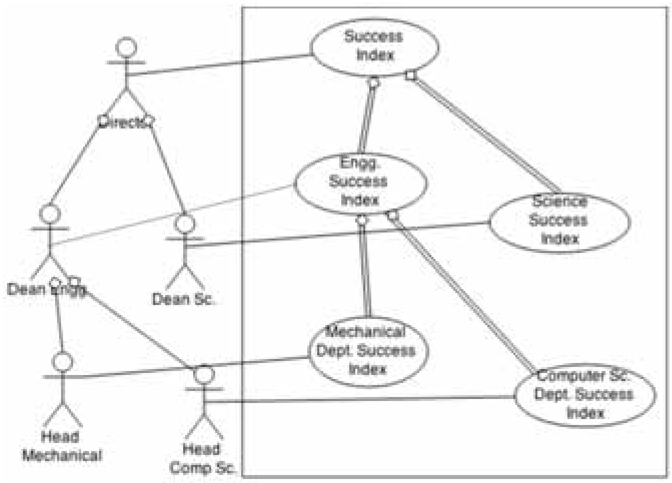
Similarly Dean Sciences also reported similar indicators as Dean Engg. but for Sciences courses.

Now we moved to the next level, the departments. Head Mechanical reported six activities during his interaction with Director and ten activities during his interaction with Dean Engg. Tables 6 and 7 show Head Mechanical - Director indicator exchange and Head Mechanical - Dean Engg. indicator exchange respectively.

A similar set of indicators (as mentioned in Table 6 and 7) was obtained from Head Computer Sc. for Dean Engg. and Director, respectively.

Let us now illustrate the last step, construction of indicator hierarchy. For the root *position* Director, the indicators 'Expenditure' and 'Receipts' were estimated from various expenditures and receipts received from Head Mechanical and Head Computer Sc. respectively. Similarly the rest of the indicators of Table 4 are picked up, and the estimation relationship is established and the one level of hierarchy is built as shown in Table 8. We can observe that some of the indicators are estimated

Figure 11. A BInd use case diagram



from indicators received from various Deans and others are estimated from various Heads of the Departments.

The foregoing builds only one level of the hierarchy. The full hierarchy is an assembly of many such. For example, consider indicator 'Engg. Success Index' that is sent by Dean Engg to Director. It is estimated from indicators received from Head Mechanical and Head Computer Sc, as in second row of Table 5. Following the procedure, we get two sub-indicators for 'Engg. Success Index', namely, a) Mechanical Department Success index, and b) Computer Sc. Department Success Index.

A multi-level hierarchy is built by exploiting commonality of child and parent indicators. Thus the Success Index (see Table 4) computed by the Director and sent to VP-PPC needs Engg. Success Index (see Table 5) computed by Dean Engg. and Science Success Index computed by Dean Science. In order to compute Engg. Success Index, Dean Engg. himself needs Mech. Departmental Success Index sent by Head Mechanical and Computer Sc. Department Success Index sent by Head Computer Sc.

Consequently we get the hierarchy, Root: Success index; Level 1: Engg. Success Index; Level 2: Mech. Department Success Index, Computer Sc. Department Success Index. The corresponding functional definitions for business indicators are as follows:

- Success Index= f1(Engg. Success Index, Science Success Index).
- Engg. Success Index= f2(Mech. Department Success Index, Computer Sc. Department Success Index,).

- Mech. Departmental Success Index= f3(Total passed mechanical students, Total mechanical students).
- Computer Sc. Dept. Success Index= f3(Total passed computer sc. students, Total computer sc. Students).

Whereas the first two business indicators are computed based on other business indicators, Departmental Success Index is self generated in the departments themselves.

BInd Use Case Diagrams

A part of the full BInd use case diagram resulting from the foregoing indicators is shown in Figure 11. This figure shows the interaction of Director with use case 'Success Index', which is *estimated from* 'Engg. Success Index' and 'Science Success Index' use cases. Similarly, Dean Engg. and Dean Sc. interact with 'Engg. Success Index' and 'Science Success Index' use cases respectively. 'Engg. Success Index' use case is *estimated from* 'Mechanical Dept. success index' and 'Computer Sc. Dept. success Index'. Both the latter use cases have interaction with Head Mechanical and Head Computer Sc respectively

Figure 11 shows that actor Director is an aggregate of two actors, Dean Engg. and Dean Sc. Similarly Dean Engg. is an aggregate of Head Mechanical and Head Comp. Sc.

We have found **two interesting observations** in our example study. These are as follows:

1. The use of the organization chart facilitates the discovery of *informs* relationship. When we tried not to use the organization chart then people and positions identification became completely ad-hoc. The organization chart provided to us a very good guide to initiate our work.
2. The *responsibility-activity* pair a) provides a guideline to elicit indicators, our experience is that without this, it was difficult to provide guidance on where to look for indicators, b) provides a means to cut down on the search space of indicators to be considered as candidates for sub-indicators while forming the indicator hierarchy.

CONCLUSION

Seeing the crucial role played by targets and business indicators in decision making, our proposal is to consider these for data warehouse requirements engineering. On the one hand, this identifies the arguments of business indicators that constitute the information relevant to the data warehouse To-be and, on the other, the packaged functionality to be built.

Our proposal is to elicit indicators of different stakeholders in an organization through our indicator exchange model. Our elicitation technique exploits the organization chart of an organization as a starting point. Taking into account the hierarchical structure of indicators, we introduce the notion of actor aggregation and *estimated from* in use case diagrams.

REFERENCES

Barone, D., Jiang, L., Amyot, D., & Mylopoulos, J. (2011). Composite Indicators for Business Intelligence, ER 2011. *LNCS*, *6998*, 448–458.

Boehnlein, M., & Ulbrich vom Ende, A. (1999). Deriving initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems. *Proc. Of Workshop on Data Warehousing and OLAP (DOLAP)*. 10.1145/319757.319780

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACMTOSEM*, *10*(4), 452–483.

Bresciani, P., Giorgini, P., Mylopoulos, L., & Perini, A. (2004). Tropos: An agent oriented software development Methodology, Journal. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203–236. doi:10.1023/B:AGNT.0000018806.20944.ef

Chalmeta,R.Grangel,R. (2005)Performance Measurement Systems for Virtual Enterprise Integration. *International Journal of Computer Integrated Manufacturing*, 73-84.

Fernández-Medina, E., Trujillo, J., Villarroel, R., & Piattini, M. (2007). Developing secure data warehouses with a UML extension. *Information Systems*, *32*(6), 826–856. doi:10.1016/j.is.2006.07.003

Frank, U., Heise, D., Kattenstroth, H., & Schauer, H. (2008). Designing and Utilizing Indicator Systems Within Enterprise Models – Outline of a Method. *MobIS*, 89-105.

Giorgini, P., Rizzi, S., & Garzetti, M. (2008). GRAnD: A goal oriented approach to requirement analysis in data warehouses. *Decision Support Systems*, *45*(1), 4–21. doi:10.1016/j.dss.2006.12.001

Golfarelli, M., & Rizzi, S. (1999). Designing the Data Warehouse: Key Steps and Crucial Issues. *Journal of Computer Science and Information Management, 2*(3).

Husemann, B., Lechtenborger, J., & Vossen, G. (2000). Conceptual Data Warehouse Design. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000).*

Kaplan, R. (1992). The Balanced Scorecard: Measures that Drive Performance. *Harvard Business Review*, *70*(1), 71–79. PMID:10119714

Kumar, M., Gosain, A., & Singh, Y. (2010). Stakeholders driven requirements engineering approach for data warehouse development. *Journal of Information Processing Systems*, *6*(3), 385–402. doi:10.3745/JIPS.2010.6.3.385

List, B., Schiefer, J., & Min Tjoa, A. (2000). Process-Oriented Requirement Analysis Supporting the Data Warehouse Design Process A Use Case Driven Approach. LNCS, 1873, 593–603.

Luján-Mora, S., Trujillo Vitae, J., & Song, I. Y. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering*, *59*(3), 725–769. doi:10.1016/j.datak.2005.11.004

Lynch, R. L., & Cross, K. F. (1991). *Measure Up!: Yardsticks for Continuous Improvement*. Wiley, Blackwell.

Organization Structure Metamodel (OSM), 2nd Initial Submission, OMG Document bmi/2006-11-02 (2006)

Prakash, D., & Gupta, D. (2014). Eliciting Data Warehouse Contents for Policy Enforcement Rules. *International Journal of Information System Modeling and Design*, *5*(2), 41–69. doi:10.4018/ijismd.2014040103

Prakash, D., & Prakash, N. (2018). A multifactor approach for elicitation of information requirements of data warehouses. *REJ, 24*(1).

Prakash, N., & Bhardwaj, H. (2012). Early Information Requirements Engineering for Target Driven Data Warehouse Development. LNBIP, 134, 188-202.

Prakash, N., & Gosain, A. (2008). *An Approach to Engineering the Requirements of Data Warehouses. Requirements Engineering Journal, 13*(1), 49–72.

Prakash, N., Prakash, D., & Sharma, Y. K. (2009). Towards Better Fitting Data Warehouse Systems. In A. Persson & J. Stirna (Eds.), *PoEM 2009. LNBIP* (Vol. 39, pp. 130–144). Heidelberg, Germany: Springer.

Prakash, N., Singh, Y., & Gosain, A. (2004). Informational Scenarios for Data Warehouse Requirements Specification. LNCS, 3288, 205-216.

Prat, N., Akoka, J., & Comyn-Attiau, I. (2006). A UML-based data warehouse design method. *Decision Support Systems*, *42*(3), 1449–1473. doi:10.1016/j.dss.2005.12.001

Shanks, G., Sharma, R., Seddon, P., & Reynolds, P. (2010). Business Analytics and Competitive Advantage: A Review and Research Agenda. *21st Australasia Conference on Information Systems*, 1-11.

Chapter 5
Random Walk Grey Wolf Optimizer Algorithm for Materialized View Selection (RWGWOMVS)

Anjana Gosain
Guru Gobind Singh Indraprastha University, Delhi, India

Kavita Sachdeva
Shree Guru Gobind Singh Tricentenary University, Haryana, India

ABSTRACT

Optimal selection of materialized views is crucial for enhancing the performance and efficiency of data warehouse to render decisions effectively. Numerous evolutionary optimization algorithms like particle swarm optimization (PSO), genetic algorithm (GA), bee colony optimization (BCO), backtracking search optimization algorithm (BSA), etc. have been used by researchers for the selection of views optimally. Various frameworks like multiple view processing plan (MVPP), lattice, and AND-OR view graphs have been used for representing the problem space of MVS problem. In this chapter, the authors have implemented random walk grey wolf optimizer (RWGWO) algorithm for materialized view selection (i.e., RWGWOMVS) on lattice framework to find an optimal set of views within the space constraint. RWGWOMVS gives superior results in terms of minimum total query processing cost when compared with GA, BSA, and PSO algorithm. The proposed method scales well on increasing the lattice dimensions and on increasing the number of queries triggered by users.

DOI: 10.4018/978-1-7998-2975-1.ch005

INTRODUCTION

OLAP (Online Analytical Processing) queries triggered by the database users, demand aggregated and compiled results instantly. Therefore, for effective decision making, information systems are required, which disseminates aggregated and instant results. Data warehouse is a kind of information system used for decision making. A data warehouse (DW) (Han & Kamber, 2001; Morse & Issac, 1998) is a data repository integrated from numerous varied sources of data, used for decision support querying and analysis. The biggest concern is to handle such datastore in a cost effective way. ETL process extracts the data from source systems, then transform the data and finally loads the data into the data warehouse. Therefore, data is stored in the warehouse as native data sources along with derived views. Such derived views are reffered as materialized views (Jain & Gosain, 2012).

Materialized views raises the processing efficiency of triggered queries, by avoiding the access to native data sources so as to reduce the total processing cost of queries. Such approach minimizes the query processing cost, but it increases the maintenance cost of chosen views. Thus, the most important task to design a data warehouse is to minimize the total query processing cost which includes both the query evaluation cost and view maintenance cost, while finding an appropriate number of views, within the storage space constraint. This problem is classified as materialized view selection (MVS) problem.

For solving the problem of MVS, various frameworks and solutions exist in literature. Frameworks like Data cube and lattice (Harinarayan et.al., 1996), Multiple View Processing Plan (MVPP) (Yang et al., 1997), AND-OR view graphs (Gupta & Mumick, 2005; Mami et al., 2011; Tamiozzo et al., 2014; Horng et al., 2003) have been used for representation of views. Lattice framework (Harinarayan et al., 1996) has the ability to capture dependency among aggregate views and thus creates data cubes at multiple dimensions. To answer a query in minimum possible time, lattice framework access the smallest data cube available. MVPP framework (Yang et al., 1997), a global query processing plan for complete set of queries, exploits the existence of common sub-expressions for most of the queries. AND-OR view graph (Gupta, 1997) is used to express all the possible execution plans for evaluating a query in the query set. To represent all the data cubes at distinct aggregation levels, lattice framework (Harinarayan et al., 1996) portrays user queries easily. Therefore, numerous studies have chosen lattice framework for representing the problem space of MVS problem. Further, to minimize the total query processing cost, various researchers have implemented numerous evolutionary optimization algorithms (EA's) like Particle Swarm Optimization (PSO) (Gosain & Heena, 2016; Sun & Ziqiang, 2009), Bee Colony Optimization (BCO) (Kumar & Arun, 2015), Adaptive Genetic algorithm (AGA) (Yu et al., 2015) etc., in selecting the materialized views. This is due to the fact that Evolutionary Algorithms (EA) works on randomly selected multiple solutions simultaneously to find out the optimum most solution and are applicable for broad range of problems.

In this work, we have implemented Random walk Grey wolf optimizer (RWGWO) (Gupta & Deep, 2019) algorithm, an improved variant of Grey Wolf Optimizer (GWO) algorithm, for materialized view selection i.e. (RWGWOMVS) on lattice framework for finding appropriate set of views under the space constraint, so as to minimize the total processing cost of triggered user queries. GWO algorithm (Mirjalili et al., 2014), is a population based algorithm in the field of swarm intelligence and is the only algorithm based on leadership hierarchy. GWO is used for solving the continuous and real world optimization problems. In GWO algorithm (Mirjalili et al., 2014), all the wolves of pack update their positions as per the leading wolves of the pack. But the main drawback is how a superior wolf will update its position and why the dominant wolf of the pack will take the guidance from inferior wolves of the pack to update its position. Due to this, the pack does not converge to global optima. RWGWO algorithm (Gupta & Deep, 2019), increases the potential of GWO algorithm, by improving the search ability of grey wolves based on random walk. RWGWO (Gupta & Deep, 2019) is a meta-heuristic algorithm inspired by grey wolves. The RWGWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature and avoids the problem of premature convergence due to the stagnation in local optima and maintains a social behavior within the pack. Three main steps of RWGWO are (1) searching for prey, (2) encircling prey and (3) attacking the prey i.e. selecting the best one, (Gupta and Deep, 2019).

Similarly, in our work, RWGWOMVS (1) searching for views, (2) selecting the materialized views, and (3) selecting the best one, are implemented. RWGWOMVS is implemented using MATLAB and experiments are administered, by executing it over standard test data available in TPC-H star schema benchmark (Neil et.al., 2007). Results are evaluated and analysed on varying the dimensions of lattice, on varying the query frequency and considering different cases of storage space. To test and validate the effectiveness of RWGWOMVS, we have compared it with other evolutionary optimization algorithms like Genetic algorithm (GA) (Lin & Kuo, 2004), Particle Swarm Optimization (PSO) (Gosain & Heena, 2016; Sun & Ziqiang, 2009), and Backtracking Search Optimization algorithm (BSA) (Civicioglu, 2013). It was found that RWGWOMVS outperforms in terms of minimum total processing cost of query and expandability of the problem.

In our proposed work, authors have used lattice framework, in which cubes correspond to the views that are to be selected, whenever a user query is invoked. The invoking frequency of a query is equivalent to the invoking frequency of a cube. Linear cost model is followed by authors for finding the answering cost of query.

This answering cost of a query corresponds to the number of rows to be accessed in the equivalent cube for the query.

Organization of the chapter is as follows: The next section explains the lattice framework and the mathematical model of materialized view selection problem. Related work is shown in further section. Next section gives the brief review of Random Walk Grey Wolf optimizer (RWGWO) algorithm. Further section presents the random walk grey wolf optimizer for materialized view selection (RWGWOMVS). Experimental results under different space constraints, on varying the dimensions of lattice, on increasing the number of user queries and comparisons with other methods is given in further section. The chapter is concluded in last section.

BACKGROUND

Materialized view selection is done for boosting the query processing efficiency of data warehouse. MVS problem is deeply studied and used in large data-centric systems. From the origin of data warehouse evolution, the main focus is on optimal selection of views. In literature, various studies have given numerous results (Halevy, 2001) and solutions (Agrawal et al., 2001; Chaudhuri et al., 1995, 1998) on the use of materialized views. View selection is categorized into - static view selection (SVS) approach and dynamic view selection (DVS) approach. In SVS, a prime set of views is selected, depending upon the static query set which is provided in advance. But, the pre-computed views may become unsuitable over time, due to unstable query patterns and growing needs of users (Choi et al., 2004). Therefore, dynamic approach to selection of views comes into play. DVS updates the pre-computed views, as per the updated query set, rejects dirty views and selects new views dynamically to manage materialized views, required by incoming queries (Kotidis & Roussopoulos, 1999; Lawrence & Rau-Chaplin, 2006).

Researchers have proposed various frameworks, methods and algorithms to find solution for materialized view selection problem. Genetic algorithm was proposed (Zhang et al., 1999) to deal with selection of materialized views. Three basic operators were used i.e. selection, crossover and mutation. As compared to heuristic approaches, it was practical, feasible and effective. A framework (Kossman et al., 2000) and greedy algorithm, integrated with query optimizer was proposed for reduction of maintenance cost (Baril et al., 2003) for MVS problem. The query optimizer selects an optimal maintenance plan out of a search space which consists of recomputation plans and incremental plans for materialized views. Proposed algorithm is costly to implement, because of its greedy nature. As per the authors, multiple query optimization methods explore a broader scope. Prototype of Genetic algorithm (Lee et al., 2001) was implemented to find efficient solution for selecting optimal set of views for maintenance-cost view selection problem. Proposed method works in context of OR graphs and significantly improves the time complexity over existing search-based methods, implementing heuristics. A hybrid evolutionary algorithm was proposed (Zhang et al., 2001) to optimize queries, to choose the best global processing plan from multiple global processing plans and to select materialized views from a given global processing plan. Experimental results suggests that, on applying evolutionary algorithm to optimization of MVS problem for a given global processing plan, significantly reduces the total processing cost of query.

Authors (Valluri et al., 2002) have proposed a framework on view relevance and algorithm on VRDS (View Relevance Driven Selection) for selecting views in data warehouse. VRDS algorithm maintains a balance between the processing cost of query and maintenance cost of views. Proposed work performs better than greedy algorithm (Yang et al., 1997; Mistry et al., 2001) when space constraint exists and update frequency is high. A randomized view selection algorithm SAVSA (Simulated Annealing View Selection Algorithm) based on simulated annealing, was presented (Kumar et al., 2012) for selecting Top-K views in a multidimensional lattice. Views were selected by performing random series of uphill and downhill moves, possessing some probability, in solution space. Proposed algorithm SAVSA, selects better quality Top-K views with minimum total view evaluation cost as compared to the views selected using greedy algorithm (Harinarayan et al., 1996). A constrained evolutionary algorithm was proposed (Yu et al., 2003) to incorporate constraints into the algorithm through a stochastic ranking method. Proposed work results in significantly better solutions in terms of minimization of query processing cost compared to previous evolutionary algorithms (Kumar et al., 2012). A greedy algorithm was proposed (Kumar et al., 2009) for selection of materialized views with emphasis on HRU (Harinarayan et al., 1996) and PGA(Polynomial Greedy algorithm) (Nadeua et al., 2002) algorithms. HRU algorithm when compared to PGA algorithm, exhibits high run time complexity as the number of possible views is exponential, whereas PGA algorithm provides a scalable solution by selecting materialized views in polynomial time and performs well on parameters like execution time, memory and CPU usage. Query Answering Greedy Algorithm (QAGA) was proposed (Kumar et al., 2010) to select the most profitable views for materialization, by using size and query frequency of a view to select the top-k profitable views from multidimensional lattice. QAGA resulted in lower total processing cost of queries, leading to improvement in the average query response time and is beneficial with respect to size and have greater capability of answering future queries.

ACA-MVS (Ant Colony based materialized View selection) algorithm (Xianqiang et al., 2010), based on evaporation mechanism and positive feedback of ant colony algorithm is proposed to achieve optimal view selection in global search area with varying space limitations. The evaporation mechanism will update pheromones and guide the ants to change their states transition. A method for materialized view creation is proposed (Ghosh et al., 2012) using statistical method by quantifying the association among different attributes in a given relation. The proposed method is independent of storage space constraint and is applicable to any data-centric system. In another work, an integer programming (IP) model (Talebi et al., 2013), was proposed for view selection, which studies the properties of views and indexes that appear in optimal solution for this model. Based on these properties, it removes a number of variables and constraints to obtain a model that gives optimal solution. Authors in (Jiyun et al., 2012) have proposed the materialized view selection based on Top-*k* query algorithm for data lineage tracing as Lineage tracing queries help to locate updated views faster in data warehouse. The selection is based on parameters like query frequency, the view storage space, query processing cost and maintenance cost. It gives better query performance than the Heuristic algorithm, for lineage tracing query.

A sampling method was proposed (Madhu et al., 2013) to estimate the size of views for materialized view selection problem, thereby saving time. Authors have proposed a hybrid estimator that measures the degree of skew in the data and combines Jacknife estimator with the Schlosser estimator to estimate the size of the view more accurately. The proposed hybrid estimator has been used to estimate the view size and has shown better estimation results as compared to individual estimators. A hybrid system combining ant based approaches and tabu search have been proposed (Habiba et al., 2012) for the generation of materialized views in a relational data warehouse environment. It is a bio-inspired approach which manages dynamically the storage to include the best views. Information retrieval techniques are used to evaluate the similarity between queries and the workload. Multiobjective Differential Evolution Algorithm Using Binary Encoded Data was proposed (Goswami et al., 2013) to select materialized views in data warehouse. This approach, maintains diversity in solution space with necessary elitism by controlling population in intermediate generations of the differential evolution process. The solutions of intermediate generations are first ranked according to their pareto dominance levels and then the diversity among solution vectors in solution space is measured. The algorithm is found to be suitable in selecting significant representative solutions from a large number of non dominating solutions of the view selection problem.

A novel approach to the selection of materialized views based on a genetic algorithm (Zhixiang et al., 2014) is proposed. Fitness function is defined and various operators for crossover and mutation are selected to avoid premature convergence for MVS at minimal cost. HAWC (History Aware Cost based optimizer), an architecture that extends a cost-based logical optimizer is proposed (Luis et al., 2014) to use history information to identify query plans, which can be used for creating materialized views. An optimization strategy is also proposed for selecting and pooling materialized views which have high probability of being recycled by future queries, subject to storage limits and other constraints.

MATERIALIZED VIEW SELECTION (MVS)

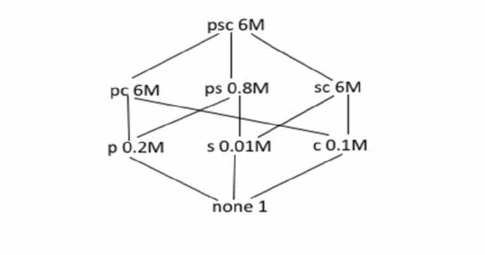
MVS problem is stated as: A given set of user queries K and space for storage Z, the MVS problem is to find a materialized view set V, which minimizes the total query processing cost for the selected view set, within the constraint that the total space taken by selected views, M is less than or equal to Z. Lattice framework is used by authors, for representing the problem of materialized view selection, which is elaborated in subsection below.

Lattice Framework

Numerous frameworks for representation, like Data cube and lattice (Harinarayan et al., 1996), Multiple View Processing Plan (MVPP) (Yang et al., 1997), AND-OR view graphs (Gupta & Mumick, 2005; Mami et al., 2011; Tamiozzo et al., 2014; Horng et al., 2009) have been proposed and used by researchers to represent the problem of view selection. Lattice framework (Harinarayan et al., 1996) has the ability to capture dependency among aggregate views and thus creates data cubes at multiple dimensions. To answer a query in minimum possible time, lattice framework access the smallest data cube available. MVPP framework (Yang et al., 1997), a global query processing plan for complete set of queries, exploits the existence of common sub-expressions for most of the queries. AND-OR view graph (Gupta, 1997) is used to express all the possible execution plans for evaluating a query in the query set.

In the present work, authors have chosen lattice framework for representing the problem space of materialized view selection, as it can easily model queries, incorporates dimension hierarchies and gives a clear picture of already materialized views (Harinarayan et al., 1996). Each lattice vertex depicts a view that aggregates numeric measures over the dimensions present in that vertex, whereas each lattice edge shows the dependencies among the adjacent views (Hung et al., 2007). Data cube (Gray et al., 1997) is defined by Group-by clause. A path exists between the two cubes ci and cj if a dependency relation c_i £ c_j, exists between the cubes c_i and c_j, signifies if a query can be answered from c_i, then it can also be answered by using cube c_j. Data cubes possible in the lattice framework is 2^N, for N dimensions fact relation. For example, sales transactions in a DW system (TPC-H Star Schema Benchmark (Neil et.al., 2007)) having three dimensions, part (P), supplier (S), and customer (C), and fact shows the sale of parts from suppliers to customers. 2^3=8 data cubes will be generated in the lattice as shown in Figure 1.

Figure 1. Lattice framework with 8 possible data cubes (Harinarayan et al., 1996)



Dependencies that exist among views are shown by the connection lines. The number beside each cube depicts its size (number of rows). Base cuboid is the cube at top i.e. psc, at lowest aggregation level. Cube with highest aggregation level is the Bottom cuboid. A query directly getting answered from cube (*,s,*), can also be answered from any of the parent cubes (p,s,*), (*,s,c) or (p,s,c) (Gosain & Heena, 2016).

Mathematical Model of Materialized View Selection Problem

MVS is crucial for designing a data warehouse efficiently. The aim is to reduce the response time of triggered user queries by selecting the views optimally, within the space and cost constraints. Lattice framework is used for representing the problem space of MVS problem. Cubes represents the views that are to be selected, whenever a query from user is triggered. The invoking frequency of a query is equivalent to the invoking frequency of a cube. Linear cost model is followed by authors for finding the answering cost of query. This answering cost of a query corresponds to the number of rows to be accessed in the equivalent cube for the query. Thus, MVS problem using lattice framework (Lin & Kuo, 2004) is given as - A lattice-cube L, with a set of c cubes $P = (p_1, p_2, \ldots\ldots, p_c)$, set of x user queries $Q = (q_1, q_2, \ldots\ldots, q_x)$, a set of values for query frequency $F = (f_{q1}, f_{q2}, ..., f_{qx})$, set of update frequency values, $UF = (l_{p1}, l_{p2}, \ldots\ldots, l_{pc})$ of P cubes, within the space constraint, Z. The objective function is to find a set of cubes (views) V to minimize the cost function shown in equation 1, within the storage space constraint $\sum_{p \in V} | P | \leq Z$.

$$\sum_{i=1}^{k} f_{q_i} * T(q_i, V) + \sum_{p \in V} l_p * M(p, V) \quad (1)$$

where $T(q_i,V)$ and $M(p,V)$ represents the evaluation cost of query q_i and the cost of maintenance of cube p, with respect to set of materialized views V.

BRIEF REVIEW OF GREY WOLF OPTIMIZER AND RANDOM WALK GREY WOLF OPTIMIZER ALGORITHM

Grey Wolf Optimizer (GWO)

Grey Wolf Optimizer (GWO) algorithm (Mirjalili et al., 2014), is a population based algorithm in the field of swarm intelligence. GWO is the only algorithm which is based on leadership hierarchy. GWO (Mirjalili et al., 2014), is used for solving the continuous and real world optimization problems. Four categories of grey wolves i.e. alpha, beta, delta, and omega are used for simulating the leadership hierarchy and social behavior of grey wolf for the process of hunting the prey. Three main steps of GWO algorithm for hunting the prey are (1) chasing the prey, (2) encircling the prey and (3) attacking the prey. Alpha wolf, α the leader wolf is the first category, can be male or female, is responsible for taking vital decisions of the pack. Wolves in second category are beta wolves β, called subservient wolves which are responsible for message delivery of α wolf to other wolves and helps the α wolf in hunting. Last type of wolves are the wolves known as omega wolves, ω are allowed to eat food in the end. Remaining wolves are categorized as delta wolves δ, which includes the caretakers, hunters and sentinels. Caretaker wolves helps to take care of wounded wolves, hunters contributes to the process of hunting of prey and sentinel wolves act as defender from external enemies.

In GWO algorithm (Mirjalili et al., 2014), grey wolf complete their hunting process by the guidance of the leaders α, β and δ wolves. Each wolf update its position with the help of leading wolves α, β and δ. Leading wolves are responsible search agents in updating the state of each wolf and gives a suitable direction towards the prey. Thus, it is important that during each iteration, leading wolves performs the best, to provide optimum guidance to approach a prey. The pseudocode of GWO algorithm is shown in Table 1.

Random Walk Grey Wolf Optimizer (RWGWO) Algorithm

In GWO algorithm (Mirjalili et al., 2014), all the wolves update their positions as per the leading wolves α, β and δ. Thus, α a leading/dominant wolf, will have to take guidance from an inferior wolf to update its position. This is the main disadvantage of GWO algorithm that a dominant wolf of the pack is taking help from low fitted wolves beta and delta of pack, to update its position. Similarly, next superior wolf of

Table 1. Pseudocode for GWO algorithm (Mirjalili et al., 2014)

```
Initialize the Grey Wolf population x_i (i=1,2,…….., n)
Initialize parameters b,µ and c
Initialize l=1, the iteration number
Evaluate the fitness of each wolf
Select α = fittest wolf of the pack
β = second best wolf
δ = third best wolf
while l < maximum number of iterations
Evaluate the fitness of each wolf
for each leader wolf
find new position y_i of the leader x_i by random walk
if f(y_i) < f(x_i)
update the leaders
end
for each wolf
update the position and apply greedy approach between current and updated positions
end
update b,µ and c
update α, β and δ wolves
l=l+1
end
```

the pack i.e. beta wolf will update its position with support from low fitted (inferior) wolf δ of the pack. Due to these drawbacks, GWO algorithm does not converge to global optima. Therefore, it is important to select the leading wolves, so as to update the position of each wolf.

RWGWO algorithm (Gupta & Deep, 2019), increases the potential of GWO algorithm, by improving the search ability of grey wolves based on random walk. RWGWO (Gupta & Deep, 2019) is a meta-heuristic algorithm inspired by grey wolves. The RWGWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature and avoids the problem of premature convergence due to the stagnation in local optima and maintains a social behavior within the pack.

In RWGWO method (Gupta & Deep, 2019), leaders will find the search space by random walks and omega wolves, ω will update their position by following them. RWGWO (Gupta & Deep, 2019), starts with initial population of wolves i.e. x_i, where i = 1, 2, ……., n. Random walk is included in each iteration, for the leading wolves α, β and δ of the population. Step size for random walk is drawn from Cauchy distribution. This is due to the fact that the variance of a Cauchy distribution is infinity, which is very effective at the time of stagnation and helpful for leading wolves α, β and δ to explore the search space for finding the prey. Table 2 gives the pseudocode for RWGWO algorithm.

Table 2. Pseudocode for RWGWO algorithm (Gupta & Deep, 2019)

Initialize the Grey Wolf population x_i (i=1,2,…….., n) **Initialize parameters** b,μ and c **Initialize** l=1, the iteration number **Evaluate** the fitness of each wolf **Select** α = fittest wolf of the pack β = second best wolf δ = third best wolf **while** l < maximum number of iterations **for** each wolf update the position **end** **update** b,μ and c **update** α, β and δ wolves l=l+1 **end**

RANDOM WALK GREY WOLF OPTIMIZER ALGORITHM FOR MATERIALIZED VIEW SELECTION (RWGWOMVS)

The proposed method for selection of materialized views Random Walk Grey Wolf Optimizer Algorithm for Materialized View Selection (RWGWOMVS) is described in a step wise manner. A multidimension lattice L, number of user queries k, total space for storage s, dimension d, of the problem, size of population n, maximum number of iterations MaxIter, parameters of RWGWO algorithm b,μ and c are provided as input to the proposed method.

Step1: Initialization: During the initialization step, population n is generated randomly and the parameters b,μ and c are initialized using equations (2) to (6) of RWGWO algorithm (Gupta and Deep, 2019). The iteration counter t is initialized to 1.

$X_{t+1} = X_{p,t} - \mu * d$ (2)

$d = | c * X_{p,t} - X_t |$ (3)

$\mu = 2*b*r_1 - b$ (4)

$c = 2*r_2$ (5)

b=2- 2*(t/max. no of iterations) (6)

Here X_{t+1} represents the position of the wolf i.e. the view at $(t+1)^{th}$ iteration,

X_t is the position of the wolf i.e. the view at t^{th} iteration,

$X_{p,t}$ depicts the position of the selected view at t^{th} iteration,

d gives the difference vector,

μ and c are coefficient vectors,

b is a linearly decreasing vector from 2 to 0 over iterations and r1, r2 are the uniformly distributed random vectors whose component lie between 0 and 1.

Step 2: Evaluation: During the evaluation phase, the fitness of each view is evaluated, by calculating its fitness function i.e. the cost function.

Step 3: Selection: During the selection phase, α, β and δ views are selected, where α represents the fittest view of the pack, β is the second best view and δ represents the third best view.

while t < maximum number of iterations

for each view, position is updated using equations (6) to (9) (Gupta and Deep, 2019).

$X_1' = X_\alpha - \mu_\alpha * d_\alpha$ (6)

$X_2' = X_\beta - \mu_\beta * d_\beta$ (7)

$X_3' = X_\delta - \mu_\delta * d_\delta$ (8)

$X_{t+1} = (X_1' + X_2' + X_3')/3$ (9)

where X_α, X_β, X_δ represents the positions approximated by α, β and δ solution views with the help of eq. (2).

end (for loop terminates here)

Step 4: Updation: During the updation step, parameters b,μ and c are updated and α, β and δ views are also updated.Increment the counter t, as t=t+1.

end (while loop terminates here)

Step 5: Output: Selected view set P, global minimum i.e. total cost of P

Produce the best view set obtained so far and calculate its total cost.

Table 3 depicts the pseudocode for RWGWOMVS algorithm.

EXPERIMENTAL RESULTS

RWGWOMVS algorithm is implemented by authors using MATLAB and experiments are conducted out using the datasets of TPC-H star schema benchmark (Neil et al., 2007). The parameters used for implementing Random Walk Grey Wolf Optimizer algorithm for Materialized View Selection (RWGWOMVS) are: size of population n=50, dimension of problem d=3, Iteration Counter, maxIter = 500, parameters of RWGWO algorithm b,μ and c initialized as per equation (2) to (6). Cost function is evaluated on changing the space constraint values, varying the dimensions of lattice to show the effectiveness of our proposed work.

Table 3. Pseudocode for RWGWOMVS algorithm

Input: A multidimensional lattice L, number of user queries k, total storage space S, problem dimension d, population size n, maximum number of iterations MaxIter, parameters of RWGWO algorithm b,μ and c
Output: selected view set P, global minimum i.e. total cost of P
Method:
Step1: Initialization: Initial population n is generated randomly, initialize the iteration counter t to 1, initialize parameters b,μ and c using equations (1) to (5),

$X_{t+1} = X_{p,t} - \mu * d$ (1)

$d = | c * X_{p,t} - X_t |$ (2)

$\mu = 2*b*r_1 - b$ (3)

$c = 2*r_2$ (4)

b=2- 2*(t/max. no of iterations) (5)

where X_{t+1} is the position of the wolf at $(t+1)^{th}$ iteration.
X_t is the position of the wolf at t^{th} iteration, $X_{p,t}$ is the position of the prey at t^{th} iteration.
d is difference vector, μ and c are coefficient vectors and b is a linearly decreasing vector from 2 to 0 over iterations and r1, r2 are the uniformly distributed random vectors whose component lie between 0 and 1.
Step 2: Evaluation: Evaluate the fitness of each view, by calculating its fitness function
Step 3: Selection:
Select α = fittest view of the pack
β = second best view
δ = third best view
while t < maximum number of iterations
for each view
update the position using equations (6) to (9)

$X_1' = X_\alpha - \mu_\alpha * d_\alpha$ (6)

$X_2' = X_\beta - \mu_\beta * d_\beta$ (7)

$X_3' = X_\delta - \mu_\delta * d_\delta$ (8)

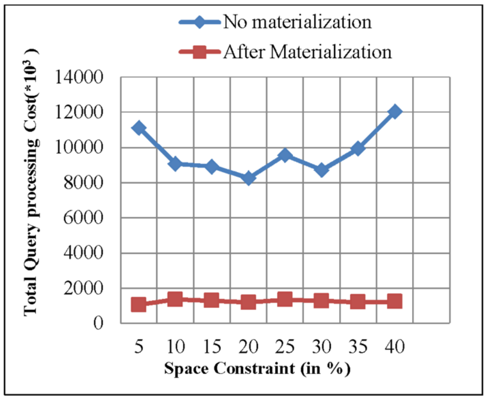
$X_{t+1} = (X_1' + X_2' + X_3')/3$ (9)

where X_α, X_β, X_δ are the positions approximated by α, β and δ solutions views with the help of eq. (1).
end
Step 4: Updation:
update b,μ and c
update α, β and δ views
t=t+1
end
Step 5: Produce the best view set obtained so far and calculate its total cost.

Total Query Processing Cost Achieved, Considering Space Constraints

Providing unlimited space to store materialized views, is not a feasible solution to obtain minimum query processing cost. Thus, authors have considered varying values of space constraints i.e. 10%, 20%, 30%, 40%, 50% and 60% to find the total query processing achieved. Figure 2 depicts the reduction in total query processing cost, when views are materialized using RWGWOMVS, as compared to the total query processing cost achieved, when views are not materialized.

Figure 2. Results of RWGWOMVS under different space constraints



Total query Processing Cost Achieved, on Varying the Lattice Dimensions and on Increasing the Number of User Queries

Authors have considered three different cases for lattice dimensions i.e. i) three dimensions ii) four dimensions and iii) five dimensions as the test data from the standard benchmark of TPC-H (Neil et al., 2007). For three dimensions authors have chosen p Product, c Customer, and s Supplier from the data set. Time t, is taken as fourth dimension and Location l, is added for considering five dimensions. The total processing cost of query attained, on changing the dimensions of lattice is shown in Figure 3. It is observed that on increasing the number of views exponentially, there is linear increase in the total query processing cost.

Fig. 4 proves the scalability of our proposed work. It has been observed that, on increasing the number of user queries, the difference between query processing cost without materialization and query processing cost with materialization decreases. There is decrease in total query processing cost, as the number of queries increased from 300 to 400 and further remains constant as shown in Figure 4. This clearly shows that RWGWOMVS is scalable.

Comparison with Other Evolutionary Algorithms

Figure 5 and Figure 6 gives the comparative results of our proposed algorithm RWGWOMVS with Particle Swarm Optimization and genetic algorithm and Backtracking Search Optimization algorithm respectively. Our proposed algorithm gives alarming results than PSO, genetic and BSA in terms of minimum total processing cost of query and on increasing the number of queries triggered by users.

Figure 3. Results of RWGWOMVS on varying lattice dimensions

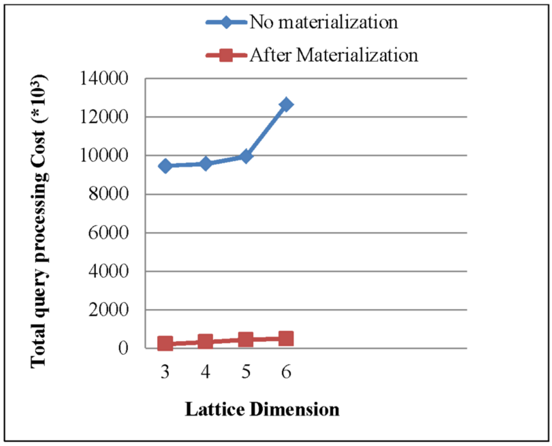
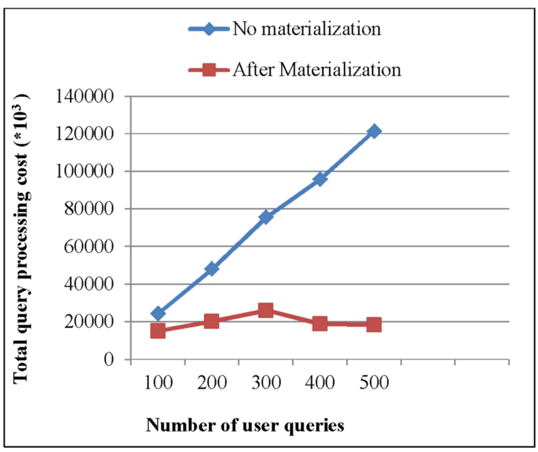


Figure 4. Number of user queries versus total query processing cost



Also, RWGWOMVS is based upon RWGWO and it avoids the problem of premature convergence due to the stagnation in local optima. Thus, RWGWOMVS is a suitable algorithm than other evolutionary algorithms in finding the materialized views.

Figure 5. Comparative results of RWGWOMVS with PSO and genetic algorithm

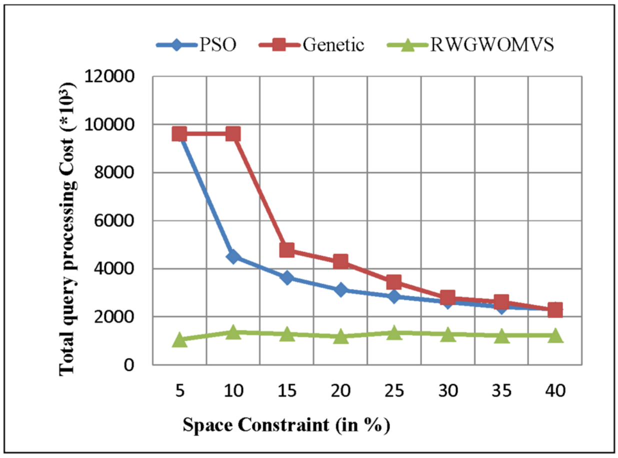
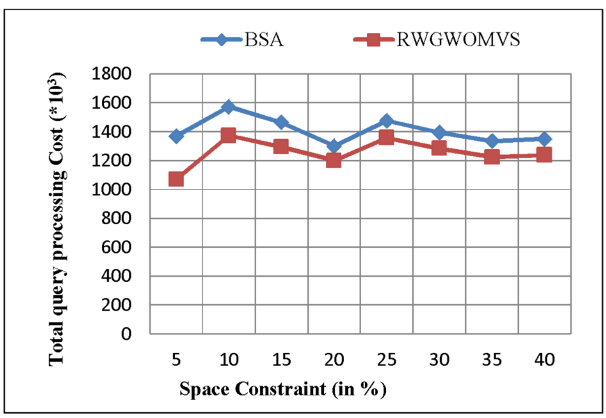


Figure 6. Comparative results of RWGWOMVS with BSA



CONCLUSION

In this paper, authors have implemented a meta-heuristic algorithm, Random walk Grey wolf optimizer (RWGWO), for materialized view selection i.e. RWGWOMVS using lattice framework to find an optimal set of views within the space constraint. RWGWO algorithm (Gupta & Deep, 2019), a meta-heuristic algorithm inspired by grey wolves is an improved variant of GWO algorithm. RWGWO (Gupta & Deep, 2019) mimics the leadership hierarchy and hunting mechanism of grey wolves in nature by improving the search ability of grey wolves based on random walk. Three main steps of RWGWO are (1) searching for prey, (2) encircling prey and (3) attacking the prey i.e. selecting the best one, (Gupta & Deep, 2019). Our proposed work, RWGWOMVS is based upon RWGWO and avoids the problem of premature convergence due to the stagnation in local optima. Three main steps (1) searching for views, (2) selecting the materialized views, and (3) selecting the best one, are implemented. Total processing cost of query is evaluated by conducting experiments on varying various factors like space constraints, dimensions of lattice and increasing the number of queries posed by user. Data sets from TPC-H benchmark were used for performing our proposed work. The results shown by RWGWOMVS are superior in evaluating the total query processing cost, when compared with Particle Swarm Optimization, Backtracking Search Optimization and genetic algorithm. It is observed that RWGWOMVS, scales well on increasing the dimensions of lattice and on increasing the number of queries triggered by users.

REFERENCES

Agrawal, S., & Chaudhuri, S. Narasayya v. (2001). Automated Selection of Materialized Views and Indexes for SQL Databases. *Proceedings of the 26th International Conference on Very Large Databases*.

Agrawal, S., Chaudhuri, S., & Narasayya, V. R. (2001). Materialized View and Index Selection Tool for Microsoft SQL Server 2000. In *Proceedings of the Special Interest Group on Management of Data Conference*. ACM SIGMOND.

Baril, X., & Bellahsene, Z. (2003). Selection of materialized views: A cost-based approach. CAiSE, 665–680. doi:10.1007/3-540-45017-3_44

Chaudhuri, S., Krishnamurthy, R., Potamianos, S., & Shim, K. (1995). Optimizing Queries with Materialized Views. *Proceedings of the 11th International Conference on Data Engineering*, 190–200. 10.1109/ICDE.1995.380392

Chaudhuri, S., & Narasayya, V. R. (1998). AutoAdmin 'What-if' Index Analysis Utility. *Proceedings of the Special Interest Group on Management of Data Conference*, 367–378. 10.1145/276304.276337

Choi, C. H., Yu, J. X., & Lu, H. (2004). A simple but effective dynamic materialized view caching. *International Conference on Web-Age Information Management*, 147–156. 10.1007/978-3-540-27772-9_16

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, *219*(15), 8121–8144. doi:10.1016/j.amc.2013.02.017

Ghosh, P., Sen, S., & Chaki, N. (2012). Materialized View Construction Using Linear Regression on Attributes. *Third International Conference on Emerging Applications of Information Technology*. 10.1109/EAIT.2012.6407899

Gosain, A., & Heena. (2016). Materialized Cube Selection using Particle Swarm Optimization algorithm. *7th International Conference on Communication, Computing and Virtualization*, 79, 2-7. 10.1016/j.procs.2016.03.002

Goswami, R., Bhattacharyya, D. K., & Dutta, M. (2013). Multiobjective Differential Evolution Algorithm Using Binary Encoded Data in Selecting Views for Materializing in Data Warehouse. *LNCS*, *8298*, 95–106.

Gou, G. (2006). A*search: An efficient and flexible approach to materialized view selection. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *36*(3), 411–425. doi:10.1109/TSMCC.2004.843248

Gray, J., Layman, A., Bosworth, A., & Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tabs and subtotals. *Data Mining and Knowledge Discovery*, *1*(Issue.1), 29–53. doi:10.1023/A:1009726021843

Gupta, H. (1997). Selection of views to materialize in a data warehouse. *Proceedings of the 6th International Conference on Database Theory*, 98-112. 10.1007/3-540-62222-5_39

Gupta, H., Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1997). Index selection for OLAP. *Proceedings of 13th International Conference on Data Engineering*, 208–219. 10.1109/ICDE.1997.581755

Gupta, H., & Mumick, I. S. (1999). Selection of views to materialize under a maintenance cost constraint. *Proceedings of the 7th International Conference on Database Theory*, 453-470. 10.1007/3-540-49257-7_28

Gupta, H., & Mumick, S. (2005). Selection of views to materialize in a data warehouse. *IEEE Transactions on Knowledge and Data Engineering*, *17*(1), 24–43. doi:10.1109/TKDE.2005.16

Gupta, S., & Deep, K. (2019). A novel Random Walk Grey Wolf Optimizer. In *Swarm and Evolutionary Computation* (Vol. 44, pp. 101–112). Elsevier.

Habiba, D. (2012). *Generating Materialized Views Using Ant Based Approaches And Information Retrieval Technologies*. IEEE.

Halevy, A. Y. (2001). Answering queries using views: A survey. *The VLDB Journal*, *10*(4), 270–294. doi:10.1007007780100054

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufman.

Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1996). Implementing data cubes efficiently. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 205–216. 10.1145/233269.233333

Horng, J. T., Chang, Y. J., & Liu, B. J. (2003). Applying evolutionary algorithms to materialized view selection in a data warehouse. *Soft Computing*, *7*(8), 574–581. doi:10.100700500-002-0243-1

Horng, U. J. T., Chang, J., Lin, B.J., & Kao, C.Y. (1999). Materialized view selection using genetic algorithms in a data warehouse system. CEC 99, *Proceedings of the 1999 Congress on Evolutionary Computation*, 3. 10.1109/CEC.1999.785551

Hung, M. C., Huang, M. L., Yang, D. L., & Hsueh, N. L. (2007). Efficient approaches for materialized views selection in a data warehouse. Information Sciences, 177(6), 1333–1348. doi:10.1016/j.ins.2006.09.007

Jain, H., & Gosain, A. (2012). A Comprehensive Study of View Maintenance Approaches in Data Warehousing Evolution. *Software Engineering Notes*, *37*(5), 1–8.

Kossmann, D. (2000). The state of the art in distributed query processing. *ACM Computing Surveys*, *32*(4), 422–469. doi:10.1145/371578.371598

Kotidis, Y., & Roussopoulos, N. (1999). Dynamat: a dynamic view management system for data warehouses. In *ACM SIGMOD Record* (Vol. 28, pp. 371–382). ACM. doi:10.1145/304182.304215

Kumar, T. V., & Ghoshal, A. (2009). A reduced lattice greedy algorithm for selecting materialized views. Information Systems, Technology and Management, 31, 6-18.

Kumar, T.V., & Haider, M. (2010). *A Query Answering Greedy Algorithm for Selecting Materialized Views*. Springer-Verlag Berlin Heidelberg.

Kumar, T. V., & Haider, M. (2012a). Materialized view selection for answering queries. In Data Engineering and Management. Springer Berlin Heidelberg.

Kumar, T. V., & Kumar, S. (2012b). Materialized view selection using simulated annealing. In Big Data Analytics. Springer Berlin Heidelberg.

Kumar, T. V. (2015). Materialized View Selection using Improvement based Bee Colony Optimization. Iin. *International Journal of Software Science and Computational Intelligence*, *7*(4), 35–61. doi:10.4018/IJSSCI.2015100103

Kumar, T. V., & Kumar, S. (2014). Materialized view selection using differential evolution. Iin. *International Journal of Innovative Computing and Applications*, *6*(2), 102–113. doi:10.1504/IJICA.2014.066499

Lawrence, M. (2006). Multiobjective genetic algorithms for materialized view selection in OLAP data warehouses. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 699-706. 10.1145/1143997.1144120

Lawrence, M., & Rau-Chaplin, A. (2006). Dynamic view selection for OLAP. *International Conference on Data Warehousing and Knowledge Discovery*, 33–44. 10.1007/11823728_4

Lee, M., & Hammer, J. (2001). Speeding up materialized viewselection in data warehouses using a randomized algorithm. *International Journal of Cooperative Information Systems*, *10*(3), 327–353. doi:10.1142/S0218843001000370

Li, J., Li, X., & Juntao, L. V. (2012). Selecting Materialized Views Based on Top-k Query Algorithm for Lineage Tracing. In *Third Global Congress on Intelligent Systems*. IEEE. 10.1109/GCIS.2012.28

Lin, W. Y., & Kuo, I. C. (2004). A genetic selection algorithm for OLAP data cubes. *Knowledge and Information Systems*, *6*(1), 83–102. doi:10.100710115-003-0093-x

Madhu Bhan, Kumar, & Rajanikanth. (2013). Materialized view size estimation using sampling. *International Conference on Computational Intelligence and Computing Research*.

Mami, I., Coletta, R., & Bellahsene, Z. (2011). Modeling view selection as a constraint satisfaction problem. In *Database and Expert Systems Applications* (Vol. 6861, pp. 396–410). Springer Berlin Heidelberg. doi:10.1007/978-3-642-23091-2_33

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Mistry, H., Roy, P., Sudarshan, S., & Ramamritham, K. (2001). Materialized view selection and maintenance using multi-query optimization. *SIGMOD Conference*, 307–318. 10.1145/375663.375703

Morse, S., & Isaac, D. (1998). *Parallel Systems in the Data Warehouse*. Upper Saddle River, NJ: Prentice Hall.

Nadeua, T. P., & Teorey, T. J. (2002). Achieving Scalability in OLAP Materialized View Selection. *DOLAP*, *02*, 28–34. doi:10.1145/583890.583895

O'Neil, O'Neil, & Chen. (2007). *The star schema benchmark (SSB)' Pat*. Academic Press.

Perez, L. L., & Jermaine, C. M. (2014). History-aware Query Optimization with Materialized Intermediate Views. *Proceedings of International conference on Data Engineering*. 10.1109/ICDE.2014.6816678

Sun, X., & Ziqiang, W. (2009). An Efficient Materialized Views Selection Algorithm Based on PSO. *Proceedings of the International Workshop on Intelligent Systems and Applications*. 10.1109/IWISA.2009.5072711

Talebi, Z. A., Chirkova, R., & Fathi, Y. (2013). An integer programming approach for the view and index selection problem. *Data & Knowledge Engineering, Elsevier*, *83*, 111–125. doi:10.1016/j.datak.2012.11.001

Tamiozzo, A. S., & Ale, J. M. (2014). A solution to the materialized view selection problem in data warehousing. *XX Congreso Argentino de Ciencias de la Computación*.

Valluri, S. R., Vadapalli, S., & Karlapalem, K. (2002). View relevance driven materialized view selection in data warehousing environment. *Australasian Database Conference*.

Xiangqian, S., & Lin, G. (2010). *An Ant Colony Based Algorithm for Optimal Selection of Materialized View*. IEEE. doi:10.1109/ICISS.2010.5654879

Yang, J., Karlapalem, K., & Li, Q. (1997). Algorithms for materialized view design in data warehousing environment. *VLDB, Proceedings of the 23rd International Conference on Very Large Data Bases*, 97, 136–145.

Yu, D., Dou, W., Zhu, Z., & Wang, J. (2015). Materialized view selection based on adaptive genetic algorithm and its implementation with Apache Hive. International Journal of Computational Intelligence Systems, 8(6), 1091–1102.

Yu, J. X., Yao, X., Choi, C.-H., & Gou, G. (2003). Materialized View Selection as constrained evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *33*(4), 458–467. doi:10.1109/TSMCC.2003.818494

Yu, J. X., Yao, X., Choi, C.-H., & Gou, G. (2003). Materialized view selection as constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics*. *Part C*, *33*(4), 458–467.

Zhang, C., & Yang, J. (1999). *Genetic Algorithm for Materialized View Selection in Data Warehouse Environments*. Springer-Verlag Berlin Heidelberg. doi:10.1007/3-540-48298-9_12

Zhang, C., Yao, X., & Yang, J. (2001). An evolutionary approach to materialized views selection in a data warehouse environment. *IEEE Transactions on Systems, Man, and Cybernetics*. *Part C*, *31*(3), 282–294.

Zhixiang, Z., & Dongjin, Y. (2014). Selecting Materialized Views based on Genetic Algorithm. In Advanced Science and Technology Letters: Vol. 45. *CCA 2014* (p. 65). SERSC.

Chapter 6
Web Services Reputation Based on Consumer Preferences

Rohallah Benaboud
ReLa(CS)2 Laboratory, University of Oum El Bouaghi, Algeria

Toufik Marir
https://orcid.org/0000-0002-8709-0680
ReLa(CS)2 Laboratory, University of Oum El Bouaghi, Algeria

ABSTRACT

With the increasing application of web services in our lives, selecting the right web service is becoming unprecedentedly difficult. Indeed, before paying the price of a web service, the customer always tries to make sure of his choice. One of the mechanisms used to put the customer in trust is to make available the opinions of other customers who have already used this web service. In the literature, many solutions for measuring the reputation of web services have been proposed. Unfortunately, they ignore certain aspects that we find important to ensure a more meaningful assessment of the reputation. Firstly, consumers do not always have the same satisfaction criteria, and as a result, they can judge the same web service differently. Thus, without knowing the consumer's preferences, it is almost impossible to give meaning to his opinion. Secondly, the qualities of a web service can be changed over time, and hence, the old ratings are no longer representative. In this chapter, the authors propose a novel reputation computation approach to deal with the problems mentioned above.

DOI: 10.4018/978-1-7998-2975-1.ch006

1. INTRODUCTION

With the growth of SOC (Service Oriented Computing), the number of Web services on the Internet is increasing. As a result, it has become difficult for a customer to select the desired Web services from those available on the Web. Web service selection is the most important task in the Web services model because they are useless if they can't be selected (Sharma & Kumar, 2011), for this, a major research effort has been undertaken to find effective solutions to select the best web services that meet the customers' needs. The selection of Web services is based essentially on their functional aspects (Inputs and Outputs), however, given the large number of Web services available and the diversity of customers, other aspects must be considered namely QoS (qualities of services) and customer preferences. Quality-of-Service is widely employed to represent the non-functional characteristics of Web services and has been considered as the key factor in service selection (Zhang et al., 2007).

Given the nature of the SOC environment, social approaches can be used in Web services selection (Billionniere et al., 2009). The social dimension is applied when people buy a product or service from a Website, such as Amazon[1]. For example, when searching for a book, a consumer can benefit from the opinions of other consumers. Consumer opinions can be used to define the reputation of this Website. With the proliferation of Web services-based applications and the consequent increase in the number of Web services, reputation systems should help address the recurring question that hinders Web service acceptance: How to find/avoid the "good"/"Bad" Web service based on past uses? (Sellami, 2011).

The reputation of an invoked Web service is a collective feedback rating of the consumers that have interacted with or used the service in the past (Wang et al., 2014). Once the Web service is invoked, the customer evaluates it by giving a rating that reflects its satisfaction with the QoS criteria of the used service. These ratings are then aggregated using an aggregation function to give a single value that estimates the credibility of the service provider or the quality of the services offered. This value, which represents the service reputation, influences the decision of other consumers when selecting services (Louati, 2015).

In the literature, many reputation systems have been proposed to accurately evaluate the reputation of Web services based on user feedback ratings (Noor, et al., 2013; Hendrikx, et al., 2015; Limam & Boutaba, 2010; Mármol & Kuhnen, 2015). Most of these systems usually calculate the reputation of Web services as the sum or the average of all given ratings. Although these methods of calculation allow an understanding of all, but, they do not allow such a detailed management of the reputation: It is not known on the basis of what criteria of QoS, the customer gave his opinion. Customers do not always have the same satisfaction criteria, and as a result, they may judge the same Web service differently. For example, a customer gives a good rating for a Web service that has a lower price QoS. If another consumer is interested in the execution time QoS, then the good rating given by the first consumer has no significance for the second one. Hence, without knowing the criterion of satisfaction of a customer, it is almost impossible to give meaning to his opinion. In our work, we propose that the customer rate the invoked Web service by mentioning their satisfaction criteria, i.e., it gives a rating to each QoS criterion. As a result, there is not only one value for the reputation score of a given Web service, but for each customer, this value is recalculated according to their preferences.

Over time, the qualities of a Web service can be changed by the service provider. In this case, the old reputation ratings are no longer representative. To cope with this situation, our approach gives more importance on recent ratings. In this paper, we propose a novel reputation computation approach to deal with the problems mentioned above. The proposed approach uses a rating model, which allows representing the information related to the rating context, especially, the ratings given by old consumers to the different QoS criteria of the invoked Web services. In order to calculate the reputation score of a Web service against a customer's preferences, this approach allows first to calculate the reputation score of a Web service against each of its QoS criteria based on the ratings given by old consumers, we give more importance to recent ratings, then the overall reputation score of a Web service against all QoS criteria, is calculated taking into account the customer's preferences.

The paper is organized as follows. To begin with, it presents related work on reputation systems in Section II, and then introduces a Rating model which enables the representation of the information needed to effectively measure the reputation of a Web service in Section III. Furthermore, we propose a novel reputation computation approach in Section IV. Finally we present an illustrative example in Section V and conclude the paper in Section VI.

2. RELATED WORK

Among the several approaches that exist for Web service selection, some approaches aim to involve consumer's opinion to discover the best Web services. In those approaches, each consumer rates the invoked Web service and shares these ratings with other new consumers. Approaches that calculate the reputation of an object as the sum of all ratings are called cumulative reputation approaches. One of the most used cumulative systems in online markets is the eBay2 system. This allows consumers to know whether potential business partners are trustworthy or not. The reputation mechanism used in eBay is based on the ratings given by buyers and sellers after the end of a transaction between them. The possible ratings are 1 (positive), 0 (neutral) and -1 (negative), and the overall rating of the reputation of an object is the sum of all the ratings given for that object. In some systems, the reputation of an object is calculated as the average of all given ratings. An example of these reputation systems is the Jurca and Faltings system (Jurca & Faltings, 2002).

The Sporas system, presented in (Zacharia et al., 2000), is a reputation system based on the following principles: Firstly, the reputation value of a former provider is never less than the reputation value of a new one. Secondly, provider with good reputation are the least likely to change reputation ratings compared to provider with a low reputation. Finally, when a provider is rated multiple times, only the most recent rating is considered. The Sporas system is more resistant to changes in the behavior of a user unlike Jurca and Faltings and Amazon systems which are based on the average of all ratings attributed to this user (Louati, 2015).

The work in (Wang et al., 2016) presents a collaboration reputation system. The reputation in this system is built on a Web service collaboration network consisting of two metrics: the invoking reputation metric which can be calculated according to other Web service's recommendation, and, the invoked reputation metric which can be assessed by the interaction frequency among Web services. The proposed trustworthy Web service selection approach based on this collaboration reputation can not only solve simple Web service selection but also complex selection.

The work in (Elabd, 2015) presents an approach that focuses on updating the reputation of Web services based on consumer confidence factors. The main idea of this approach is to ensure that the rating given by the consumer is true based on his confidence level, all consumers who gave ratings close to his rating, and the margin allowed between this rating and the current rating. The work presented in (Wang et al., 2011) proposes a solution to the problem of the malicious ratings. The proposed approach consists of two phases. The first phase allows detecting false ratings using the CSM method "Cumulative Sum Method". In the second phase, Pearson's correlation coefficient is used to detect and reduce the feedback effect of different users.

The work presented in (Gunasekaran et al., 2016) proposes a trust model for selecting the most trustworthy provider agent. This model which is based on three components: reputation, disrepute, and conflict, can enrich the trust in multi-agent systems by considering previous positive and also negative behaviours of providers in a heterogeneous multiagent environment. In (Fan, 2018), the author proposes a cooperation model based on trust rating in dynamic infinite interaction environment. The trust rating is proposed to control agents' selection selfishness and reputation, making them do the rational decision.

In (El-Kafrawy, 2015), the authors propose a Web Service discovery and selection model based on reputation model that consider consumer trust factor when calculating Web Service Reputation. The proposed discovery mechanism performs three main functions when the consumer's requests are received. Firstly, it finds the services that match the consumer's requirements. Secondly, the Reputation Module computes the trust factor of the consumers that rate to services. Finally, the discovery mechanism returns the matching list of services to the consumers. The work presented in (Tibermacine et al., 2015) proposes a bootstrapping mechanism for evaluating the reputation of newcomer Web services based on their initial Quality of Service (QoS) attributes, and their similarity with "long-standing" Web services.

After analyzing the different works that use the reputation systems, we find that they ignore certain aspects that we find important to ensure a more meaningful assessment of the reputation:

- Consumers do not always have the same satisfaction criteria, and as a result, they can judge the same Web service differently. If a consumer gives a good rating for a Web service that meets the criterion that interests them, this rating has no meaning for another consumer who is interested in another criterion. Thus, without knowing the criterion of satisfaction of a consumer, it is almost impossible to give meaning to his opinion.
- Over time, the qualities of a Web service can be changed by the service provider. In this case, the old ratings are no longer representative

3. RATING MODEL

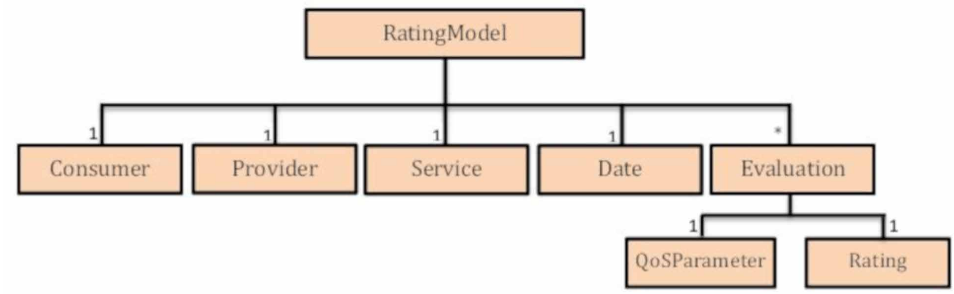
The rating for evaluating a Web service represents the degree of customer satisfaction with the Web service. It is represented by an integer between 0 and 4, and the meaning of each value is as follows:

- The value '4': Very satisfied.
- The value '3': Satisfied.
- The value '2': Little satisfied.
- The value '1': Dissatisfied.
- The value '0': Very dissatisfied.

In order to make better use of ratings, we propose a rating model, as depicted in Figure 1, which allows representing the following information related to the rating context:

- The identifier of the customer who gave the rating (usually, the client's IP address is used).
- The rated Web service.
- The Web service provider.

Figure 1. Rating model



- The date of the rating.

The ratings attributed to Web service or Web service provider according to the different QoS criteria. The consumer must give a rating for each QoS according to his degree of satisfaction.

If a customer rates the same Web service he has already rated, then the new rating replaces the old one. Figure 2 shows an instance in the ratings database according to the rating model:

4. THE PROPOSED APPROACH

In order to calculate the reputation score of a Web service against a customer's preferences, two parameters must be considered: The ratings given by old consumers to the different QoS criteria of this Web service, and the degree of importance of these QoS criteria for that new customer. The calculation of the reputation score goes through several steps as depicted in Figure 3.

First, we have to calculate the reputation score of the Web service against each of its QoS criteria. Secondly, each reputation score must be normalized in order to have a value in the range [0, 1]. Finally, we calculate the global reputation score of the Web service against all the QoS criteria.

4.1. Reputation Score Computing Against a Single QoS Criterion

The first step in computing the Web service reputation score is to calculate the reputation score of a Web service against each of its QoS criteria. The formula (1) shows how to calculate the reputation score of a service Sk against a QdS criterion

Figure 2. Example of an instance in the ratings database

```
<r:Rating>
 <r:Consumer rdf:about="#consumerC1"/>
 <r:Provider  rdf:about="#serviceProvider1"/>
 <r:Service rdf:about="#serviceS1"/>
 <r: RatingDate rdf:datatype="&xsd;date">01-01-2018</r:
 RatingDate>
 <r: ExecutionPriceScore rdf:datatype="&xsd;integer">2 </r:
 ExecutionPriceScore >
 <r: RespenseTimeScore rdf:datatype="&xsd;integer">4 </r:
 RespenseTimeScore >
 <r: ReliabilityScore rdf:datatype="&xsd;integer"> 4 </r:
 ReliabilityScore >
 </r.............................................. . .
</r:Rating>
```

Figure 3. Reputation process

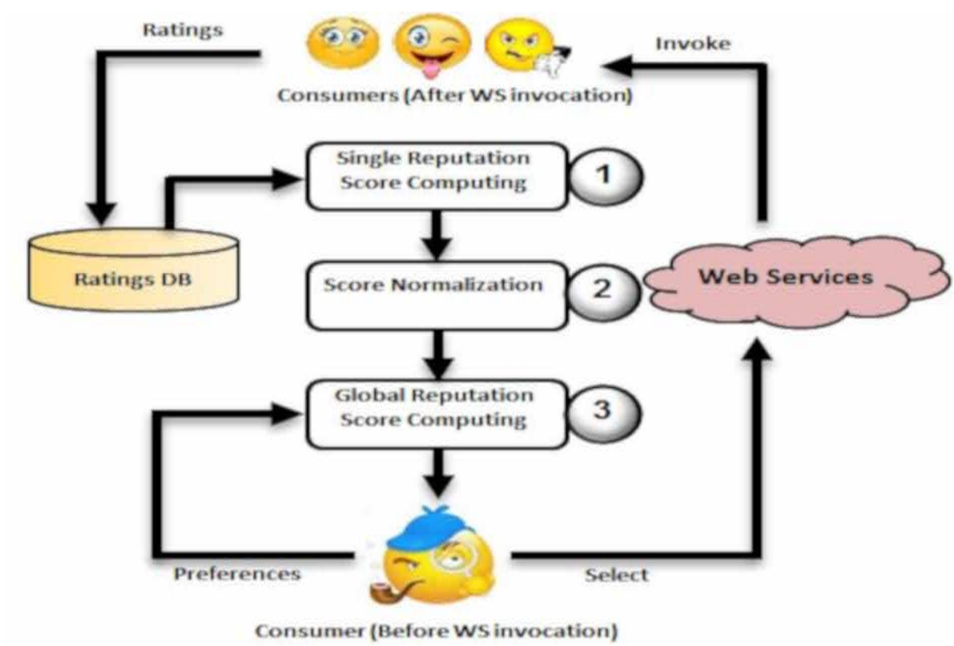


Table 1. Rating given to WS1 and WS2 (d= days)

	200 d	150 d	100 d	50 d	20 d
WS1	1	2	3	3	4
WS2	4	4	3	2	1

Qj, where Rate (Qj, i) is the ith score attributed to Sk against Qj, and N is the number of the given ratings which also represents the number of the clients who rated this Web service.

$$\mathrm{SingleRate}\left(\mathrm{Sk},\mathrm{Qj}\right)=\frac{\sum_{i=1}^{N} Rate\left(Qj,\ i\right)}{\mathrm{N}} \tag{1}$$

Table 1 shows the ratings given to two Web services WS1 and WS2 for the same QoS criterion in different dates, giving the number of days between the rating date and the current date.

By examining the evolution of the ratings, it is easy to see that the reputation of the Web service WS1 is increasing over time, while the reputation of the Web service WS2 is declining. This evolution encourages us to choose the Web service WS1 rather than the Web service WS2. On the other hand, if we apply the formula (1), the Web service WS1 will have a reputation score equal to 2.6, while the Web service WS2 will have a reputation score equal to 2.8; which means that WS2 is better than WS1.

To deal with this situation, we propose to give more importance to recent ratings. The formula (2) makes it possible to update the i^{th} score given to a Web service S_k against a QdS criterion Q_j by taking into account Dd, the number of days between the rating date and the current date. Naturally, the influence of a score will be decreased over time. Hence, we used the common logarithm function that expresses this influence on Dd. For normalization reason, we added 10 to Dd in order to eliminate]-¥, 1[values that make the updated rating value a wrong one. Table 2 shows the evolution of a rating equals to 3 according to Dd.

Table 2. Evolution of a rating equals to 3

	0 d	20 d	50 d	100 d	200 d
Updated ratings	3	2.30	1.76	1.50	1.3

Table 3. Updating rating for WS1 and WS2

	200 d	150 d	100 d	50 d	20 d
WS1	0.43	0.9	1.47	1.67	2.7
WS2	1.72	1.81	1.47	1.12	0.67

$$\text{Updated Rate}\left(\text{Sk}, Qj, i\right) \cdot \frac{Rate\left(Qj, i\right)}{\log_{10}\left(10 + d\right)} \quad (2)$$

Table 3 shows the notes given to Web services WS1 and WS2 after the update by the formula (2). The calculation of the reputation score gives for WS1 the value 1.43 and for WS2 the value 1.36, which means that WS1 is better than WS2.

To calculate the reputation score of a Web service S_k against a QoS criteria Qj, the new formula (3) takes into consideration both formulas (1) and (2) as follows:

$$\text{SingleRate}\left(\text{Sk}, \text{Qj}\right) = \frac{\sum_{i=1}^{N} \text{UpdatedRate}\left(\text{Sk}, Qj, i\right)}{\text{N}} \quad (3)$$

Score Normalisation

In this step, each SingleRate(Sk, Qj) score must be normalized in order to have homogeneous values in the range [0,1]. We consider the reputation score as a positive QoS criterion and we apply the formula (4) for normalization.

$$\text{NormalizedValue(SingleRate .=} \begin{cases} 1 - \frac{\text{SingleRate.}\max - \text{SingleRate}}{\text{SingleRate.}\max - \text{SingleRate.}min} & if\left(SingleRate.\text{max} \neq \text{SingleRate.min}\right) \\ 1 & if\left(SingleRate.\max = \text{SingleRate.min}\right) \end{cases}$$

(4)

Where, the values SingleRate.max and SingleRate.min respectively represent the maximum and minimum value of a single rating score within all given ratings.

Table 4 shows the normalized ratings of WS1 and WS2 given previously in Table 3.

Table 4. Normalized rating for WS1 and WS2

	200 d	150 d	100 d	50 d	20 d
WS1	0	0.20	0,45	0,55	1
WS2	0.92	1	0.70	0.39	0

Table 5. Example of ratings against 2 QoS criteria

WSs	QoS	200 d	150 d	100d	50 d	20 d
WS1	response time	2	1	2	1	1
	execution price	4	4	3	4	4
WS2	response time	4	4	4	4	4
	execution price	2	3	3	2	2
WS3	response time	2	1	0	1	1
	execution price	1	0	1	0	1

4.2. Reputation Score Computing Against all QoS Criteria

The overall reputation score of a Web service S_k against all QoS criteria, is calculated by the formula (5) as the weighted sum of the reputation scores of each QoS criteria, where weight (j) is the customer defined weight for QdS Qj, and M is the number of QoS criteria. This step aims to take into account the customer's preferences in the calculation of the reputation score, therefore, the reputation score of a web service S_k must be recalculated each time for a new customer.

$$\text{OverallReputation}\left(\text{Sk}\right) = \sum_{j=1}^{M} \text{Weight}\left(\text{j}\right) \text{NormalizedValue}\left(\text{SingleRate}\left(\text{Sk, Qj}\right)\right)$$

(5)

5. ILLUSTRATIVE EXAMPLE

Let us suppose two customers C1 and C2 who are interested in Web services having as QoS criteria the response time and the execution price. For C1, the response time is most important than the execution price, for this purpose it gives them the weights 0.8 and 0.2, respectively. Whereas, for C2, the response time is less important than

Table 6. Example of reputation score computing

Web Service	QoS	200d	150d	100d	50d	20d	Average Score	Normalized score
WS1	response time	0.86	0.45	0.98	0.56	0.67	0.7	0.13
	execution price	1.72	1.81	1.47	2.25	2.7	1.99	1
WS2	response time	1.72	1.81	1.96	2.25	2.7	2.08	1
	execution price	0.86	1.36	1.47	1.12	1.35	1.23	0.55
WS3	response time	0.86	0.45	0	0.56	0.67	0.5	0
	execution price	0.43	0	0.49	0	0.67	0.31	0

the execution price, therefore it gives them the weights 0.3 and 0.7 respectively. Table 5 shows the ratings given by all consumers to three Web services WS1, WS2 and WS3 against these two criteria.

The results of the application of formulas (1), (2), (3), and (4) are presented in Table 6.

Regardless of the preferences of both consumers, the results show that the best service is WS2 and the worst service is WS3. On the other hand, applying the formula (5), in order to take into account consumers' preferences, the reputation score of each Web service for each consumer is given as follows:

- Web Service WS1:
  ¬ Consumer C1: OverallReputation(WS1)= 0.8×0.13 + 0.2×1=0.30
  ¬ Consumer C2: OverallReputation(WS1)= 0.3×0.13 + 0.7×1=0.74
- Web Service WS2:
  ¬ Consumer C1: OverallReputation(WS2)= 0.8×1 + 0.2×0.55=0.91
  ¬ Consumer C2: OverallReputation(WS2)= 0.3×1 + 0.7×0.55=0.68
- Web Service WS3:
  ¬ Consumer C1: OverallReputation(WS3)= 0.8×0 + 0.2×0= 0
  ¬ Consumer C2: OverallReputation(WS3)= 0.3×0 + 0.7×0= 0

The end results show that the best Web service for the C1 consumer is WS2, while for the C2 consumer, WS1 is the best Web service.

6. CONCLUSION

With the growing number of Web services on the Web, accurate reputation measurement plays an important role for selecting the best service that meets the consumer preferences. Reputation can be considered as a collective measure of

Web service or service provider credibility based on the referrals or ratings from past consumers.

In this paper, we proposed a novel reputation computation approach for accurately measuring the reputation of Web services. Our approach takes into account the satisfaction criteria of each client. To do this, the reputation system asks the customer to give the degree of importance of each QoS criterion for him and to rate the invoked Web service according to each of the QoS criteria offered.

For future work, we intent to increase the number of consumers and Web services used for the experimentation, and also implement the proposed approach in the real-world environment.

REFERENCES

Billionniere, E., Greiman, D., & Gosha, K. (2009, December). A comparison of social service selection techniques. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing* (pp. 260-265). IEEE. 10.1109/DASC.2009.24

El-Kafrawy, P., Elabd, E., & Fathi, H. (2015). A trustworthy reputation approach for web service discovery. *Procedia Computer Science*, *65*, 572–581. doi:10.1016/j.procs.2015.09.001

Elabd, E. (2015). A dynamic reputation-based approach for web services discovery. *International Journal of Information Technology and Computer Science*, *7*(8), 31–36. doi:10.5815/ijitcs.2015.08.05

Fan, S. (2018). A Multiagent Cooperation Model Based on Trust Rating in Dynamic Infinite Interaction Environment. *Mathematical Problems in Engineering*.

Gunasekaran, A., Subramanian, N., Tiwari, M. K., Majd, E., & Balakrishnan, V. (2016). A reputation-oriented trust model for multi-agent environments. *Industrial Management & Data Systems*.

Hendrikx, F., Bubendorfer, K., & Chard, R. (2015). Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, *75*, 184–197. doi:10.1016/j.jpdc.2014.08.004

Jurca, R., & Faltings, B. (2002, July). Towards incentive-compatible reputation management. In *Workshop on Deception, Fraud and Trust in Agent Societies* (pp. 138-147). Springer.

Limam, N., & Boutaba, R. (2010). Assessing software service quality and trustworthiness at selection time. *IEEE Transactions on Software Engineering*, *36*(4), 559–574. doi:10.1109/TSE.2010.2

Louati, A. (2015). *Une approche multi-agents pour la composition de services Web fondée sur la confiance et les réseaux sociaux* (Thèse de doctorat). Université Paris Dauphine.

Mármol, F. G., & Kuhnen, M. Q. (2015). Reputation-based Web service orchestration in cloud computing: A survey. *Concurrency and Computation*, *27*(9), 2390–2412. doi:10.1002/cpe.3177

Noor, T. H., Sheng, Q. Z., Zeadally, S., & Yu, J. (2013). Trust management of services in cloud environments: Obstacles and solutions. *ACM Computing Surveys*, *46*(1), 12. doi:10.1145/2522968.2522980

Sellami, M. (2011), *Découverte d'annuaires de services Web dans un environnement distribué* (Thèse de doctorat). Télécom Sud Paris et l'Université d'Evry-Val d'Essonne.

Sharma, V., & Kumar, M. (2011). Web service discovery research: A study of existing approaches. *Int. Journal on Recent Trends in Engineering & Technology*, *1*, 106–109.

Tibermacine, O., Tibermacine, C., & Cherif, F. (2015, June). Regression-based bootstrapping of web service reputation measurement. In *2015 IEEE International Conference on Web Services* (pp. 377-384). IEEE. 10.1109/ICWS.2015.57

Wang, S., Huang, L., Hsu, C. H., & Yang, F. (2016). Collaboration reputation for trustworthy Web service selection in social networks. *Journal of Computer and System Sciences*, *82*(1), 130–143. doi:10.1016/j.jcss.2015.06.009

Wang, S., Zheng, Z., Sun, Q., Zou, H., & Yang, F. (2011, July). Evaluating feedback ratings for measuring reputation of web services. In *2011 IEEE International Conference on Services Computing* (pp. 192-199). IEEE. 10.1109/SCC.2011.32

Wang, S., Zheng, Z., Wu, Z., Lyu, M. R., & Yang, F. (2014). Reputation measurement and malicious feedback rating prevention in web service recommendation systems. *IEEE Transactions on Services Computing*, *8*(5), 755–767. doi:10.1109/TSC.2014.2320262

Zacharia, G., Moukas, A., & Maes, P. (2000). Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, *29*(4), 371–388. doi:10.1016/S0167-9236(00)00084-1

Zhang, L. J., Zhang, J., & Cai, H. (2007). *Services computing*. Beijing: Tsinghua University Press.

ENDNOTES

1. https://www.amazon.com/
2. http://www.eBay.com/

Chapter 7
Industry 4.0-Based Enterprise Information System for Management and Energy Efficiency

K. S. Sastry Musti
https://orcid.org/0000-0003-4384-7933
Namibia University of Science and Technology, Namibia

Helvi Iileka
Namibia Energy Institute, Namibia

Fenni Shidhika
Namibia Energy Institute, Namibia

ABSTRACT

The supply demand gap in energy sector in any country is a major challenge. Demand side management (DSM) and energy efficiency (EE) are the well-known solutions in the short term, and capacity addition is the long-term solution. However, both DSM and EE initiatives require significant investment and logistics if implemented in the traditional approach. The contemporary Industry 4.0 principles can be effectively applied to resolve several issues. This chapter proposes a novel enterprise information system (EIS) by treating the modern power systems as cyber physical system and to manage the processes of DSM and EE. A prototype system is suggested to pave the path for EIS, and the functional characteristics are illustrated with a few data visualizations.

DOI: 10.4018/978-1-7998-2975-1.ch007

INTRODUCTION

Modern power systems are very complex in the sense, they are fed from different forms of renewable energies through micro and smart grids (on the generation side) and supply linear and non-linear loads (at the customer end) through different power network components such as lines, transformers etc. It is normal to see that some of the generation plants are owned and /or operated by different companies, which are also known as independent power producers (IPPs). Both generation plants and customers are geographically wide-spread and values of generation and loads dynamically vary. In other words, there are several electrical parameters such voltages, currents and powers at several different locations and these are continuously changing. In general, utilities may have several different forms of measuring instruments, right from legacy analog equipment (non-smart, cannot communicate) to modern equipment (capable of computing and communicating) that collect data at regular intervals (typically every 15 or 30 minutes) for storage and processing. Over the time, this data gets huge and hence is big-data.

Now, IPPs will have different energy production costs and selling tariffs as specified by power purchase agreements (PPAs). Utilities specify different tariff mechanisms for different customers such as industrial, commercial and domestic. Even Time of Use (ToU) based tariff schemes with four different prices for electrical power in a single day is being implemented in different countries. This measure is part of 'Demand Side Management (DSM)' to influence customer power usage pattern. DSM, in a sense, encourages customers to exercise caution in using the electric power (Khripko, 2017). On the other hand, it is important that electrical loads have good efficiency and are of good quality to ensure they deliver the expected performance at reduced costs. In other words, inefficient or low quality (or equipment not confirming to set quality standards) may consume higher energy to deliver same, expected output and even may cause fire accidents. The aspect of energy efficiency of power apparatus and / or customer appliances is studied under the broad title Energy Efficiency (EE). Though DSM and EE are different in several aspects, in reality both of them depend on each other. In real world setting, electrical load parameters at the customer end are dynamically changing, which need to be measured by meters (Hemapala et al., 2012). However the data as measured by the meters needs to be processed and stored; and then should be made ready for various forms of computations and analysis. Both EE and DSM also require periodical energy audits (Roshan et al., 2014) to develop or to modify the implementation strategies.

Regulatory bodies or organizations specify various quality standards for customer service and even for power network maintenance activities. Both IPPs and power utilities have to follow the specifications set by respective state regulatory bodies.

Customer awareness and free flow of information exchange is utmost important. Most regulatory bodies across the world entrust this responsibility to IPPs and utilities.

Different strategies for DSM and EE have been adopted by various power utilities across the world as can be seen from the literature. These strategies depend on various factors including local power tariffs, power network capabilities, and economic status of stakeholders with customers. However, for successful implementation of DSM and/or EE a lot of data (which is authentic) is required be collected and processed from the customers. Big data tools have been recently used in power system monitoring control (Zobba, 2018; Zhou et al., 2016) for various purposes. Several computing architectures (Zhang, 2108) and data analytics have been suggested for composite systems. However, enterprise information systems, or data driven intelligence and analytics specifically for DSM and EE have not been presented in detail.

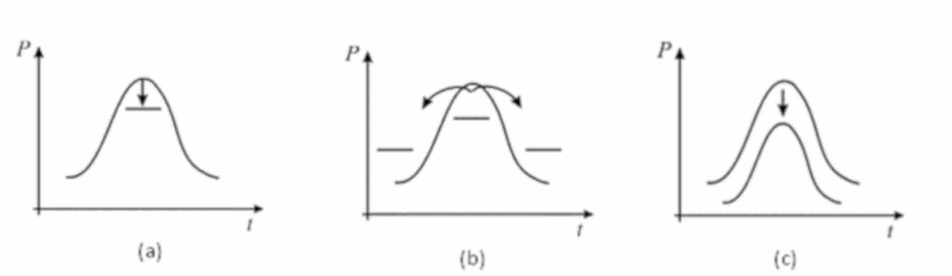
From the above, it is clear:

- There is a significant interest in implementation of DSM and EE.
- Advising and educating customers on DSM and EE is now an obligation.
- DSM and EE are data driven.
- There is a significant interest in applying big-data and data analytics in power systems.
- A well-defined enterprise information system is required to assist various stakeholders in implementing DSM and EE.

This chapter makes an attempt to explain both DSM and EE from the big-data point of view and suggests a framework for an intelligent information system that accomplishes the complete cycle from data specification, collection, processing to the end usage of the information obtained from the analytics. Also, the key issue of measurement and verification of the data will be taken up with a novel IoT based system. Versatile visualizations specific to power utilities, IPPs and customers will be presented. To accomplish these objectives, this chapter provides a comprehensive review of recent literature in three areas – viz, DSM and EE implementation experiences, information system frameworks for power systems, data visualizations and analytics for specific goals.

Overall content of this chapter is divided into various sections. Firstly, DSM is explained in simple terms, then salient aspects of frameworks adopted by various countries in the adoption and implementation of DSM. Then the chapter focuses on characteristics of a general EIS and explains how DSM requires such EIS. A prototype EIS for DSM is then presented that has several components. Two smartphone applications are proposed for the first time in line with industry 4.0 principles and also to assist energy auditors and wider public to simplify the processes and also as per the latest societal norms and user expectations. A small part of the data is

Figure 1. (a) Peak clipping (b) Load shifting (c) Energy conservation



then considered to illustrate common visualization requirements. The section titled discussion provides insights to various aspects of DSM, infrastructure, usage and data related issues.

DEMAND SIDE MANAGEMENT

In a typical power system consumers are supplied through 'feeders', that are essentially power distribution lines which run from transformers in substations. In a typical urban setting, there will be several different feeders with different voltages (11kV, 400V, 220V etc) for different types of consumers. Each feeder may supply hundreds of consumers. Most often, planning engineers might want to know if it is really beneficial to implement DSM on a specific feeder. For instance, if 40% of the consumers on a feeder are already having solar thermal systems, then base load will be less. In other words, there is no need to push for DSM on that feeder. Similarly, if 30% of the consumers have installed solar PV systems, then peak load on the feeder is already clipped and again there is no need to push for peak-clipping part of DSM. Similarly, a large scale institution or industry installs water chillers based central air-conditioning system, then energy consumption goes down significantly. This is due to the fact that water chillers consume 40% of energy, only a third of water when compared to conventional electric air conditioners. And their efficiency is very high. Further, replacing the electric air conditioners with water chillers is one of the popular DSM activities too. Figure 1 shows three important DSM strategies; viz., peak clipping, load shifting and energy conservation.

From the above, it can be see that it is important to understand how existing consumers are operating and what their energy consumption patterns are. Interestingly, utilities also want to know how much energy is consumed for what purposes and also what are the appliances residential consumers have (Ahmed et al., 2018).

How to obtain such detailed information? Energy audit provides an excellent inputs to both DSM and Energy Efficiency – Appliance Labeling (EE-AL). Several authors (Gan, 2012; You et al., 2018; Padhy, 2002) clearly indicated that a majority of the DSM implementations have seen poor results due to lack of implementation experience and lack of awareness. However, energy audits and related analysis are very data oriented. Firstly customers are more in number, data (such as details of all the appliances, monthly billing information etc) needs to be collected from each and every customer. And then consumers may add new apparatus, replace existing ones and then bill amounts vary every month. This indicates that the nature of consumer data is voluminous and slowly changing with respect to time; and that energy audits should be undertaken periodically at least once in three years. If the consumers are generating electricity, then data needs to be collected too. Typically solar PV data is collected once in every 15 minutes or even less like 10 minutes, due to random nature of the class of solar energy. Hence energy audit by itself is a heavily data driven process and indeed is continuous process.

To advise consumers properly about energy efficient apparatus, life-span, quality and performance ratings etc; a national policy needs to be in place. And then there should be a laboratory facility that is equipped to test various apparatus and properly accredited for quality processes. Such facility should proactively undertake testing of various appliances that belong to different brands and different capacities. In the days of competition, it is possible to have understated (or under quoted) name plate capacities at the lower prices. Poor making and low quality insulation might result in apparatus failures and/or even result in safety issues such as fire accidents. Results of all such testings along with the labels should be properly made available to the public so that they can plan on their shopping. Further consumers want detailed information on planning certain aspects. For instance, a commercial consumer wants to add solar PV system, but before that wants to investigate cost-benefit analysis. Such analysis depends on ToU tariffs; rules for net-metering/ feed-in metering; size of the PV system, battery, charge controller etc. Even this analysis is also data driven. It is also usual that consumers may want to visit existing installations and discuss with owners to understand the ownership experience. Having a comprehensive information system of various appliances, real life case studies and scenarios is essential (Roshan et al., 2014; Shipman et al., 2013). Testing labs need to stay active as many new products come into the market round the year; and hence testing (and updating information) is continuous process.

Expectations Of DSM

DSM is implemented in most countries through a national policy framework (Apajalahti et al., 2015; Stotzer et al., 2011). Framework policy is a general and more abstract set of principles and long-term objectives that guide the development of and form the basis of specific programs. It creates institutions establishing relevant laws and programs. The framework policy can also be in the form of a binding quantitative national savings target. Targets may be set as a capped value or volume, typically for a long-term period in the future or as a goal compared to a base year or projected baseline. They are defined in terms of a specific energy or demand goal (kWh or kW), savings (% as compared to base or baseline), emissions reduction or energy intensity (kWh per GDP). Detailed discussion of DSM policy frameworks of individual countries is out of the scope. However, following points summarize internationally adopted frameworks and hence are important for making a case for EIS for DSM. Information processed/ produced by EIS for DSM should help in:

- A plan to achieve energy independence and promote the efficient and rational use of electricity in all sectors of the economy.
- Achieving energy efficiency market transformation through standard practice based on trends.
- Providing data driven energy audits for end-users.
- Conducting energy efficiency demonstration projects.
- Creating case studies of large energy-intensive industries,, corrective actions with penalties for non-compliance.
- Reporting an annual energy consumption footprints of large consumers in terms of tonnes of oil equivalent as part of national and international expectaions.
- Creating an assorted repository of energy ratings, EE labels for most commonly sold appliances along with energy performance certificates for various commercial, residential and government buildings.
- Providing the information on various DSM initiatives undertaken by utilities and other stakeholders; and the corresponding results/ benefits.
- Assisting analysts, energy managers, energy auditors, regulator and common citizens with usable, and reliable information.

DSM can get as complex as it can be. Utilities across the globe found the compelling need for engaging a separate organization to deal with DSM. Education and capacity building measures are soft measures that help minimize possible rebound effects and induce long-term behavioral changes of end users. They include education, public outreach and awareness campaigns, training programs, detailed billing and disclosure programs, etc. However, education and / or training wider community is not an immediate function of the utilities. At present energy information centers have been established, however a properly designed EIS for DSM can potentially integrate and digitally transform such centers to eliminate the requirement of human intervention. Since the EIS for DSM umbrella is vast, the need for a separate organization to champion the cause of DSM arises. However, a few operational challenges need to be resolved. For example, the SCADA and real-time metering infrastructure remain with the utilities. Data collection and energy audits need legitimate access to customer premises. Such factors point to the need of greater cooperation between various stakeholders in seamless exchange of data and infrastructure.

Characteristics of Data of a Typical Power Distribution System

It is essential to examine the data characteristics so that appropriate platforms and technologies can be chosen to process the data to extract information. In power distribution systems data is measured and transmitted to respective engineering centers for processing purpose. Contemporary power distribution systems will have metering infrastructure as part of Supervisory Control And Data Acquisition (SCADA) systems. Future power systems will employ Industrial Internet of Things (IIoT) based measuring infrastructure. This infrastructure may co-exist along with SCADA or even may function independently.

Volume and Velocity: Power networks expand regularly; new customers get added continuously and hence number of data acquisition points only increase over the time. As stated earlier, data is collected once in 30 minutes or even 15 minutes from each measuring point. Hence the data that comes from very large number of points quickly adds to the volume, though frequency and velocity is almost constant.

Some portions of the data may be redundant; so care must be taken while processing the data to eliminate redundancy. Usually redundancy of data measurements may be deliberate (especially in case of electrical power distribution) to overcome the situations of faulty measuring equipment; or to detect power pilferage; or even for verification purposes in some cases. It can be understood that data volumes can be very high, which is one of the major attribute for this data to be classified as 'bigdata'.

Variability: The patterns of customer load consumptions do follow a general l trend over the day, though the trend itself may (will) vary from place to place depending on type of consumers. This variation of power consumption is typically represented with 'daily load curves'. Usually during the midday (from 11am to 3am) power consumption will be peaking in most cases. However, averages, peaks of the power consumptions will be different from month to month, depending on local climatic conditions. For instance, during summer season, air-conditioner loads will be predominantly causing higher consumptions, similarly during winter season the heating loads will result in higher loads. Utilities monitor peaks and averages such as 'daily peaks', 'monthly peaks', 'monthly averages' etc. Also it is important monitor the peaks and averages over the weekends. The reason behind looking for such information is different tariff structures. Based on the usage patterns, tariff structures for the next year will be determined. Hence, the data of a power distribution system varies, but not a high frequency. And variation of the data is essential to the utility and even for the state regulator.

Variety: The key parameters in any power systems are current (Amps), Voltage (Volts), active power (kW), reactive power (kVAR), power factor. Measured values of these parameters from various feeders, transformers or major load points constantly relayed. In the modern smart systems, same data for various loads (lighting loads, cooling/ heating loads etc) may be measured and relayed. However, it should be noted that measurement or individual loads is almost not possible and practicable. Hence tools such as energy audits are used to estimate the numbers, types and purpose of various loads in the customer premises. Such variety is essential for DSM to deploy appropriate measures. For instance, it is prudent to push for solar water heaters where electrical energy is predominantly used for heating the water. It can be seen that variety of data is present, but the variety itself is constant. Another important data is from network switching. Engineers do switch on/off circuit breakers and/or switches frequently as part of real-time operation for various reasons. With this, relationship between network components might change. For example, a group of consumers may be put on a different feeder over a period for some maintenance purposes. This does not necessarily add to the spread of variety, however is different data from above stated parameters. One of the DSM initiatives - ripple control (explained in later sections) is an example of network switching on/off.

Another aspect of variety comes into effect when DSM initiatives are taken up. For instance, installation of solar water heaters (SWH) in industry 4.0 setting will bring a host of parameters that are not essentially electrical, but physical. Similarly solar PV installations require measuring the energy generated over the sun-shine hours.

Accuracy and Authenticity: Measurement and Verification (M&V) process is the heart of the data collection in power distribution. For this meter will be periodically checked and even calibrated on a random basis. This field practice needs to continue to ensure trust on the data. However, in big data context, it is possible to deploy various algorithms to detect any abnormalities. Even block-chain technologies are suggested by a few authors, however such discussion is out of scope for this chapter.

Veracity: As above characteristics of data tend to get stronger, the confidence or trust in the data gets reduced. Usually the questions– how reliable or how meaningful this data is for business analysis? ; Can we rely on the trends provided by the information to take power purchase decisions over the next year?; what methodologies were followed in summarization (or information creation) - need to find convincing answers. Hence it is important that the data is processed carefully and the information is properly acknowledged (periodically) by field engineers.

End Use and Business Value: Several core business requirements of power utilities such as plotting various load curves, determination of network loading conditions, power flows through various distribution lines, generating wide ranging information for reporting to public and the regulator etc; depend on the data and information. These activities are time consuming and data intensive. Load forecast, day ahead planning, fund allocation for future purchases (in short and long terms), determination of tariffs (time of use, real time and critical peak timing) etc. depend on the information generated by the data. Hence, end use and business value of data collected in a typical power distribution environment is very high.

From the above it can be seen that data of a typical power distribution system has most common features of bigdata. Hence, appropriate bigdata platforms and tools need to be used for developing EIS for DSM so that information can be extracted in an efficient manner.

Impact of DSM Initiatives on Data Monitoring and Logging

Though, there are many DSM initiatives, three major aspects viz., installation of solar water heaters, solar-PV systems, ripple control and replacing existing lamps with energy efficient ones; will be taken up for discussion.

Solar water heaters (SWH) are very popular in energy saving. Hence, one of the major DSM initiatives is to install solar thermal systems on roof-tops to reduce electricity use for water heating purposes. Solar thermal systems are proven means of saving energy costs. Typical payback periods of solar installations will be anywhere between 2 to 7 years depending on the capacity, overall costs and local climatic conditions. However, there are associated challenges do exist. Consumers need precise information about payback times, maintenance issues, ownership experiences and even information on how to select the size. These challenges obviously point to lack of information and awareness. Then to encourage the consumers on installing solar water heaters, it is important to provide a simple handout (or even a smartphone app) that helps consumers in sizing the system for their specific purpose. Further, showing all the solar water heaters in their own community both in numbers and also on digital maps will have a positive impact. Naturally modern day consumers want to see all such cost saving measures and installations in smartphones with wide raging smart visualizations and apps. This obviously creates a need for a custom smartphone app. Though there are a very a few apps that are available at present, however they need local climatic data especially the monthly meteorological data to start with. DSM enterprise systems need local meteorological data in any case for accomplishing the tasks like short term and long term capacity addition.

Even the modern day SWHs are smart and do come with an array of sensors. The parameters that are continuously monitored include - flow rate of the hot water in liters/hour, water flow rate in various other loops, power of hot water consumption in kW, temperature of the hot water in the collector, temperature of cold water in the storage tank. Most importantly the local climatic parameters such as – global radiation in the specific location, wind speed etc., will also be monitored and actually need to be processed to estimate the cost benefits over the time. On the other hand, summary of this information needs to be properly channeled to the consumers to educate and thus encourage them invest in SWHs.

From the above it is clear that every component such as a solar thermal system, that is added to the system, needs to be monitored. Essentially each component will have its own parameters and hence a lot of data will be generated on a continuous basis. The data should be collected in real time for processing, storage and further analysis.

Ripple control is a known solution and hence indeed an integral part of DSM. This technique is used to control a large part of the uniform loads such as street lights, water heaters in a given area. As stated earlier, ripple control involves network switching and hence data acquisition is essential to determine the overall savings. Ripple control also results in reducing the energy consumption as well as in shifting load from peak time to off peak time or standard time by remotely switching off electrical geysers at households and business at specific times. It is essential to provide the information related to energy savings through versatile data visualizations, as shape(s) of load curve(s) will be altered significantly.

Enterprise Information Systems

An enterprise information system (EIS) is broadly defined as “a software package that helps a business perform and oversee certain processes, gathers data about these processes for analytics and maintains clear transparent records of these transactions” (Smyth, 2019). From the above sections of this chapter, it can be seen that DSM involves significant data collection, storage operations, computations and analytics. Also, there are means and ways that the data for DSM is collected, particularly in modern smart grid environments. Measurement and verification is major component of DSM and indeed is the driver of the trust for energy sales. On top of that smart control techniques are also part of DSM. Hence, in the context of DSM, EIS can be defined as “a large scale cyber-physical system that helps various stakeholders in the smart, modern energy markets that oversees various processes and provides information through data visualizations in a seamless manner”.

The chief characteristic of EIS is the centralized database and real time operation. DSM is all about integrating several databases (power network database, available generation data, consumer database, tariff database, historical data, consumer load data, consumer own generation data etc) and data computing models and information extraction processes.

The three major business categories (Smyth, 2019) of generic EIS are (customer relationship management (CRM), supply chain management (SCM) and enterprise resource planning (ERP). Even in the case of DSM all the three categories are present. The aspect of CRM in power utilities is critical and well published area (Rahman et al., 2016; Sastry, 2007). Worldwide, it is a standard for the utilities to serve customers through a dedicated call center to address grievances. CRM aspect alone requires a well-designed information system that essentially integrates various databases and e-CRM of power utilities indeed need to operate in real time. Some DSM activities include encouraging the customers to install solar thermal systems. For that several case scenarios need to be developed to advice on size of the system, capital costs, return period etc.

SCM in modern, smart grids refers to continuously varying quantities of load and generation. In case of smart grids, even consumers also produce energy and thus reduce the supply-demand gap particularly during day time or peak load hours. Utilities are expected to operate their systems carefully through proper real-time monitoring and control. At times, utilities need to resort to load shedding to disconnect the customers to save the power networks from overloading conditions. This again might result in revenue losses in terms of energy sales; however capital damages can be avoided.

ERP in DSM context is a continuous activity. Both load shedding and supply-demand gaps indicate the need for adding new capacity at the earliest. However, utility needs compressive information (obtained from reliable data/ accurate M&V). Cost benefit and payback time studies are required to properly plan the capacity addition. However capacity additional requires time and the interim solution to manage the system is the DSM. Usually utility engineers need to go through costing, projections, revenue estimates for planning the expansion of the system (Koh et al., 2016).

EIS For DSM

This section proposes a novel EIS framework for DSM. This EIS framework consists of a central database (that extracts data from different sources, including real-time M&V, a custom built test facility), IIoT based M&V system, two smartphone apps, computing models and data visualizations. The overall objectives of EIS for DSM include assisting consumers in:

- Selecting an appropriate size and energy efficient electrical apparatus with details of energy label /ratings.
- Understanding the aspects of net-metering and/ or feed-in-tariff systems.
- Understanding daily energy consumption patterns in various ToU tariff zones, monthly and yearly values of energy units consumed as well.
- Understanding energy saving opportunities through cost benefit analysis based on historical data for investing in solar PV or solar thermal systems.
- Evaluating their appliances for energy efficiency through smartphone apps.
- Obtaining energy ratings for the existing power apparatus.
- Shopping for new apparatus or replacing the existing ones.

EIS should also assist utilities in:

- Planning ripple control circuits on the power network and devise strategies to implement.
- Undertaking ripple control on a daily basis during overloading or low tariff times.
- Adjusting ToU tariffs based on consumer energy consumption patterns in a given day, month or season.
- Announcing energy saving schemes with incentives for those consume more.
- studying network overloading and thus determine specific interim DSM strategies such as peak clipping, load shifting.
- determining the overall energy savings for each control action through before and after case scenarios.
- Planning the capacity addition through additional energy resources such as solar PV or wind.
- Determining the need to add a base load plant and/or a load following power plant.
- Periodical assessment of revenues and operating expenses.

Then DSM is also expected to assist regulatory authorities in following various trends energy consumption and generation, flow of cash and revenues in a seamless fashion (Rosenow & Bayer, 2016). Further, it is expected to assist energy auditors in:

- Undertaking a detailed customer load survey to capture types, numbers and capacities of apparatus and appliances.
- Capturing the data directly from the customer location using smartphones.
- Providing information on energy efficiency of existing customer apparatus.

Assist organizations (like NEI) in organizing educational, informative workshops to wider public, which is one of the major DSM initiatives. With huge data, a well-designed EIS can convey a lot of knowledge through wide ranging possible case scenarios.

Careful planning is required to develop utility standard EIS for DSM, just as the case for EIS for any other purpose. Interestingly, building a prototype is the significant part since user requirements can vary widely. In other words, it is not possible to capture requirements that are 100% satisfying to everyone. And the users are from different walks of society, as utility engineers, common public, owners of IPPs, industry consumers, energy auditors and regulatory officials etc. Process of building EIS becomes more complex especially when a separate organization such as NEI takes up the responsibility, as real-time monitoring and control systems are owned and operated by the utility. Hence it is important to design and develop a prototype with most expected features and then let it mature through exhaustive usage, testing and improvising processes.

Prototype EIS for DSM

Building a functional prototype for EIS for DSM is critical. To begin with, firstly, the biggest challenge lies in difficulty in identifying accurate, final user requirements. Secondly, the subject of DSM is too wide to comprehend and most consumers may not have complete understanding in any case. Hence, it is not very well possible to capture user requirements easily in the initial phases. If there exists a prototype, then the users operate and then may be able to suggest their needs or even possible customizations. Figure 2 shows a typical schematic for the proposed prototype.

As can be seen from figure 2 the prototype consists of DSM servers - a database server (for prototype, any license free database such as MySql can be considered), application server for data abstraction, computing, information processing, creating visualizations; and then different databases such as electrical power network, customer information, tariff structures etc. IoT based M&V infrastructure involves measurement of various parameters at key places in the network. For instance, real time readings are required at the upstream, that is at the substation. This may initially look redundant as utility already has the metering infrastructure. However, it is important to have IoT based, low cost infrastructure that is part of the prototype for comparison and benchmarking purposes. The parameters measured here should be tallied with its own readings at various points in the downstream, including at the consumer locations. It is recommended that utility grants access to prototype infrastructure to be installed after the current transformers and voltage transformers located in the substations. Similarly IoT based M&V apparatus is required to be placed at all industrial and commercial consumers as their tariff structures are different from residential consumers. Indeed, revenues from these two category consumers to the utilities are quite significant when compared to residential consumers. (Koh et al., 2016). Details of circuitry and IoT infrastructure are avoided, as this chapter focuses on overall framework for EIS.

EIS Prototype for DSM should have a majority of the functional characteristics of utility scale EIS. For instance, functions like - data collection, fog / edge computing, pre-stored case scenarios and visualizations etc should be more or less same. Utility scale EIS will essentially be equipped with big data and high performance computing platforms; where as prototype may be designed to operate on scaled down platforms such as desktop/ network technologies (Yu et al, 2015, Tu et al, 2017). Prototype EIS may be initially designed to operate with static (or slowly changing) but huge volumes of data. During the transition to utility scale many hardware and software layers may have to be changed for various reasons mainly for data processing and computing efficiencies. Big data tools, platforms and technologies have to be essentially used in corporate scale EIS (Munshi et al., 2017; Schuelke-Leech et al., 2015). A significant investment needs to be made towards wide area communications and cyber security purposes. It is also common to see special budgeting needs to be put in place for additional cooling for data storage facilities and routine maintenance activities as the scale of system grows over the time.

MOBILE APPS FOR ENERGY AUDITS AND ENERGY EFFICIENCY

Energy audit and power apparatus labeling are time consuming activities that require a lot of human efforts and involve other logistics. For instance, energy audit teams need to visit each customer premises to make a comprehensive record of each and every load. Two smartphone based apps are proposed for the first time; one for conducting energy audit and other for knowing the energy efficiency of the power apparatus and appliances. Figure 3 schematic of the EIS along with the two proposed apps and the data flow. Essentially data is collected energy audit app and sent as input to the EIS. Energy efficiency app serves the consumers in understanding patterns of their own consumption (if they have home automation systems) and the efficiencies of various appliances; from the stored information as collected from previously conducted lab tests and labels given.

The energy efficiency app essentially works on both fog and edge computing principles to show the necessary information. For instance, customer end data is analyzed locally within their own automation infrastructure and only the summary (or required) information is sent to the EIS. The app will extract the required information on each update (for information on various apparatus) from the cloud

Figure 2. Schematic diagram for prototype EIS

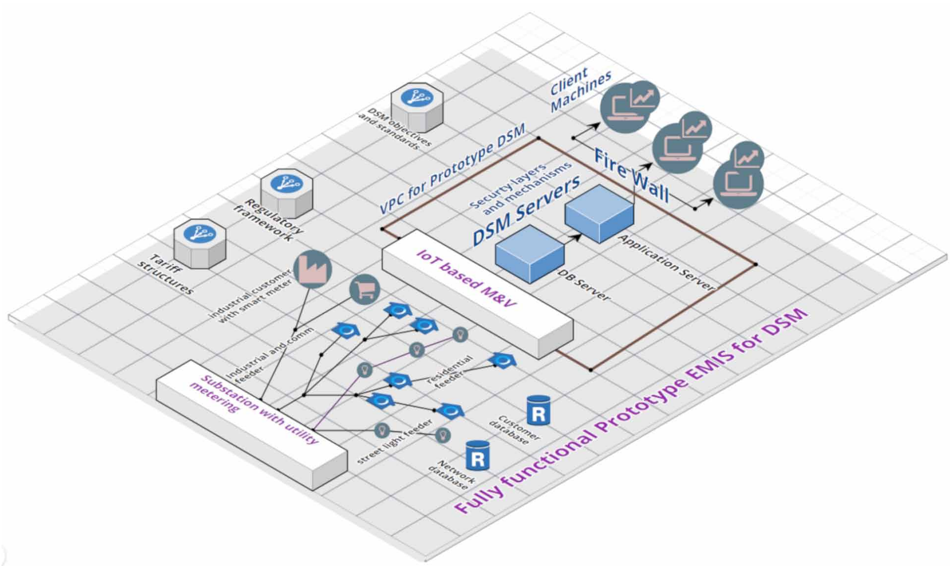
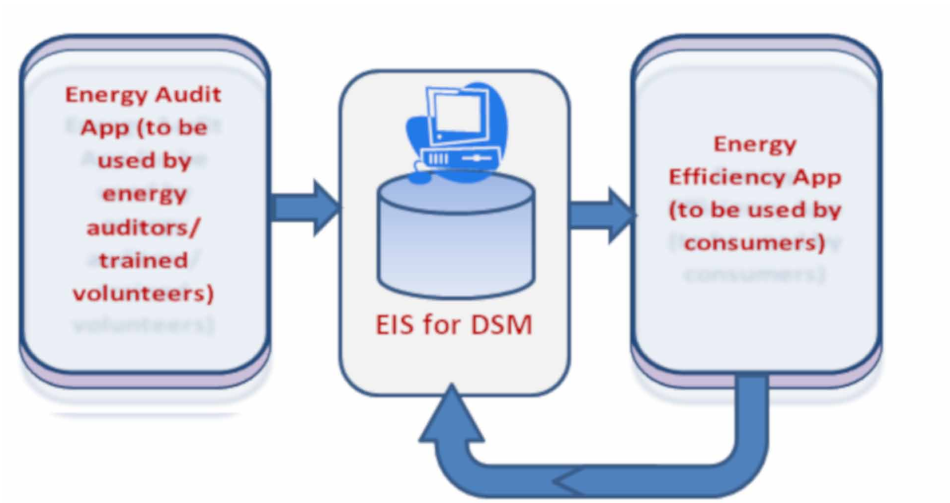


Figure 3. Schematic for smartphone apps for energy audit and energy efficiency



that EIS operates. The feedback loop from energy efficiency loop is meant for knowing what type of apparatus and what systems consumers are looking for. With such information, EIS can be further augmented in its functionality.

Table 1. Customer information

Customer Type:	Meter Number:
Address:	Name of the Consumer:
Average Monthly Utility Bill:	Email / Phone Contact:

The energy audit app essentially should be used by trained and certified energy auditors and volunteers. The reason why this cannot be used as ‘crowd sourcing’ app to collect the data is – integrity of the data is very important. Data cleaning alone takes up significant efforts. Data collection using the contemporary crowd sourcing principles may not be necessarily suitable for DSM / EE-AL. Then EE-AL app is open to everyone as it does not accept any inputs from users and essentially to advice on various aspects as discussed above.

Table 1 shows an example of the information that is usually collected for energy audit purposes. It should be noted that this is only an example for the purpose of illustration and in an ideal situation, more detailed information may be required to be obtained in order to fulfill the objectives of energy audit and energy efficiency studies.

Thanks to the technological revolution, the penetration of smartphones is very high even in the semi-illiterate population. Android platform and the apps are so intuitive that users do not require any formal training to operate. It is possible to use smartphone apps to undertake the data collection for both energy audit as well as energy efficiency studies. The Energy Audit and Apparatus Labeling (EAAL) app should be designed to collect the details of types of lighting loads, their numbers, ratings of power apparatus such as washing machines, air-conditioners, microwaves, electric geysers, electric stoves, entertainment / electronic loads such as television, music systems etc. Table 2 shows a sample for a typical template that can be used for customer load survey.

Users must be able to input the details by themselves and also upload the images (using the camera of the smartphone) of name plates showing the electrical ratings of their apparatus. Even the address can be obtained by location feature of the phone. The app should be simple to operate collect the information of the consumer information. Appropriate care should be taken in handling the data (name, email / phone contacts and location etc) and users must be provided assurance when they download / install the app. To sensitize about the app, local authorities should issue notifications through print and electronic media.

Table 2. Template for customer load data

	Quantity	Total Watts
Incandescent Lamps		
Fluorescent Lamps		
LED Lamp		
Refrigerator		
Air-conditioner		
Microwave		
Water Heater		
Solar Water Heater		
Room Heater		
Electrical Stove		

The information collected from the EAAL app should be used to provide diverse insights and solutions that can benefit consumers as well as utility. Once user base increases, EAAL app will be collecting huge amounts of information about different types, brands and sizes of the electrical apparatus. The energy efficiency test center (EETC) would have conducted the necessary tests under standard operating conditions on various brands and would have designated the appropriate labels. Now, the back office personnel need to examine each consumer load to attach the energy rating label. This information is valuable to the customer and also to the utility in general; in terms of replacing inefficient apparatus with more efficient ones. The benefits can be provided as 'Information as Service'. Typically for individual consumers, the app should be able to provide information on ratings/ or labels of each existing device based on the photos of the name plates and also the devices themselves. In case consumers want to buy a new apparatus (for example an air-conditioner) In doing so, the EAAL app may provide links to local market vendors with costs. This brings in some revenue to the EAAL app as well besides guiding the consumers.

Over the time, consumers may add new apparatus and/ or replace the existing apparatus with new ones. Hence, EAAL should be designed in such a way that it is friendly enough for the users to update their information on regular basis. From the above, it can be seen that an app like EAAL that works on crowd sourcing principles has good potential to assist consumers and also increase their awareness. The information generated by the data of EAAL can be great resource for the utility as well. It is possible to observe the relation between load consumption (from the bills and meters) to the EAAL data. If majority consumers in a specific area are using electrical geysers, then utility may plan for ripple control. On the other

Figure 4. Data for a small urban feeder (partly shown)

Time of the day	0	0.15	0.3	0.45	1	1.15	1.3	1.45	2	2.15	2.
Lighting Load (kW)	30	30	30	30	30	30	33	35	35	38	4
Heating / Cooling (kW)	40	42	40	40	45	50	55	57	60	65	6
Water Heating (kw)	10	8	10	10	12	15	25	25	25	30	3
Other Loads (kw)	40	42	42	45	40	43	45	42	45	43	5
Total Load (kW)	120	122	122	125	127	138	158	159	165	176	19
Solar PV generation (kW)	0	0	0	0	0	0	0	0	0	0	
Net load (kW)	120	122	122	125	127	138	158	159	165	176	19

hand, it is possible to observe if there is any pilferage or significant changes in the power consumption in the feeders. For example, sudden drop in the consumption is generally flagged for further probing towards pilferage. However, if a good number of consumers on a feeder install solar water heaters, then it leads reduced power consumption. Energy audits also help utility to plan expansion of existing and/or new power networks.

DATA VISUALIZATION FOR DSM

Now, the output of DSM is as seen from above, is data visualization intensive. To illustrate load research, a typical city feeder is considered along with its daily load data. It supplies a small urban area with a combination of small-scale industries, commercial establishments and residential customers. Using this example, data visualizations are developed to demonstrate the effectiveness of the prototype, its data models and computing features. Details of integration of data sources, database design and relational aspects are avoided.

Figure 4 shows a part of a sample data. for both load and solar PV generation at every 15 minute interval. Now for the purpose of utilities (or other stakeholders), overall load composition over the day is shown. Then DSM is always represented with daily load curves and hence even the EIS (or the prototype) is expected to produce same visualization. Figure 5 shows the composition of the overall daily load. This information is useful to determine the extent of ripple control and even energy conservation parts of DSM.

Figure 6 shows actual load, values of PV generation and the net load over the day. It can be seen that the visualization matches the theoretical approach specifically in this case, the achievement of peak clipping.

Now, if a small number of customers have installed solar thermal systems then there will be lesser energy consumption. Data for this case is shown in figure 7 and the composite visual is shown in figure 8

Figure 5. Representing composition of overall feeder load

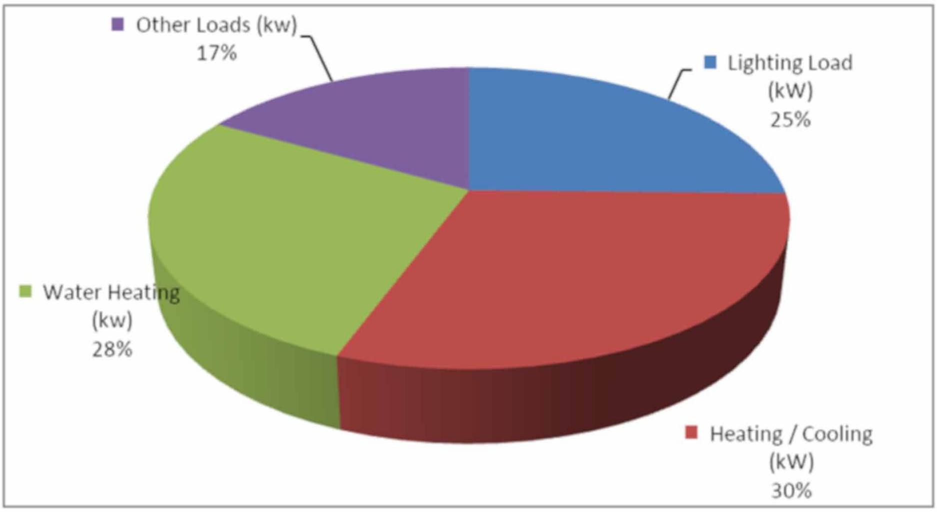
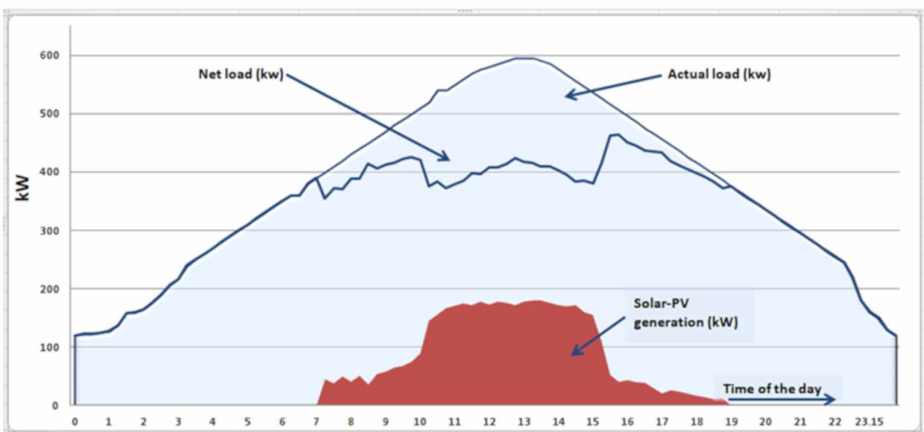


Figure 6. Load curves with and without peak clipping

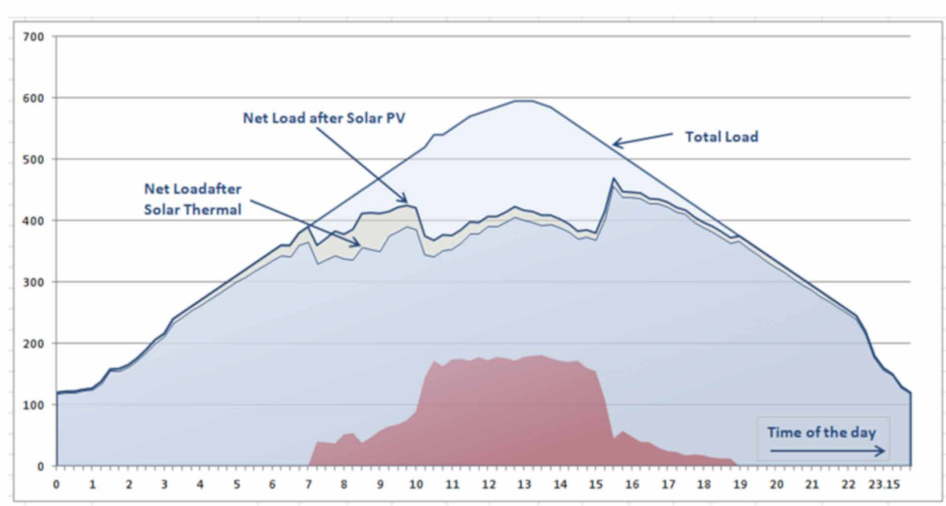


The above visuals (figures 5 to 8) show the information over a single day and for the entire feeder from the readings captured at the upstream, in the substation. It should be noted that data visualizations shown above are mainly useful for utility and regulatory personnel. For consumers, data visualizations will be different.

Figure 7. Load data with solar thermal systems included (partly shown)

Time of the day	0	0.15	0.3	0.45	1	1.15	1.3	1.45	2
Lighting Load (kW)	30	30	30	30	30	30	33	35	35
Heating / Cooling (kW)	40	42	40	40	45	50	55	57	60
Water Heating (kw)	10	8	10	10	12	15	25	25	25
Other Loads (kw)	40	42	42	45	40	43	45	42	45
Total Load (kW)	120	122	122	125	127	138	158	159	165
Solar PV generation (kW)	0	0	0	0	0	0	0	0	0
Net load (kW)	120	122	122	125	127	138	158	159	165
Load reduction due to Solar - Thermal	-2	-2	-2	-2	-2	-3	-4	-4	-4
Net load with S-T (kW)	118	120	120	123	125	135	154	155	161

Figure 8. Load curves with and without peak clipping as well as energy conservation



TESTING FACILITIES AND ENERGY EFFICIENCY LABELING

Energy efficiency and apparatus labeling require good facilities to test various power apparatus under standard, uniform and unbiased operating conditions. For this a centralized energy efficiency test center (EETC) needs to be established on specific requirements that depend on local market conditions and wide ranging apparatus that might be in use; in general. Testing standards and protocols must be developed and then revised over the time based on the needs. The common parameters that need to be tested include, but not limited to current drawn, energy consumed over a fixed time and overheating of the surface etc. Further, quality of the build, electrical insulation and general safety issues will need to inspected. After that standard labeling may be done as per the set standard processes. In a way modern smart grids can be benefited with contemporary big data analytics and related technologies (Schuelke-Leech, 2015). Indeed it will be a paradigm shift (Akhavan-Hejazi, 2018).

BIG DATA OPERATIONS

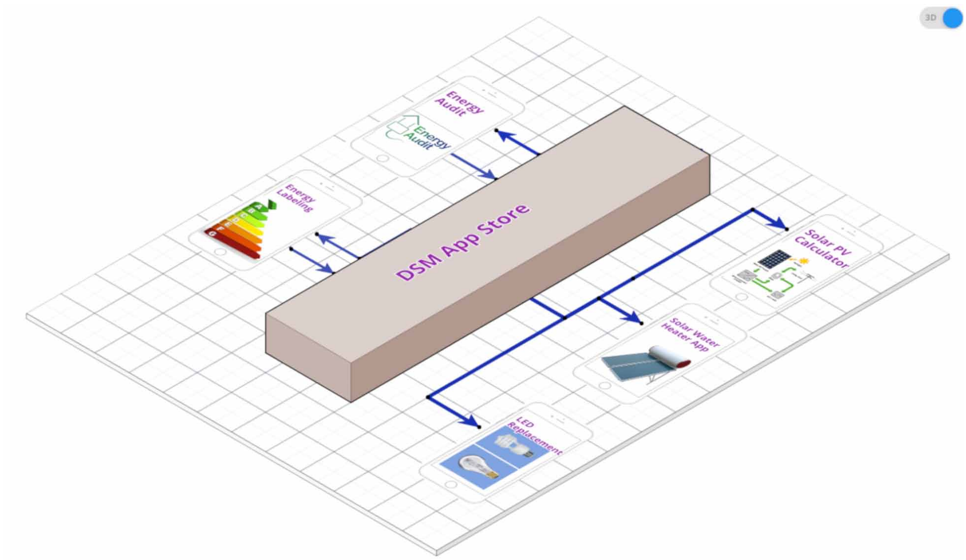
Considering the EIS data characteristics, several big data related operations need to be undertaken.

- Processing the huge volumes of data generated by various smart measuring equipment requires 'dimensionality reduction' to reduce communication costs, computing complexity, and data storage requirements; etc. Random projection technique can be effectively applied to provide a reduced sketch (information) of the measured data.
- K-Means algorithms based on Fuzzy C-means, ANNs can be used for load classification and even for forecasting the load for short term and long term planning.
- For determination of network losses, power flows on huge data sets, high performance computing, distributed and parallel computing techniques need to be used.

In any urban area, several feeders will be distributing the electric power. Nature of data collected from the feeders is more or less same. Hence for data aggregation, MapReduce concepts along with parallel processing need to be employed for providing the overall summary. Platforms such as Cassandra database can be considered to store large datasets. Besides these, several opportunities do exist in developing cloud applications based on fog and edge computing models to reduce the communication and storage burden. In fact, the EIS even can deploy a repository for smartphone apps. Figure 9 shows a few apps as already discussed. Using these apps, consumers can investigate the cost benefit prospects of investing in solar PV, SWHs, energy efficient lights etc. Auditors can use the apps for information collection and energy labeling processes.

As can be seen from the above discussion, there are several opportunities to develop various algorithms, platforms, smartphone apps etc for data processing, analyzing and visualization. One of the interesting aspects is the custom GIS environment.

Figure 9. DSM app store



It is common to use google-maps to display geographical location in user interfaces; and indeed this feature is one of the essential components in data visualizations. As far as DSM is concerned, geographical location is important for several reasons. Some of the reasons include – consumer location, network component (such as substation) location, mapping of SWHs in a given area, network zones effected by ripple control, mapping of solar PV systems etc. System planners and even consumers may want to know the areas that have heavy penetration of SWH or solar PV installations for various reasons. Since early 2018, google maps are no longer free and are priced for most usage purposes (Maps, 2019). This poses challenge to the system development as GIS platform is now priced as per the usage, though mobile usage of the google-maps is still free. This challenge naturally leads to a new opportunity for developers to consider designing custom GIS platforms. Though this can be laborious, at the end, sustainable software development environment can be achieved.

DISCUSSION

From the above sections, it can be seen that DSM strategies in modern smart grids are different and require even a smart EIS. Data collection, real-time M&V and information visualization are the essential elements. Then there are several challenges in implementing DSM too. In a typical market driven conditions, utilities depend heavily on electricity sales to the consumers. More the energy consumed, more is the revenue to the utility. When DSM is implemented, consumers tend to 'shift' to lower tariff times in the day to save on energy bills simply due to ToU tariffs. Due incentives and/or encouraging conditions a good number of consumers might have installed solar thermal systems; and thus consume less energy, which accounts up to 20% to 30% of their original consumption. With ripple control method, heavy consumer loads such as air-conditioners and water heaters will be switched off; and hence very low energy consumption again. Replacing and/or using LED lamps will lead to lower energy consumption too. From the above it can be seen that DSM actually results in lesser revenues to the utility. Even though there are known methods (such as decoupling the tariffs) to compensate the revenue losses, most utilities tend to show lesser interest in DSM implementation. On the other hand, if the utility is able to add more generation (which is very expensive than energy saving) and consumers so that it can sell more energy then it will be an excellent revenue proposition. Important point that needs to be noted is adding generation that has load following characteristic (such as solar PV) will be of great help. Hence, utilities must consider establishing more solar PV plants to supply the loads that peak during mid-day. DSM also might result in load shifting (as consumers move to cheaper tariff times) that in turn will push the base load up. In this case, utilities are left with only option of adding base load generation capacity (such as coal / thermal power plants)to meet the load that exists round the clock. Normally, cities expand continuously due to people migrating from rural areas and thus energy demand in the cities always increases and hence, DSM is a continuous process. DSM results can vary from place to place and may depending on various parameters. The role of EIS is vital as it should be able to quantify the benefits in an appropriate manner using reliable data. The success and usability of EIS depends on the user participation and dependability of its output. To achieve the best results, prototype EIS should be put to rigorous tests and then appropriate modifications to make it acceptable to all the stakeholders.

Some aspects of energy audit and energy efficiency from regulatory framework are to be noted here. Time of Use (ToU) tariffs are applied to both industrial and commercial consumers. Utility bills are very high when compared to domestic consumers due to ToU tariffs. Usually these consumers are expected to undertake energy audit studies through licensed firms/ or consultants /or energy auditors on regular intervals and furnish the reports to the regulator. Energy auditors also examine safety aspects also during the inspections. Energy audit reports are expected to contain the Energy Saving Opportunities (ESOs), safety aspects, and actions to taken or to be taken to achieve those opportunities or to improve the performance of operation. With such regulatory requirements in place, most industrial and commercial consumers will be able to meet energy efficiency and operational safety requirements. This process also help consumers to save significant amount of money as the overall electrical power consumption reduces.

Domestic consumers are typical category in the sense, they are in majority in numbers and ToU tariffs generally are not applied. Physical visits to the customer premises require advance notification to the public through news and print media to sensitize about the purpose of Energy Audit and Energy Efficiency projects. Customer reactions and level of cooperation may differ greatly from individual to individual. It should be noted that visits to commercial and industrial consumers will be generally, relatively easier when compared to domestic consumers due to various reasons including socio-economic and educational backgrounds. On the other hand, the major objective of DSM is spread awareness about lessening the energy usage and increasing the energy efficiency.

Combining home automation with DSM will be the next stage of development. As principles of fourth industrial revolution (or simply Industry 4.0) are applied in every walk of life, chances are high for utilities to implement DSM in near future from the industry 4.0 context. DSM requires large databases, computing technologies, continuous, real-time monitoring (eg: energy measurements, status of apparatus etc), physical control loops (eg: ripple control), versatile data visualization formats, mobile apps and of course communication systems etc. Thus DSM is a perfect cyber-physical system and indeed is a good example candidate for industry 4.0. IIoT can be effectively used for real-time monitoring and control purposes. Cloud based home automation systems that provide access to clients through mobile apps to control their home appliances are already available. This is better known as 'Home Energy Management as a Service HEMaaS'. Implementing DSM from the point-of-view of industry 4.0, will be more cheaper as IoT can help mitigating costs in lieu of expensive (and legacy) SCADA systems.

The modern market driven energy environment is very competitive and success of the utilities depends on cost effective capacity addition as a long term solution; and meeting the energy needs of the consumers through effective DSM initiatives. In most situations, utilities need to engage both long term and short term strategies simultaneously. EIS plays a significant role in accomplishing both of the tasks. Though utilities may be opposing aggressive form of DSM (due to fears of losing revenues), the success depends on finding new consumers and channeling the saved energy, let us from peaking clipping.

CONCLUSION

DSM and EE are indeed very complex, especially in modern smart grids that are driven by market conditions. Relevant international trends, practices and experiences from various countries have been presented. A strong case is made for EIS for DSM, as it provides an essential solution for managing various processes such as data collection, real-time M&V and information visualization. Significant logistical and physical initiatives such as solar thermal installations, ripple control have been explained along with associated impact on data generation. Various challenges in implementing DSM and EE are discussed. Data visualizations based on day's information are provided to illustrate the effectiveness of the prototype. With due diligence and careful planning, it is possible to build a prototype EIS and then further develop it to utility scale EIS for wider acceptance. Further, a prototype EIS can essentially support educational awareness and information dissemination.

REFERENCES

Ahmed, N., Levorato, M., & Li, G. P. (2018). Residential consumer-centric demand side management. *IEEE Transactions on Smart Grid*, *9*(5), 4513–4524. doi:10.1109/TSG.2017.2661991

Akhavan-Hejazi, H., & Mohsenian-Rad, H. (2018). Power systems big data analytics: An assessment of paradigm shift barriers and prospects. Energy Reports, 4, 91–100. doi:10.1016/j.egyr.2017.11.002

Apajalahti, E., Lovio, R., & Heiskanen, E. (2015). From demand side management (DSM) to energy efficiency services : A Finnish case study. doi:10.1016/j.enpol.2015.02.013

Diamantoulakis, P. D., Kapinas, V. M., & Karagiannidis, G. K. (2015). Big Data Analytics for Dynamic Energy Management in Smart Grids. Big Data Research, 2(3), 94–101. doi:10.1016/j.bdr.2015.03.003

E. C. B. (2018). *City of Windhoek electricity tariffs 2017 / 2018*. Author.

Gan, C. K. (2012). A review on micro-grid and demand side management and their related standards. *Power and Energy Conversion Symposium (PECS)*.

Hemapala, K. T. M. U., & Kulasekera, A. L. (2012). Demand Side Management for Microgrids through Smart Meters. doi:10.2316/p.2012.768-060

Khripko, D., Morioka, S. N., Evans, S., Hesselbach, J., & De Carvalho, M. M. (2017). Demand side management within industry : A case study for sustainable business models. *Procedia Manufacturing*, *8*, 270–277. doi:10.1016/j.promfg.2017.02.034

Koh, S. L., & Lim, Y. S. (2016). Evaluating the economic benefits of peak load shifting for building owners and grid operator. Proceedings - 2015 International Conference on Smart Grid and Clean Energy Technologies, ICSGCE 2015, 30–34. 10.1109/ICSGCE.2015.7454265

Li, C., Yu, X., Yu, W., Chen, G., & Wang, J. (2017). Efficient Computation for Sparse Load Shifting in Demand Side Management. *IEEE Transactions on Smart Grid*, *8*(1), 250–261. doi:10.1109/TSG.2016.2521377

Lin, C. C., Deng, D. J., Liu, W. Y., & Chen, L. (2017). Peak Load Shifting in the Internet of Energy With Energy Trading Among End-Users. *IEEE Access : Practical Innovations, Open Solutions*, *5*(c), 1967–1976. doi:10.1109/ACCESS.2017.2668143

Maps. (2019). Google maps is no longer free. Retrieved from https://www.shoredigitalinc.com/google-maps-is-no-longer-free/

Merrick, J., Ye, Y., & Entriken, R. (2017). Assessing the System Value of Optimal Load Shifting. *IEEE Transactions on Smart Grid*, ▪▪▪, 1–10.

Munshi, A. A., & Mohamed, Y. A. R. I. (2017). Big data framework for analytics in smart grids. *Electric Power Systems Research*, *151*, 369–380. doi:10.1016/j.epsr.2017.06.006

Ozadowicz, A. (2017). A new concept of active demand side management for energy efficient prosumer microgrids with smart building technologies. *Energies*, *10*(11), 1771. doi:10.3390/en10111771

Padhy, N. P. (2002). Demand Side Management and Distribution System Automation : A Case Study on Indian Utility. *National Power Systems Conference (NPSC)*, 308–311.

Prodanuks, T., & Blumberga, D. (2015). Methodology of demand side management Study course. experience of case studies. *Agronomy Research (Tartu)*, *13*(2), 520–525.

Rahman, A., Islam, R., Sharif, K. F., & Aziz, T. (2016). Developing Demand Side Management Program for Commercial Customers : A Case Study. 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 1–6. 10.1109/CEEICT.2016.7873061

Rosenow, J., & Bayer, E. (2016). *Costs and Benefits of Energy Efficiency Obligation Schemes. The Regulatory Assistance Project*. RAP.

Roshan, C., Kinley, W., & Sastry, M. K. S. (2014), Home Energy Audit: A Case Study of Phuentsholing, *Bhutan. IEEE International Conference on Communication Systems and Network Technologies (CSNT)*. DOI: 10.1109/CSNT.2014.208

Sastry, M. K. S. (2007). Integrated Outage Management System: An Effective Solution for Power Utilities to Address Customer Grievances. *International Journal of Electronic Customer Management*, *1*(1), 30–40.

Schuelke-Leech, B. A., Barry, B., Muratori, M., & Yurkovich, B. J. (2015). Big Data issues and opportunities for electric utilities. *Renewable & Sustainable Energy Reviews*, *52*, 937–947. doi:10.1016/j.rser.2015.07.128

Shipman, R., Gillott, M., & Naghiyev, E. (2013). The Mediterranean Green Energy Forum 2013, MGEF-13 SWITCH : Case studies in the demand side management of washing appliances. *Energy Procedia*, *42*, 153–162. doi:10.1016/j.egypro.2013.11.015

Shyam, R., Ganesh, H. B. B., Kumar, S. S., Poornachandran, P., & Soman, K. P. (2015). Apache Spark a Big Data Analytics Platform for Smart Grid. Procedia Technology, 21, 171–178. doi:10.1016/j.protcy.2015.10.085

Stötzer, M., Gronstedt, P., & Task, E. T. G. (2011). Demand side management potential a case study for Germany. *21st International Conference on Electricity Distribution*.

Tu, C., He, X., Shuai, Z., & Jiang, F. (2017). Big data issues in smart grid – A review. *Renewable & Sustainable Energy Reviews*, *79*, 1099–1107. doi:10.1016/j.rser.2017.05.134

You, A., Be, M. A. Y., & In, I. (2018). Assessment of utility side financial benefits of demand side management considering environmental impacts Assessment of Utility Side Financial Benefits of Demand Side Management Considering Environmental Impacts. doi:10.1063/1.5022897

Yu, N., Shah, S., Johnson, R., Sherick, R., Hong, M., & Loparo, K. (2015). Big data analytics in power distribution systems. *2015 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference*. doi:10.1109/ISGT.2015.7131868

Zhang, Y., Huang, T., & Bompard, E. F. (2018). Big data analytics in smart grids: A review. Energy Informatics, 1(1), 1–24. doi:10.1186/s42162-018-0007-5

Zhou, K., Fu, C., & Yang, S. (2016). Big data driven smart energy management: From big data to big insights. *Renewable & Sustainable Energy Reviews*, *56*, 215–225. doi:10.1016/j.rser.2015.11.050

Zobaa, A. F. (2018). *Big Data Analytics in Future Power Systems* (1st ed.). CRC Press; doi:10.1201/9781315105499.

Chapter 8
On Dynamically Reconfiguring IoT Architectures

Amira Hakim
University of Badji Mokhtar Annaba, Algeria

Abdelkrim Amirat
University of Souk Ahras, Algeria

Mourad Chabane Oussalah
University of Nantes, France

ABSTRACT

The world is moving towards a new generation of internet, the internet of things (IoT). This technological jump is improving the way we live by creating a bridge between the physical and the virtual world. Researchers are curious about the field from different perspectives especially managing of the complexity and dynamicity of these systems. The main problem being targeted in this chapter is that IoT systems are exposed to many structural and behavioral changes due to internal and external factors. This study is about the necessity of having a mechanism that enables IoT systems to perform without breaks or shutdowns regardless of context changes. The solution consists of a contextual dynamic reconfiguration process implemented by a reflexive multilayered architecture. This process is based on the autonomic computing loop. The authors also integrated evolution styles to make reusable the reconfigurations applied on the architecture of the system. Validation of the proposed approach was made on an e-health scenario, which was simulated using Cisco Packet Tracer before performing real development.

DOI: 10.4018/978-1-7998-2975-1.ch008

INTRODUCTION

When developing software systems, many aspects should be considered including satisfying user needs, respecting the deployment environment's properties, making the use of the software as simple as possible independently of the user's skills.

A key aspect to establish a good software system is the ability of managing its evolution at any stage of the life cycle.

Software evolution refers to the process of updating the system timely for various reasons: new requirements emerge when the software is used; objects gets in and out the system, or it may be related to improvements or just setups of the system.

The systems we are targeting by this study are connected object systems. Cyber-physical systems, Internet of Things are terms that refer to the current technological revolution. This term was first appeared in 1999 by Kevin Ashton to define the interconnection between the Internet and everyday objects. The new Internet emerges and is supposed to change the world: it is not only about connecting people, but also connecting various things or objects. It can be a person with a heart implant that transmits data, an animal carrying a smart chip, a car with tire pressure sensors or any other object, whether man-made or not, to which an IP address is assigned and which can transmit information.

Since the last two decades, we are witnessing a fast proliferation of the Internet of Things to reach and influence almost all sides of the daily life. This wide spread is leading to a noteworthy growth of systems complexity. Therefore, a big amount of data will be running all over this network of objects.

The problem addressed by this chapter is the need for a mechanism that allows IoT systems to keep performing without stops or breaks, regardless of the changes that affect their context.

BACKGROUND

IoT comes to trigger a revolution in the era of Internet and communication by connecting all surrounding entities together. The environment surrounding them often influences connected objects, they are induced frequently to undergo updates in response to context changing where some objects have to disappear and others will be detected. These changes cannot be foreseen at design time, being applied at runtime, will promote the system and the user with more efficiency and comfort of use.

Several research in ubiquitous computing focus on the analysis of newly developed applications properties including mobility, evolvability and context-awareness. Thus dealing with dynamic reconfirmation of ubiquitous systems (i.e. IoT based systems), our paper scope many aspects including context awareness, reflectiveness and mobility. Supporting real time dynamic adaptations by IoT systems is a property of a tremendous importance. Thus; it should be handled within the development process of such systems because of the multiplicity and variability of context they have to deal with. The concept of context has first appeared associated to ambient intelligence in1991 in Mark Weiser's work (Weiser, 1991). Dey defines context as "Any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" (Dey, 2001).

The spread of connected objects in the world of technology is ever increasing. Nevertheless, some questions remain to be settled in order to establish the Internet of Things as a good practice. (Pirotte, 2014) and (Jain 2015) have enumerated some challenges having an immense impact on the good use of IoT like addressing devices, the efficient management of energy resources, sharing and allocation of spaces to store objects in the cloud, the analysis of a very large mass of information coming from a huge number of sensors, heterogeneity and scale invariance, data security and confidentiality and dynamics and high availability.

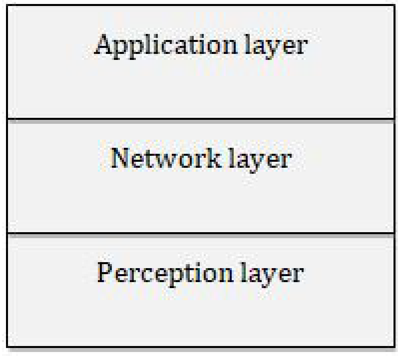
We address in this work the challenge that concerns the dynamic aspect and high availability of connected objects. Behind this challenge are engendered the occupations cited below.

The main problem dealt with in this chapter is the dynamic reconfiguration of connected object systems at the architectural level. The proposal of a solution on this subject gave rise to several sub-issues:

1. There is not yet a standardized architecture that supports all aspects of the Internet of Things.
2. A very large mass of data will be acquired, what mechanisms to filter this data?
3. How to model the contextual information acquired and how to store it so that it is easy to query when making decisions.
4. How to decide if context changes require reconfiguration and which strategy to invoke.
5. What mechanisms to use to reconfigure the system, and then how to reuse reconfiguration knowledge in a recurring situation.
6. How to ensure the coherence of the system after its evolution.

Starting by the first point of interest which is the standardized architecture, according to (Ray, 2018), IoT is able to interconnect billions of objects via the Internet. It is very difficult to manage entire objects on the Internet. So, there comes the need for a standardized architecture.

Figure 1. IoT's basic architecture



The basic architecture of IoT consists of three layers (shown in Figure 1). Analyzing this structure of the Internet of Things, (Wu & Lu et al., 2010) suggests that the three-layer architecture describes the loT structure from a technical perspective, which is reasonable at the initial stage of development. This structure cannot express all features and connotations of the Internet of Things.

According to (Khan & Khan et al., 2012), the IoT is built on five main layers allowing its operation, including the perception layer consisting of physical objects and sensors. This layer deals with the identification of objects and the collection of contextual data, the network layer having the role of transferring the information of the sensors to the information processing system by means of wired or wireless transmission media and technologies such as 3G, infrared, Wi-Fi, ZigBee, etc. The middleware layer deals with the management of the services implemented by the devices, the application layer manages the applications implemented by the IoT, such as smart health, intelligent agriculture, smart home, smart city, transportation intelligent, etc. and finally the business layer that is responsible for managing the IoT global system, including applications and services.

Guthikonda & Chitta et al. (2014) state that the previously proposed architectures for IoT are not sufficient for better use and deployment, especially for multi-industry system architectures. To remedy this, the authors propose the architecture illustrated in Figure 3.

Through the "performance management" module, this architecture supports reconfiguration according to the quality of service criterion by negotiating with the user. Nevertheless, there is no guarantee on the operation of the system after the reconfiguration.

Figure 2. Architecture of the internet of things (Khan & Khan et al., 2012)

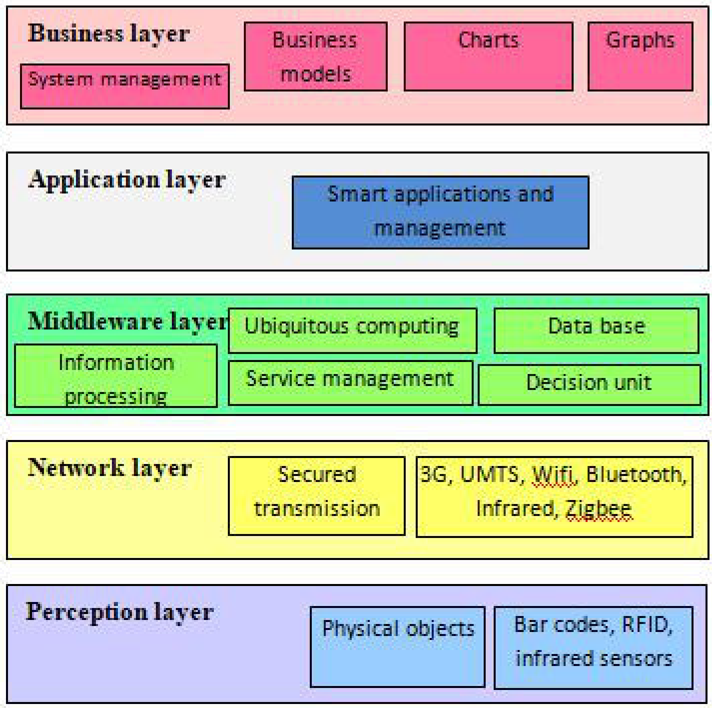


Figure 3. IoT architecture (Guthikonda & Chitta et al., 2014)

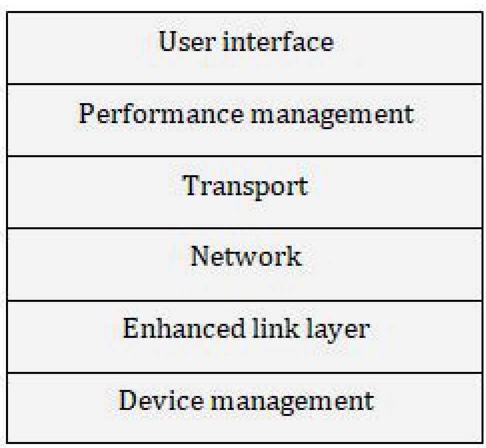
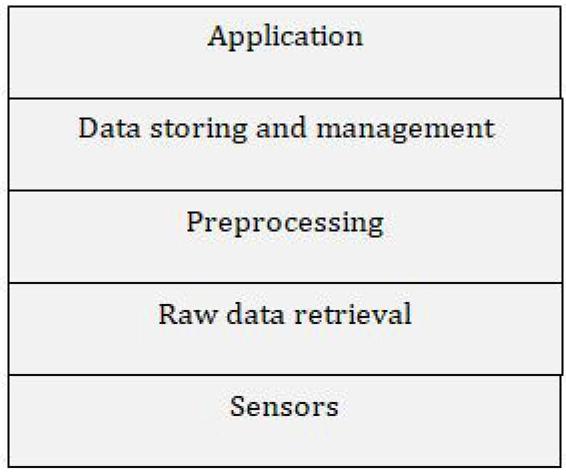


Figure 4. Layered conceptual framework of context sensitive systems



The second thing to handle is the data, how it is acquired, how to filter the big mass of the gathered information to get rid of the non prominent one, where to stock it for better use of the system. For this reason, we'll explore in this section the structure of a context aware system.

In this context (Baldauf & Dustdar et al., 2007) propose the multilayer architecture presented in Figure 4.

In what follows, we will explain the specifications and functionalities of each framework layer such proposed by (Baldauf & Dustdar et al., 2007).

Sensors

The first layer consists of a collection of different sensors. It should be noted that the word "sensor" refers not only to the detection equipment, but also to all sources of data that can provide usable contextual information.

Regarding how the data is captured; the sensors can be classified into three groups including: physical sensors, virtual sensors, logical sensors.

Raw Data Retrieval

The second layer is responsible for extracting raw context data. It uses appropriate drivers for physical sensors and APIs for virtual and logical sensors. Query functionality is often implemented in reusable software components that make low-level details of hardware access transparent by providing more abstract methods such as getPosition (). By using interfaces for components responsible for identical context types, these components become exchangeable. Therefore, it is possible to replace an RFID system with a GPS system without any major modification of the current and higher layers.

Pre-Processing of Contextual Data

It can provide useful information if the raw data is too coarse. The pretreatment layer is responsible for the reasoning and interpretation of the contextual information. Sensors surveyed in the underlying layer most often return technical data that application designers cannot use. Therefore, this layer raises the results of layer two to a higher level of abstraction. The transformations include extraction and quantization operations. For example, the exact GPS position of a person may not be useful for an application, but the name of the room where it is located may be.

Storage and Management of Contextual Data

The fourth layer organizes the collected data and offers them via a public interface to the client. Customers can have access in two different ways, synchronous and asynchronous. Synchronously, the client queries the server for changes made through remote method calls. Therefore, it sends a message requesting some kind of data proposed and pauses until it receives the response from the server. Asynchronous mode works through subscriptions. Each customer subscribes to specific events that interest him. During one of these events, the customer is simply warned or a method of the customer is directly involved by a reminder. In most cases, the asynchronous approach is more appropriate because of rapid changes in the underlying context.

Application

The client is implemented in this fifth layer. The actual reaction on different events and instances of context is implemented here. Sometimes information retrieval and application-specific context and reasoning management are encapsulated as agents, which communicate with the context server and act as an additional layer between preprocessing and the application layer. An example of a client-side logical context is the display on mobile devices: Since a light sensor detects bad lighting, the text can be displayed with a higher color contrast.

SOFTWARE EVOLUTION

To effectively exploit contextual data in a situation, the context lifecycle shows how that data moves from one phase to another in the system. (Bernardos, Tarrio et al. 2008) identified three phases in a typical context management system: context acquisition, information processing, reasoning and decision.

Talking about context management system drives us to ask about software evolution. Dey et al. (1998) introduced the notion of adaptation by defining context awareness as work leading to the automation of software based on the user's context knowledge.

We speak of evolution to designate a maintenance operation, adaptation, reconfiguration or versioning of a system. Adaptations are the result of evolutions. A system is able to evolve by itself and automatically thanks to auto-reconfigurations.

The IEEE software maintenance standard defines evolution as "The modification of a software after its delivery, in order to correct failures, improve its performance or other attributes or adapt it following environment changes" (1219-1993 1993).

(Sicilia & Cuadrado et al., 2005) define evolution by: "The totality of the activities that are required to provide support, at the best possible cost, of software. Some activities start before the software is delivered, so during the initial design, but the majority of activities take place after final delivery (when the development team considers that it has finished its work)."

The definition given by (FIPS, 1984) is the following: "All the activities required to keep the software in a state of operation following its operational delivery".

IOT'S DYNAMIC CONTEXTUAL RECONFIGURATION

The main goal of this study is to propose an efficient approach to perform reconfiguration actions in a transparent and dynamic way on IoT systems. First of all, we must work on the subject by trying to answer the four dimensions that define the evolution in these systems and that will lead to build the corresponding process. The features that we think are important for addressing the evolution of software architectures are introduced in the form of questions: who, what, when and how.

The "What" corresponds to the object of evolution. This dimension specifies the architectural elements which are concerned by the evolution. In this study, we are interested in IoT systems, so the evolution can relate to objects, connections between them or both.

In this research, we are interested in complex software systems based on connected objects. These systems contain a set of entities interconnected by network connections for the purpose of communicating, sharing data and resources for better Internet exploration. These entities may be real world people, places or objects (Abowd & Atkeson et al., 1997). The connections between these entities differ according to the speakers (human-machine connection, man-to-man connection, machine-to-machine connection (Tan & Wang, 2010).

From a structural and functional point of view, each connected object contains a unique identifier, a name, a location and other attributes describing its properties (Atzori & Iera et al., 2010). It contains several sensors according to its type and its characteristics. These sensors serve as contextual data acquirers using common features, including acquisition, interpretation, and response to collected data (Kortuem & Kawsar et al., 2010). Smart objects are connected to each other by network and technology connections.

The "Who", this dimension corresponds to who intervenes in these objects. In other words, who is responsible for triggering an evolution or why evolve a system. In our case, evolution is the result of changes in the context of the system. The nature of the studied systems means that the evolutions are unforeseen and must therefore be carried out dynamically in a non-intrusive way.

The raw context information will be translated and evaluated to decide whether to trigger or ignore a reconfiguration operation. Therefore, the triggering of a reconfiguration operation is mainly related to changes in the system context, this is called context sensitivity (Schilit & Adams et al., 1994).

The "How" corresponds to the reconfiguration techniques used to keep the system in a stable situation or how does a context aware system work.

Primarily, any reconfiguration operation goes through three phases (Kakousis & Paspallis et al., 2010), namely:

- **Context Variation Detection / Notification**: This requires a mechanism to detect and report relevant changes in the environment, such as decreasing network bandwidth, direct action of a human user on an interface man-machine.
- **Decision-making**: After notifying a variation of the context, the adapter must decide based on an adaptation policy what actions to take.
- **Execution**: The adapter executes the decided modification actions. They must be applied to the application concerned by the variations (on its components and / or structure) or on the underlying infrastructure and / or on simultaneous applications.

With regard to the programming techniques and concepts used for adaptation, we have according to (Chefrour, 2005), (Ketfi & Belkhatir, 2004), (Aissaoui & Atil et al., 2013): the reflection technique, aspect oriented programming, ADLs based programming, component based programming .

The fourth dimension is the "When", it corresponds to the time of the intervention. An evolution can be triggered during the development phase, deployment or during the execution of the system (Ketfi & Belkhatir, 2004; Chefrour, 2005).

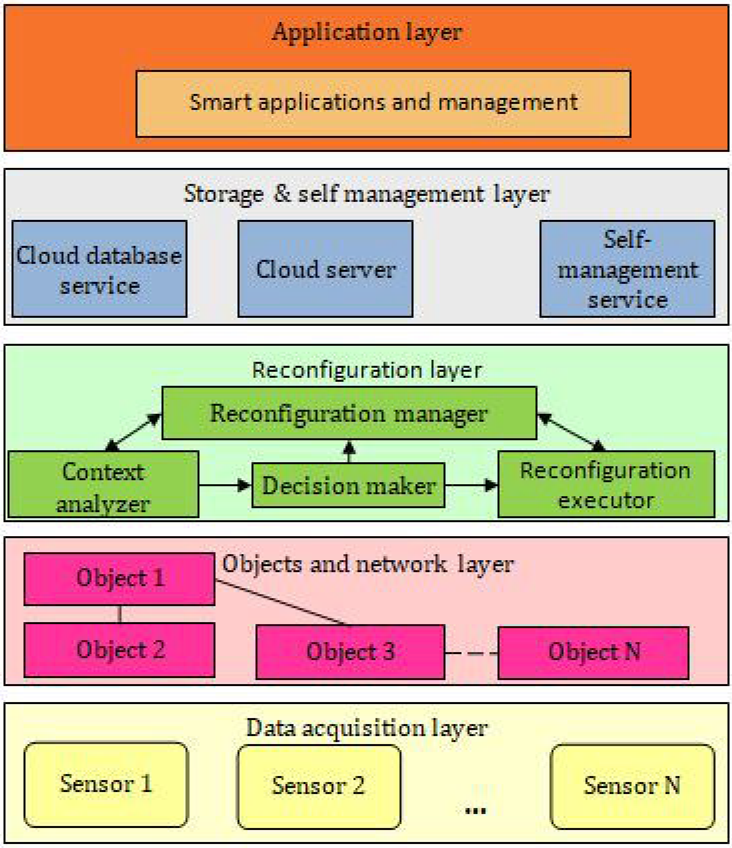
- **Design Time (Static):** The software change is for the source code of the system. Therefore, the software must be recompiled for the changes to become available. The simplest but most expensive technique is rewriting all or part of the code to accommodate the application to the particular conditions of its environment.
- **Loading Time**: The software change occurs when the software items are loaded into an executable system. This is particularly the case when it comes to porting an application to a different platform than originally planned.
- **Execution Time (Dynamic)**: The software change takes place during its execution. In order to ensure the proper functioning of the applications despite the variations of their environments, they must be adapted while they are running. This dynamic adaptation can be difficult to implement because, in addition to writing the application's business code, developers must write the code that observes the variations of the resources and makes the necessary modifications while preserving the integrity of the application.

PROPOSED ARCHITECTURE

In order to adapt the IoT system to contextual changes, we propose a generic architecture model. The multi-layered architecture (Figure 5) is inspired by the one presented by Khan & Khan et al. (2012) in terms of the elements and technologies essential to IoT. This reflective architecture model (proposed in (Hakim & Amirat et al., 2018) implements a non-intrusive dynamic contextual reconfiguration process for connected object systems. It is composed of four main layers including:

Data Acquisition Layer: It contains many sensors that serve as acquirers of dynamic raw context data. The first layer represents the context of execution of the connected objects, this context can be categorized according to Ferry and Lavirotte et al. (2008) into four classes related to the user of the system i.e., it concerns all the information that can be related to a user, his location, his mood, the task he is performing. It can also be related to the temporal aspect which deals with everything related to moments, dates, and history. It also concerns the properties and variations of the machine which is interested in the execution environment, the resources available for the devices of the system. The environmental context is related to everything surrounding the system, brightness, noise, temperature etc.

Figure 5. IoT's reconfiguration architecture



This categorization will allow us to define the main families of conditions that can trigger an adaptation.

The contextual information is represented in the first layer in a raw way, it will then be disseminated to the upper layers for analysis and to take into account the most relevant and appropriate information.

Objects and Network Layer: It represents the functional aspect of the system. The latter is composed of a set of intelligent objects linked together. These objects are represented simply by a class diagram where each object is defined by the values of these attributes which are the identity, the name, the location and all the sensors it contains. Objects have common functionalities (Kortuem & Kawsar et al., 2010) that characterize their intelligences including acquisition, interpretation of contextual data, and reaction to a given policy. Nevertheless, each object has other features and activities that depend on its nature and role. The objects are connected to each other by links and network technologies mostly wireless such as wifi, Bluetooth, NFC (Near Field Communication), ZigBee, 6LowPan and Z-Wave etc.

Reconfiguration Layer: it manages the reconfiguration operations. It contains four main tools: reconfiguration manager, context data analyzer, decision maker, and reconfiguration executor.

A huge amount of contextual data is acquired through the sensors or probes, this data is sent to the engine or manager of the contextual data to analyze and filter them and share the most relevant data to the decision engine which, in turn analyzes these data and decides according to a certain policy to trigger or ignore a reconfiguration operation. If the decision is positive and the decision engine deems the context changes important, the decision is sent to the reconfigurator who will execute the operation and act on the connected objects layer either by adding, deleting or modifying the implementation of an object or a connector.

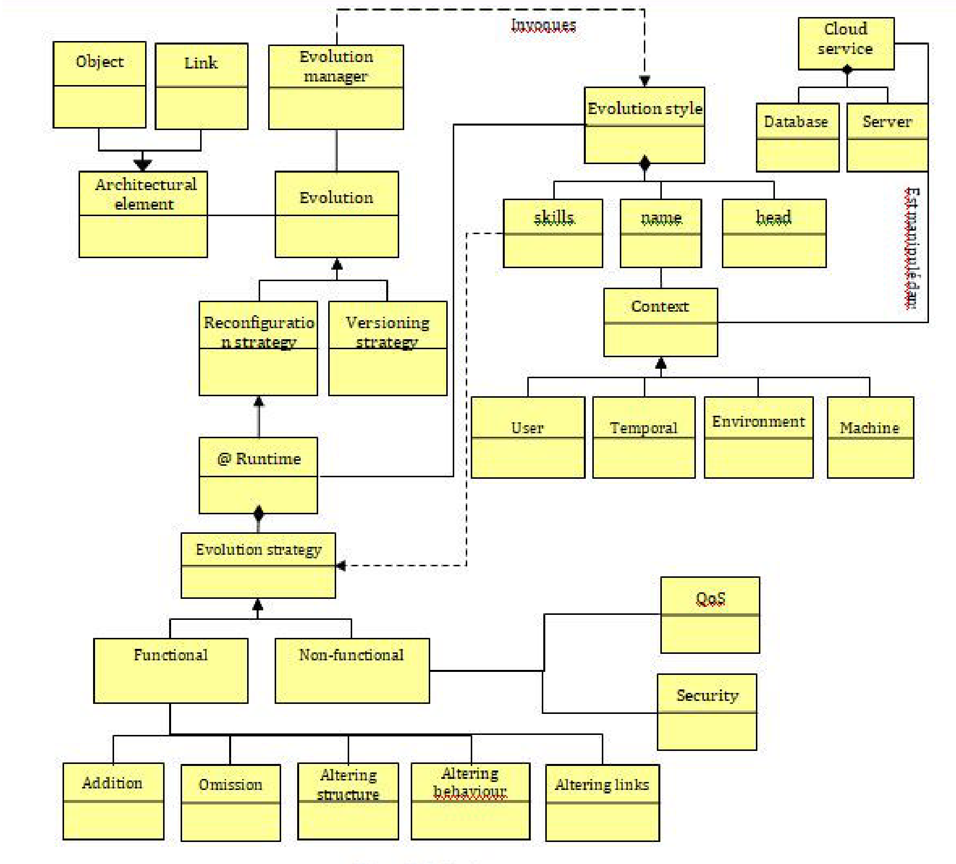
Storage and Self Management Layer: It is responsible for storing and processing context data passed by the Context Data Analyzer to the Cloud. It also contains the knowledge base on the static context. This layer thus provides a self-management service (reflexivity) whose role is to maintain the consistency and efficiency of the entire system after the reconfiguration.

Application Layer: This layer provides global management of the application based on the objects information and types. The applications implemented by IoT can be smart health, smart farming, smart home, smart city, intelligent transportation, etc.

We consider that the network layer of Khan's architecture (Khan & Khan et al., 2012) is integrated into the application layer. We aim to improve the aforementioned architecture by adding:

- Two cloud services, one delegated to store contextual information and the other for processing.
- Tools running a dynamic contextual reconfiguration process (reconfiguration manager, context data analyzer, decision maker, reconfiguration executor).
- A service to ensure the reflexivity of the system and its self-management (the self-management service in the upper layer).
- Mechanisms to check the non-functional properties of the system after its reconfiguration via the automatic management service.
- Use of evolution styles to make architectural evolutions reusable to exploit the skills of an architect.

Figure 6. Evolution manager structure



RECONFIGURATION PROCESS

The operating mechanism of the proposed approach consists of a reconfiguration process applied to an architectural element of the system, regardless of its granularity (object, set of objects or the whole system).

By default (definition of an object in the object-oriented vision), each element of architecture is defined by its structure (fields and attributes) and its behavior (methods and procedures).

In this research, we are interested in IoT systems, which are inherently scalable due to frequent changes in their context. We propose to associate to each architectural element an “Evolution” part. This part will be responsible for the management of system transitions while performing evolutions, in particular reconfigurations.

IoT's Evolution Manager Structure

The evolution of connected object systems is represented as a hierarchical process, where the first layer is represented by the architecture proposed in Figure 5. The second layer is the operating mechanism of the process which is the contextual dynamic reconfiguration. The third layer guarantees the reflexivity of the system and its autonomy. The key elements of this process are identified as well as the relationships between them in the meta-model presented in Figure 6.

Figure 6 Highlights the internal structure of the evolution manager. Each architectural element is associated with one or more evolutions. An evolution can be a reconfiguration strategy, a versioning strategy or any other type of evolution. In this work, we will focus on the dynamic reconfiguration strategy. By dynamic, we mean that it can be completed at runtime without requiring system breaks or shutdowns. The reconfiguration actions supported by our process include: adding a new element, deleting an existing element, updating the structure / behavior of objects, updating links, revision of the non-functional properties of the system.

The novelty of our work relies in the use of evolution styles. The concept, introduced for the first time by Le Goaer et al. (2007) and Le Goaer (2009), by analogy with architectural styles, aims to make architectural evolutions reusable to exploit the skills of an architect . An evolution style consists of a name, a header, and a set of skills. The header has an informal description of the objective and publishes a list of parameters and assertions. A context bucket is described in the header of an evolution style to express the context of the evolutionary element. The heading declares the descriptive elements of the evolution, including its pre and post evolution conditions. The evolution style skills indicate how to proceed with evolution.

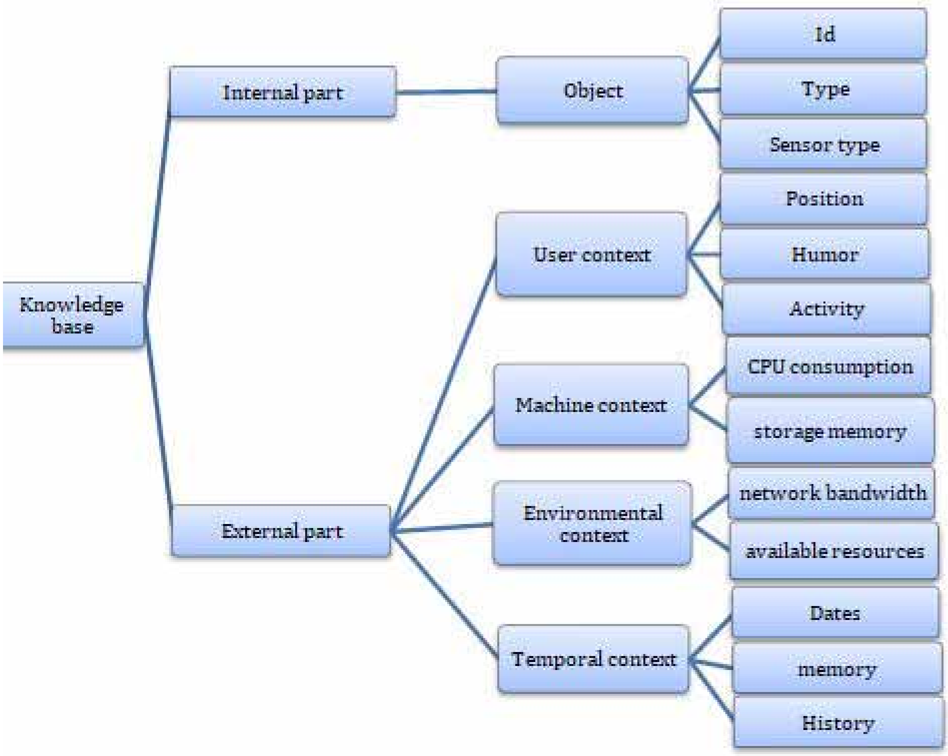
We chose to manage context variations by applying the MAPE/K loop that monitors, analyzes, plans, and executes the reconfiguration appropriately. In this section we discuss how the process works step by step.

1. Contextual Data Acquisition

An IoT system is a context-sensitive system, which means that it is strongly influenced by changes and variations in the context. Each connected object contains one or more sensors enabling it to acquire context data and become aware of the environment to which it belongs.

Contextual information can come from a wide variety of sources. Although much of the previous research in the context sensitivity field focuses only on the detected information, other context models have been found, they are richer: they integrate the detected, static, and derived information provided by the user.

Figure 7. Representation of contextual data using the XML schema



According to Ferry et al. (2008) the context can be classified into four categories to define the conditions that can trigger an adaptation: user context, machine context, temporal context and environmental context. While Bettini et al (2010) found that rich context models incorporating detected, static, user-provided and derived information were most useful.

Other sources of contextual information may be related to the characteristics of the system or, in other words, to knowledge of its own structure, architecture, sudden addition or removal of existing node / entity.

Referring to the context sensitive system structure, the raw data detected must be processed and filtered. Here, a standard data format is required to unify the large amount of data from variable sources.

To facilitate querying contextual information and the identification of the current situation, the XML format is adapted (Figure 7). The contextual information of the IoT systems is divided into an internal part that includes the objects themselves, each object being characterized by a unique identifier, a type and a set of sensors.

And an external part describing the context information relating to the user, the time, the environment and the machine.

Decision Making

Upon receipt of a notification indicating a change of context of the system, resulting from the update of the knowledge base (XML file), the decision maker analyzes this data in the following algorithm to identify the situation in which the manager of the reconfiguration would trigger or ignore the reconfiguration action.

The decision maker constantly listens to the reconfiguration manager. When the knowledge base file changes, the reconfiguration manager sends a notification to the decision maker. The first step to be performed by the latter is to query the functional part of the system by comparing the current number of objects to the previous versions (which are stored and sorted by date in the cloud database service). There are three possibilities:

- **Case 1**: If the number of objects increases, the reconfiguration executor invokes the addition policy. Here, a new object will be added to the connected objects. The objects are distinguished by an identification number, the links between the old and the new object are to be regulated.
- **Case 2**: If the number of objects decreases, there are two possibilities:
  1. An object is no longer part of the system: in this case, the object omission policy is executed.
  2. The network bandwidth is decreasing or a non-functional object has triggered an anomaly.
- **Case 3**: If the number of objects does not change, there are two possibilities:
  1. Some existing objects modify their structures or behaviors,
  2. The non-functional concerns of the system have been affected.

Decision Making Algorithm

Begin

```
While true do {
Listen_to_the_reconfiguration_manager()
Receive_notification ()
Int nb:= old_number_of_objects
Int count=0
foreach (xml_node n in knowledge_base.functional_part) {
count += 1
if (count < nb) then
```

9. functional_reconf. add_object ()
10. **End if**
11. **else**
    if (count > nb) **then**
12. **if** network_bandwidth. IsOk () **continue**;
13. **else** functional_reconf.remove_object();
14. **End if**
15. **else if** (count==nb) **then**
16. functional_reconf. check_objectsStructure ()
17. **End if**
18. non_functional_reconf. CheckAllparameters ()
    End.

2. Reconfiguration Execution

The reconfiguration process will be triggered in response to context changes. When notified by a reconfiguration event, from the context analyzer, the planner selects a strategy to meet the new requirements using the shared knowledge base. Upon receipt of the reconfiguration plan, the executor adapts the system to its current context. The reconfiguration actions supported by our process include:

- **Adding New Objects**: Create new objects and assign them a unique identifier, connecting sensors to objects based on their functionality.
- **Deleting Existing Objects**: This action will delete its links with other objects.
- Altering the structure of an object: The modification of the structure of an object mainly consists of modifying its visibility, adding or removing attributes. To do this, temporarily delete the object, replace it with a backup object, modify the object concerned, and replace the backup object with the modified object.
- **Altering the Behavior of an Object**: The behavior of an object is described by Kortuem & Kawsar et al. (2010) as "They sense, log, and interpret what's happening within themselves and the world, act on their own, intercommunicate with each other, and exchange information with people". Based on this definition, behavior change can affect how the object captures contextual information or how it shares the information it captures. It can also affect the interpretation of the events it receives. Support for this action is the same as editing the structure.
- **Editing Links**: Smart objects are connected by network and by technological connections such as Wi-Fi, Bluetooth, NFC, RFID, ZigBee, Z-wave, 6lowPan, etc. The changes in this phase are: Change the type of connection used to

connect system objects. The change may also affect the signal quality or the bandwidth of the links between the connected objects.

- **Maintaining Non-functional Properties of the System (Quality of Service, Security, Flexibility, etc.)**: Once the reconfiguration process is completed, a check of the system consistency in terms of non-functional properties is performed. This last obligatory step guarantees the performances of the system and preserves the quality of service and, finally, the good functioning of the connected objects.

Each reconfiguration action is a strategy. For example, the maintenance of an object requires replacing it with a backup object in order to guarantee the transparency of the system for its users during its maintenance operations.

PROCESS EVALUATION

We proposed a dynamic contextual reconfiguration process. The functioning of the process and its principles has been detailed in the previous section. We have yet to propose a possible realization and to conduct an experiment around our proposal.

We have opted for a particular case to study the behavior of the proposed process with regards to a concrete scenario in the field of electronic health.

We will evaluate our approach to a mobile system in the field of electronic health. The system is responsible for monitoring the glycemic level of a diabetic patient and instructing him / her to inject insulin on time.

CASE STUDY

A 60-year-old type 1 diabetes patient who lives alone needs a system to take care of his life, regardless of his location and state of health, that is, when he is at home or outside.

ComfortCareIoT is a system we offer to manage the life of the aforementioned diabetic patient by making the home smarter, taking care of him by reminding him of his insulin injections and preventing damage that may occur at home while he is sleeping. It should be noted that diabetics are characterized by a deep sleep).

It has happened that the system sends a notification to the patient to remind him of his insulin injection at 9 o'clock in the evening. The notification has been sent to his Smartphone and the system is waiting for confirmation of receipt.

Five minutes later, the system unusually did not receive confirmation from the patient. Thus, a call has been transferred to his Smartphone. In responding, he confirmed that he had not received any notification. In this case, the system manager reminds the patient to take his injection and triggers a reconfiguration action.

The reconfiguration actions consist of temporarily replacing the defective Smartphone with the smart watch, asking the patient to bring the phone to the maintenance service of the online health center.

The smart house we are proposing belongs to a sick elderly person who lives alone. Operating the system in a context-sensitive style ensures that user intervention is not required. Users with limited capabilities can easily use the system. In the scenario we propose, the patient may be asleep or unconscious, but his life is always monitored automatically.

MODELING AND REALIZATION

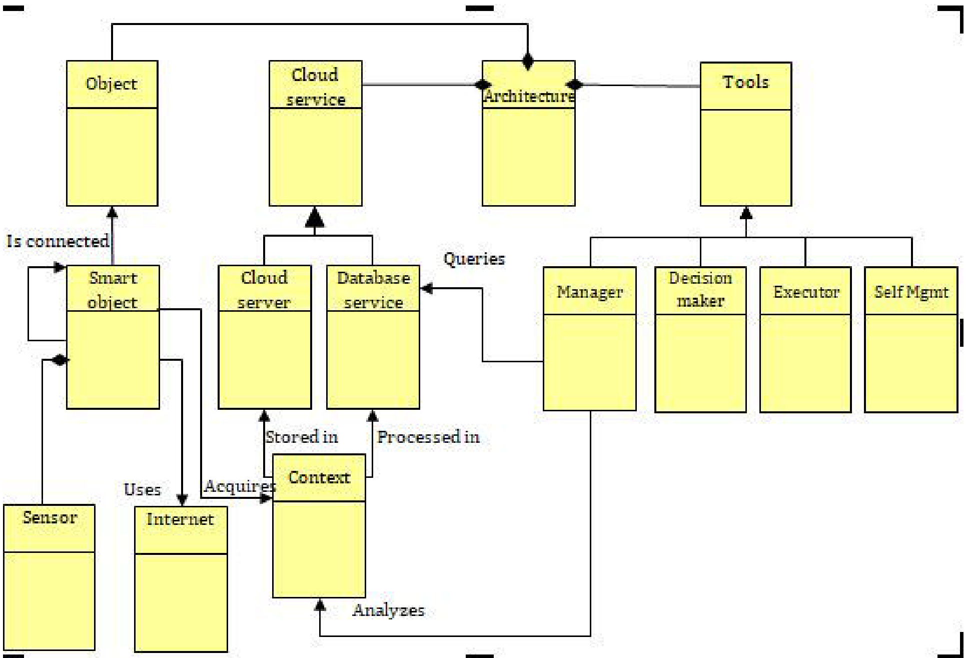
To implement the dynamic reconfiguration process on connected object systems, we have proposed a generic architecture model (Figure 5).

Using EMF, we created two types of models; on the one hand a meta model (Figure 8) defining the concepts of the architecture discussed previously and on the other side a model instantiating its concepts. The latter is described using a palette generated using Eugenia's Emfatic tool under Eclipse.

Our ambition behind the proposed approach was to build a generic and extensible architecture model. For this, several properties are targeted by our model:

- **Genericity**: The model we propose is designed in a way that allows it to adapt to the area of application in which the system will be deployed. This feature provided by the use of meta-modeling does not limit our approach to a simple language or modeling notation.
- **Reflexivity**: This property makes the system take care of its own adaptation. The detection of the relevant change requiring adaptation, we aim to equip the system with the ability to auto configuration, self optimization and self protection. This feature will reduce the costs of external interventions in terms of time and effort. Reflexivity is ensured by the self-management module implemented in the fourth layer of the proposed architecture.
- **Scalability**: One of the characteristics of the Internet of Things is the large number of connected objects. The nature of these objects implies that it must be flexible to add new objects and new services to existing objects. Due to dynamic reconfiguration, adding running objects does not influence system availability.

Figure 8. Meta model of the proposed architecture

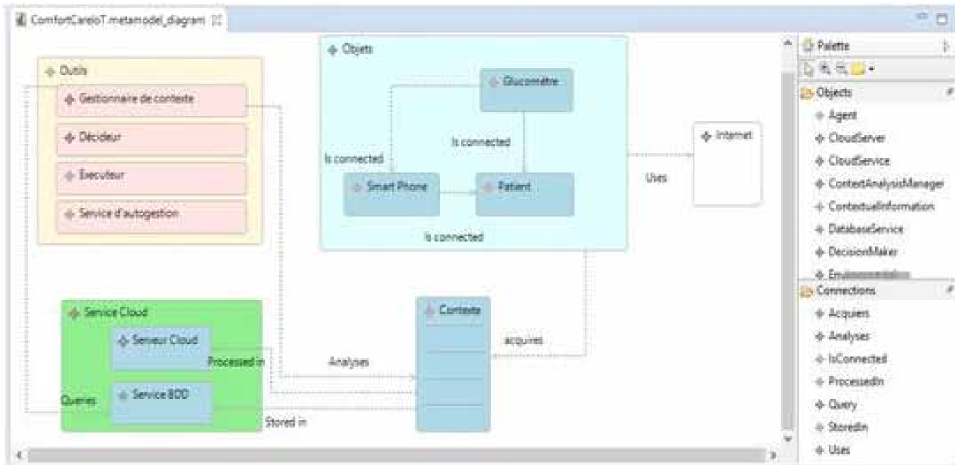


- **Reuse**: The high level of abstraction of the proposed architecture, as well as the use of evolution styles, allow the reuse of know-how concerning possible evolution strategies on the system, for example the addition, deleting and modifying connected objects.
- **Context Awareness**: The proposed model allows the system to adapt to changes in its environment, which is made possible by the context-sensitive quality offered by IoT systems. Contextual information is very important for improving application performance.

Cloud computing for storing context information: Connected objects have limited resources in terms of calculations and storage: the battery life is limited, the objects consume energy which reduces the performance of the application. We propose that storage and data processing be at the cloud level.

By instantiating this model, we obtain a language specific to the domain of the Internet of Things in the field of electronic health in the form of an Eclipse plugin (Figure 8) and a palette containing the instances of the elements of the previous meta-model (elements conform to this meta-model).

Figure 9. Eclipse plugin of the ComfortCareIoT model



Regarding the behavior or the functioning of the process described by the concrete example of the failure arrived at the cell phone of the patient, the maintenance strategy is highlighted in the following sequence diagram. This type of reconfiguration triggers several other actions (addition, deletion) which constitute an effective scenario.

In response to the failure of the Smart Phone object, the reconfiguration manager invokes the update policy. It consists in temporarily removing the defective object during its repair and replacing it with a backup object in order to preserve the stability of the system and its transparency for the end user.

To capture the knowledge about the maintenance strategy of an object, we recommend the use of an evolution style corresponding to the reconfiguration strategy "Updating the behavior of an object". The style is as explained previously consists of a name, a header and a set of skills. We illustrate in Figure 10 the evolutionary style corresponding to the scenario of the proposed case study.

SIMULATION

A validation of the proposed scenario is performed in a simulated environment before real development. A smart home is implemented (shown in Figure 11) using Cisco Packet Tracer simulation software, which includes various smart objects used for home automation, such as a smart fan, smart window, smart door, an intelligent light, a fire extinguisher, a turf extinguisher, etc.

Figure 10. Evolution style "maintenance of an object"

Behavioral update of the Smartphone object
Context: User, Environmental, Temporal, Machine Pre: The smart phone no longer receives notifications Post: The smart phone is reconnected to the system
• Create a backup object to replace the smart phone • Delete smart phone object • Replace the smart phone with the smart watch (the backup object) • Repair the defective object • Delete the backup object • Reconnect the smart phone object to the system • Check the non-functional properties of the system

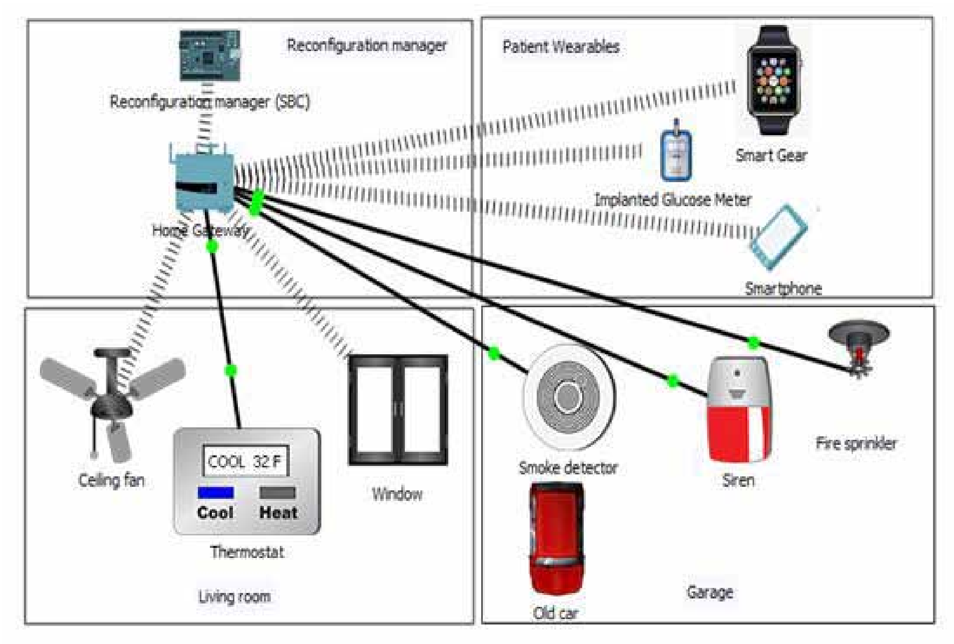
Cisco Packet Tracer is a very powerful tool that emulates a network topology. It does not require expensive hardware or wiring hours. Packet Tracer simulates the operation of a network by exchanging and viewing Ethernet frames. Two simulation modes are proposed by the tool created:

- **Real-time Simulation**: Allows to immediately visualize all the sequences that take place in real time.
- **Detailed Simulation**: Allows you to view sequences in slow motion between two or more devices.

DEVICES USED IN THE SMART HOME

A variety of peripherals and sensors are installed inside the smart home, using wireless connections and a DHCP mode to assign IP addresses through the home gateway. These devices are: a home gateway used to record smart objects and assign IP addresses, a ceiling fan to ventilate the home environment according to certain conditions, a window, a thermostat, an old car, an automatic fire extinguisher that rejects water detected by a smoke detector, a siren emitting a sound in case of

Figure 11. Smart home simulation using Cisco Packet Tracer 7.1



danger and a smart phone belonging to the person allowing him to control the smart objects that are recorded there. The patient also has a smart gear and an implanted glucose meter.

The proposed scenario invokes many reconfiguration actions, including adding objects, changing the structure and behavior of objects. Here are some examples of actions mentioned:

- The siren and fire extinguisher are activated once smoke is detected.
- The window opens or closes depending on the temperature detected by the thermostat.
- The ceiling fan lights up if the temperature exceeds 15° C.
- The smart phone needs to be replaced by its smart equipment.

The reconfiguration manager is installed inside the single-board computer object (simulates a Rasperry Pi). To implement the maintenance strategy, we worked on the realization of the aforementioned scenario, that is to say that the reconfiguration manager had to send a notification to the smart phone of the patient as a scheduled task written in Java.

However, if the smart phone is defective, the smart phone will be replaced by the backup object (the smart watch) for the system to work.

Table 1. Devices used in the smart home

Device	Utility
Home gateway	Used to record smart objects and assign them IP addresses
Seiling fan	Used to ventilate the environment of the house according to certain conditions
Window	Used to control the window remotely
Thermostat	Used to detect the temperature of the house
Old car	Used to simulate different scenarios in the design of a house, as far as it affects, the level of CO, CO2 and smoke
Fire sprinkler	Rejects water when the effects of a fire have been detected
Smoke detector	Used to detect the level of smoke
Siren	Provide sound for some events at home
Smartphone	To control registered smart objects and provide different server features
Smart gear	Used to receive notifications instead of Smartphone
Implanted glucose meter	Used to monitor the patient's glycemic level
Single Board Computer SBC	Simulates a Raspberry Pi, the reconfiguration manager must be installed in the SBC

Figure 12. Java method to notify the patient

```
public static void Send_notification () {
        Calendar calendar = Calendar.getInstance() ;
        calendar.set (Calendar.HOUR_OF_DAY, 21);
        calendar. set (Calendar.MINUTE, 00);
        calendar.set (Calendar.SECOND, 00);
        Date time = calendar.getTime() ;
        Timer timer = new Timer();
        timer.schedule(new MaTask(), time);
}
```

Instead of writing programs, another solution is to use the Google Assistant to ensure good communication between the system and the patient through keyboard input or a natural voice. The virtual assistant launched by Google can search the Internet, schedule events and alarms, adjust the user's device hardware settings, and view information from their Google Account. The Google Assistant extends to just

Figure 13. Invocation of the "send_notification" method

```
public static void main (String args [] ) {

          if (!SmartPhone.Send_notification()){
          System.out.println("It seems that an error occurred on your mobile,
please contact the maintenance department");
          SmartGear.Send_notification();
          }
    }
```

about every modern Android device. It is also available on Wear OS, Android TV and Nvidia Shield devices as well as Android Auto.

The reconfiguration manager must be associated with Google Assistant, which is distributed in the users' smart devices, according to the IoT features.

Regarding the interface, the Google wizard will be used as it is. The user is free to use it via natural voice or keyboard input.

Using Google Assistant in conjunction with the Reconfiguration Manager will enhance system interactivity by facilitating communication between the user and the system. This will also make the process of collecting user data relative to your Google Account effective.

CONCLUSION

The emergence of the Internet of Things in different areas of daily life improves the way humans interact with their environment. The ultimate goal is to create a better world for human beings, where the objects around us know what we love, what we want and what we need and act without explicit instructions.

We presented an approach to dynamic reconfiguration of connected object-based software systems. The main goal behind this approach is to allow an IoT system to continue to function as it evolves.

The approach we have proposed is to propose a generic architecture model for connected object systems. Then, the proposal of a process that will be implemented by this model to allow dynamic contextual reconfiguration.

The model is described using a four layer reflective architecture, including a contextual layer, an object and network layer, a dynamic reconfiguration layer, and an automatic control layer.

The reconfiguration process implemented by the proposed architecture is inspired by the autonomous computing loop; It consists of four stages: context acquisition, reasoning, planning and execution. After performing a reconfiguration, it is important to check the non-functional properties of the system in question to ensure that it always retains the same level of efficiency and accuracy.

To validate the proposed approach, a case study in the field of electronic health has been proposed. The corresponding model was realized using the Eclipse modeling framework (EMF) and its instantiation through Eugenia's Emfatic tool under Eclipse. A simulation is performed using the Cisco Packet Tracer simulator associated with the Java language to evaluate the system before proceeding with the actual development. The Google Assistant is used by the user to integrate and manage the proposed house.

FUTURE PERSPECTIVES

To conclude, we must admit that the results of this study must be considered in the light of certain limitations on the modeling plan as well as on the implementation plan. To the best of our knowledge, the model we have proposed lacks mechanisms to ensure the delivery of results in real time. We must also use the safety aspect which is a very important point in these systems; information sharing requires a certain level of security, end-users must benefit from secure and confidential privacy.

This work allowed the proposition of a model for the evolution of software architectures of connected object systems. It consists of an architecture implementing a contextual dynamic reconfiguration process of IoT systems. Nevertheless, this study can be extended to ensure other aspects of connected object systems. In what follows, we cite some open avenues of exploration to enrich our approach.

Real-time Reconfiguration: We aim to limit the reaction of the system by a very short delay of an order of milliseconds. Nevertheless, the system must not simply deliver results within set deadlines; it must deliver them with a high level of accuracy.

Better Cloud Exploitation: It's clear that using the cloud comes with many benefits in terms of resource consumption of objects such as battery, storage space and device performance. However, sending contextual data to the cloud for processing and storage is expensive in terms of time. The implementation of the real-time techniques will then be required in this case to respect the time and the temporal aspect.

Development of Other Evolution Styles: The know-how that we capture at each reconfiguration operation is saved (by analogy with the concept of Components Off the Shelf (COTS) to be able to reuse them in similar situations, we aim to enrich the style library to support all context change situations that can affect one or more connected objects.

Maintenance of Non-functional Properties: In our proposal, we insisted on checking the functional properties of the application after its evolution. However, we focused only on the consistency of the application and more precisely on the quality of the services offered by each object. We aim to give more importance to the security and confidentiality aspect. On the one hand, the data transmitted to the Cloud must be secured to prevent the loss of frames of data. On the other hand, the privacy of users must be respected; we must manage the visibility of the properties of objects in the network.

REFERENCES

Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, *3*(5), 421–433. doi:10.1023/A:1019194325861

Aissaoui, O., Atil, F., & Amirat, A. (2013). Towards a Generic Reconfigurable Framework for Self-adaptation of Distributed Component-Based Application. In Modeling Approaches and Algorithms for Advanced Computer Applications. Springer. doi:10.1007/978-3-319-00560-7_43

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, *54*(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010

Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, *2*(4), 263–277. doi:10.1504/IJAHUC.2007.014070

Bernardos, A. M., Tarrio, P., & Casar, J. R. (2008). A data fusion framework for context-aware mobile services. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*. IEEE. 10.1109/MFI.2008.4648011

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, *6*(2), 161–180. doi:10.1016/j.pmcj.2009.06.002

Chefrour, D. (2005). *Plate-forme de composants logiciels pour la coordination des adaptations multiples en environnement dynamique*. Université Rennes 1.

Dey, A. K. (1998). Context-aware computing: The CyberDesk project. *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, *5*(1), 4–7. doi:10.1007007790170019

Ferry, N., Lavirotte, S., Rey, G., & Tigli, J.-Y. (2008). Adaptation Dynamique d'Applications au Contexte en Informatique Ambiante. Rapport Technique, Laboratoire I3S (Universite de Nice Sophia Antipolis/CNRS), numero I3S/RR-2008-20-FR, Sophia Antipolis, France.

Guthikonda, Chitta, Tekawade, & Attavar. (2014). *Comparative analysis of IoT architectures*. TLEN 5710 Capstone, Tech. Rep.

Hakim, A., Amirat, A., & Oussalah, M. C. (2018). Non-intrusive contextual dynamic reconfiguration of ambient intelligent IoT systems. *Journal of Ambient Intelligence and Humanized Computing*. doi:10.100712652-018-1127-2

Jain, R. (2015). Internet of Things: Challenges and Issues. In *20th Annual Conference on Advanced Computing and Communications*. IEEE.

Kakousis, K., Paspallis, N., & Papadopoulos, G. A. (2010). A survey of software adaptation in mobile and ubiquitous computing. *Enterprise Information Systems*, *4*(4), 355–389. doi:10.1080/17517575.2010.509814

Ketfi, A., & Belkhatir, N. (2004). A metamodel-based approach for the dynamic reconfiguration of component-based software. In *International Conference on Software Reuse*. Springer. 10.1007/978-3-540-27799-6_22

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. IEEE. 10.1109/FIT.2012.53

Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, *14*(1), 44–51. doi:10.1109/MIC.2009.143

Le Goaer, O. (2009). *Styles d'évolution dans les architectures logicielles, Université de Nantes; Ecole Centrale de Nantes (ECN)(ECN)(ECN)*. ECN.

Le Goaer, O., Oussalah, M., Tamzalit, D., & Seriai, A.-D. (2007). *Evolution dirigée par les styles. Atelier RIMEL*. Rétro-Ingénierie, Maintenance et Evolution des Logiciels.

Pirotte, J. (2014). *Les quatre défis à surmonter pour l'Internet des Objets*. Academic Press.

Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 291–319. doi:10.1016/j.jksuci.2016.10.003

Schilit, B., Adams, N., & Want, R. (1994). Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. IEEE. 10.1109/WMCSA.1994.16

Weiser, M. (1991, Sept.). *The computer for the twenty-first century*. Scientific American.

Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., & Du, H.-Y. (2010). Research on the architecture of Internet of Things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*. IEEE.

Chapter 9
Blockchain Governance for Collaborative Manufacturing

Marty Kelley
Johns Hopkins University, USA

ABSTRACT

The manufacturing industry is rapidly changing due to widespread adoption of information and communication technologies. This new landscape, described as the fourth industrial revolution, will be characterized by highly complex and interdependent systems. One particular aspect of this shift is horizontal integration, or the tight coupling of firms within a value chain. Highly interconnected and interdependent manufacturing systems will encounter new challenges associated with coordination and collaboration, specifically with regards to trust. This purpose of this chapter is to explore the potential of blockchain to address these challenges. Survey data collected from manufacturing professionals suggests that the perceived nature of trust and resource value can be bounded and controlled. Concepts from game theory, systems theory, and organizational economics are used to augment this research data and inform a collaborative manufacturing blockchain model and architecture.

INTRODUCTION

Much has been written on how the manufacturing industry, through technology adoption and innovation, is facing fundamental changes. Accordingly, organizations like Platform Industry 4.0 and the National Institute of Standards and Technology (NIST) have laid out roadmaps and published works describing what the future holds for manufacturers, customers, and nations (Federal Ministry of Economic Affairs

DOI: 10.4018/978-1-7998-2975-1.ch009

and Energy, 2016; Lu, Morris, & Frechette, 2016). Whether these fundamental shifts are economically influenced, socially inspired, or technologically inevitable, the effects will be far reaching. Manufacturing, at its core, is a simple process with a raw material input, transformative process, and a final product output. However, manufacturing value chains are anything but simple with cyberphysical systems, complicated processes, and cutting edge operational technologies coming together to form vast networks of value creating partnerships. The boundaries of the traditional manufacturing firm have been completely blurred as the industry leverages platforms, social networks, and the sharing economy to produce high quality and affordable end products in a lot size of one. Just as the previous industrial revolutions influenced more than industry and technology, the fourth industrial revolution will impact social and economic structures. As engineers design and build systems that will interact in a complex network of systems, new considerations must be made with respect to how these systems will interact, coordinate, and collaborate.

INDUSTRY 4.0 AND HORIZONTAL INTEGRATION

For nearly a decade, industrial nations have been allocating funds for the reinvigoration and advancement of the manufacturing sector (Kang et al., 2016). With the belief that new technologies and cheaper computing can inject new levels of global competitiveness in an aging industry, nations like the United States, China, and Germany have launched strategic initiatives, participated in private/public consortiums, and funded incubators with the hope of achieving the promise of advanced manufacturing. Not surprisingly, these initiatives are remarkably similar to one another (Kagermann, Wahlster, & Helbig, 2013; Kang et al., 2016; National Science and Technology Council, 2012). Governments recognize that achieving the vision of advanced manufacturing will require participation from industry, academia, and entrepreneurs working together to exploit the collision of information, communication, and operational technologies.

The German initiative, "Industrie 4.0," is a good illustration of what these industrialized nations are working toward. Self-described as a "vision" where products control their own production, the authors of the first official Industry 4.0 publications describe a future enabled by the convergence of cyberphysical systems and the Internet (Pfeiffer, 2017). The Platform Industrie 4.0 working group, a government-led group of academics, industry partners, and government officials, has gone many steps forward from just describing a vision. In 2013, the working group published a final report detailing many aspects of Industry 4.0, including three features that must be implemented to fully benefit from the fourth industrial revolution (Kagermann et al., 2013). One of these features is horizontal integration across the value chain.

Horizontal integration of the value chain in an Industry 4.0 context is described as the collaboration between nodes in a production network. This type of integration has traditionally taken the form of strategic alliances, mergers, and acquisitions as a means to increase production capacity or strategically enter new markets. Industry 4.0 value chain networks, however, can include numerous types of decentralized nodes such as people, individual machines, information systems, or complex systems. It is expected that the automation and codification of discrete processing steps will eventually be dynamically distributed to a variety of business entities rather than a single organization. According to the Platform Industry 4.0, this raises several issues like financing, development, reliability, risk, liability, and the protection of ingress protection (IP). Coordination between value chain partners will need to consider how responsibility for these elements will be assigned, incentivized, verified, and documented (National Academy of Sciences, 2016).

Distributed manufacturing is not a new concept. Despite the challenges outlined by Platform Industrie 4.0, progress has been made in terms of realizing adaptable, reconfigurable, and autonomous manufacturing systems. (See Kuhnle [2010] for an exhaustive review of distributed manufacturing.) However, implementation details are largely along a single technical dimension. There are other design considerations to enable the integration of a value network as described by Industry 4.0. The terminology most used in the academic literature for horizontal integration of this scale is collaborative networked organizations (Brettel, Friederichsen, Keller, & Rosenburg, 2014). These types of networks can possess very high levels of complexity both in terms of how coordination and collaboration is controlled and also managing the interests of the participating entities (Lockett, 2011). The presence of these complexities has led to a number of different academic research fields, including the impacts of information technology, enhancing supplier/customer relationships, and the impacts of the product lifecycle on product service systems (Durugbo & Riedel, 2013).

The Platform Industry 4.0 publications describe how central platform services (CPS) platforms will need to “orchestrate” various “perspectives” of the value chain over and above the actual technical services. The use of the term orchestrate is exceedingly broad and is meant to be so. Orchestration refers to coordination, collaboration, and cooperation across enterprises, which has economic and organizational considerations (Kagermann et al., 2013). In order to analyze the impact of horizontal integration on system design decisions, an understanding of how these systems interact organizationally is necessary.

Organizational Economics

A highly connected and networked value chain will result in new collaboration paradigms which may look nothing like traditional value chains. Today, firms are generally organized in a hierarchical manner, interacting with each other through market price mechanisms, authoritative power, and complex contracts. This coordination structure is theorized as an efficient way to reduce the costs associated with participating in a market (Coase, 1960). Transaction cost economics theorizes why firms exist and how their boundaries are defined; transaction costs related to economic activity, such as production, can be minimized if they are coordinated under a single governance structure (Coase, 1937). However, the boundaries of a firm should now be under consideration as new collaboration structures are evolving between companies enabled by information and communication technologies. These new technologies can further reduce coordination problems and information asymmetry between economic agents from different firms. Traditionally, companies make complex make/buy decisions and strategic partnership decisions, creating complicated contracts. New collaborative platforms, however, can automate coordination and drastically reduce transaction costs. In fact, many real-world organizational structures have been evolving into complex networks of interacting agents because traditional forms of collaborations are insufficient to address the evolving challenges and demands of industry (Powell, 1990).

The collaboration mechanisms for these complex structures are fundamentally different from neoclassical and even transactional cost economic theories. Cooperation in networks is "trust based" whereas traditional hierarchical organizations rely on price mechanisms and authoritative power to allocate scarce resources. Elsner, Hocker, and Schwardt (2010) argued that:

With complex structure, evolutionary process, and emergence, the boundaries of the firm become not only shifting but also fuzzy, and both the vertical and horizontal boundaries then do not only depend on given technological conditions, but on complementary capabilities, absorptive and learning capacities, and path dependence, and the external relations of the firm become crucial to the firm's existence and development. (p.2)

The implications of this are not subtle; a collaborative networked structure operates on a different set of rules. Elsner et al. (2010) clarified:

Progressive value-added chain and supplier network governance rules would require levels of learned trust that allow for the outflow of positive externalities among firms so that profits may be reduced by inevitable outflows of knowledge but will be compensated by corresponding inflows. (p.18)

This, however, will only be feasible in the case of trusting, coordinated, and, particularly, cooperative action. Elsner et al. (2010) suggested institutionalizing trust as a way to solve complex problems with evolutionary processes.

Trust

Designing trust into a system has some significant challenges. Studies suggest that once trust is lost between agents, it is very difficult to get back (Yan, Zhang, & Vasilakos, 2014). This fact alone can discourage agents from investing the necessary resources to attain trust and decide to operate within the predictable confines of a hierarchy. Also, trust can be its own worst enemy. According to Adler (2001, p. 226), "Trust can fail us because it makes betrayal more profitable." In a trusting environment, participating agents risk more resources making betrayal more profitable for free riders (Adler, 2001). Yet another challenge is the tendency for trusted environments to become exclusive. A main creator of trust is shared norms and path dependence. These can lead to "clan like" exclusivity and elitist institutions, which further inhibit knowledge sharing and innovation (Adler, 2001). Powell (1990) suggested that networks relying on trust will sustain collaboration as long as there is common background among participants. In addition, as diversity of participants increases, both trust and collaboration recede (Powell, 1990).

A 2011 study by Jung and Lake (2011) used an agent-based model to simulate and analyze how independent agents create different organizational forms. The simulation suggested that sustainable networks are created based on social affinity rather than information sharing. In fact, agents who formed information-sharing networks eventually leave the network for the safety of a hierarchy. While this study reinforces the idea that reciprocity and path dependence play a major role in network forming, there is a dark side as well: bad actors have the capacity to dissolve the network entirely by eroding collective trust. It appears that although trust is necessary for network forming, its very nature may inhibit network longevity, prohibiting the capture of any economic benefits of collaboration.

As changes in the industrial landscape lead to more distributed and decentralized manufacturing systems, a higher degree of trust is required to achieve the level of integration described by Industry 4.0. Miller and Drexler (1988) succinctly addressed this in stating, "Security is needed where trust is lacking, but security involves overhead; this provides an incentive for trust" (p. 165). Unfortunately, trust is expensive and time consuming to develop by traditional means. A distributed computing infrastructure may offer at least a partial solution to these challenges. Indeed it stands to reason that a distributed manufacturing environment may do well to operate on a distributed infrastructure. Whether as a standalone solution or a component in a deeper technology stack, distributed ledger technology, or blockchain, may provide solutions to some of Industry 4.0's trust problems.

Blockchain

To achieve the Industry 4.0's vision of horizontal integration and address the challenge of enabling collaboration in this highly interconnected landscape, trust either needs to be quickly and predictably obtained or designed out of the equation systemically. This is where distributed ledger technology or blockchain may play a role.

Blockchain is an information technology infrastructure that allows people to coordinate and conduct transactions without necessarily trusting one another or an independent authoritative third party. This is accomplished by replicating a ledger of transactions and storing it on all nodes of the network rather than one centralized copy. The records are immutable; changes must be approved by mathematical consensus (Dhillon, Metcalf, & Hooper, 2017). Of course, cryptographic security is an important aspect of the system. However, the decentralized structure promotes transparency and discourages opportunistic behavior.

As discussed, traditional hierarchical systems rely on specific mechanisms, including complex contracts and clear boundaries, to control and manage processes. However, new methods are needed to enable interconnected networks of diverse participants contributing to a joint benefit. The objective of this chapter is to explore the potential of blockchain to enable collaboration at the Industry 4.0 scale.

BACKGROUND

Blockchain, originally proposed by Nakamoto (2008), may be best known as the technology behind Bitcoin. Blockchain, or at least some variant of distributed ledger technology, is present in all cryptocurrencies, the initial use case of the technology. Recently, other use cases have been explored. Various applications of blockchain technology have been written about and implemented to address challenges in several industries. The breadth of new use cases for blockchain in development or implementation is vast. New applications are being explored every day. Any investigation into how blockchain can solve or address issues related to horizontal integration among manufacturing value networks must first examine how distributed ledger technology is being used in these other domains. A review of the literature yields novel applications of blockchain technology and gaps in our understanding of how to meet requirements for collaborative manufacturing.

Supply Chain, Internet of Things (IoT), and Manufacturing

One field that has received a lot of attention is the application of blockchain to the supply chain. The main focus of these applications is the provenance of assets transferred between companies and the verification of asset quality. Chen et al. (2017) wrote that self-interested supply chain partners, information asymmetry, and the cost of inspections are the leading supply chain challenges facing companies today. They propose the use of a blockchain to record all transactions in a distributed manner to increase data transparency and reduce the cost of redundant inspections. Toyoda, Mathiopolos, Sasase, and Ohtsuki (2017) proposed a model designed to use a blockchain in a counterfeit parts avoidance system where algorithms are used to incentivize behavior and verify asset provenance. Tian (2017) suggested the use of radio-frequency identification (RFID) and global positioning system (GPS) data, particularly around critical or hazardous assets, as a means to publish transactions to a blockchain, thereby increasing transparency and increasing trust. Similarly, Bocek, Rodrigues, Strasser, and Stiller (2017) proposed the introduction of sensors on products that must meet regulatory standards, such as temperature and vibration. These sensors would publish data to a blockchain visible to all supply chain participants.

Supply chain use cases of blockchain focus on transferring the characteristics of a blockchain to transactions performed between different business entities. Supply chain use cases almost universally cite data transparency and independence from third parties as the main value of implementing a blockchain. This is a similar requirement for Industry 4.0 horizontal integration in that manufacturing systems and processes will span traditional firm boundaries and may involve sensitive information not normally trusted to third parties. Certain elements of these implementations could benefit the collaborative manufacturing domain. However, the core issue of trust is not completely addressed.

Supply chain management systems have existed since the introduction of the Internet. Designed to share and visualize information, these systems have had limited success in the supply chain, largely due to lack of trust among participants. The blockchain models mentioned may address specific elements of the supply chain. However, the fact remains that traditional contracts in the form of purchase agreements and capacity agreements are still preferred to open collaboration. Despite many interesting blockchain models proposed in the literature for supply chain use cases, there is little research to suggest successful adoption.

Another emerging application of blockchain technology is the IoT. Although the term is many years old, only recently has the true impact of ubiquitous connectivity and smart products been realized. Along with incredible promise, a reality where millions of sensors and actuators are connected on the Internet comes with challenges. Conoscenti, Vetro, and DeMartin (2016) conducted a systematic literature review in which numerous blockchain IoT use cases are discussed, including data storage, identification management, and rating systems.

Scalability and anonymity are key aspects of blockchain technology that become apparent when reviewing IoT use cases. The volume of devices potentially involved in a horizontally integrated value chain necessitate careful consideration of these characteristics. The authors conclude there is a direct correlation between security and scalability; trade-off decisions must be made at an architectural level. Christidis and Devetsikiotis (2016) echoed this relationship, exploring blockchain models for IoT and smart contracts. They argued that the addition of smart contracts does more than just facilitate transactions. It enables interaction through automated and deterministic contract execution. This, according to the authors, holds tremendous value when considering IoT where many time-consuming and laborious workflows can be automated and distributed in a cryptographically secure manner (Christidis & Devetsikiotis, 2016). Bahga and Madisetti (2016) took this idea further and incorporated blockchain into a cloud environment where IoT devices can publish and call transactional data via smart contracts. This is a popular model in the literature as it leverages service-oriented architectures with the immutability and transparency of a blockchain.

Although limited in the literature, novel uses of blockchain technology in manufacturing use cases can be found. Lehmann, Schock, and Kufner (2017) set out to describe blockchain requirements specific to small- and medium-sized manufactures with an Industry 4.0 context. Due to the scale of digitization coming in the manufacturing industry, they believed there is potential for decentralized infrastructures like blockchain to overcome the burden of costly, centralized IT systems (Lehmann et al., 2017). Despite this broad claim, however, the authors admit there is no one-size-fits-all solution in manufacturing for a blockchain.

Zhang, Liu, and Shen (2017) proposed a manufacturing blockchain model specific to smart factories. In this complex environment, people, products, and machines interact on an industrial IoT. The model allows for information and physical resources to be shared on an enterprise blockchain that would interface to external blockchains (Zhang et al., 2017).

Backman, Valtanen, Yrjola, and Mammela (2017) offered a novel use of blockchain as a network of machines publishing spare capacity for other network participants to lease. This mimics the methodology used by 5G mobile network brokers, creating virtual instantiations of physical networks to increase overall capacity. In the study, blockchain with a variety of smart contracts act as a broker to reconfigure networks of machines to maximize capacity (Backman et al., 2017).

Perhaps the most complete manufacturing use case for blockchain comes from Angrish, Crver, Hasan, and Starly (2018). Their FabRec model is a blockchain model for manufactures to publish machine and design data. FabRec uses a structure of smart contracts to orchestrate cross-company manufacturing processes. The authors specifically mention trust as a costly but necessary element of doing business (Angrish et al., 2018).

However, if true virtual manufacturing potential is to be realized, new technologies, specifically distributed ledger technologies, will need to be adopted to address trust. Liu, Jiang, and Leng (2017) a step further and describe a "production credit mechanism" to systemically assign reputation values to participants to facilitate data sharing. Termed "social manufacturing," it includes requirements for voting and complete autonomy among participants to create and change rules (Liu et al., 2017).

Self-governance is an important idea, at least according to Andrew, Broby, Paul, and Whitfield (2018), who explain that although blockchain has been called the trustless technology, it does not completely eliminate the need for trust. The authors lay out a set of integration requirements on the premise that any use of blockchain technology in manufacturing will need to be integrated with many other systems, all of which have their own interdependencies (Andrew et al., 2018). In light of this complexity and the realization that trust may never be eliminated, governance will play a crucial role in blockchain's success.

It is important to note that many of the blockchain models reviewed do not transfer assets or currency. Instead, they are used as a data layer where transactional information is shared with the security and immutability of a blockchain. There is value in sharing transactional data in a trusted way, especially if considering a future business relationship. This is not typically how a blockchain is used in other industries, but using a blockchain this way is still being explored for other enterprise applications.

Shared Resources, Trust, and Self-Governance

A review of supply chain, IoT, and manufacturing blockchain models in the literature yields a set of novel applications in which the technology, at least theoretically, addresses challenges of trust. Indeed, many of the authors cited specifically identify trust, either in third parties or partners, as the main driver for using blockchain technology. However, there is little evidence to suggest that the mere implementation of a blockchain will improve collaboration or facilitate more transparent resource sharing.

Many existing models address technical details but leave little assurance that adoption would actually lead to improvement. Similar to supply chain management systems developed in the post-Internet era, technical feasibility does not necessarily lead to successful adoption. If blockchain technology is to address the trust challenges presented in a horizontally integrated collaborative manufacturing environment, mechanisms must be designed to solve, or at least limit, the social dilemmas faced by system users.

Although the literature on resource sharing is vast, a useful analog can be found in the work of Ostrom (1990). In her 1990 publication *Governing the Common*, Ostrom (1990) described shared resource systems in which users successfully overcome challenges of trust through self-governance. Despite the knowledge that cooperation and collaboration can lead to maximized joint benefits, opportunistic behavior on the part of users can (and often does) degrade or destroy the resource system. This concept is germane to collaborative manufacturing networks that, despite knowing the benefits of collaboration, resort to costly, time-consuming, and incomplete contracts to dictate behavior. Ostrom challenged the idea that shared resources must be either privatized or supervised by an authority such as a government. Instead, her empirical field research shows that resource systems can be successfully managed through self-governance, thereby institutionalizing trust (Dietz, Ostrom, & Stern, 2003).

A common misconception in much of the blockchain literature to date is the idea that the inherent trustless nature of a blockchain will be automatically transferred to any use case presented. Instead, a more realistic interpretation is that the characteristics of a blockchain can be leveraged for any given use case through configuration and incentive design. For common pool resource systems, Ostrom (1990) developed a set of design principles present in many successful, self-governing communities. A detailed look at these design rules is beyond the scope of this chapter, but a subset can be used to aid in the configuration and design of a collaborative manufacturing blockchain. For example, clear boundaries delineating authorized users and resources must be defined (Wilson, Ostrom, & Cox, 2013). This may go beyond mere access control and could dictate the use of linked blockchains where necessary. This idea was explored by Shackelford and Myers (2017) who explained that blockchain developers need to consider the authority given to users when architecting a solution. The authors analyze each of Ostrom's rules as they apply blockchain development. They concluded that not all apply, but many have the potential to improve the collective action necessary for many use cases (Shackelford & Myers, 2017).

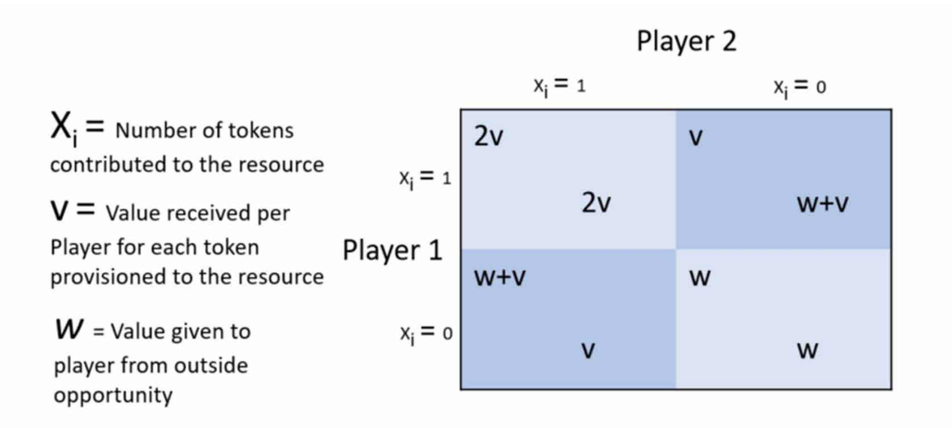
Before fully investigating the applicability of Ostrom's design principles to a collaborative manufacturing blockchain, certain elements must be analyzed, namely, the nature of the resources themselves. Ostrom's work was focused on a unique type of resource called common pool resources. Common pool resources are subtractable in nature, that is, two people cannot use the same resource unit. It is also difficult to exclude their use from unauthorized users. Examples would be a forest or fishery. These characteristics have implications on how the resources are managed (Ostrom, 2010). By this definition, knowledge and data would not be considered a common pool because they are not subtractable. However, Ostrom remarked that with current information technology, there are new ways to restrict, authorize, and trace information which could affect how it is managed (Hess & Ostrom, 2007). In addition, Ostrom's lab work confirmed that there are specific social dilemmas encountered in common pool resource systems that are also encountered by public goods or goods that are not subtractable like data and information (Ostrom, Gardner, & Walker, 1994). Her research shows that provisioning of public goods can be modeled using game theory where dilemmas and opportunism exist despite the resource being a common pool resource or public good (Ostrom et al., 1992).

PROVISIONING PUBLIC GOODS AND THE BLOCKCHAIN

In many ways, a blockchain used by a network of collaborating manufacturers resembles a public good. The data shared on the network is not subtractable. Once access has been granted to any individual entity, human, or machine, their exclusion becomes limited. Similar to a public good, the participants on the blockchain will be responsible for maintaining it as a resource and ensuring it is adequately provisioned. Transactions on the blockchain need to be mined, which will cost some amount of computing resources to verify the validity of the transaction. Similarly, smart contracts will cost some amount of computing resources to execute the code. In addition, there is a cost associated with sharing data with the blockchain from the perspective of the originator, especially in the event that some competitive advantage is risked.

For a group of manufacturers interested in collaborating on a blockchain, these costs must be offset by the value gained from participation. However, due to the nature of the resource and of the blockchain itself, individual participants may be incentivized to behave opportunistically. In the laboratory, Ostrom, Gardner, and Walker (1994) modeled this behavior using game theory. This approach allowed the researchers to isolate specific variables to determine their effect on outcomes in a variety of scenarios. Blockchain configuration and design to address issues of trust must be informed by these models and their insight.

Figure 1. Provision of public good game matrix



Public Goods Games

Any system where users must provide or maintain the resource can be modelled as a "provisioning a public good" game (Ostrom et al., 1994). In these situations, a resource system depends on the contributions of entities which extract value from it. An example would be Wikipedia. The Website is free but depends on the contributions of individuals to keep it up to date, relevant, and accurate. There is a point, however, where if not enough people contribute, the Website will lose its value. Similarly, a blockchain's collaborative tool will only provide enough value to the network participants if they contribute a sufficient amount of resources albeit data or computing power. However, ensuring the appropriate levels of resource contribution is no easy task. There are many context-specific variables to consider when designing a system that involves public goods.

Consider a simple example of two individuals deciding to contribute a fixed number of tokens *xi* to either a shared resource or some outside opportunity with a known return *w*. The individuals must compare the value of *w* and the yield of the resource, which for this example is *v*, for each token provisioned for the resource. The type of this particular resource is a public good. Therefore, the yield of the resource system is shared between both participants regardless of who contributes. Figure 1 summarizes the possible outcomes of a one-shot decision.

This, of course, is a simplistic representation. However, the basic elements are exposed. Equilibrium is dependent upon the relationship between *v* and *w*. It there is no way to exactly calculate payoffs, then decisions are made based on perceived values. In this example, there are two possible outcomes contingent on how the players perceive the value of the variables. If they believe that the payout of the resource is greater than some outside opportunity, then they will both contribute and joint benefit is maximized. If they perceive the outside opportunity to be greater than v but less than $2v$, the dominant strategy is to not contribute. This is what is known as a prisoner's dilemma. Its implications to a collaborative blockchain are obvious. Despite the knowledge of maximized joint benefits, adoption at the individual level can lead to suboptimal outcomes for the system as a whole. Of course, as was explained, if there is a high level of trust among participants, the network is capable of reaching an optimal equilibrium. However, trust at this level is costly and time consuming to develop. It cannot possibly keep pace with the dynamic reality of a horizontally integrated value network described by Industry 4.0.

A simple modification can be introduced to the previous example, exposing yet another possible equilibrium state. In some resource systems, the contribution levels have a lower limit in which if it is not reached, there is no value at all and any contributions made will be completely lost. An example of this is a tunnel in which two people must decide whether to contribute their resources to dig and meet at the middle. If one person contributes and the other does not, there is no tunnel. Value is gained only if both contribute.

In this scenario, the lower limit of required resource contributions is known as a provision point. In laboratory experiments, the introduction of a provision point can change the strategy of participants and lead to maximum joint benefit. This strategy is known as assurance. The flip side is also true as participants will not contribute if they believe others will not. In laboratory experiments, low provision points were shown to lead to maximized joint benefits after repeated rounds of play (Isaac, Schmitdz, & Walker, 1989). High provision points were shown to be less successful and encouraged freeriding in later rounds.

Another experimental approach to solving the public goods problems is known as a payback mechanism. A payback mechanism incentivizes contribution by providing a way for unsuccessful contributions to be repaid if a provision point is not met. Payback mechanisms do not always work. Most are successful in situations where providing additional resources over the provision point does not significantly increase payoffs.

The concepts of provision points and contribution mechanisms provide an avenue to explore how blockchain can be configured to overcome dilemmas facing partners in a collaborative manufacturing network. As stated, the technological capacity of existing models is robust and offers novel solutions for enterprises interested in using blockchain technology. However, adoption and success of these complex and interdependent systems is more than just a technical problem. Configuration and design of a blockchain must also address incentive challenges and dilemmas faced by the potentially large number of collaborating entities. The public goods game models, coupled with Ostrom's design principles, leads to new ways of architecting a blockchain for manufacturing.

Blockchain Architecture for Collaborative Manufacturing

Blockchains exist in all shapes and sizes. They have been architected for a wide variety of use cases. Once the double spend problem was addressed by Bitcoin's proof of work consensus protocol, architectural variations began to emerge to address security, scalability, and performance characteristics. In addition, as blockchain technology began to be applied to noncrypto currency use cases, integration to other systems in an enterprise landscape became extremely relevant.

From a manufacturing perspective, there is no shortage of transactional systems operating throughout the traditional automation pyramid that will need to integrate outside traditional firm boundaries in an Industry 4.0 capacity (Vogel-Houser, Kegel, Bender, & Wucherr, 2009). The challenge becomes identifying which components of a blockchain need to be modified to support a collaborative manufacturing use case. Considering the variety of blockchain models in the literature, perhaps a more prudent question is: Can adoption dilemmas be addressed through configuration? As discussed, there are specific elements both from Ostrom's field work with common pool resources and various laboratory experiments concerning public good provisioning that provide insight into the challenges described by Industry 4.0. These elements act as requirements for future blockchain development to ensure specific dimensions of collaboration and self-governance are met by the architecture.

User Boundaries: Ostrom's fieldwork with shared resource systems upheld previously held notions that defining boundaries around authority to use a resource is of critical importance. This may seem obvious. However, it is important to explain the specifics of why, particularly for a public good.

Unclear boundaries create uncertainty in how the resource is managed. It also promotes the incentive to free ride if the opportunity presents itself. As long as boundaries are clear, the ability to monitor, manage, and sanction undesirable behavior becomes more realistic. This is critical for any self-governed system (Shackelford & Myers, 2017). In terms of a blockchain, there are specific architecture trade of decisions that relate to user boundaries. First, a decision is needed on the level of decentralization required for the use case. Of course, a fully centralized infrastructure negates the need for a blockchain. A fully decentralized architecture implies no user boundaries. Instead, a permissioned blockchain offers the ability to include rules for defining who has access to the network. This can be done by either a trusted arbitrator or the physical network infrastructure itself. In addition, many permissioned blockchains include verifiers or outside entities that have been given the authority to verify events outside the blockchain. An example could be a regulatory body. The verifier does not necessarily need to be a single entity. However, the function of verification can be distributed among a variety of entities (Xu et al., 2017).

For a collaborative manufacturing use case, user boundary decisions have some obvious dependencies. For example, a permissioned blockchain can exist as purely private, an industry-wide consortium, or even product-centric if all the participants in a product value chain decide to share resources. This has implications for authority to join, permit access, or deny access.

In much the same way, verification roles depend on how manufacturers use the blockchain. The design of a verification function may be different for a blockchain comprised of a single value chain vs. industry-wide containing competitors. As discussed, for participants to be willing to share resources and invest in the operation of a collaborative blockchain, there must be perceived value greater than outside investment opportunities. In addition, participants must be assured that other participants will not erode the value of the blockchain by free riding. This perception will directly influence user boundaries and the architecture of the blockchain.

Resource Boundaries: In addition to user boundaries, resource boundaries are of critical importance to the design of a manufacturing blockchain. Ostrom did not initially identify resource boundaries as a separate design principle, opting instead to group it with user boundaries. However, Wilson et al. (2013) created a subprinciple calling attention to the importance of separating the boundaries of the shared resource and the broader social-ecological systems that may be present.

The dizzying amount of transactional data that could potentially be shared on a manufacturing blockchain deserves special attention when contemplating architecture and design. Specifically, data storage and computation can be accomplished in a variety of ways, none of which are trivial with respect to system operation. For example, data can be stored on a blockchain or off the blockchain in a standalone system. Data stored on the blockchain impacts performance but could potentially simplify integration requirements with other systems. Conversely, if data pertinent to the use case is stored off the blockchain, accessing it will require infrastructure.

Computation can also be achieved on the blockchain, most notably by smart contracts. It is perfectly conceivable to use a blockchain as merely a data layer or a registry with all associated computation done off the blockchain. However, the inclusion of smart contracts allow automated execution of transactions, permitted all the necessary inputs are available (Xu et al., 2017). This could be a significant feature for collaborators, specifically in an industry where automation is a necessity.

Manufacturing operations, even on a small scale, can utilize many systems and tools, all of which generate data. In addition, products themselves can generate data through the use of sensors and IoT. Any discussion regarding blockchain integration to a wider ecosystem must consider the data and the value of that data to its owners. On one hand, blockchain design decisions with regard to storage and computation is about cost because mining blocks and executing smart contracts will be impacted by these decisions. On the other hand, the perceived value of this data can dictate how the blockchain needs to be configured to ensure collaboration and self-governance. Any risk taken by network participants by sharing data must be offset by the perceived joint benefits of participation. To configure a blockchain appropriately to limit opportunistic behavior, data types and data sources need to be investigated in terms of their perceived value to owners.

Provision Point: As discussed, a provision point is a minimum value of resources needed to produce a public good. In public good game theory models, the existence of a provision point can lead to an equilibrium state with joint maximum benefit. This strategy is known as assurance. Laboratory experiments also show that the relative level of this provision point can effect the contributions of participants in both early and later rounds of a game (www.elementsproject.org).

A provision point designed into a collaborative blockchain may replicate these findings provided the level of provisioning is well understood. Provision points are usually found when the public good cannot be consumed unless it is appropriately provisioned for. This seems to contradict the nature of collaborative data sharing blockchain. If only one participant publishes transactions, the other participants are free to use the data. The public good has effectively been provisioned. However, in a permissioned blockchain, some level of provisioning could be mandated to gain access. This, in essence, acts as a provision point. It also acts as a guarantee to participants that other users have shared a similar amount of data with the network. This may influence individual strategy and lead to an assurance equilibrium.

A provision point designed into the architecture of a blockchain will most certainly depend on resource boundaries. The value of shared data is not easily quantified. Any individual participant providing data to the blockchain will need assurances that others are providing a sufficient amount of resources.

Another alternative for designing in the provision point is the use of side chains (or separate blockchains) for different data types. This effectively introduces multiple provision points depending on the perceived value of the data in each chain. For example, the base blockchain could consist of historical data with a requirement to provide 12 months' worth of data for any given asset. A separate side chain could be created for real-time data from the same assets. Isolating data types based on value could impose additional provision points as well as flexibility in governing access. Given the large amount of potential data elements, research is needed to understand how perceived values of data types can translate into a design element.

Payback Mechanism: Payback mechanisms refer to the action of paying back resources to participants if some provision point is not met. The purpose is to reduce the risk of individual participants investing in the provision of a public good so as to encourage participation and facilitate cooperation. In certain laboratory settings, payback mechanisms are shown to successfully provide the public good in up to 98% of cases (Bagnolli & McKee, 1991). Of course, in a laboratory setting, resource values are quantified and known among participants who have clear objectives.

Designing a payback mechanism is no trivial task for a collaborative manufacturing blockchain given the complexity of systems, data, and strategic objectives of the participants. However, there are aspects of the architecture that could be analyzed for applicability to a payback mechanism. For example, the process of mining requires incentives because mining blocks costs resources. In most tokenized blockchains, this is done through the creation or payment of currency. It is conceivable that the resources spent mining blocks of transaction for inclusion into the blockchain could be reimbursed if some provision point is not met. Again, this type of mechanism design would have important dependencies on how data is valued by the network participants and the structure of the blockchain itself.

SOLUTIONS AND RECOMMENDATIONS

As discussed, Ostrom's fieldwork and laboratory experiments yield a set of requirements that any collaborative manufacturing blockchain design must meet in order to address challenges of trust. User boundaries, resource boundaries, provision points, and payback mechanisms require the gathering of additional data from potential users in order to intelligently inform design decisions. Research was conducted to address this gap in the literature, specifically blockchain as a resource-sharing application in collaborative manufacturing networks.

The dynamics of individual dilemmas in public goods games are understood. However, the extension of these dynamics to the industrial domain is not. Are they the same? Can participants in a network collaborate through self-governance as suggested by Ostrom's empirical evidence?

In addition to the nature of dilemmas faced by value chain partners, questions pertaining to blockchain technology's ability to address trust is relevant to this study. This section discusses the method of data collection, analyzes the data, and introduces three architectural elements to address the challenges.

Data Collection Methodology

Horizontal integration from an Industry 4.0 perspective is multidisciplinary. The complexity of the problem requires research to be flexible in understanding its nature and the many perspectives it can be viewed. The diversity of specific situations that participants may face in a collaborative network is vast. For this reason, both quantitative and qualitative data were collected and analyzed. For the quantitative phase of the study, a survey was developed for professionals in the manufacturing and supply chain domain to assess attitudes and perceptions around key blockchain architecture elements. As discussed, user and resource boundaries, provision points, and payback mechanisms have been identified as key inputs into a collaborative manufacturing blockchain design.

Once survey data was collected, an analysis was performed to translate results into specific blockchain design elements. These elements were reviewed with a subset of the population in an open-ended interview. This phase of the study enhanced understanding of the survey results and the ultimate blockchain design.

Survey Results and Analysis

The survey consisted of eight questions. Two questions were designed to assess responder attitudes toward the usefulness and applicability of a blockchain, two questions were allocated to determining resource boundaries, and two explored provision points. User boundaries and payback mechanisms were each allocated one question. The survey was sent to eight manufacturing professional associations and distributed to attendees at a professional manufacturing and supply chain technology conference.

A total population of 44 responses were collected for analysis. Results from the initial questions pertaining to general blockchain attitudes describe a fairly optimistic population in terms of the potential of blockchain for their respective enterprise. When asked if they believed a blockchain would yield a positive return on investment, 59% replied "agree" or "strongly agree" (see Figure 2). In terms of adoption challenges, the privacy and security concerns option was cited in 84% of responses (see Figure 3).

Figure 2. Will collaborating on a blockchain result in positive ROI?

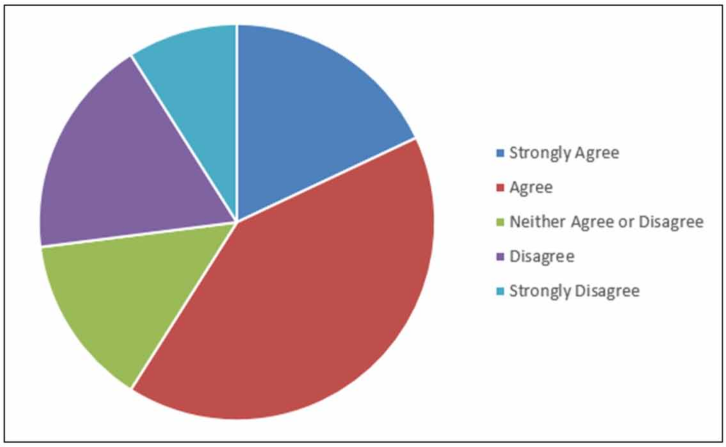
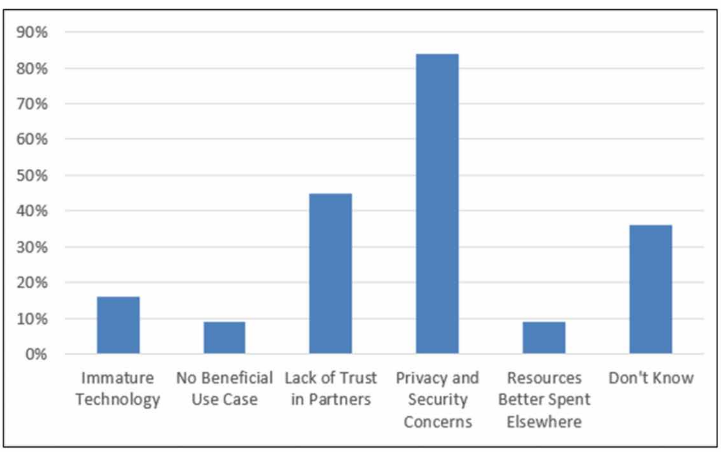
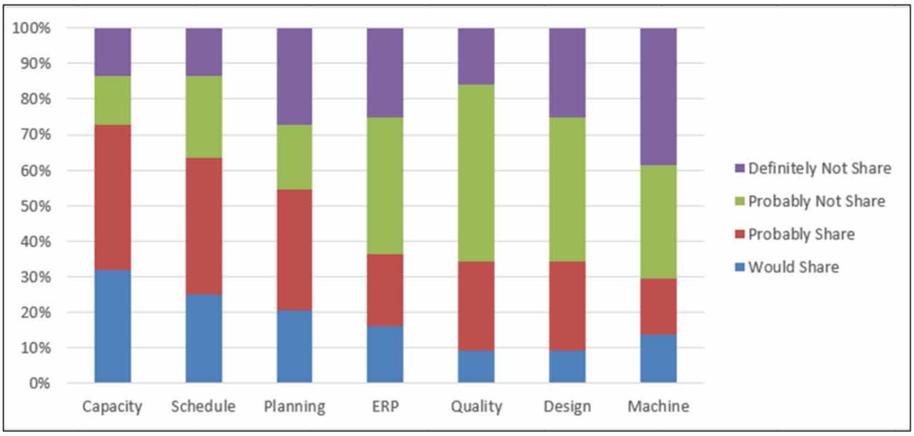


Figure 3. Challenges with participating in a collaborative blockchain



Professionals were asked to assign a value of high, low, or none to different potential partner types with respect to who they would be willing to share data with. Ninety-five percent of responders assigned either high or low value to both customers and suppliers. This reinforces the idea that sharing information outside

Figure 4. Willingness to share data with partners by data type



of firm boundaries is perceived to have a nonzero value. Sixteen percent felt sharing data industry-wide could produce value; only 9% felt an open access blockchain would provide any value at all. Perhaps surprisingly, 48% of responders felt sharing data with competitors could provide value.

In open-ended interviews with a subset of the population, professionals from small- and medium-sized companies were more willing to share data with competitors citing a need to be more collaborative when dealing with large and powerful customers. Interviews also revealed a diversity of context-specific scenarios in which sharing data with various partner types could yield value. Anecdotally, one interviewee explained that, despite indicating "no value" for an industry-wide blockchain, an industry-wide manufacturing blockchain designed for tracking material provenance could be of immense value.

Participants in the survey were also asked to assess various data types in terms of their willingness to share with outside entities on a blockchain (see Figure 4). There are clear delineations between data types and perceived value to the responders. For three data types, "definitely would" and "probably would" exceed 50% of responses. There is a relatively sharp drop for the remaining answer choices. When asked which data types would be beneficial if shared from other network participants, similar patterns result (see Figure 5). Particularly, "capacity" and "schedule" data types are perceived to be both valuable and shareable. Conversely, responders are unwilling to share "quality" and "machine" data types, finding little value in this data from partners.

Figure 5. Benefit of data type from partners

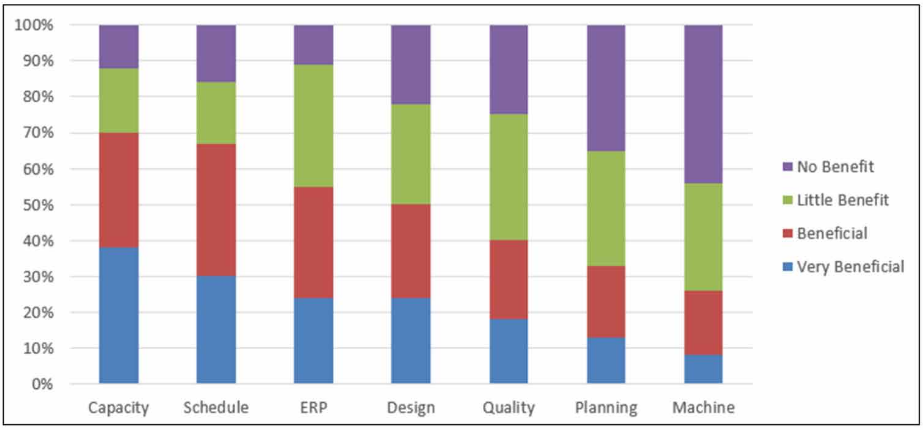
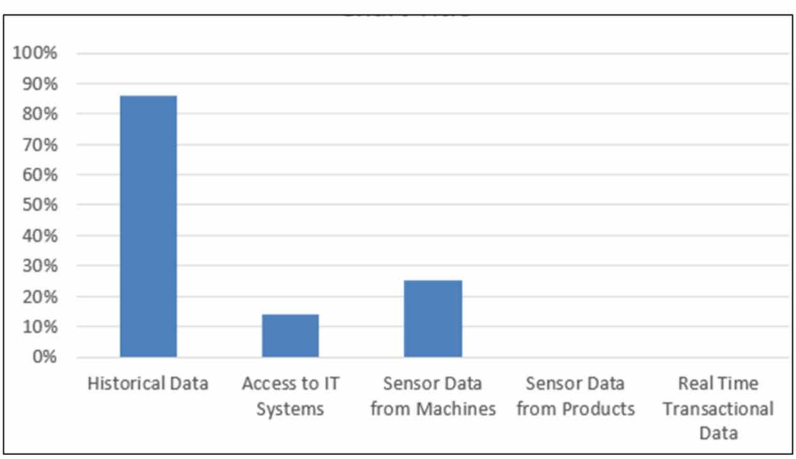


Figure 6. Initial access by data type



Interviews with responders yield additional insight with regard to data types. “Capacity” and “schedule” data were perceived as being lower risk, especially if data elements were not plant-wide and specific to a single product line or machine. Not surprisingly, “ERP” data was reported to be too risky to share. However, integrated ERP systems was cited as having potential benefits given the appropriate circumstances.

An important finding in the game theory literature in terms of provisioning a public good is the concept of a provision point. Responders were asked a series of questions to explore potential provision points for a collaborative blockchain. When asked to identify data sources they would be willing to share to gain access to a blockchain (see Figure 6), a vast majority (86%) answered that historical data was the only data they would be willing to share. Sensor data from production machines and access to internal information technology systems received 25% and 14%, respectively. Both the real time transactional data option and sensor data from products option received zero responses.

Responders were then asked if their willingness to provide this data would change if they were guaranteed other participants were providing the same. Sixty-one percent responded “maybe” with the remaining responses split between “yes” and “no” at 20% and 19%, respectively. When asked about this in the interviews, responders felt that historical data was an appropriate risk to take without knowing exactly what value could be gained from full participation in the blockchain. With regard to data source, many interviewees noted that it helped to hear a guarantee that all participants were sharing the same data. However, they felt it was more important to have guarantees that risks would have payoffs.

Survey responses for assessing opinions on payback mechanisms were inconclusive. While 18% favored some level of reimbursement, 41% did not and 41% were unsure. The concept of a payback mechanism is not easily translated to a general survey population because the details of how participants would be paid back is very specific to the system design. In addition, all three interviewees gave similar feedback, reporting that the effectiveness of a payback mechanism would need to be assessed on a case-by-case basis. The general suggestion that the cost of computational resources, or mining, would be reimbursed based on some level of participation did not yield any significant response.

Collaborative Manufacturing Blockchain Structure

The data from the previous section leads to a number of conclusions with respect to blockchain design decisions. First, architecture development for a collaborative blockchain requires a degree of flexibility in terms of how entities connect with one another on the network. Data from the survey verifies speculation that a permissioned blockchain would be most appropriate. Although manufacturers could participate in a variety of blockchain consortiums based on specific use cases, a single blockchain utilizing a permission granting mechanism could be a more adaptable and efficient alternative. Second, perceived value of data types can best be addressed with a multitiered resource boundary concept. The relationship between perceived risk and value can potentially be managed through a structure of smart contracts that

Table 1. Matrix of survey questions, requirements, and architecture elements

Question	Requirement	Architecture Element
Who would you be willing to share data on a collaborative blockchain?	User Boundary	Smart Contracts: Use of Off Chain Oracle to verify partner types (i.e., via cage codes)
Would you be willing to share more data with the network if you could be reimbursed the cost of providing it?	Payback Mechanism	Data inconclusive, no element represented
What type of data would you be willing to share on a collaborative blockchain?	Resource Boundary	Smart Contracts and Confidential Transactions: Data type parameters used in smart contract execution in conjunction with blinding keys
What type of data from your value chain could benefit your organization?	Resource Boundary	Smart Contracts and Confidential Transactions: Data type parameters used in smart contract execution in conjunction with blinding keys
How much data would you be willing to share in order to gain access to a collaborative blockchain? Mark all that apply.	Provision Point	Verification and Access: Trusted verifier nodes with access control based on initial data requirements
Would you be willing to share more data if you were guaranteed other participants were sharing the same amount?	Provision Point	Smart Contracts: Use of Off Chain Oracle to facilitate gradual data sharing based on type

enforces trust building through the sharing of low-risk resources. Third, assurance appears to be a feasible strategy for participants. Segregating more valuable data types with respect to partner type improves perception of shared risk.

As a result of this synthesis, three specific architecture elements are proposed to address the requirements of user boundaries, resource boundaries, and provision points with the intention of constructing an enabling platform for collaborative manufacturing. Table 1 provides a matrix relating the survey questions and requirements to architecture elements.

Verification and Access: Initially, any participant will need to gain permission to the blockchain. This will require some initial investment of data verified by a decentralized, multisignature federation of verifiers (see Figure 7). From the data analysis, this initial operating requirement can be historical capacity and schedule data. In traditional transactional systems, this type of data can have a wide variety of nomenclatures. However, the meaning is essentially the same.

Work cells, cost centers, tool usage, deliveries, and release dates are all potential data values for the capacity and schedule data type. Once verification of the participant is complete, access to the blockchain is granted and the historical transactions are published for all participants. This initial data requirement acts as both a provision point and a boundary for users and data. The action of providing data to gain

Figure 7. "M of N" verification, in this case

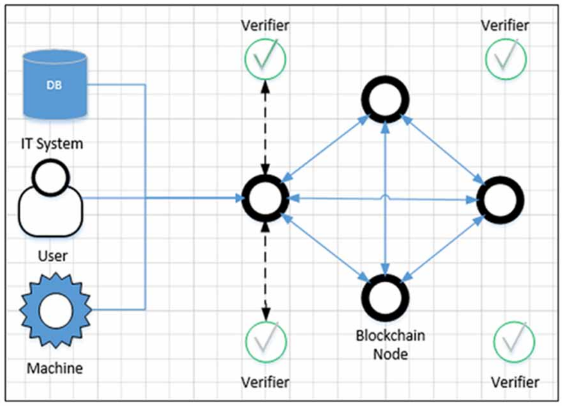


Table 2. Transaction data model

Element	Description
Sender	Signed public key of sender account
From	Data origination field
To	Data destination
DataType	Specify the type of data being transacted
DataSource	Specify the source of data being transacted
Data	String containing message data or initialize contract
TimeStamp	Time when transaction was executed

access should improve the participants willingness to collaborate by invoking an assurance strategy. Similarly, all users who are granted access have been verified by a federation of verifiers.

The use of verifiers, although centralized in nature, is common in permissioned blockchains as a way to inject trust in the system. "M of N" is a common model used to partially decentralize the function of verification by requiring approval from a predefined number of verifiers, M, out of the total population of verifiers, N. Verifiers can be any off-chain entity with the information necessary to complete the verification.

Data Model and Confidential Transactions: Every transaction published to a blockchain is a cryptographically signed composite of information from one entity to another. This must follow a predefined data schema. Based on the data collected, certain elements can be added to the transaction data model to enable the creation of useful smart contracts. Table 2 shows the complete list of data elements for the transaction.

Given the difference in perceived risk and value of the actual data, the datatype and datasource transaction elements can be used as conditions by smart contracts. In addition to the standard transaction data model, the research survey suggests that more valuable types of data be segregated and shared with participants who have taken the risk to share the same data type.

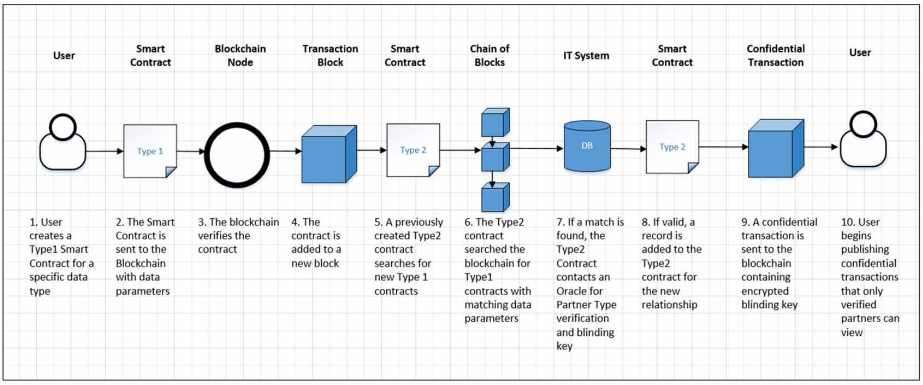
There are various ways to address this, one of which is to use confidential transactions. Although anonymity is preserved on a blockchain, privacy is not. All transactions are visible on the blockchain to all participants even if the identity of the transactors is unknown. Some transaction, however, may need to have the content decipherable only to those entities that are allowed to see it. In the case of more valuable data types (ERP data or design data), the contents of a transaction can be encrypted using a blinding key (Elements, n.d.). This key is hashed to the private address key of the users and can be used to unencrypt sensitive data. Although the transaction will still be visible to all on the blockchain, the contents of the transaction is kept confidential. When used in conjunction with smart contacts, confidential transactions can address challenges specific to this use case.

Smart Contracts: As discussed, user boundaries, resource boundaries, and provision points are necessary components of a successful self-governed resource system. Using the data gathered from manufacturing and supply chain professionals, a series of smart contracts can be created to manage this information and automatically execute transactions. Contracts are not themselves transactions. Instead, they are functions and data stored on the blockchain as bytecode, usually compiled by another program like solidity (Bambara & Allen, 2018).

To facilitate resource sharing in a self-governed environment, these contracts can be predefined and automatically executed without engaging in lengthy contract discussions with individual partners. In addition, a smart contract can be created to access off-chain data. It can be published based on conditions predefined by users, effectively establishing an automated provision point. As transactions are published to the blockchain, smart contracts can automatically manage access for predetermined data types between partners in a secure way. Based on the data gathered, two types of smart contracts can be defined and used in concert to create such a system.

The first type, Type 1, can be created by a network participant for each data type and data source combination that they are willing to share above the initial data requirements for gaining access to the blockchain. The contract will contain a

Figure 8. Smart contract flow



series of parameters that can be used by other contracts. Those parameters include a unique address, data type, and data source.

The second type, Type 2, will be a contract on the blockchain that looks at every new block for a newly created Type 1 contract. The Type 2 contract will then use the parameters of the Type 1 contract to match other partners who have previously created Type 1 contracts with the same data parameters. When matches are found, the Type 2 contract will send parameter data to an off chain Oracle for verification of the partner types.

If the participants are in the same value network, or are suppliers or customers of one another, a confidential transaction is published with a hash of the blinding key and the private keys of the user. The user can then unencrypt the blinding key using their private key and begin publishing confidential transactions of the new data type. Note, this construct permits data sharing of certain data types only between partners who have created Type 1 contracts and provided access to their own data (see Figure 8).

This combination of smart contracts and confidential transactions effectively creates a mechanism for gradual collaboration based on the participant's perceived risk and data value. The Type 2 smart contract creates a record of users, machines, or systems that have agreed to share transactional data of a specific type with the knowledge that all other nodes have made the same assessment of risk and value. This way, a series of provision points are created to establish user and resource boundaries.

FUTURE RESEARCH DIRECTIONS

Given the complexity of potential ineractions among value chain partners, much more research is needed. Specifically, simulations in a laboratory environment to observe behavior within the context of a manufacturing environment would provide valuable insight into blockchain designs and architecture. Similarly, this study did not address computational limitations of architecture decisions relevant to the use case. Specifics with regards to consensus algorithms, verification activities, tokenization, and mining operations all play a part in the tradeoff between security and resources. In addition, various threats to validity exist for this partcular research survey. For example, the sample size is small and the sample population may contain measurable bias. More data from a larger set could use more rigourous computational methods to classify survey responses.

CONCLUSION

The complexity of a horizontally integrated value network at the scale described by Industry 4.0 cannot be understated. People, machines, and systems have the technical capacity to connect, share data, and provide value cheaper, faster, and better than before. The impacts of this have already begun to change the way manufacturers meet customer demands. This technical capacity, however, must be scrutinized on an institutional level as decades of infrastructure, policy, and rules of thumb have left manufacturers entrenched in business as usual.

Trusted third parties, hierarchical authority, and complicated, yet incomplete, contracts dominate value chains despite the overwhelming potential of existing information and communication technologies. This is changing, however, as industry trends and consumer demands are forcing the hand of many manufacturers in order to meet business needs. Economics appears to be a driving force of change. It is perhaps appropriate to use an economic lens to analyze the potential use of blockchain technologies in this domain. Data collected in this study sheds light on how specific game theoretic dilemmas can be translated to architectural elements and how clear delineations exist in the perceptions of data types, data sources, and partner types in terms of both value and risk. Although various solutions can found to some of the specific adoption challenges discussed, a simple set of smart contracts with a privacy mechanism is sufficient to adequately bound users and resources while enabling an assurance strategy via gradually increasing provision points.

REFERENCES

Adler, P. S. (2001). Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, *12*(2), 215–234. doi:10.1287/orsc.12.2.215.10117

Andrew, C., Broby, D., Paul, G., & Whitfield, I. (2018). *Utilizing financial blockchain technologies in advanced manufacturing*. Glasgow, UK: Strathprints.

Angrish, A., Crver, B., Hasan, M., & Starly, B. (2018). A case study for blockchain in manufacturing: FabRec: A prototype for peer to peer network of manufacturing nodes. *North American Manufacturing Research Conference*. 10.1016/j.promfg.2018.07.154

Backman, J., Valtanen, K., Yrjola, S., & Mammela, O. (2017). *Blockchain network slice broker in 5G*. IEEE.

Bagnolli, M., & McKee, M. (1991). Voluntary contribution games: Efficient private provision of public goods. *Economic Enquiry*, *29*(2), 351–366. doi:10.1111/j.1465-7295.1991.tb01276.x

Bahga, A., & Madisetti, V. (2016). Blockchain platform for the industrial Internet of things. *Journal of Software Engineering and Applications*, *9*(10), 533–546. doi:10.4236/jsea.2016.910036

Bambara, J., & Allen, P. (2018). *Blockchain: A practical guide to developing business, law and technology solutions*. McGraw Hill.

Bocek, T., Rodrigues, B., Strasser, T., & Stiller, B. (2017). Blockchains everywhere: A use case of blockchains in the pharma supply chain. *International Symposium on Integrated Network Management*. 10.23919/INM.2017.7987376

Brettel, M., Friederichsen, N., Keller, M., & Rosenburg, M. (2014). How virtualization, decentralization and network building change the manufacturing landscape: An Indutsry 4.0 perspective. *International Journal of Information and Communication Engineering*.

Chen, S., Shi, R., Ren, Z., Yan, J., Shi, Y., & Zhang, J. (2017). A blockchain based supply chain quality management framework. *IEEE International Conference on e-Business Engineering*. 10.1109/ICEBE.2017.34

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of things. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 2292–2303. doi:10.1109/ACCESS.2016.2566339

Coase, R. (1937). The nature of the firm. *Economica*, *4*(16), 386–405. doi:10.1111/j.1468-0335.1937.tb00002.x

Coase, R. (1960). The problem of social cost. *The Journal of Law & Economics*, *3*, 1–44. doi:10.1086/466560

Conoscenti, M., Vetro, A., & DeMartin, J. (2016). *Blockchain for the Internet of Things: A systemic literature review*. IEEE.

Dhillon, V., Metcalf, D., & Hooper, M. (2017). *Blockchain enabled applications: Understand the blockchain ecosystem*. Orlando, FL: Springer. doi:10.1007/978-1-4842-3081-7

Dietz, T., Ostrom, E., & Stern, P. (2003). The struggle to govern the commons. *Science*, *302*(5652), 1907–1912. doi:10.1126cience.1091015 PMID:14671286

Durugbo, C., & Riedel, J. (2013). Readiness assessment of collaborative network organizations for integrated product service systems. *International Journal of Production Research*, *51*(2), 598–613. doi:10.1080/00207543.2012.658529

Elsner, W., Hocker, G., & Schwardt, H. (2010). Simplistic vs complex organization: Markets, hierarchies and networks in an organizational triangle. *Journal of Economic Issues*, *44*(1), 1–30. doi:10.2753/JEI0021-3624440101

Federal Ministry of Economic Affairs and Energy. (2016). *Aspects of the research roadmap in application scenarios*. Berlin: Federal Ministry of Education and Research.

Hess, C., & Ostrom, E. (2007). Studying the knowledge commons. In *Understanding knowledge as a commons: From theory to practice* (pp. 1–27). Hong Kong: MIT Press.

Isaac, M., Schmitdz, D., & Walker, M. (1989). The assurance problem in a laboratory market. *Public Choice*, *62*(3), 217–236. doi:10.1007/BF02337743

Jung, D. F., & Lake, D. A. (2011). Markets, hierarchies, and networks: An agent based organizational ecology. *American Journal of Political Science*, *55*(4), 972–990. doi:10.1111/j.1540-5907.2011.00536.x

Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative Industrie 4.0*. Frankfurt: National Academy of Science and Engineering.

Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., . . . Noh, S. D. (2016). Smart manufacturing: Past research, present findings, and future decisions. *International Journal of Preision Engineering and Manufacturing - Green Technology, 3*(1), 111-128.

Kuhnle, H. (2010). *Distributed manufacturing: Paradigm, concepts, solutions and examples*. London, UK: Springer. doi:10.1007/978-1-84882-707-3

Lehmann, C., Schock, C., & Kufner, T. (2017). Requirements for blockchain applications in manufacturing small and medium sized enterprizes. *International Conference on Production Research*.

Liu, J., Jiang, P., & Leng, J. (2017). A framework of credit assurance mechanism for manufacturing services under social manufacturing context. *IEEE Conference on Automation Science and Engineering*. 10.1109/COASE.2017.8256072

Lockett, H., Johnson, M., Evans, S., & Bastl, M. (2011). Product service systems and supply network relationships: An explanatory case study. *Journal of Manufacturing Technology Management*, *22*(3), 293–313. doi:10.1108/17410381111112684

Lu, Y., Morris, K., & Frechette, S. (2016). *Current standards landscape for manufacturing systems*. National Institute of Standards and Technology. doi:10.6028/NIST.IR.8107

Miller, M. S., & Drexler, E. K. (1988). Markets and computation: Agoric open systems. *The Ecology of Computation,* 133-176.

Nakamoto, S. (2008). *Bitcoin: A peer to peer electronic cash system.* Retrieved from https://bitcoin.org/bitcoin.pdf

National Academy of Sciences. (2016). Platform Industrie 4.0: Progress report. Berlin: Author.

National Science and Technology Council. (2012). *A national plan for advanced manufacturing*. Washington, DC: Author.

Ostrom, E. (1990). *Governing the common*. Boston, MA: Cambridge University Press. doi:10.1017/CBO9780511807763

Ostrom, E. (2010). Beyond markets and states: Polycentric governance of complex economic systems. *The American Economic Review*, *100*(3), 641–672. doi:10.1257/aer.100.3.641

Ostrom, E., Gardner, R., & Walker, J. (1992). Covenants with and without a sword: Self governance is possible. *The American Political Science Review*, *86*(2), 404–417. doi:10.2307/1964229

Ostrom, E., Gardner, R., & Walker, J. (1994). *Rules, games and common pool resources*. University of Michigan Press. doi:10.3998/mpub.9739

Pfeiffer, S. (2017). The vision of Industrie 4.0 in the making - A case of future told, tamed and traded. *NanoEthics*, *11*(1), 107–121. doi:10.100711569-016-0280-3 PMID:28435474

Powell, W. W. (1990). Neither market nor hierarchy: Network forms of organization. *Research in Organizational Behavior*, *12*, 295–326.

Shackelford, S., & Myers, S. (2017). *Block by block: Leveraging the power of blockchain technology to build trust and promote cyber peace*. *Yale Journal of Law and Technology*.

Tian, F. (2017). *A supply chain traceability system for food safety based on HACCP, blockchain and the Internet of things*. Vienna: Academic Press.

Toyoda, K., Mathiopolos, T., Sasase, I., & Ohtsuki, T. (2017). A novel blockhain based product ownership management system for anticounterfeit in the supply chains. *IEEE Access: Practical Innovations, Open Solutions*, *5*, 17465–17477. doi:10.1109/ACCESS.2017.2720760

Vogel-Houser, B., Kegel, G., Bender, K., & Wucherr, K. (2009). Global information architecture for industrial automation. *Standardisierung,* 108-115.

Wilson, D., Ostrom, E., & Cox, M. (2013). Generalizing the core design principles for the efficacy of groups. *Journal of Economic Behavior & Organization*, *90*, S21–S32. doi:10.1016/j.jebo.2012.12.010

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., ... Rimba, P. (2017). A taxonomy of blockchain based systems for architecture design. *IEEE International Conference on Software Architecture*. 10.1109/ICSA.2017.33

Yan, Z., Zhang, P., & Vasilakos, A. (2014). A survey on trust management for the Internet of things. *Journal of Network and Computer Applications*, *42*, 120–134. doi:10.1016/j.jnca.2014.01.014

Zhang, F., Liu, M., & Shen, W. (2017). Operation modes of smart factory for high end equipment manufacturing in the internet and big data era. *IEEE International Conference on systems, Man and Cybernetics*. 10.1109/SMC.2017.8122594

ADDITIONAL READING

Adler, P. S. (2001). Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, *12*(2), 215–234. doi:10.1287/orsc.12.2.215.10117

Coase, R. (1937). The nature of the firm. *Economica*, *4*(16), 386–405. doi:10.1111/j.1468-0335.1937.tb00002.x

Durugbo, C., & Riedel, J. (2013). Readiness assessment of collaborative network organizations for integrated product service systems. *International Journal of Production Research*, *51*(2), 598–613. doi:10.1080/00207543.2012.658529

Elsner, W., Hocker, G., & Schwardt, H. (2010). Simplistic vs complex organization: Markets, hierarchies and networks in an organizational triangle. *Journal of Economic Issues*, *44*(1), 1–30. doi:10.2753/JEI0021-3624440101

Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative Industrie 4.0*. Frankfurt: National Academy of Science and Engineering.

Nakamoto, S. (2008). *Bitcoin: A peer to peer electronic cash system*. Retrieved from https://bitcoin.org/bitcoin.pdf

Ostrom, E. (1990). *Governing the common*. Boston, MA: Cambridge University Press. doi:10.1017/CBO9780511807763

Ostrom, E. (2010). Beyond markets and states: Polycentric governance of complex economic systems. *The American Economic Review*, *100*(3), 641–672. doi:10.1257/aer.100.3.641

Ostrom, E., Gardner, R., & Walker, J. (1992). Covenants with and without a sword: Self governance is possible. *The American Political Science Review*, *86*(2), 404–417. doi:10.2307/1964229

Powell, W. W. (1990). Neither market nor hierarchy: Network forms of organization. *Research in Organizational Behavior*, *12*, 295–326.

KEY TERMS AND DEFINITIONS

Collaboration: The process of two or more entities reaching a common goal or outcome with joint benefits and/or costs. This process requires more than just orchestrated actions. It usually involves sharing resources, information, and costs associated with reaching the goal.

Horizontal Integration: The tight integration of processes, resources, and information between entities in a common value chain. Horizontal integration refers to the interdependence that emerges from such tight integration. It must be implemented systemically rather than contractually.

Industry 4.0: The fourth industrial revolution characterized by the convergence of cyberphysical systems and the internet of things (IoT).

Payback Mechanism: Any feedback mechanism that compensates individuals who have invested in a public good that fails to be provisioned. Payback mechanism design is a risk reduction technique to incentivize the provisioning of public goods.

Provision Point: A quantifiable limit that must be reached in order for a public good to be created. If the provision point is not met, regardless of the resources previously invested in it, the public good will not be created.

Smart Contracts: Name given to executable code stored on a blockchain. Smart contracts behave like transactions in that they must be mined and appended to a block. They can store simple code, which deterministically executes predefined logic.

Chapter 10
On Ontology of Integration Between Access Control and Business Process Deep Structure

Sérgio Luís Guerreiro
Instituto Superior Técnico, Universidade de Lisboa, Portugal & Instituto de Engenharia de Sistemas e Computadores – Investigação e Desenvolvimento, Portugal

ABSTRACT

Access control models (ACM) offers the guarantee that only the qualified users can gain access to the artifacts contained in business processes. Business processes are designed, implemented, and operated using many industrial standards that challenge the interoperation with access control standards. Enterprise engineering (EE) introduces rigorous capabilities to design and implement the essential concepts related with the dynamic of business processes. ACM deals with the systematic design and implementation of dynamic and static access control concepts to qualify the access of the users to the artifacts. This chapter proposes an ontological integration between EE and ACM concepts in order to enable the discussion of access control in the deep structure of the business processes. ACM integrated with EE allow the run-time qualification of the actors while they perform all the business process steps and not only at invocation time. The proposal encompasses business process designed with DEMO ontology and role-based access control concepts using a mathematical model logic description.

DOI: 10.4018/978-1-7998-2975-1.ch010

1. INTRODUCTION

Authorization is present in every form of information technology today and is concerned with the ways which users can access resources in computer systems or, informally speaking, with *'who can do what'*. Role-based access control (RBAC) (Ravi et al., 1996) models the classical concepts for access control encompassing: *user*, *role*, *permission*, *constraint* and *session*. It represents an evolution from Discretionary Access Control (DAC), Mandatory Access Control (MAC) and other policies due to less provisioning effort needed. In RBAC users are assigned to a role; each role has a set of associated permissions and changing the permissions only affects the users associated with each role. Some well-known RBAC constraints are: separation of duties (SoD), conflict of interest (CoI), delegation of duties (DoD), binding of duties (BoD), history-based separation of duties (HsD) or newly identified constraints in social networks such as context constraints, *e.g.*, (Carminati et al., 2011). RBAC is applicable to organizational silos, however, it is limited to only one kind of organizational artifact at a single time. It is broadly used in a single architectural layer such as applications or databases (Ferraiolo, Kuhn, & Chandramouli, 2007). This limitation is identified as a first problem which demands a large effort to manage the configuration of accesses in business processes performed by an organization over time (Gaaloul, Guerreiro, & Proper, 2014; Guerreiro, Vasconcelos, & Tribolet, 2011).

Besides this limitation, Hung and Karlapalem (2003) point to the problem of how to enforce access control in workflow management systems (WfMS). To solve this problem, Bertino, Ferrari, and Atluri (1999, 1997) propose a combination of static, dynamic and hybrid constraints to split the enforcement of RBAC in WfMS: the static constraints are processed offline and the dynamic constraints requires an execution engine to monitor the workflow sessions. It is a computationally intensive process that cannot get in the way with the WfMS execution itself.

Developments in attribute-based access control (ABAC) approaches such as the one proposed by Wang, Wijesekera, and Jajodia (2004) extend the RBAC capability, granting access to the artifacts based on the attributes possessed by the requester. The advantage when compared with RBAC is that management of the roles and permissions is not needed. The access configuration is specified by a set of rules that uses the attributes that are issued in the client request. A change in the access policy is performed by a simple change in the rules. In this context, Kuhn, Coyne, and Weil (2010) add that ABAC approach is still evolving from the previous definition of RBAC and not emerging from previous unknown. Moreover, the distributed nature of the web services and the service-oriented architecture (SOA) environments are identified as a potential application area for the development of the ABAC concepts (Shen & Hong, 2006).

In general, the solutions offered by access control models (ACM) decouple the access control concepts from the organization artifacts. The presented solutions usually state that assets exist within the organization, without specifying which kind of assets are and how to integrate with them. The current solutions are mainly concerned with a specific enterprise architectural layer (usually the application layer) and do not satisfy the concerns related to improvements in cyber security as stated, for instance, by the U.S. Department of Homeland Security (DHS, 2019), or the promotion and research of the norms and procedures to protect classified issues as defined in the European community space by the European Network and Information Security Agency (ENISA, 2019). Moreover, regarding the application domains for organization control, security has been a major concern in the enterprise's agendas. (NIST, 2019) reports that future work to be done in the scope of information technology security refers to a special concern that is given to the conceptualization: *'Therefore, there is a need for a security ontology that can clearly define security related concepts and their relationships, and which can then be used to do quantitative risk analysis for enterprise information systems'*. In line with this point of view, (Barker, 2009) emphasizes on the need for the definition of a meta-model for access control rather than specifying multiple instances. The goal is to identify the basic concepts and thus leading to a set of concepts that could be used by all the ACM community. This proposal is based in the categories theory to define the concepts and uses first order logic to relate the concepts. In this line of reasoning, Guerreiro (2018) identifies the challenges for the access control meta modelling design. Other insight that ontological approaches for the enforcement of security in organizations are needed is presented in (Parkin, Moorsel, & Coles, 2009). These authors propose an ontology-based approach to comply the behaviors of the organizational actors with the external standards that are being pushed in order to guarantee the effectiveness of information security measures. A practical result of this ontological study is to identify the actors' behavior, and therefore to align it with the security management policies that are established within the scope of the organization. A password policy is demonstrated.

To deal with the above described problem, this paper grounds a solution in enterprise engineering (EE) concept and integrate them with ACM. An enterprise is a large, complex and dynamic system whose dynamics encompasses both the business processes and the organizational structure's systemic definition. From this point of view, to deal properly with the systemic definition of an enterprise, scientific foundations with new theories and methodologies that enable the understanding, design and implementation of the organization are needed. The foundational concepts from RBAC, which are defined and used successfully for years, are applied in this paper to control the run-time execution of business processes specified by EE. In specific, DEMO theory and methodology contained in Enterprise Ontology (EO) (Dietz, 2006; 2017) is used.

This paper is organized as follow. Section 2 presents the related work with access control enforcement in organizations. Section 3 introduces how business processes are ontological described in DEMO ontology. Then, section 4 integrates the ground concepts from RBAC and DEMO business processes using a S5 modal logic description. Section 5 indicates the contributions achieved in this paper. Finally, section 6 concludes the paper.

2. RELATED WORK

This section summarizes state of the art work related with the enforcement of access control in organizations.

Park et al. (2004) proposes to unify two RBAC layers by separating the organizational roles and the system roles. OR-SRA (Organizational Role System Role Assignment) maps the Organizational level and the System level separately, using the concepts of RBAC: constraints, roles, permissions, sessions and hierarchy. A connection is established between the Organizational and the System level using a predefined set of relations mappings between the roles, requiring an extra effort of the role engineering (Atluri, 2008). However, the main focus of this proposal is in the role models but does not contribute to the enforcement of the model itself.

An alternative solution is proposed by Kang, Park, and Froscher (2001) under the scope of inter-enterprises business processes execution using a central system to enforce the domain of each remote service invocation between organizations rather than a distributed domain provider within each organization. Each enterprise has its own RBAC and the role domain of each one is passed through the communications. The workflow execution also needs to define the concept of a roleDomain for each task. The authors argue that their approach *(i)* separates the application-level from the organization-level, *(ii)* achieve a fine-grained control and *(iii)* supports dynamic constraints. However, this proposal does not encompass the time concerns neither the roles between different models.

Zhang (2008); Zhixiong, Xinwen, and Sandhu (2006) extends the RBAC standard (Ravi et al., 1996) with the concepts of Organization and Asset, justified by the incapacity of the model for the cross-organization RBAC and named it as ROBAC. Here, the aim is to create a ROBAC between organizations that possess different assets. ROBAC represents a deep contribution to the development of the RBAC standard (Ravi et al., 1996), which has a wide potential within the security enforcement of IS yet needs further development regarding the overall enterprise complexity. The roles are mapped with the assets and each asset has its own permission set, but the assets are not further detailed neither the relation between the assets. The cross-organization concern is then performed by the role's application to each one of the previous assigned domains. This proposal also lacks in the assumption that the organizations have comparable roles. To enforce it uses a manifold that implements a virtual organization which refers to the involved assets. By security concerns, in the end it is deleted. The author presents a fully detailed analysis about the different approaches for administering the RBAC implementation, for instance, TMAC (team-base role-based access control), roles templates, organizational units and credential-based access control.

Smith (1997) simplifies ACM enforcement whereas the role of the users is the same used by the technological applications. However, as also stated by Monakova and Schaad (2011), this simplification does not suit real organizational case studies where organizational roles and business processes models are completely separated. Security concerns should be enforced since business process modeling phase.

An approach to apply access control models to individual enterprise architecture (EA) layers of business tasks, information and applications is presented in Guerreiro, Vasconcelos, and Tribolet (2010). Only a local solution is achieved at each EA layer. To obtain global control a set of rules are computed using the logging data from each architecture layer.

3. DEMO BUSINESS TRANSACTIONS

The DEMO theory and methodology for enterprise is proposed by Dietz (2006, 2008). It consists of an essential ontology that is compatible with the communication and production, acts and facts that occur in reality between actors in the different layers of the organization. The DEMO represents a rigorous theory that provides a solid understanding of organizations, based upon the -theory that stands for Fact-information theory (FI), -theory that stands for Technology, Architecture and Ontology (TAO), and -theory, that stands for Performance in Social Interaction (PSI). DEMO is founded on five distinct axioms: *(i)* the Operation Axiom, *(ii)* the Transaction Axiom, *(iii)* the Composition Axiom, *(iv)* the Distinction Axiom, and *(v)* the organization Theorem. World is composed of: *(i)* the enterprise system which includes state space and transition space, and *(ii)* the entire surrounding environment.

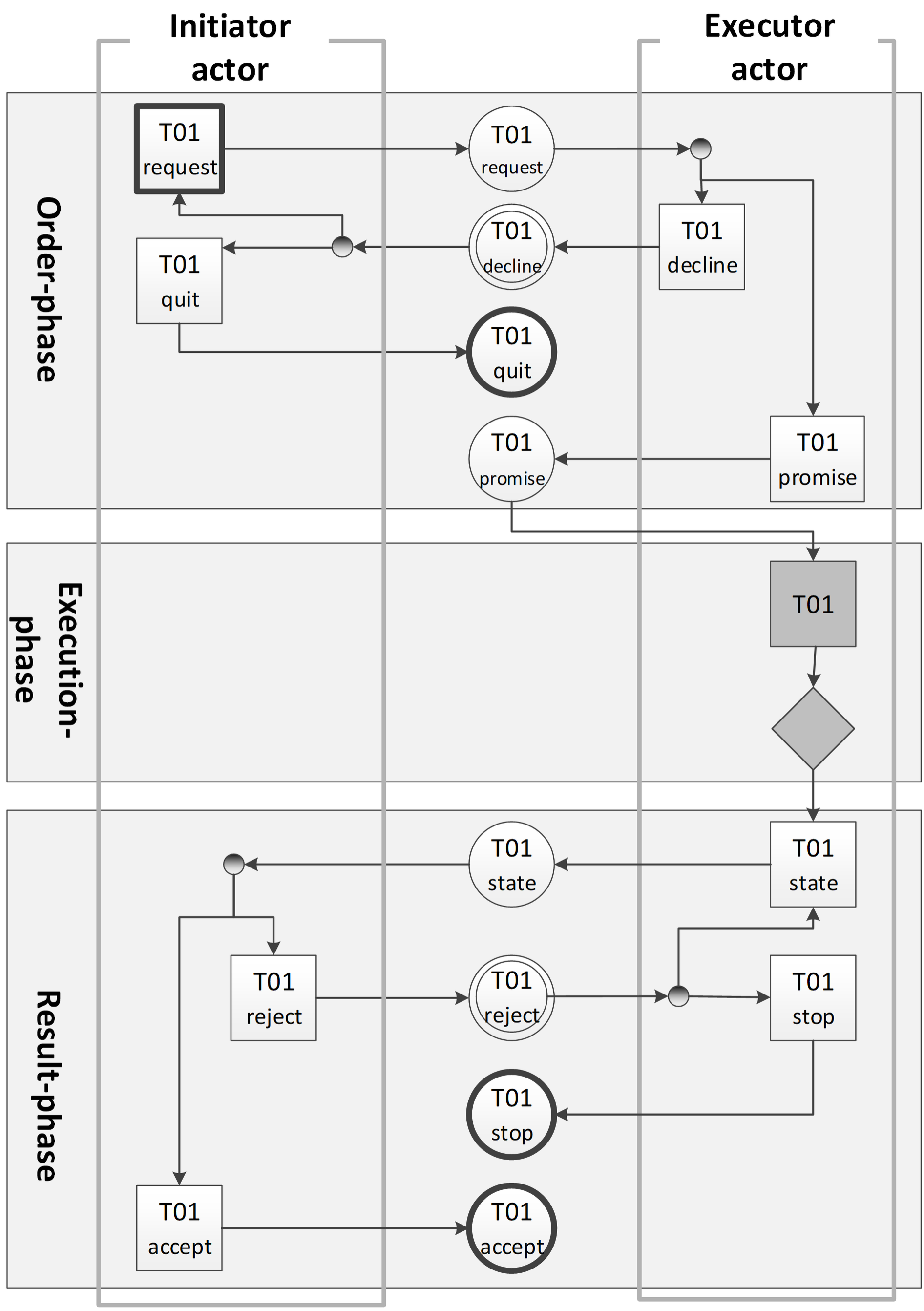
3.1. Transition Space

Focusing in the -theory, the DEMO standard pattern of a transaction is depicted in Figure 1. in this paper a business transaction is considered equivalent to the concept of a business process. However, to avoid semantic manipulation with DEMO concepts from this point forward business transaction is referred. Two distinct actor roles are identified in this transactional pattern: The Customer and the Producer. The goal of performing such a transaction pattern is to obtain a new fact. The transactional pattern is performed by a sequence of coordination and production acts that leads to the production of the new fact, encompassing three phases: *(i)* the order phase (O-phase) that encompasses coordination and production acts of request, promise, decline and quit, *(ii)* the execution phase (E-phase) that includes the production act of execution and the execution of the new fact it self and *(iii)* the result phase (R-phase) that includes coordination and production acts of state, reject, stop and accept.

Firstly, when the Customer desires a new fact then he requests a new fact. Afterward, the fact has been requested a promise to produce the fact is delivered by the Producer. Then, after producing the fact, the Producer states that it has already been finished. Finally, the Customer accepts the new fact produced. DEMO basic transaction pattern is universal in the sense that it summarizes all the coordination and production acts that could occur between persons to persons, machines to machines and also between persons and machines. Sometimes, parts of this transaction pattern are performed tacitly. DEMO basic transaction pattern aims at specifying the transition space of a system that is given by the set of allowable sequences of transitions. Every transition is not only dependent from the previous sequence of transitions but also from the actual and the previous states. The organizational actors are the ones that are responsible for performing the production and/or coordination acts in the scope of a business transaction. A production act (P-act) contributes to bringing the goods and/or services (material or immaterial) that are delivered to the environment of the enterprise, and a coordination act (C-act) enters into and complies with commitments toward each other regarding the performance of the production acts. The P-acts and C-acts have effect in the correspondingly two separate worlds: the production world (P-world) and the coordination world (C-world) (Dietz, 2006). Referring to Figure, coordination acts lies outside the elementary actor roles because communication always happens through a communication channel that exists in the surrounding environment. The most primitive form of communication is verbal speaking. A more technological form is, for instance, an electronic mail message or a web service invocation. When a coordination act is designed with a bold line then it means that is a final coordination act where the transaction pattern ends. When a production act is designed with a bold line then it means that is an initial production act.

DEMO basic transaction pattern is a conversational pattern to be used between actors and not a set of activities within the scope of a business transactions. The DEMO transaction pattern lies in the ontological world; hence its implementation requires more detail and analysis. A single transactional process step, *e.g. Tx/request*,

Figure 1. The DEMO standard pattern of a transaction between 2 actors with separation between communication and production acts (adapted from Dietz, 2006)



would be implemented by a web-based software application (Guerreiro et al., 2012; Dudok et al., 2015).

3.2. State Space

A state space model is not a database relationship model, neither an entity-relationship diagram nor an UML (OMG, 2019) class diagram, but rather a conceptual presentation that would be used to be shared among the enterprise's actors in order to exist an unique conceptual schema of a given world.

By Dietz (2006) state space is understood as the set of allowed or lawful states. It is specified by means of the state base and the existence laws. The state base is the set of facts that are existentially dependent. It is an inherent property of a thing or an inherent relationship between things. The existence laws of facts are timeless. They come into existence dependent on an existentially independent fact existence.

Regarding-theory and narrowing a specific diagram which is the Object Fact Diagram (OFD). It is founded by the Halpin Terry PhD thesis's (Halpin, 1989) and then developed with more contemporary developments within the ORM contribution (Halpin, 1998). Afterward and Dietz (2006, 2005) introduced the WOSL for integrating the state space of the P-world with the transition space. For clarification some WOSL concepts are introduced. A core concept in the state model is the *type* (Dietz, 2006). A type is a generic concept that can be instantiated. Every instantiated object conforms with the type. A fact type represents an elementary state of affairs in a world. All the concepts presented in the state model are fact type, which are then divided in two kinds: intensionally and extensionally. The intensionally term is used when fact type specialization is required. A new type is intensionally defined as a subtype of another type, by adding characteristics. The extensional term is used when fact type generalization is required. A new type is extensionally defined as a supertype of two or more other types, by uniting the constituting types. Figure exemplifies an extensional declaration of a fact type. The unary fact type *Person* is also called a category. An external fact type lays outside the state space of the considered ontology.

Figure 3 presents an intensional declaration regarding an unary, a binary and a ternary fact type. The unary fact type is equivalent to the notation presented in the extensional declaration of Figure 2. X, Y and Z are called roles.

An intensional declaration can also be an unary event type to be used in design of the transition space. Figure 4 exemplifies this possibility.

The notation of the reference law for the unary statum type *b* and for the binary statum type *c* is exemplified in Figure 5. The aim of this concept is to relate extensional and/or intensional declarations. Besides the graphical notation presented herein, the state model could also be expressed using S5 modal logic (Epstein 1990). The

Figure 2. Extensional declaration. Notation of the unary fact type Person and notation of the external unary fact type Year (adapted from Dietz, 2006).

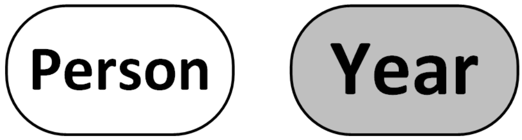


Figure 3. Intensional declaration. Notation of an unary user, a binary owns and a ternary fact type timetable (adapted from Dietz, 2006).

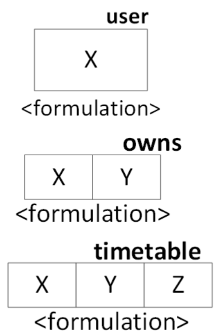
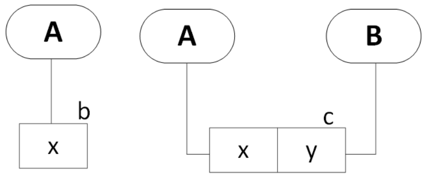


Figure 4. Event type declaration. Notation of an intensional declaration of the unary event type R04 - Session S has been observed (adapted from Dietz, 2006).

following basic relations are recommended: negation () also represented by (), strict implication () also represented by (), and (), or (), it also includes necessity (), possibility (), equivalence (), universal quantifier () and existential quantifier ().

Figure 5. Reference law declaration. In left side: notation of the reference law for the extensional declaration of the unary fact type b. In the right side: notation of the reference law for the extensional declaration of the binary fact type c (adapted from Dietz, 2006).



4. RUN-TIME BUSINESS TRANSACTIONS DYNAMIC ACCESS CONTROL

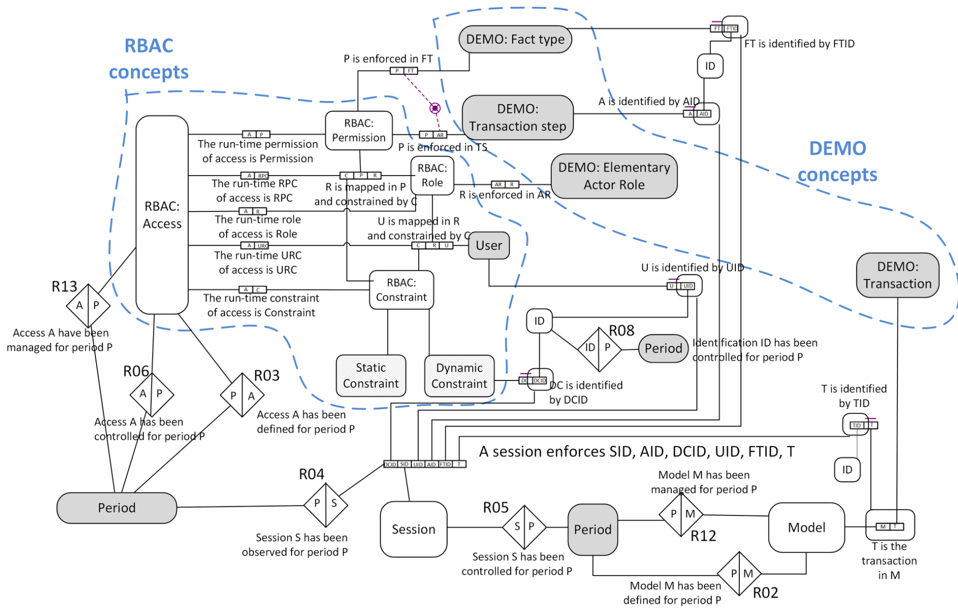
The solution to control the run-time user's access to organizational artifacts within the scope of organizational business transactions is designed in Figure [fig:runtimeaccesscontrol] using a DEMO state space description. During the following textual description, the concepts are written in *italics*. RBAC (Ravi et al., 1996) is modeled by the concepts of: *Permission*, *Role*, *User* and *Constraint*. The relations between them are presented by an equivalent representation in modal logic:

$$
\begin{aligned}
ur\left(u,r,c\right)\text{Þ}User\left(u\right)\,\&\,Role\left(r\right)\,\&\,Constra\,\mathrm{int}\left(c\right)\\
User\left(u\right)\text{ÞÇ}ur\left(u,\text{-},\text{-}\right)\\
Role\left(r\right)\text{ÞÇ}ur\left(\text{-},r,\text{-}\right)\\
Constra\,\mathrm{int}\left(c\right)\text{ÞÇ}ur\left(\text{-},\text{-},c\right)\\
rp\left(r,p,c\right)\text{Þ}Role\left(r\right)\,\&\,Permission\left(p\right)\,\&\,Constra\,\mathrm{int}\left(c\right)\\
Role\left(r\right)\text{ÞÇ}rp\left(r,\text{-},\text{-}\right)\\
Permission\left(p\right)\text{ÞÇ}rp\left(\text{-},p,\text{-}\right)\\
Constra\,\mathrm{int}\left(c\right)\text{ÞÇ}rp\left(\text{-},\text{-},c\right)
\end{aligned}
$$

The *RBAC:Access* is limited to a single *Period* in the time of the organization. Outside the defined *Period* the *RBAC:Access* is no longer valid.

Business transactions are modeled by transition space and state space as in section 3. Each *Transaction step* follows the P-act and C-act performed by actors in the operation of the organization. P-act brings the goods and/or service (material or immaterial), *i.e.*, assembling a package, and a C-act enters into and complies with commitments toward the actors, *i.e.*, send an email or making a phone call to

Figure 6. OFD of the run-time user's access within business transactions. Integration between RBAC and DEMO concepts is represented.



someone. By its turn, each P-act or C-act produces a correspondingly P-fact or C-fact (represented in Figure [fig:runtimeaccesscontrol] by *Fact type*) that are represented in the state model. Thus, the organizational assets that should be considered in the scope of business transactions are *DEMO:Fact type* and *DEMO:Transaction step*. To enforce them with RBAC, we use the Zhang (2008) and Zhixiong, Xinwen, and Sandhu (2006) proposals which is related with the organizational assets. The author states that the organizational assets should be related with the *RBAC:Permission*. A mutual exclusion restriction is also added to guarantee that each *RBAC:Permission* is solely related with a *DEMO:Fact type* or with a *DEMO:Transaction step*. The concept relations are presented by an equivalent representation in modal logic:

$$
\begin{aligned}
&SessionObserved(dcid, sid, uid, aid, ftid, tid) \Rightarrow \\
&\Box Session(sid) \,\&\Box part\_AID(aid) \,\&\Box part\_DCID(dcid) \\
&\&\Box part\_UID(uid) \,\&\Box part\_FTID(ftid) \\
&\&\Box part\_TID(tid) \\
&part\_DCID(dcid) \Rightarrow \Diamond SessionObserved(dcid, -, -, -, -, -) \\
&Session(sid) \Rightarrow \Diamond SessionObserved(-, sid, -, -, -, -) \\
&part\_UID(uid) \Rightarrow \Diamond SessionObserved(-, -, uid, -, -, -) \\
&part\_AID(aid) \Rightarrow \Diamond SessionObserved(-, -, -, aid, -, -) \\
&part\_FTID(ftid) \Rightarrow \Diamond SessionObserved(-, -, -, -, ftid, -) \\
&part\_TID \Rightarrow \Diamond SessionObserved(-, -, -, -, -, tid)
\end{aligned}
$$

$$
\begin{aligned}
&P\_in\_ft(p, ft) \Rightarrow \Box Permission(p) \,\&\Box FactType(ft) \\
&FactType(ft) \Rightarrow \Diamond P\_in\_ft(-, ft) \\
&Permission(p) \Rightarrow \Diamond P\_in\_ft(p, -) \\
&P\_in\_ts(p, ar) \Rightarrow \Box Permission(p) \,\&\Box TxStep(ar) \\
&TxStep(ar) \Rightarrow \Diamond P\_in\_ts(-, ts) \\
&Permission(p) \Rightarrow \Diamond P\_\mathrm{in}\_ts(p, -) \\
&P\_in\_ft(p, -) \Rightarrow \sim \Diamond P\_\mathrm{in}\_ts(p, -) \\
&P\_in\_ts(p-) \Rightarrow \sim \Diamond P\_\mathrm{in}\_ft(p, -)
\end{aligned}
$$

Moreover, each business transaction to be complete requires the specification of an initiator actor and an executor actor, as depicted in Figure 1. *DEMO:Elementary actor role* is the concept used to specify the initiator and the executor, and is directly related with the *RBAC:role*. In this way, there is no need to reinvent the organizational roles since the business transactions modeling in DEMO demands role definition. Hence, our approach reuse the *DEMO:elementary actor role* from the DEMO ontological design. The concept relations are presented by an equivalent representation in modal logic:

$$
\begin{aligned}
&R\_in\_ar\left(ar, r\right) \Rightarrow \Box Role\left(ar\right) \,\&\Box DEMO : Role\left(r\right) \\
&Role\left(ar\right) \Rightarrow \Diamond R\_in\_ar\left(ar, -\right) \\
&DEMO : Role\left(r\right) \Rightarrow \Diamond R\_in\_ar\left(-, r\right)
\end{aligned}
$$

The *Model* of an organization is valid within a given *Period* of time and encompasses all the *DEMO: Transaction* that are performed by the actors. The concept relations are presented by an equivalent representation in modal logic:

$$T\_in\_M(m,t) \Rightarrow \Box Model(m) \,\&\, \Box DEMO: Transaction(t)$$

$$Model(m) \Rightarrow \Diamond T\_in\_M(m,-)$$

$$DEMO: Transaction(t) \Rightarrow \Diamond T\_in\_M(-,t)$$

$$ModelDefinition = \{(Model, DEMO: Transaction) \mid \exists m,t : T\_in\_M(m,t)\}$$

Until this point of the paper, the business transactions prescriptions are enforced with the RBAC prescriptions. Next step is to consider the operation of an organization. In practice, business transactions occur at run-time across a distributed network of actors that constitutes an enterprise (Dietz et al., 2013). Every occurrence of the previously identified concepts must be observed, stored and used to decide upon the next action to be taken by the dynamic access control solution. The former prescriptions are useful to understand what restrictions that a run-time business transaction environment must meet. Our solution proposes to continuously observe the following five concepts: *DEMO: Transaction*, *FactType*, *Transaction step*, *User* and *Dynamic constraint*. Each of these concepts is related with the *ID* concept. The relationships between the concepts are represented by the following set of expressions in modal logic:

$$AID(a, aid) \Rightarrow \Box TxStep(a) \& \Box ID(aid)$$
$$TxStep(a) \Rightarrow \Diamond AID(a-)$$
$$ID(aid) \Diamond AID(-, aid)$$
$$DCID(dc, dcid) \Rightarrow \Box DynamicConstra\,\mathrm{int}(dc) \& \Box ID(dcid)$$
$$DynamicConstra\,\mathrm{int}(sc) \Rightarrow \Diamond DCID(dc, -)$$
$$ID(dcid) \Rightarrow \Diamond DCID(-, dcid)$$
$$UID(u, uid) \Rightarrow \Box User(u) \& \Box ID(uid)$$
$$User(u) \Rightarrow \Diamond UID(u, -)$$
$$ID(uid) \Rightarrow \Diamond UID(-, uid)$$
$$FTID(ft, ftid) \Rightarrow \Box FactType(ft) \& \Box ID(ftid)$$
$$FactType(ft) \Rightarrow \Diamond FTID(ft, -)$$
$$ID(ftid) \Rightarrow \Diamond FTID(-, ftid)$$
$$TID(t, tid) \Rightarrow \Box DEMO : Transaction(t) \& \Box ID(tid)$$
$$DEMO : Transaction(t) \Rightarrow \Diamond \mathrm{T}ID(t, -)$$
$$ID(tid) \Rightarrow \Diamond \mathrm{T}ID(-, tid)$$

In order to unequivocally identify each occurrence of the previous concepts, further modal logic expressions are used to restrict repeated occurrences:

$$\sim \Diamond \left(AID(x, y) \& AID(x, z) \& y \neq z \right)$$
$$\sim \Diamond \left(DCID(x, y) \& DCID(x, z) \& y \neq z \right)$$
$$\sim \Diamond \left(UID(x, y) \& UID(x, z) \& y \neq z \right)$$
$$\sim \Diamond \left(FTID(x, z) \& FTID(y, z) \& x \neq y \right)$$
$$\sim \Diamond \left(TID(x, y) \& TID(x, z) \& y \neq z \right)$$

Then, to optimize the state space for each concept's identification, the following five partitions are created:

$$part\_AID = \{aid \mid \exists a : AID(a, aid)\}$$
$$part\_DCID = \{dcid \mid \exists dc : DCID(dc, dcid)\}$$
$$part\_UID = \{uid \mid \exists u : UID(u, uid)\}$$
$$part\_FTID = \{ftid \mid \exists ft : FTID(ft, ftid)\}$$
$$part\_TID = \{tid \mid \exists t : TID(t, tid)\}$$

Finally, all the concepts previously observed and unequivocally identified are related with *Session* concept. *Session* differs from the Session concept defined in RBAC standard (Ravi et al., 1996) in the sense that it is not only used to relate the User with the Role but used to relate all above partitions. The sum of the above partitions represents the overall picture of DEMO business transaction run-time execution. The *Session* concept is thus represented by the following set of expressions in modal logic:

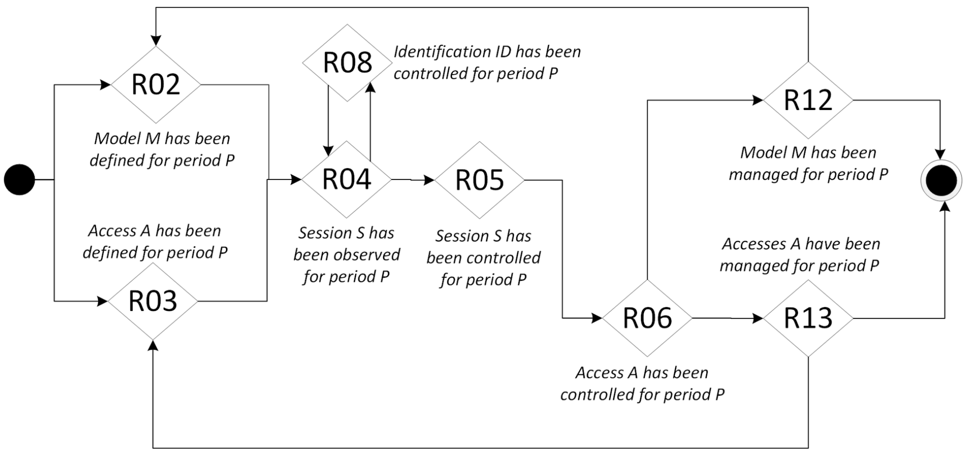
$$SessionObserved(dcid, sid, uid, aid, ftid, tid) \Rightarrow$$
$$\Box Session(sid) \,\&\Box part\_AID(aid) \,\&\Box part\_DCID(dcid)$$
$$\&\Box part\_UID(uid) \,\&\Box part\_FTID(ftid)$$
$$\&\Box part\_TID(tid)$$
$$part\_DCID(dcid) \Rightarrow \Diamond SessionObserved(dcid, -, -, -, -, -)$$
$$Session(sid) \Rightarrow \Diamond SessionObserved(-, sid, -, -, -, -)$$
$$part\_UID(uid) \Rightarrow \Diamond SessionObserved(-, -, uid, -, -, -)$$
$$part\_AID(aid) \Rightarrow \Diamond SessionObserved(-, -, -, aid, -, -)$$
$$part\_FTID(ftid) \Rightarrow \Diamond SessionObserved(-, -, -, -, ftid, -)$$
$$part\_TID \Rightarrow \Diamond SessionObserved(-, -, -, -, -, tid)$$

To finalize Figure 5 explanation, eight event fact types are designed. An event fact type uses the notation shown in Figure 4, and represents the transition space of *P-facts*. By other words, event fact types identify which are the fact types that are involved in each transition. The following sets of event fact types are identified: *(i)* Definition of organizational prescriptions to be followed: R02 (Model) and R03 (Access), *(ii)* Observation of run-time business transactions: R04 (Session), *(iii)* Control of run-time business transactions: R05 (Session), R06 (Access) and R08 (ID), and and *(iv)* Management of run-time business transactions to change prescriptions when needed: R13 (Access), R12 (Model). These event fact types follow the transition state machine presented in Figure 6. This representation offers a view on the sequence of transitions that are required to control the access to business transitions artifacts.

5. CONTRIBUTIONS

This paper uses the body of knowledge taken from ACM integrated with EE concepts to exemplify the enforcement of user's qualification in the scope of business transactions execution. Qualification's enforcement is actually a major concern in

Figure 7. Transition state machine of the dynamic access control enforcement in business transactions. Result types are extracted from OFD in Figure 5.



the scope of the contemporary organizations, that is not fully solved. Considering the EE community, in specific DEMO theory and methodology, a first effort to link the security concern with DEMO is presented by (Marginas, 2006). Author present a manifesto regarding the importance of the enforcement of security in DEMO but does not reveal how can this be achieved. Neither it defines what kind of concepts are needed, nor how to enforce them in the ontology. A second effort in the DEMO community is presented by (Bobbert & Mulder, 2018). However, it does not specify how to enforce a set of posed questions in the DEMO checklist, such as, among others: *Is this business transaction critical?* or *Is there an access control policy and are control methods in place?* The major innovative contribution of this paper is the conceptualization of an ACM approach (namely RBAC) in the enterprise dynamic systems with a focus in the run-time business transactions. Control is presented in this paper as a continuous orchestration of low-level control actions taken by the organizational actors while performing business transactions. As a secondary contribution, this paper addresses the open issues identified in the scope of the ORBAC approaches. An issue that is not addressed in the work of Park et al. (2004) and Zhixiong, Xinwen, and Sandhu (2006) is the capability of encompassing the detailed enterprise ontology artifacts. Role mapping concept in those proposals is essential to enforce the RBAC concepts in real organizations. However, a comprehensive effort to map the roles from the different layers is needed. In a broader discussion, the difficulty that has been identified in the solutions offered by this ACM scientific community is that there exists a decoupling between the access control concepts and the organization artifacts. The presented solutions only state that assets exist in the organization, without specifying which kind of assets or how to integrate them with access control. The actual solutions are strictly concerned with a specific enterprise architectural layer (usually the application layer) and do not satisfy the concerns related to improvements in cyber security as stated, for instance, by the U.S. Department of Homeland Security (DHS, 2019), or the promotion and research of the norms and procedures to protect classified issues as defined in the European community space by the European Network and Information Security Agency (ENISA, 2019).

6. CONCLUSIONS

Access control in the scope of business transactions performed by organizations is demanded to guarantee that only the qualified users are able to gain access to the involved artifacts. ACM deals with the systematic design and implementation of access control concepts and EE deals with the capabilities to design and implement the concepts of business transactions. ACM integrated with EE enables the discussion of access control in the deep structure of the business transactions. This discussion would benefit the organizations that are pursuing an approach to achieve a fine-grained organizational access control model. For instance, to allow run-time qualification of actors while they perform all the different business transactions steps (*e.g.:* all the data store accesses) and not only at invocation of the business transaction (*e.g.:* a web service call). To summarize, we identify that access control ground concepts (using RBAC as an example) should be integrated in the following three ground concepts of business transactions (using DEMO business transaction as an example): *(i)* each *DEMO: Transaction step* defined within the transition space of each *DEMO: Transaction* (following the description in Figure 1) is related with the *RBAC: Permission*, *(ii)* each *DEMO: Fact type* defined within the state space of each *DEMO: Transaction* is related with the *RBAC: Permission*, and finally *(iii)* each *DEMO: Elementary actor role* defined within each *DEMO: Transaction* is related with the *RBAC: Role*. This integration proposal takes benefit of business transactions prescription to provision the RBAC configuration, and thus decreasing the effort required when decoupling the business transactions design process from the RBAC configuration process. Moreover, this approach represents an enforcement based in ground concepts that are developed in the two scientific communities. As actual limitations, we identify *(i)* the lack of automatic tools to support the modeling and control of the run-time business transactions in the scope of software systems and *(ii)* the high volume of data that is intercepted in the *Session*, implementation approaches should be researched. Further work is demanded to implement and test this proposal in more real case studies and with other access control models approaches.

Moreover, the described state solution describes the mandatory concepts to implement the ontology. Also, more research is needed to the solution of the transition space.

REFERENCES

Atluri, V. (2008). Panel on Role Engineering. In *SACMAT '08: Proceedings of the 13th Acm Symposium on Access Control Models and Technologies*, (pp. 61–62). New York, NY: ACM.

Barker, S. (2009). The Next 700 Access Control Models or a Unifying Meta-Model? In *SACMAT '09: Proceedings of the 14th Acm Symposium on Access Control Models and Technologies, (pp. 187–96)*. New York, NY: ACM. doi:10.1145/1542207.1542238

Bertino, E., Ferrari, E., & Atluri, V. (1997). A Flexible Model Supporting the Specification and Enforcement of Role-Based Authorization in Workflow Management Systems. In *RBAC '97: Proceedings of the Second Acm Workshop on Role-Based Access Control*, (pp. 1–12). New York, NY: ACM.

Bertino, E., Ferrari, E., & Atluri, V. (1999). The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Transactions on Information and System Security*, *2*(1), 65–104. doi:10.1145/300830.300837

Bobbert, Y., & Mulder, H. (2018). Enterprise Engineering in Business Information Security. *Enterprise Engineering Working Conference*, 88–112.

Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., & Thuraisingham, B. (2011). Semantic Web-Based Social Network Access Control. *Computers & Security*, *30*(2-3), 108–115. doi:10.1016/j.cose.2010.08.003

DHS. (2019). *Department of Homeland Security Strategic Plan Fiscal Years 2008-2013*. Homeland Security. Retrieved from http://www.dhs.org

Dietz, J. L. G. (2005). A World Ontology Specification Language. In On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. Springer. doi:10.1007/11575863_88

Dietz, J. L. G., Hoogervorst, J. A. P., Albani, A., Aveiro, D., Babkin, E., Barjis, J., ... Winter, R. (2013). The Discipline of Enterprise Engineering. *International Journal of Organisational Design and Engineering*, *3*(1), 86–114. doi:10.1504/IJODE.2013.053669

Dudok, E., Guerreiro, S., Babkin, E., & Pergl, R. (2015). Enterprise Operational Analysis Using Demo and the Enterprise Operating System. In *Enterprise Engineering Working Conference*, (pp. 3–18). Springer. 10.1007/978-3-319-19297-0_1

ENISA. (2019). *European Network and Information Security Agency*. Retrieved from http://www.enisa.europa.eu/

Epstein, R. L. (1990). The Semantic Foundations of Logic Volume 1: Propositional Logics. Kluwer Academic Publishers. doi:10.1007/978-94-009-0525-2

Ferraiolo, D., Kuhn, D., & Chandramouli, R. (2007). *Role-Based Access Control* (2nd ed.). Artech House.

Gaaloul, Guerreiro, & Proper. (2014). Modeling Access Control Transactions in Enterprise Architecture. *2014 IEEE 16th Conference on Business Informatics*, *1*, 127–34.

Guerreiro, S. (2018). Challenges of Meta Access Control Model Enforcement to an Increased Interoperability. In *Encyclopedia of Information Science and Technology* (4th ed.; pp. 651–661). IGI Global. doi:10.4018/978-1-5225-2255-3.ch056

Guerreiro, S., Van Kervel, S., Vasconcelos, A., & Tribolet, J. (2012). Executing Enterprise Dynamic Systems Control with the Demo Processor: The Business Transactions Transition Space Validation. In *7th Mediterranean Conference on Information Systems*, (pp. 97–112). Springer-Verlag Berlin Heidelberg. 10.1007/978-3-642-33244-9_7

Guerreiro, S., Vasconcelos, A., & Tribolet, J. (2010). Adaptive Access Control Modes Enforcement in Organizations. In ENTERprise Information Systems. Springer Berlin Heidelberg. doi:10.1007/978-3-642-16419-4_29

Halpin, T. (1998). Handbook on Architectures of Information Systems. In P. Bernus, K. Mertins, & G. Schmidt (Eds.), *Springer-Verlag*. Berlin.

Hung, P. C. K., & Karlapalem, K. (2003). A Secure Workflow Model. In *ACSW Frontiers '03: Proceedings of the Australasian Information Security Workshop Conference on Acsw Frontiers 2003*, (pp. 33–41). Darlinghurst, Australia: Australian Computer Society, Inc.

Kang, M. H., Park, J. S., & Froscher, J. N. (2001). Access Control Mechanisms for Inter-Organizational Workflow. In *SACMAT '01: Proceedings of the Sixth Acm Symposium on Access Control Models and Technologies*, (pp. 66–74). New York, NY: ACM.

Kuhn, D. R., Coyne, E. J., & Weil, T. R. (2010). Adding Attributes to Role-Based Access Control. *Computer*, *43*(6), 79–81. doi:10.1109/MC.2010.155

Marginas, I. (2006). *DEMO and Security, an Investigation of the Connections Between Security and the Design & Engineering Methodology for Organizations (Demo)* (Master's thesis). Delft: Delft University of Technology, Faculty of Electrical Engineering, Mathematics and, Computer Science, Information Architecture Group.

Monakova, G., & Schaad, A. (2011). Visualizing Security in Business Processes. In *Proceedings of the 16th Acm Symposium on Access Control Models and Technologies*. New York, NY: ACM. doi:10.1145/1998441.1998465

NIST. (2019). *2010 Annual Report, Computer Security Division. National Institute of Standards and Technology (Nist)*. Retrieved from http://csrc.nist.gov/publications/nistir/ir7751/nistir-7751_2010-csd-annual-report.pdf

OMG. (2019). *Object Management Group*. Retrieved from Http://Www.omg.com

Park, J. S., Costello, K. P., Neven, T. M., & Diosomito, J. A. (2004). A Composite Rbac Approach for Large, Complex Organizations. In *SACMAT '04: Proceedings of the Ninth Acm Symposium on Access Control Models and Technologies*. New York, NY: ACM. doi:10.1145/990036.990063

Parkin, S. E., van Moorsel, A., & Coles, R. (2009). An Information Security Ontology Incorporating Human-Behavioural Implications. In *Proceedings of the 2nd International Conference on Security of Information and Networks*, (pp. 46–55). New York, NY: ACM.

Ravi, Edward, Hal, & Charles. (1996). *Role-Based Access Control Models*. IEEE Press. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.3867

Shen & Hong. (2006). An Attribute-Based Access Control Model for Web Services. *Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on*, 74–79. . doi:10.1109/PDCAT.2006.28

Smith, C. (1997). A Survey to Determine Federal Agency Needs for a Role-Based Access Control Security Product. *Software Engineering Standards, International Symposium on*, 222.

Wang, L., Wijesekera, D., & Jajodia, S. (2004). A Logic-Based Framework for Attribute Based Access Control. In *Proceedings of the 2004 Acm Workshop on Formal Methods in Security Engineering*, (pp. 45–55). New York, NY: ACM.

Zhang, Z. (2008). *Scalable Role & Organization Based Access Control and Its Administration* (PhD thesis). Fairfax, VA: George Mason University.

Zhixiong, Z., Zhang, X., & Sandhu, R. (2006). ROBAC: Scalable Role and Organization Based Access Control Models. *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 50.

Chapter 11
Towards Convergence in Information Systems Design

Deepika Prakash
https://orcid.org/0000-0001-8404-3128
NIIT University, India

ABSTRACT

Three technologies—business intelligence, big data, and machine learning—developed independently and address different types of problems. Data warehouses have been used as systems for business intelligence, and NoSQL databases are used for big data. In this chapter, the authors explore the convergence of business intelligence and big data. Traditionally, a data warehouse is implemented on a ROLAP or MOLAP platform. Whereas MOLAP suffers from having propriety architecture, ROLAP suffers from the inherent disadvantages of RDBMS. In order to mitigate the drawbacks of ROLAP, the authors propose implementing a data warehouse on a NoSQL database. They choose Cassandra as their database. For this they start by identifying a generic information model that captures the requirements of the system to-be. They propose mapping rules that map the components of the information model to the Cassandra data model. They finally show a small implementation using an example.

INTRODUCTION

Business Intelligence (BI), Big Data and Machine Learning (ML) are three among the major technological developments in the last 15 years. Business Intelligence encompasses query reporting, data mining in the context of providing decision support. It is based on Data Warehouse (DW) technology. Traditionally, Data

DOI: 10.4018/978-1-7998-2975-1.ch011

Warehouse (DW) star schemas are implemented either using a relational database which allows ROLAP operations or on a multi-dimensional database that allows MOLAP operations. While the data in the former is stored in relational tables, the data in the latter are stored in multidimensional databases (MDB). MDBs use either multi-dimensional array or hypercubes to store this data. A number of RDBMS offer support for building DW systems and for ROLAP queries. MOLAP engines have proprietary architectures. This results in niche servers and is often a disadvantage.

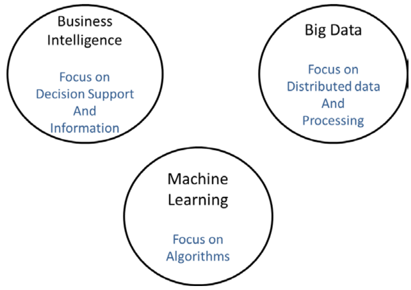
One of the early views of Big Data is that any data satisfying the properties of Velocity, Volume, Variety is big data; this was expanded to include Veracity. Clearly, based on this definition there are two major concerns (a) building a repository for storage of large amounts of data, (b) accommodating a variety of data. To address (a), there was a shift away from vertical scaling to what is called horizontal scaling. Unlike vertical scaling, horizontal scaling is done using commodity machines. Horizontal scaling leads to a repository of data which is distributed across nodes and datacenters. Now, to address (b), variety includes structured, semi-structured and unstructured data. While traditional relational databases are able to store structured data, unstructured data can be stored as a BLOB. The BLOB does not allow full range of querying and processing. Thus, a new model and architecture for databases was required that also provided horizontal scaling. The answer was found in NoSQL databases.

The third technological development is Machine Learning (ML). The area develops and applies algorithms enabling a system to learn. Notice, the system learns by itself without any additional explicit program being written. This may be done through learning patterns or inference rules. The aim of this learning is to gain insights and improve user experience. ML algorithms make no commitment to data storage and management.

If we compare the three technologies from the query viewpoint, we find that BI is oriented to provide business information; Big Data systems improve execution of unstructured and distributed data; and finally ML improves the quality of data in the hands of the user. The first relies on an explicit data storage and architecture of a data warehouse, the second relies on the NoSQL data storage and architecture whereas the ML de-emphasizes the data aspects but deals with the processing aspects almost exclusively. It can be seen that these three technologies reflect the tension between data orientation and process orientation in information systems with BI and Big Data at the data end and ML at the process end.

The three technologies developed independently and at different times (see figure 1): BI was the earliest followed by Big data and ML that were developed almost at the same time. Notice that these three technologies, in so far as they address different domains, are isolated from one another. Yet, there is no reason why these could not benefit from cross fertilization. Indeed there is a case for convergence of these three.

Figure 1. The three technological islands



Notice that ML algorithms depend on "lots of data" to effectively run the algorithms. In fact, they not only need bulk storage of data but also historical data. For example, they may need voice data for the last 4 years for analysis. Further they require a system that enables quick random "reads". Based on the white paper (TDWI, 2018) there are eight requirements for storage of ML data some of which are need for scalability, durability and parallel architecture. Notice, these requirements are satisfied by a Big Data system. The input to an ML algorithm can be unstructured or structured data. Outputs are typically smaller and output storage can be often handled easily.

Some examples of application of ML algorithms to unstructured data are document classification systems where the document is the unstructured data and classification is achieved using Natural Language Processing and ML algorithms. Other ML algorithms are, for example, extracting images from documents and voice based applications. Evidently, it is possible to exploit NoSQL for such data. Where ML deals with structured data, any data set contained in an RDBMS would be acceptable. Most of the traditional ML algorithms can be used here. There is another type of data, semi-structured data, where there are a few semantic tags associated with the data. ML algorithms are applied and any NoSQL database can be used for such storage.

It can also be seen that BI can benefit from both Big Data and ML. While the former can be used to address the issues surrounding diversity and distributed data, using the latter we will be able to produce relatively more intelligent responses to queries. ML has been applied by (Büsch et al., 2017) to classify data warehouse data to support Information Lifecycle Management. Though we do find some applications like the one mentioned here, this does not reflect any definitive trend towards integration of these technologies.

The application of Big Data to BI is also in its nascent stage with only a handful of methodologies being proposed. However, there seems to be two emerging schools of thought for building a DW using Big data and NoSQL databases. One starts from

Table 1. Issues with ROLAP implementation of DW and the solution

DW Issues	ROLAP
Dimensions in Facts may be: • Optional • Not applicable • Unknown	Requires special rows in dimension tables Use of NULL
Varying data types (DW 2.0) • Text, images • Audio, video	Limited
Efficient ETL	Slow because data got into fixed standard form
Distribution	Negative to horizontal scaling

the multi-dimensional model and directly fits facts, measures and dimensions into the NoSQL database. Proposals by (Chevalier et al., 2015; Dehdouh et al., 2015) are examples of this. The second school of thought implements the data warehouse on a NoSQL platform directly from the requirements stage, without arriving at the multi-dimensional model (Prakash, 2019). In spite of the differences, the motivation of using a NoSQL database for both the methodologies lies in overcoming the disadvantages of relational database implementations.

In this chapter we develop a framework to implement a data warehouse using the second school of thought. ***By analogy with ROLAP and MOLAP we refer to the NoSQL approach as NOSOLAP***. We choose Cassandra as our NoSQL database. Out of the four types of NoSQL databases namely column store, document store, key-value pair and graph databases, Cassandra is a column oriented database.

The chapter is organized as follows. In the next section, we discuss our motivation to address convergence of BI and NoSQL databases. Subsequently, we review the existing NOSOLAP techniques. In the next section, we identify our information model from the various DW requirements engineering approaches. Thereafter, we propose mapping rules from the information model to the Cassandra data model. We finally present our conclusion.

MOTIVATION TO CONVERGE BI AND NOSQL DATABASES

As stated in the Introduction, the motivation to implement a data warehouse on a NoSQL platform stems from the disadvantages of a ROLAP implementation. Table 1 below highlights the issues.

Consider the first row of Table 1. Dimensions as implemented in ROLAP are in a 1-M relationship with the fact. It is generally assumed that there will be at least one tuple in the dimension table that can be associated with every fact tuple. However, it may be the case that certain dimensions are *optional* in the fact tuple. Further, it may be the case that some piece of data is not present or in underlying data sources at the time of extraction (ETL). Thus, the dimension is *unknown* to a fact tuple. There may also be dimensions that are *not applicable* to a fact tuple.

As shown in the first row of Table 1, there are two ways of handling this in a relational database engine. One is by adding special rows in the dimension tables with values like -1, 0. The problem with this method is that often only one special row per dimension is created. This makes it impossible to understand the actual reason for the variation. Another solution is to use a NULL 'value'. This causes major difficulties particularly in the use NULL as a dimension value and also as a foreign key value in fact tables. Designers of star schemas have outlined a number of problems associated with these practical solutions to the problem of NULL values. Some of these are, for example, difficulty of forming queries, and misinterpretation of query results. For facts that have NULL values in their foreign keys, introduction of special dimension rows in dimension tables is often proposed as a possible solution to ease the NULL problem.

The second row of Table 1 shows the second issue. DW 2.0 (Inmon, 2010) states, "DW must cater to storing text, audio, video, image and unstructured data "as an essential and granular ingredient". A relational database fails when it comes to unstructured data, video, audio files.

A third issue is that ETL for a relational implementation of a DW is a slow, time consuming process (row 3 of Table 1). This is because data from disparate sources must be converted into one standard structured format of the fact and dimension tables.

Some other minor issues with ROLAP are that the join operations, between facts and dimensions or between normalized dimensions as in the case of snowflake schema, are heavy making the performance of the system slow at query time. Further, relational databases focus on ACID properties Since a DW largely caters to a read-only, analytic environment with changes restricted to ETL time, enforcement of ACID is irrelevant.

We believe NoSQL databases can help mitigate the DW issues identified above. The essential characteristics of such databases are as follows:

1. Flexible data types: They capture multiple data types.
2. Do not have NULL values.
3. Distributed data.
4. Elastically scalable.

Table 2. Mitigation of DW issues by using Cassandra

DW Issues	Cassandra
Dimensions in Facts may be: • Optional • Not applicable • Unknown	All NULLs captured naturally through absence of columns
Varying data types (DW 2.0) • Text, images • Audio, video	Stores unstructured as well as structured data
Efficient ETL	Fast Writes at ETL
Distribution	Highly supportive to horizontal scaling

As stated above, we base this chapter on Cassandra, a blend of Dynamo and BigTable. It gets its distributed feature from Dynamo and provides a peer-to-peer architecture, and handles replication through asynchronous replication.

Table 2 shows the use of Cassandra to resolve DW issues. Recall, dimensions may either be optional, not applicable, unknown as shown in the first row of Table 2. A NULL in Cassandra is captured by the absence of a column. In other words, it completely omits data that is missing. Thus, the problem of NULLS in the data cube is removed.

As mentioned in the Introduction, NoSQL databases were developed to store unstructured and structured data. Thus, text, images, audio, video etc. can be stored in Cassandra thus mitigating the earlier problem.

ETL requires a "write" operation on the database. Cassandra performs fast writes thus making the ETL more efficient. The last major issue of distribution is also resolved using Cassandra as it supports horizontal partitioning.

Further, there were two minor issues pointed out. One was the issue of joins during querying. This is automatically resolved as data modeling in Cassandra is done such that no joins are performed. Second issue was the imposition of ACID properties on a DW making it slow. All NoSQL databases implement BASE which provides the necessary flexibility and, indeed, may lead to better DW performance.

In conclusion, the use of Cassandra helps mitigate the problems encountered with ROLAP.

EXISTING NOSOLAP METHODOLOGIES

As shown in Figure 2 (a), one way to implement DW on NoSQL platform is as a two-step process, where we first arrive at facts and dimensions and then convert the resulting star schema into NoSQL form. The NoSQL databases considered by various authors are column, and document databases. Thus, the figure mentions them as the target database. Let us consider some existing work that follows this two-step process.

(Chevalier et al., 2015) converted the MD schema into a NoSQL column oriented model as well into a document store model. We will consider each of these separately. Regarding the former, each fact is mapped to a column family with measures as the columns in this family. Similarly each dimension is mapped to separate column families with the dimension attributes as columns in the respective column families. All families together make a table which represents one star schema. They used HBase as their column store.

The work of (Dehdouh et al., 2015) is similar to the previous work but they introduce simple and compound attributes into their logical model. (Santos et al., 2017) transforms the MD schema into HIVE tables. Here, each fact is mapped to a primary table with each dimension as a descriptive component of the primary table. The measures and all the non-key attributes of the fact table are mapped to the analytical component of the primary table. As per the query needs and the lattice of cuboids, derived tables are constructed and stored as such.

Now let us consider the conversion of (Chevalier et al., 2015) into document store. Each fact and dimension is a compound attribute with the measures and dimension attributes as simple attributes. A fact instance is a document and the measures are within this document. Mongodb is the document store used.

Since the foregoing proposals start off from a model of facts and dimensions, they suffer from limitations inherent in the fact dimension model. This is shown in Figure 2(a) against the step from star schema to column or document store. Some of these limitations are as follows:

1. Since aggregate functions are not modelled in a star schema, the need for the same does not get translated into the model of the NoSQL database.
2. Features like whether history is to be recorded, what is the categorization of the information required, or whether a report is to be generated are not recorded in a star schema.

Evidently, it would be a good idea to start from a model that makes no commitments to the structural, fact-dimension, aspects of a data warehouse and yet captures the various types of data warehouse contents. This brings us to the second methodology

Figure 2(a). Implementing NOSOLAP as a two-step process

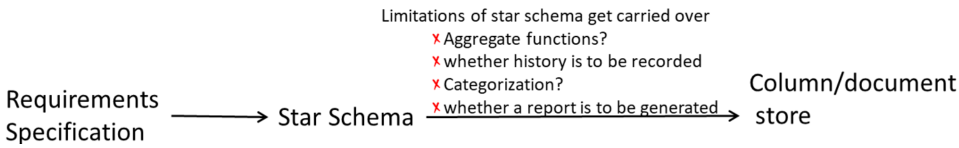
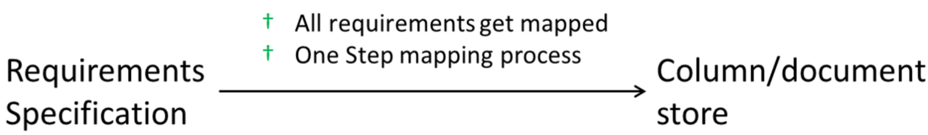


Figure 2(b). Implementing NOSOLAP as a single-step process



of NOSOLAP, that is, *to move from a high-level, generic, information model to NoSQL databases directly, without the intervening star schema being created*. The consequence of this direct conversion is the elimination of the step of converting to a star schema. This is shown below in figure 2(b).

To realize our position, we will look for a DWRE approach that outputs data warehouse contents in a high-level information model that captures in it all the essential informational concepts that go into a data warehouse.

DATA WAREHOUSE REQUIREMENTS ENGINEERING (DWRE) APPROACHES

In the early years of data warehousing, the requirements engineering stage was de-emphasized. Indeed, both the approaches of Inmon (2005) and Kimball (2002) were for data warehouse design and, consequently, do not have an explicit requirements engineering stage. Over the years, however, several DWRE methods have been developed. Today, see Figure. 1, there are three broad strategies for DWRE, goal oriented (GORE) approaches, process oriented (PoRE) techniques and DeRE, decisional requirements engineering.

Some GORE approaches are (Prakash & Gosain, 2003; Prakash et al., 2004; Giorgini et al., 2008). In the approach of (Prakash & Gosain, 2003; Prakash et al., 2004) there are three concepts, goals, decisions and information. Each decision is associated with one or more goals that it fulfils and also associated with information relevant in taking the decision. For this, information scenarios are written. Facts and dimensions are identified from the information scenarios in a two step process.

In GrAND (Giorgini et al., 2008) Actor and Rationale diagrams are made. In the goal modelling stage, goals are identified and facts associated as the recordings that have to be made when the goal is achieved. In the decision-modeling stage, goals of the decision maker are identified and associated with facts. Thereafter dimensions are associated with facts by examining leaf goals.

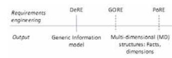
PoRE approaches include Boehnlein and Ulbricht (Boehnlein & Ulbrich-vom, 1999; Boehnlein & Ulbrich-vom, 2000). Their technique is based on the Semantic Object model, SOM, framework. The process involves goal modelling, modelling the business processes that fulfil the goal. Entities of SERM are identified which is then converted to facts and dimensions; facts are determined by asking the question, how can goals be evaluated by metrics? Dimensions are identified from dependencies of the SERM schema.

Another approach is by Bonifati et al. (2001). They carry out goal reduction by using the Goal-Quality-Metric approach. Once goal reduction is done, abstraction sheets are built from which facts and dimensions are identified. In the BEAM* approach (Corr & Stagnitto, 2012) each business event is represented as a table. The attributes of the table are derived by using the 7W framework. 7 questions are asked and answered namely, Who is involved in the event? (2) What did they do? To what is done? (3) When did it happen? (4) Where did it take place? (5) Why did it happen? (6) HoW did it happen – in what manner? (7) HoW many or much was recorded – how can it be measured? Out of these, the first six form dimensions whereas the last one supplies facts.

Whereas both GORE and PORE follow the classical view of a data warehouse as being subject oriented, DeRE takes the decisional perspective. Since the purpose of data warehouse is to support decision-making, this approach makes decision identification the central issue in DWRE. The required information is that which is relevant to the decision at hand. Thus, DeRE builds data warehouse units that cater to specific decisions. Information elicitation is done in DeRE using a multi factor approach (Prakash, 2016; Prakash and Prakash, 2019). For each decision its Ends, Means, and Critical Success Factors are determined and these drive information elicitation

Figure 3 shows the two approaches to representation of the elicited information, the multi-dimensioned and the generic information model approaches. The former lays emphasis on arriving at the facts and dimensions of the DW to-be. The latter, on the other hand, is a high level representation of the elicited information. The dashed vertical lines in the figure show that in both GORE and PoRE the focus of the requirements engineering stage is to arrive at facts and dimensions. On the

Figure 3. DWRE Techniques and their outputs



other hand, the DeRE approach represents the elicited information in a generic information model.

(Prakash & Prakash, 2019) model information as early information. Information is early in the sense that it is not yet structured into a form that can be used in the DW to-be. An instantiation of this model is the identified information contents of the DW to-be.

Since the model is generic, it should be possible to produce logical models of a range of databases from it. This is shown in Figure 4. As shown, this range consists of the multi-dimensional mode and NoSQL data models. Indeed, an algorithmic approach was proposed in (Prakash, 2018) to identify facts, dimensions and dimension hierarchies from the information model.

In accordance with our one-step approach, we will need to reject all Data Warehouse Requirements Engineering, DWRE, techniques that produce facts and dimensions as their output. We propose to use the information model of (Prakash, 2018) to arrive at the logical schema of NoSQL databases.

THE INFORMATION MODEL

The information model of (Prakash & Prakash, 2019), shown in Figure 5, tries to capture all information concepts that is of interest in a data warehouse. Thus, instead of just focusing on facts and dimensions, it details the different kinds of information.

Information can either be detailed, aggregate or historical. Detailed information is information at the lowest grain. Aggregate information is obtained by applying aggregate functions to either detailed or other aggregate information. For example, items that are unshipped is detailed information whereas total revenue lost due to

Figure 4. Converting the information model

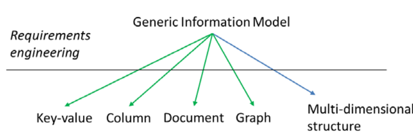
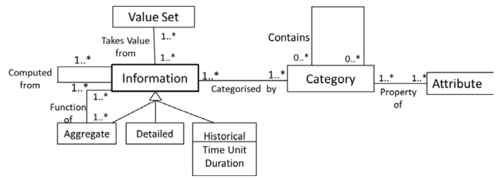


Figure 5. Information model



unshipped items is aggregate information as function sum need to be applied to the revenue lost from each unshipped item. Information can also be historical. Historical information has two components, time unit that states the frequency of capturing the information and duration which states the length of time the data needs to be kept in the data warehouse.

Information can be computed from other detailed, aggregate or historical information. Information is categorized by category which can have one or more attributes. A category can contain zero or more categories. Information takes its values from a value set.

Let us consider an example from the medical domain. The average waiting time of the patients is to be analysed. This is calculated as the time spent by the patient between registration and consultation by a doctor. Let us say that waiting time has to be analysed department wise and on a monthly basis. Historical data of 2 years is to be maintained and data must be captured daily. Instantiation of the information model gives us:

Information: Waiting time of patients
Computed from: Registration time, consultation time
Aggregate functions: Average
Category: Department wise, Month-wise
Category Attributes: Department: code, name
Month: Month
History: Time Unit: Daily
Duration: 2 years

MAPPING RULES

Having described the information model to be used as input to our conversion process, it is now left for us to identify the mapping rules.

Cassandra organizes data into tables with each table having a partitioning and clustering key. The language to query the tables is CQL which has aggregate functions like min, max, sum, average. Cassandra has the possibility of supporting OLAP operations. In this paper, our concern is the conversion to the Cassandra logical model and we do not take a position on OLAP operations. However, we notice that the SLICE OLAP operation can be defined as a set of clustered columns in a partition that can be returned based on the query. Cassandra does indeed support the SLICE operation well. This is because it is very efficient in 'write' operations having its input/output operations as sequential operations. This property can be used for the SLICE operation.

We give a flavour of our proposed rules that convert the information model of the previous section to the logical model of Cassandra. These are as follows:

Rule 1: Each Information, I, of the information model, becomes a table of Cassandra named I. In other words, each instantiation of the information model of figure 5 gives us one table.

Rule 2: Each 'Computed from' Ɛ I is an attribute of the Cassandra table, I.

Rule 3: Each category attribute, attr, of the information model, becomes attributes of Cassandra table, I.

Rule 4: Let us say that category C contains category c. In order to map a 'contains' relationship, two possibilities arise:

a) There is only one attribute of the category c. We propose the use of *set* to capture all the instances of c. The set, s, thus formed is an attribute of Cassandra table, I. For example, consider that a Department contains Units and Units has one attribute unit-name. Set, named say unit_name, will be an attribute of table I. Set unit_name is defined as

unit_name set <text>

where, each name is of data-type text.

The set 'unit_name' will capture all the units under the department.

Notice that since *Set*s are used to store a group of values, it is a suitable data structure for this case.

b) There is more than one attribute of the category c. Since a *Map* is used to store key-value pairs, we propose to use *Map* data structure of Cassandra. The *Map* created will form an attribute of the Cassandra table, I. Consider the same example of a Department contains Units. But this time let Units have attributes, code, name and head. A *Map* will be defined as

units_map <key:value1, key:value2, key:value3>
where the values code, name and head can be stored using data-type int, text and text respectively.

Rule 5: The category of the information model makes up the primary key of table I. Notice, that the model allows more than one category for information, I. Now, the primary key of Cassandra has two components namely, partitioning key and clustering key. Therefore, we need to specify which category maps to which key of Cassandra. There are two possibilities:

a) If there is only one category, then that category is the partitioning key.
b) In the second case, there may be more than one 'category'. Note that the partitioning key is used to identify and distribute data over the various nodes. The clustering key is used to cluster within a node. Thus, the decision on which category or pair of 'category' becomes the partition key and which becomes the clustering key influences the performance of the DW system. We recommend that all the categories should become partitioning keys. However, notice that after selecting a category as the partitioning key, a specific attribute of the category must be selected to designate it as a key. This task will have to be done manually.

For deciding on the clustering key, any attribute of the Cassandra table can be a clustering key. This will have to be determined by the requirements engineer in consultation with domain experts.

There is a special case to (b). There can be certain categories for which the value of their attribute is taken from the system date and time. Such attributes get mapped to Cassandra's timestamp/timeuuid datatype. Assigning such a category/ category attribute as a partitioning key creates as many partitions as the number of dates and time. To minimize the number of partitions we recommend to not use such attributes as the partitioning key. This is because such partitions provide no real value. Instead, we recommend that such a category attribute be assigned as a clustering key.

Applying our broad rules to the example taken in above, we get a table named waiting time. There are two 'computed from' information pieces, registration time and consultation time. Thus, these two become attributes of table waiting time. Further, category attributes: department code and department name belonging to category department also become attributes of the table.

Based on Rule 5, department and month are the partitioning keys. Let us say, department_code attribute represents the department. So, the partitioning key will be (department code, dateMonth). Registration time is the clustering key. The table is created with the CQL statement shown below in Table 1. Notice that there is a column dateMonth that represents month wise category.

Figure 6. The output of applying partition tokens to our data set

department_code	datemonth	registration_time	consultation_time	unit_set	system.token(department_code, datemonth)
Med	201901	1599f7a0-1efa-11e9-8ca9-0fdba4c6d409	2019-01-12 03:48:22.000000+0000	{'U1', 'U2'}	2881630029079461779
Med	201901	216427e0-1efa-11e9-8ca9-0fdba4c6d409	2019-01-12 04:05:02.000000+0000	{'U1', 'U2'}	2881630029079461779
Sur	201901	33e90fc0-1efa-11e9-8ca9-0fdba4c6d409	2019-01-12 04:05:02.000000+0000	{'S1', 'S2', 'S3'}	6975182804555251421
Sur	201901	4317a0b0-1efa-11e9-8ca9-0fdba4c6d409	2019-01-12 04:26:02.000000+0000	{'S1', 'S2', 'S3'}	6975182804555251421

In order to clearly understand the partition based on department code and month of registration, let us look at the partition token obtained when rows are inserted into the table.

The result of running the above statement can be seen in figure 6. For all rows that have the same partition token Cassandra creates one row for each department and each month. So, for example, Medicine department and month of January will be one row. All the associated attributes will be stored as columns of the row.

Now, the aggregate function of the example is to be mapped. Aggregate functions of the information model can be directly mapped to the aggregate functions of Cassandra. In our example, waiting time is calculated as the difference between registration and consultation time. Since there is no difference operator in Cassandra, we wrote a function, datetime_diff, to calculate the same.

Once this was done, the aggregate function, avg, was applied and group by department code and month. Figure 7 shows the CQL code for the same.

Suppose we want information about waiting time but with a different categorization say, patient wise in addition to department wise and month wise. For the information model of figure 5, a completely new piece of information, compared to the earlier one, is generated. Thus, when we map this new information to Cassandra, a new table will be created with its own attributes and partition keys. In other words, each Cassandra table caters to one instantiation of the information model.

A tool to map the information model to Cassandra logical model based on the rules proposed is being developed.

Figure 7. Sample code for creating a user-defined function in CQL

```
CREATE FUNCTION hospital.datetimediff(t1 timestamp, t2 timestamp)
    CALLED ON NULL INPUT
    RETURNS bigint
    LANGUAGE java
    AS $$return (t1.getTime() - t2.getTime())/3600000;$$;
```

CONCLUSION

In our quest for convergence between BI, Big Data and ML, we see that unification of the first two is indeed potentially realizable. Further, if ML can obtain its data from data warehouse or NoSQL data bases then there can be full convergence.

Convergence of BI and Big Data starts off with a recognition of the drawbacks of ROLAP and MOLAP. The use of multi-dimensional database in MOLAP gives high performance but the downside is that thse databases are proprietary databases making a server niche.

ROLAP implements facts and dimensions of a data warehouse as relational tables and consequently suffers from the problems of the relational model being sensitive to NULLs, not catering to all the different types of data that is to be stored, involving heavy join operations which impacts system performance. Additionally, the limitations of RDBMS technology imposes limitations like he difficulty of delivering distributed data bases and horizontal scalability.

Therefore, we have proposed to use a NoSQL database. After all the information needs of the DW to-be have been identified by the requirements engineering phase, the proposed mapping rules to take us to the logical model of NoSQL databases. Our preliminary work is for Cassandra, a column oriented database. Once the mapping is complete, OLAP operations can be performed.

It is well known that the ETL process, while being at the heart of data warehouse delivery is also very complex and difficult to be standardized. Broad approaches and directions for building the ETL process are available and some ETL packages are also available. However, these are difficult to use and adapt to specific situations requiring considerable effort in building the required ETL process. It needs to be seen as to how the ETL process shall be delivered in NOSOLAP systems. Similarly, the manner in which the basic operations of BI like roll-up, drill down etc. will be delivered has not been considered in this chapter.

Yet, we have outlined a basic possibility of convergence of the two technologies in NOSOLAP.

REFERENCES

Boehnlein, M., & Ulbrich-vom Ende, A. 1999, November. Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems. In *Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP* (pp. 15-21). ACM. 10.1145/319757.319780

Böhnlein, M., & Ulbrich-vom Ende, A. (2000). Business process oriented development of data warehouse structures. In *Data Warehousing 2000* (pp. 3–21). Heidelberg, Germany: Physica. doi:10.1007/978-3-642-57681-2_1

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACM Transactions on Software Engineering and Methodology*, *10*(4), 452–483. doi:10.1145/384189.384190

Büsch, S., Nissen, V., & Wünscher, A. (2017). Automatic classification of data-warehouse-data for information lifecycle management using machine learning techniques. *Information Systems Frontiers*, *19*(5), 1085–1099. doi:10.100710796-016-9680-8

Chevalier, M., El Malki, M., Kopliku, A., Teste, O., & Tournier, R. 2015, April. How can we implement a Multidimensional Data Warehouse using NoSQL? In *International Conference on Enterprise Information Systems* (pp. 108-130). Springer. 10.1007/978-3-319-29133-8_6

Corr, L., & Stagnitto, J. (2011). *Agile data warehouse design: Collaborative dimensional modeling, from whiteboard to star schema*. DecisionOne Consulting.

Dehdouh, K., Bentayeb, F., Boussaid, O., & Kabachi, N. 2015, January. Using the column oriented NoSQL model for implementing big data warehouses. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)* (p. 469). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Duda, J. (2012). Business intelligence and NoSQL databases. *Information Systems Management*, *1*(1), 25–37.

Giorgini, P., Rizzi, S., & Garzetti, M. (n.d.). A Goal-Oriented Approach to Requirement Analysis in Data Warehouses. *Decision Support Systems (DSS) Journal,* 4-21.

Inmon, W.H. (1995). What is a data warehouse? *Prism Tech Topic, 1*(1).

Inmon, W. H., Strauss, D., & Neushloss, G. (2010). *DW 2.0: The architecture for the next generation of data warehousing*. Elsevier.

Kimball, R. (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses* (Vol. 1). New York: John Wiley & Sons.

Prakash, D. (2016). Eliciting Information Requirements for DW Systems. In *CAiSE*. Doctoral Consortium.

Prakash, D. (2018, September). Direct Conversion of Early Information to Multi-dimensional Model. In *International Conference on Database and Expert Systems Applications* (pp. 119 126). Springer. 10.1007/978-3-319-98812-2_8

Prakash, D., & Prakash, N. (2019). A multifactor approach for elicitation of Information requirements of data warehouses. *Requirements Engineering*, *24*(1), 103–117. doi:10.100700766-017-0283-9

Prakash, N., & Gosain, A. (2003, June). Requirements Driven Data Warehouse Development. In CAiSE Short Paper Proceedings (Vol. 252). Academic Press.

Prakash, N., Singh, Y., & Gosain, A. (2004, November). Informational scenarios for data warehouse requirements elicitation. In *International Conference on Conceptual Modeling* (pp. 205-216). Springer. 10.1007/978-3-540-30464-7_17

Santos, M. Y., Martinho, B., & Costa, C. (2017). Modelling and implementing big data warehouses for decision support. *Journal of Management Analytics*, *4*(2), 111–129. doi:10.1080/23270012.2017.1304292

Stonebraker, M. (2012). New opportunities for new sql. *Communications of the ACM*, *55*(11), 10–11. doi:10.1145/2366316.2366319

TDWI. (2018). Retrieved from https://tdwi.org/whitepapers/2018/06/adv-all-cloudian-eight-storage-requirements-for-ai-and-machine-learning.aspx

Compilation of References

Abadi, D. J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., & Zdonik, S. (2003). Aurora: A new model and architecture for data stream management. *The VLDB Journal*, *12*(2), 120–139. doi:10.100700778-003-0095-z

Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., & Pinkerton, M. (1997). Cyberguide: A mobile context-aware tour guide. *Wireless Networks*, *3*(5), 421–433. doi:10.1023/A:1019194325861

Adler, J. L., & Blue, V. J. (2002). A cooperative multi-agent transportation management and route guidance system. *Transportation Research Part C, Emerging Technologies*, *10*(5-6), 433–454. doi:10.1016/S0968-090X(02)00030-X

Adler, P. S. (2001). Market, hierarchy, and trust: The knowledge economy and the future of capitalism. *Organization Science*, *12*(2), 215–234. doi:10.1287/orsc.12.2.215.10117

Agrawal, S., & Chaudhuri, S. Narasayya v. (2001). Automated Selection of Materialized Views and Indexes for SQL Databases. *Proceedings of the 26th International Conference on Very Large Databases*.

Agrawal, S., Chaudhuri, S., & Narasayya, V. R. (2001). Materialized View and Index Selection Tool for Microsoft SQL Server 2000. In *Proceedings of the Special Interest Group on Management of Data Conference*. ACM SIGMOND.

Aguilar, W., Santamar'ıa-Bonfil, G., Froese, T., & Gershenson, C. (2014). The past, present, and future of artificial life. *Frontiers in Robotics and AI*, *1*, 8. doi:10.3389/frobt.2014.00008

Ahmed, N., Levorato, M., & Li, G. P. (2018). Residential consumer-centric demand side management. *IEEE Transactions on Smart Grid*, *9*(5), 4513–4524. doi:10.1109/TSG.2017.2661991

Aissaoui, O., Atil, F., & Amirat, A. (2013). Towards a Generic Reconfigurable Framework for Self-adaptation of Distributed Component-Based Application. In Modeling Approaches and Algorithms for Advanced Computer Applications. Springer. doi:10.1007/978-3-319-00560-7_43

Akhavan-Hejazi, H., & Mohsenian-Rad, H. (2018). Power systems big data analytics: An assessment of paradigm shift barriers and prospects. Energy Reports, 4, 91–100. doi:10.1016/j.egyr.2017.11.002

Albiero, F., Fitzek, F. H., & Katz, M. (2007). Cooperative power saving strategies in wireless networks: an agent-based model. In *2007 4th international symposium on wireless communication systems* (pp. 287–291). 10.1109/ISWCS.2007.4392347

Allen, J. F. (1983). Maintaining Knowledge About Temporal Intervals. *ACM-Communiction*, *26*(11), 832–843. doi:10.1145/182.358434

Anderson, J. R. (1996). Act: A simple theory of complex cognition. *The American Psychologist*, *51*(4), 355–365. doi:10.1037/0003-066X.51.4.355

Andrew, C., Broby, D., Paul, G., & Whitfield, I. (2018). *Utilizing financial blockchain technologies in advanced manufacturing*. Glasgow, UK: Strathprints.

Andrighetto, G., Governatori, G., Noriega, P., & van der Torre, L. W. (2013). *Normative multi-agent systems* (Vol. 4). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Angrish, A., Crver, B., Hasan, M., & Starly, B. (2018). A case study for blockchain in manufacturing: FabRec: A prototype for peer to peer network of manufacturing nodes. *North American Manufacturing Research Conference*. 10.1016/j.promfg.2018.07.154

Apajalahti, E., Lovio, R., & Heiskanen, E. (2015). From demand side management (DSM) to energy efficiency services : A Finnish case study. doi:10.1016/j.enpol.2015.02.013

Arasu, A. B., Babu, S., & Widom, J. (2006). The CQL continuous query language: Semantic foundations and query execution. *The VLDB Journal*, *15*(2), 121–142. doi:10.100700778-004-0147-z

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., & Motwani, R. (2003a). Stream: The stanford stream data manager. *IEEE Data Eng. Bull.*, *26*(1), 19–26.

Arasu, A., Babu, S., & Widom, J. (2003b). CQL: A language for continuous queries over streams and relations. In *International Workshop on Database Programming Languages* (pp. 1-19). Springer.

Atluri, V. (2008). Panel on Role Engineering. In *SACMAT '08: Proceedings of the 13th Acm Symposium on Access Control Models and Technologies*, (pp. 61–62). New York, NY: ACM.

Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, *54*(15), 2787–2805. doi:10.1016/j.comnet.2010.05.010

Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002). Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 1-16). ACM. 10.1145/543613.543615

Backman, J., Valtanen, K., Yrjola, S., & Mammela, O. (2017). *Blockchain network slice broker in 5G*. IEEE.

Bagnolli, M., & McKee, M. (1991). Voluntary contribution games: Efficient private provision of public goods. *Economic Enquiry*, *29*(2), 351–366. doi:10.1111/j.1465-7295.1991.tb01276.x

Bahga, A., & Madisetti, V. (2016). Blockchain platform for the industrial Internet of things. *Journal of Software Engineering and Applications*, *9*(10), 533–546. doi:10.4236/jsea.2016.910036

Baldauf, M., Dustdar, S., & Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, *2*(4), 263–277. doi:10.1504/IJAHUC.2007.014070

Baltes, S. (2013). *Study on usage of sketches and diagrams in software engineering practice*. Retrieved from http://empirical-software.engineering/projects/sketches/

Baltes, S., & Diehl, S. (2014). Sketches and Diagrams in Practice. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*,(pp. 530-541). Academic Press.

Baltes, S., & Wagner, S. (2016). Effects of Sketching on Program Comprehension. *Proceedings of the 17th International Conference on Agile Processes in Software Engineering and Extreme Programming*. 10.1007/978-3-319-33515-5_26

Bambara, J., & Allen, P. (2018). *Blockchain: A practical guide to developing business, law and technology solutions*. McGraw Hill.

Barbieri, D. F., Braga, D., Ceri, S., Valle, E. D., & Grossniklaus, M. (2010). Querying rdf streams with c-sparql. *SIGMOD Record*, *39*(1), 20–26. doi:10.1145/1860702.1860705

Baril, X., & Bellahsene, Z. (2003). Selection of materialized views: A cost-based approach. CAiSE, 665–680. doi:10.1007/3-540-45017-3_44

Barker, S. (2009). The Next 700 Access Control Models or a Unifying Meta-Model? In *SACMAT '09: Proceedings of the 14th Acm Symposium on Access Control Models and Technologies, (pp. 187–96)*. New York, NY: ACM. doi:10.1145/1542207.1542238

Barone, D., Jiang, L., Amyot, D., & Mylopoulos, J. (2011). Composite Indicators for Business Intelligence, ER 2011. *LNCS*, *6998*, 448–458.

Bauer, E. (2009). *The Business Rule Approach.* Retrieved from https://pdfs.semanticscholar.org/dbc3/25febb3fa29bbf2bc0b8f7fad08895e46d00.pdf

Bedau, M. A. (2003). Artificial life: Organization, adaptation and complexity from the bottom up. *Trends in Cognitive Sciences*, *7*(11), 505–512. doi:10.1016/j.tics.2003.09.012 PMID:14585448

Bedau, M. A., McCaskill, J. S., Packard, N. H., Rasmussen, S., Adami, C., Green, D. G., ... Ray, T. S. (2000). Open problems in artificial life. *Artificial Life*, *6*(4), 363–376. doi:10.1162/106454600300103683 PMID:11348587

Begoli, E., Camacho-Rodríguez, J., Hyde, J., Mior, M. J., & Lemire, D. (2018). Apache calcite: A foundational framework for optimized query processing over heterogeneous data sources. *Proceedings of the 2018 International Conference on Management*. 10.1145/3183713.3190662

Beni, G. (2004). From swarm intelligence to swarm robotics. In *International workshop on swarm robotics* (pp. 1–9). Academic Press.

Bernardos, A. M., Tarrio, P., & Casar, J. R. (2008). A data fusion framework for context-aware mobile services. In *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*. IEEE. 10.1109/MFI.2008.4648011

Bertino, E., Ferrari, E., & Atluri, V. (1997). A Flexible Model Supporting the Specification and Enforcement of Role-Based Authorization in Workflow Management Systems. In *RBAC '97: Proceedings of the Second Acm Workshop on Role-Based Access Control*, (pp. 1–12). New York, NY: ACM.

Bertino, E., Ferrari, E., & Atluri, V. (1999). The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Transactions on Information and System Security*, *2*(1), 65–104. doi:10.1145/300830.300837

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., & Riboni, D. (2010). A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, *6*(2), 161–180. doi:10.1016/j.pmcj.2009.06.002

Bhandari, G., & Gupta, R. (2017). UML Class Diagram from Object Oriented Program for Reverse Engineering Purpose. In *International Conference on Innovative Entrepreneurship and Startup* (pp.147-150). Academic Press.

Billionniere, E., Greiman, D., & Gosha, K. (2009, December). A comparison of social service selection techniques. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing* (pp. 260-265). IEEE. 10.1109/DASC.2009.24

Bobbert, Y., & Mulder, H. (2018). Enterprise Engineering in Business Information Security. *Enterprise Engineering Working Conference*, 88–112.

Bocek, T., Rodrigues, B., Strasser, T., & Stiller, B. (2017). Blockchains everywhere: A use case of blockchains in the pharma supply chain. *International Symposium on Integrated Network Management*. 10.23919/INM.2017.7987376

Boehnlein, M., & Ulbrich vom Ende, A. (1999). Deriving initial Data Warehouse Structures from the Conceptual Data Models of the Underlying Operational Information Systems. *Proc. Of Workshop on Data Warehousing and OLAP (DOLAP)*. 10.1145/319757.319780

Boella, G., & Lesmo, L. (2002). *A game theoretic approach to norms and agents*. Academic Press.

Böhnlein, M., & Ulbrich-vom Ende, A. (2000). Business process oriented development of data warehouse structures. In *Data Warehousing 2000* (pp. 3–21). Heidelberg, Germany: Physica. doi:10.1007/978-3-642-57681-2_1

Bonabeau, E., Marco, D. R. D. F., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems (No. 1)*. Oxford University Press.

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACM Transactions on Software Engineering and Methodology*, *10*(4), 452–483. doi:10.1145/384189.384190

Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACMTOSEM*, *10*(4), 452–483.

Bresciani, P., Giorgini, P., Mylopoulos, L., & Perini, A. (2004). Tropos: An agent oriented software development Methodology, Journal. *Autonomous Agents and Multi-Agent Systems*, *8*(3), 203–236. doi:10.1023/B:AGNT.0000018806.20944.ef

Brettel, M., Friederichsen, N., Keller, M., & Rosenburg, M. (2014). How virtualization, decentralization and network building change the manufacturing landscape: An Indutsry 4.0 perspective. *International Journal of Information and Communication Engineering.*

Bu¸soniu, L. Babuˇska, R., & De Schutter, B. (2010). Multi-agent reinforcement learning: An overview. In Innovations in multi-agent systems and applications-1 (pp. 183–221). Springer.

Büsch, S., Nissen, V., & Wünscher, A. (2017). Automatic classification of data-warehouse-data for information lifecycle management using machine learning techniques. *Information Systems Frontiers*, *19*(5), 1085–1099. doi:10.100710796-016-9680-8

Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., & Tzoumas, K. (2015). Apache flink: Stream and batch processing in a single engine. *A Quarterly Bulletin of the Computer Society of the IEEE Technical Committee on Data Engineering*, *36*(4).

Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., & Thuraisingham, B. (2011). Semantic Web-Based Social Network Access Control. *Computers & Security*, *30*(2-3), 108–115. doi:10.1016/j.cose.2010.08.003

Castelfranchi, C., Dignum, F., Jonker, C. M., & Treur, J. (1999). Deliberative normative agents: Principles and architecture. In *International workshop on agent theories, architectures, and languages* (pp. 364–378). Academic Press.

Chakraborti, A., Challet, D., Chatterjee, A., Marsili, M., Zhang, Y.-C., & Chakrabarti, B. K. (2015). Statistical mechanics of competitive resource allocation using agent-based models. *Physics Reports*, *552*, 1–25. doi:10.1016/j.physrep.2014.09.006

Chalmeta,R.Grangel,R. (2005)Performance Measurement Systems for Virtual Enterprise Integration. *International Journal of Computer Integrated Manufacturing*, 73-84.

Chandrasekaran, S. C. (2003). Telegraphcq: Continuous dataflow processing for an Uncertain world. In Cidr (Vol. 2, p. 4). Academic Press.

Chaudhuri, S., & Narasayya, V. R. (1998). AutoAdmin 'What-if' Index Analysis Utility. *Proceedings of the Special Interest Group on Management of Data Conference*, 367–378. 10.1145/276304.276337

Chaudhuri, S., Krishnamurthy, R., Potamianos, S., & Shim, K. (1995). Optimizing Queries with Materialized Views. *Proceedings of the 11th International Conference on Data Engineering*, 190–200. 10.1109/ICDE.1995.380392

Chefrour, D. (2005). *Plate-forme de composants logiciels pour la coordination des adaptations multiples en environnement dynamique*. Université Rennes 1.

Chen, S., Shi, R., Ren, Z., Yan, J., Shi, Y., & Zhang, J. (2017). A blockchain based supply chain quality management framework. *IEEE International Conference on e-Business Engineering.* 10.1109/ICEBE.2017.34

Chevalier, M., El Malki, M., Kopliku, A., Teste, O., & Tournier, R. 2015, April. How can we implement a Multidimensional Data Warehouse using NoSQL? In *International Conference on Enterprise Information Systems* (pp. 108-130). Springer. 10.1007/978-3-319-29133-8_6

Choi, C. H., Yu, J. X., & Lu, H. (2004). A simple but effective dynamic materialized view caching. *International Conference on Web-Age Information Management*, 147–156. 10.1007/978-3-540-27772-9_16

Christidis, K., & Devetsikiotis, M. (2016). Blockchains and smart contracts for the Internet of things. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 2292–2303. doi:10.1109/ACCESS.2016.2566339

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation*, *219*(15), 8121–8144. doi:10.1016/j.amc.2013.02.017

Claus, C., & Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI, 1998*(746-752), 2.

Coase, R. (1937). The nature of the firm. *Economica*, *4*(16), 386–405. doi:10.1111/j.1468-0335.1937.tb00002.x

Coase, R. (1960). The problem of social cost. *The Journal of Law & Economics*, *3*, 1–44. doi:10.1086/466560

Colmerauer, A., & Roussel, P. (1996). *History of programming languages II: The birth of Prolog*. New York: ACM.

Computing, A. (2006). An architectural blueprint for autonomic computing. *IBM White Paper, 31*(2006), 1–6.

Conoscenti, M., Vetro, A., & DeMartin, J. (2016). *Blockchain for the Internet of Things: A systemic literature review*. IEEE.

Corr, L., & Stagnitto, J. (2011). *Agile data warehouse design: Collaborative dimensional modeling, from whiteboard to star schema*. DecisionOne Consulting.

Cranor, C. J. (2003). *Gigascope: a stream database for network applications. In 2003 ACM SIGMOD international conference on Management of data* (pp. 647–651). ACM. doi:10.1145/872757.872838

Dayarathna, M., & Perera, S. (2018). *How to Choose a Stream Processor for your App.* Retrieved from www.infoq.com

Dehdouh, K., Bentayeb, F., Boussaid, O., & Kabachi, N. 2015, January. Using the column oriented NoSQL model for implementing big data warehouses. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)* (p. 469). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Deshmukh, J., & Srinivasa, S. (2015). Evolution of cooperation under entrenchment effects. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems* (pp. 1717–1718). Academic Press.

Dey, A. K. (1998). Context-aware computing: The CyberDesk project. *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, *5*(1), 4–7. doi:10.1007007790170019

Dhillon, V., Metcalf, D., & Hooper, M. (2017). *Blockchain enabled applications: Understand the blockchain ecosystem*. Orlando, FL: Springer. doi:10.1007/978-1-4842-3081-7

DHS. (2019). *Department of Homeland Security Strategic Plan Fiscal Years 2008-2013*. Homeland Security. Retrieved from http://www.dhs.org

Di Paolo, E. A. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences*, *4*(4), 429–452. doi:10.100711097-005-9002-y

Diamantoulakis, P. D., Kapinas, V. M., & Karagiannidis, G. K. (2015). Big Data Analytics for Dynamic Energy Management in Smart Grids. Big Data Research, 2(3), 94–101. doi:10.1016/j.bdr.2015.03.003

Dietz, J. L. G. (2005). A World Ontology Specification Language. In On the Move to Meaningful Internet Systems 2005: OTM 2005 Workshops. Springer. doi:10.1007/11575863_88

Dietz, J. L. G., Hoogervorst, J. A. P., Albani, A., Aveiro, D., Babkin, E., Barjis, J., ... Winter, R. (2013). The Discipline of Enterprise Engineering. *International Journal of Organisational Design and Engineering*, *3*(1), 86–114. doi:10.1504/IJODE.2013.053669

Dietz, T., Ostrom, E., & Stern, P. (2003). The struggle to govern the commons. *Science*, *302*(5652), 1907–1912. doi:10.1126cience.1091015 PMID:14671286

Dignum, F., Morley, D., Sonenberg, E. A., & Cavedon, L. (2000). Towards socially sophisticated bdi agents. In *Proceedings fourth international conference on multiagent systems* (pp. 111–118). 10.1109/ICMAS.2000.858442

Dorigo, M., & Di Caro, G. (1999). Ant colony optimization: a new metaheuristic. In *Proceedings of the 1999 congress on evolutionary computationcec99 (cat. no. 99th8406)* (Vol. 2, pp. 1470–1477). 10.1109/CEC.1999.782657

Doyle, J. (1977). *Truth maintenance systems for problem solving* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Duda, J. (2012). Business intelligence and NoSQL databases. *Information Systems Management*, *1*(1), 25–37.

Dudok, E., Guerreiro, S., Babkin, E., & Pergl, R. (2015). Enterprise Operational Analysis Using Demo and the Enterprise Operating System. In *Enterprise Engineering Working Conference*, (pp. 3–18). Springer. 10.1007/978-3-319-19297-0_1

Durugbo, C., & Riedel, J. (2013). Readiness assessment of collaborative network organizations for integrated product service systems. *International Journal of Production Research*, *51*(2), 598–613. doi:10.1080/00207543.2012.658529

E. C. B. (2018). *City of Windhoek electricity tariffs 2017 / 2018*. Author.

Egbert, M. D., & Barandiaran, X. E. (2011). *Quantifying normative behaviour and precariousness in adaptive agency*. Ecal.

Elabd, E. (2015). A dynamic reputation-based approach for web services discovery. *International Journal of Information Technology and Computer Science*, *7*(8), 31–36. doi:10.5815/ijitcs.2015.08.05

El-Kafrawy, P., Elabd, E., & Fathi, H. (2015). A trustworthy reputation approach for web service discovery. *Procedia Computer Science*, *65*, 572–581. doi:10.1016/j.procs.2015.09.001

Elsner, W., Hocker, G., & Schwardt, H. (2010). Simplistic vs complex organization: Markets, hierarchies and networks in an organizational triangle. *Journal of Economic Issues*, *44*(1), 1–30. doi:10.2753/JEI0021-3624440101

ENISA. (2019). *European Network and Information Security Agency*. Retrieved from http://www.enisa.europa.eu/

Epstein, R. L. (1990). The Semantic Foundations of Logic Volume 1: Propositional Logics. Kluwer Academic Publishers. doi:10.1007/978-94-009-0525-2

Ernest, J. F. H. (2008). JESS Rule Engine. Retrieved from http://www.jessrules.com/jess/docs/71

Faber, A., Valente, M., & Janssen, P. (2010). Exploring domestic microcogeneration in the netherlands: An agent-based demand model for technology diffusion. *Energy Policy*, *38*(6), 2763–2775. doi:10.1016/j.enpol.2010.01.008

Fan, S. (2018). A Multiagent Cooperation Model Based on Trust Rating in Dynamic Infinite Interaction Environment. *Mathematical Problems in Engineering*.

Federal Ministry of Economic Affairs and Energy. (2016). *Aspects of the research roadmap in application scenarios*. Berlin: Federal Ministry of Education and Research.

Ferber, J., & Weiss, G. (1999). *Multi-agent systems: an introduction to distributed artificial intelligence* (Vol. 1). Addison-Wesley Reading.

Fernández-Medina, E., Trujillo, J., Villarroel, R., & Piattini, M. (2007). Developing secure data warehouses with a UML extension. *Information Systems*, *32*(6), 826–856. doi:10.1016/j.is.2006.07.003

Ferraiolo, D., Kuhn, D., & Chandramouli, R. (2007). *Role-Based Access Control* (2nd ed.). Artech House.

Ferry, N., Lavirotte, S., Rey, G., & Tigli, J.-Y. (2008). Adaptation Dynamique d'Applications au Contexte en Informatique Ambiante. Rapport Technique, Laboratoire I3S (Universite de Nice Sophia Antipolis/CNRS), numero I3S/RR-2008-20-FR, Sophia Antipolis, France.

Forgy, C. (1982). Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, *19*(1), 17-37.

Frank, U., Heise, D., Kattenstroth, H., & Schauer, H. (2008). Designing and Utilizing Indicator Systems Within Enterprise Models – Outline of a Method. *MobIS*, 89-105.

Franklin, S., & Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International workshop on agent theories, architectures, and languages* (pp. 21–35). Academic Press.

Gaaloul, Guerreiro, & Proper. (2014). Modeling Access Control Transactions in Enterprise Architecture. *2014 IEEE 16th Conference on Business Informatics*, *1*, 127–34.

Gan, C. K. (2012). A review on micro-grid and demand side management and their related standards. *Power and Energy Conversion Symposium (PECS).*

Ganek, A. G., & Corbi, T. A. (2003). The dawning of the autonomic computing era. *IBM Systems Journal*, *42*(1), 5–18. doi:10.1147j.421.0005

Gaweł, B., & Skalna, I. (2012). Model driven architecture and classification of Business rules modelling languages. In *Federated Conference on Conference on Computer Science and Information Systems* (pp. 949-952). Academic Press.

Ghosh, P., Sen, S., & Chaki, N. (2012). Materialized View Construction Using Linear Regression on Attributes. *Third International Conference on Emerging Applications of Information Technology*. 10.1109/EAIT.2012.6407899

Giorgini, P., Rizzi, S., & Garzetti, M. (n.d.). A Goal-Oriented Approach to Requirement Analysis in Data Warehouses. *Decision Support Systems (DSS) Journal,* 4-21.

Giorgini, P., Rizzi, S., & Garzetti, M. (2008). GRAnD: A goal oriented approach to requirement analysis in data warehouses. *Decision Support Systems*, *45*(1), 4–21. doi:10.1016/j.dss.2006.12.001

Giurca, A. (2007). *Using URML and R2ML in a Financial Services Use Case (implemented with JBoss Rules and JenaRules).* Retrieved from http://www.macs.hw.ac.uk/bisel/rewerse/I1/oxygen.informatik.tu-cottbus.de/rewerse-i1/%40q%3Dnode_2f33.html#head-6

Goertzel, B., Looks, M., & Pennachin, C. (2004). Novamente: an integrative architecture for artificial general intelligence. *Proceedings of aaai symposium on achieving human-level intelligence through integrated systems and research.*

Goertzel, B., & Pennachin, C. (2007). The novamente artificial intelligence engine. In *Artificial general intelligence* (pp. 63–129). Springer. doi:10.1007/978-3-540-68677-4_3

Golab, L., & Özsu, M. T. (2003b). Processing sliding window multi-joins in continuous queries over data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29* (pp. 500-511). VLDB Endowment. 10.1016/B978-012722442-8/50051-3

Golab, L., & Ozsu, M. (2005). Update-pattern-aware modeling and processing of continuous queries. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data* (pp. 658-669). ACM. 10.1145/1066157.1066232

Golab, L., & Özsu, M. T. (2003a). Issues in data stream management. *SIGMOD Record*, *32*(2), 5–14. doi:10.1145/776985.776986

Golfarelli, M., & Rizzi, S. (1999). Designing the Data Warehouse: Key Steps and Crucial Issues. *Journal of Computer Science and Information Management, 2*(3).

Goranko, V., & Galton, A. (2015). Temporal Logic. In *The Stanford Encyclopedia of Philosophy.* Retrieved from http://plato.stanford.edu/archives/ sum2015/entries/logic-temporal accessed on 08/012/2016

Gosain, A., & Heena. (2016). Materialized Cube Selection using Particle Swarm Optimization algorithm. *7th International Conference on Communication, Computing and Virtualization*, 79, 2-7. 10.1016/j.procs.2016.03.002

Goswami, R., Bhattacharyya, D. K., & Dutta, M. (2013). Multiobjective Differential Evolution Algorithm Using Binary Encoded Data in Selecting Views for Materializing in Data Warehouse. *LNCS*, *8298*, 95–106.

Gotts, N. M., Polhill, J. G., & Law, A. N. R. (2003). Agent-based simulation in the study of social dilemmas. *Artificial Intelligence Review*, *19*(1), 3–92. doi:10.1023/A:1022120928602

Gou, G. (2006). A*search: An efficient and flexible approach to materialized view selection. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *36*(3), 411–425. doi:10.1109/TSMCC.2004.843248

Gray, J., Layman, A., Bosworth, A., & Pirahesh, H. (1997). Data cube: A relational aggregation operator generalizing group-by, cross-tabs and subtotals. *Data Mining and Knowledge Discovery*, *1*(Issue.1), 29–53. doi:10.1023/A:1009726021843

Grzegorz, J. (2012). UML Representation for Rule-Based Application Models with XTT2-based Business Rules. *International Journal of Software Engineering and Knowledge Engineering*, *22*(04), 485–524. doi:10.1142/S021819401250012X

Guerreiro, S., Van Kervel, S., Vasconcelos, A., & Tribolet, J. (2012). Executing Enterprise Dynamic Systems Control with the Demo Processor: The Business Transactions Transition Space Validation. In *7th Mediterranean Conference on Information Systems*, (pp. 97–112). Springer-Verlag Berlin Heidelberg. 10.1007/978-3-642-33244-9_7

Guerreiro, S., Vasconcelos, A., & Tribolet, J. (2010). Adaptive Access Control Modes Enforcement in Organizations. In ENTERprise Information Systems. Springer Berlin Heidelberg. doi:10.1007/978-3-642-16419-4_29

Guerreiro, S. (2018). Challenges of Meta Access Control Model Enforcement to an Increased Interoperability. In *Encyclopedia of Information Science and Technology* (4th ed.; pp. 651–661). IGI Global. doi:10.4018/978-1-5225-2255-3.ch056

Gunasekaran, A., Subramanian, N., Tiwari, M. K., Majd, E., & Balakrishnan, V. (2016). A reputation-oriented trust model for multi-agent environments. *Industrial Management & Data Systems*.

Gunda, S. e. (2019). *What is Azure Stream Analytics?* Retrieved from https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction

Gupta, H. (1997). Selection of views to materialize in a data warehouse. *Proceedings of the 6th International Conference on Database Theory*, 98-112. 10.1007/3-540-62222-5_39

Gupta, H., Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1997). Index selection for OLAP. *Proceedings of 13th International Conference on Data Engineering*, 208–219. 10.1109/ICDE.1997.581755

Gupta, H., & Mumick, I. S. (1999). Selection of views to materialize under a maintenance cost constraint. *Proceedings of the 7th International Conference on Database Theory*, 453-470. 10.1007/3-540-49257-7_28

Gupta, H., & Mumick, S. (2005). Selection of views to materialize in a data warehouse. *IEEE Transactions on Knowledge and Data Engineering*, *17*(1), 24–43. doi:10.1109/TKDE.2005.16

Gupta, S., & Deep, K. (2019). A novel Random Walk Grey Wolf Optimizer. In *Swarm and Evolutionary Computation* (Vol. 44, pp. 101–112). Elsevier.

Guthikonda, Chitta, Tekawade, & Attavar. (2014). *Comparative analysis of IoT architectures*. TLEN 5710 Capstone, Tech. Rep.

Habiba, D. (2012). *Generating Materialized Views Using Ant Based Approaches And Information Retrieval Technologies*. IEEE.

Hakim, A., Amirat, A., & Oussalah, M. C. (2018). Non-intrusive contextual dynamic reconfiguration of ambient intelligent IoT systems. *Journal of Ambient Intelligence and Humanized Computing*. doi:10.100712652-018-1127-2

Halevy, A. Y. (2001). Answering queries using views: A survey. *The VLDB Journal*, *10*(4), 270–294. doi:10.1007007780100054

Halpin, T. (1998). Handbook on Architectures of Information Systems. In P. Bernus, K. Mertins, & G. Schmidt (Eds.), *Springer-Verlag*. Berlin.

Han, J., & Kamber, M. (2001). *Data Mining: Concepts and Techniques*. San Francisco, CA: Morgan Kaufman.

Harinarayan, V., Rajaraman, A., & Ullman, J. D. (1996). Implementing data cubes efficiently. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 205–216. 10.1145/233269.233333

Hemapala, K. T. M. U., & Kulasekera, A. L. (2012). Demand Side Management for Microgrids through Smart Meters. doi:10.2316/p.2012.768-060

Hendrikx, F., Bubendorfer, K., & Chard, R. (2015). Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, *75*, 184–197. doi:10.1016/j.jpdc.2014.08.004

Hess, C., & Ostrom, E. (2007). Studying the knowledge commons. In *Understanding knowledge as a commons: From theory to practice* (pp. 1–27). Hong Kong: MIT Press.

Hogg, L. M., & Jennings, N. R. (1997). Socially rational agents. In *Proc. aaai fall symposium on socially intelligent agents* (pp. 8–10). Academic Press.

Horng, U. J. T., Chang, J., Lin, B.J., & Kao, C.Y. (1999). Materialized view selection using genetic algorithms in a data warehouse system. CEC 99, *Proceedings of the 1999 Congress on Evolutionary Computation*, 3. 10.1109/CEC.1999.785551

Horng, J. T., Chang, Y. J., & Liu, B. J. (2003). Applying evolutionary algorithms to materialized view selection in a data warehouse. *Soft Computing*, *7*(8), 574–581. doi:10.100700500-002-0243-1

Horrocks, I., Peter, F., Boley, H., Tabet, S., Grosof, S., & Dean, M. (2004). *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved from http://www.w3.org/Submission/SWRL/

Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—Degrees, models, and applications. *ACM Computing Surveys*, *40*(3), 7. doi:10.1145/1380584.1380585

Huhns, M. N., & Bridgeland, D. M. (1991). Multiagent truth maintenance. *IEEE Transactions on Systems, Man, and Cybernetics*, *21*(6), 1437–1445. doi:10.1109/21.135687

Hung, M. C., Huang, M. L., Yang, D. L., & Hsueh, N. L. (2007). Efficient approaches for materialized views selection in a data warehouse. Information Sciences, 177(6), 1333–1348. doi:10.1016/j.ins.2006.09.007

Hung, P. C. K., & Karlapalem, K. (2003). A Secure Workflow Model. In *ACSW Frontiers '03: Proceedings of the Australasian Information Security Workshop Conference on Acsw Frontiers 2003*, (pp. 33–41). Darlinghurst, Australia: Australian Computer Society, Inc.

Husemann, B., Lechtenborger, J., & Vossen, G. (2000). Conceptual Data Warehouse Design. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000).*

Husemann, S., & Schafer, M. (2006). Building Flexible eHealth Processes using Business Rules. In *European Conference on eHealth* (pp. 25-36). Academic Press.

Hutter, M. (2001). Towards a universal theory of artificial intelligence based on algorithmic probability and sequential decisions. In *European conference on machine learning* (pp. 226–238). 10.1007/3-540-44795-4_20

Inmon, W.H. (1995). What is a data warehouse? *Prism Tech Topic, 1*(1).

Inmon, W. H., Strauss, D., & Neushloss, G. (2010). *DW 2.0: The architecture for the next generation of data warehousing*. Elsevier.

Ioana, A., & Ileana, A. (2007). Business Rules – Important Artefacts of Software Development. *Ecological Informatics*, *1-4*, 11–14.

Iqbal, M. H., & Soomro, T. R. (2015). Big data analysis: Apache storm perspective. *International Journal of Computer Trends and Technology*, *19*(1), 9–14. doi:10.14445/22312803/IJCTT-V19P103

Isaac, M., Schmitdz, D., & Walker, M. (1989). The assurance problem in a laboratory market. *Public Choice*, *62*(3), 217–236. doi:10.1007/BF02337743

Jain, R. (2015). Internet of Things: Challenges and Issues. In *20th Annual Conference on Advanced Computing and Communications*. IEEE.

Jain, H., & Gosain, A. (2012). A Comprehensive Study of View Maintenance Approaches in Data Warehousing Evolution. *Software Engineering Notes*, *37*(5), 1–8.

Jayasekara, S., Perera, S., Dayarathna, M., & Suhothayan, S. (2015). Continuous analytics on geospatial data streams with WSO2 complex event processor. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems* (pp. 277-284). ACM. 10.1145/2675743.2772585

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, *1*(1), 7–38. doi:10.1023/A:1010090405266

Johnston, J., Moscovitch, M., & Umilt'a, C. (2008). *The allure of machinic life: Cybernetics, artificial life, and the new ai*. MIT Press. doi:10.7551/mitpress/9780262101264.001.0001

Jung, D. F., & Lake, D. A. (2011). Markets, hierarchies, and networks: An agent based organizational ecology. *American Journal of Political Science*, *55*(4), 972–990. doi:10.1111/j.1540-5907.2011.00536.x

Jurca, R., & Faltings, B. (2002, July). Towards incentive-compatible reputation management. In *Workshop on Deception, Fraud and Trust in Agent Societies* (pp. 138-147). Springer.

Kagermann, H., Wahlster, W., & Helbig, J. (2013). *Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative Industrie 4.0*. Frankfurt: National Academy of Science and Engineering.

Kakousis, K., Paspallis, N., & Papadopoulos, G. A. (2010). A survey of software adaptation in mobile and ubiquitous computing. *Enterprise Information Systems*, *4*(4), 355–389. doi:10.1080/17517575.2010.509814

Kang, H. S., Lee, J. Y., Choi, S., Kim, H., Park, J. H., Son, J. Y., . . . Noh, S. D. (2016). Smart manufacturing: Past research, present findings, and future decisions. *International Journal of Preision Engineering and Manufacturing - Green Technology, 3*(1), 111-128.

Kang, M. H., Park, J. S., & Froscher, J. N. (2001). Access Control Mechanisms for Inter-Organizational Workflow. In *SACMAT '01: Proceedings of the Sixth Acm Symposium on Access Control Models and Technologies*, (pp. 66–74). New York, NY: ACM.

Kant, J.-D., & Thiriot, S. (2006). Modeling one human decision maker with a multi-agent system: the codage approach. In *Proceedings of the fifth international joint conference on autonomous agents and multiagent systems* (pp. 50–57). 10.1145/1160633.1160639

Kaplan, R. (1992). The Balanced Scorecard: Measures that Drive Performance. *Harvard Business Review*, *70*(1), 71–79. PMID:10119714

Kardasis, P. (2001). *Business Rule Modeling* (Ph.D. Thesis). Department of Computation, Manchester, UK.

Kennedy, J. (2006). Swarm intelligence. In *Handbook of nature-inspired and innovative computing* (pp. 187–219). Springer. doi:10.1007/0-387-27705-6_6

Kephart, J. O. (2005). Research challenges of autonomic computing. In *Proceedings of the 27th international conference on software engineering* (pp. 15–22). Academic Press.

Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41–50. doi:10.1109/MC.2003.1160055

Ketfi, A., & Belkhatir, N. (2004). A metamodel-based approach for the dynamic reconfiguration of component-based software. In *International Conference on Software Reuse*. Springer. 10.1007/978-3-540-27799-6_22

Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: the internet of things architecture, possible applications and key challenges. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. IEEE. 10.1109/FIT.2012.53

Khripko, D., Morioka, S. N., Evans, S., Hesselbach, J., & De Carvalho, M. M. (2017). Demand side management within industry : A case study for sustainable business models. *Procedia Manufacturing*, *8*, 270–277. doi:10.1016/j.promfg.2017.02.034

Kiesling, E., Gu¨nther, M., Stummer, C., & Wakolbinger, L. M. (2012). Agentbased simulation of innovation diffusion: A review. *Central European Journal of Operations Research*, *20*(2), 183–230. doi:10.100710100-011-0210-y

Kimball, R. (1996). *The data warehouse toolkit: practical techniques for building dimensional data warehouses* (Vol. 1). New York: John Wiley & Sons.

Kim, S., Lee, K., Cho, J. K., & Kim, C. O. (2011). Agent-based diffusion model for an automobile market with fuzzy topsis-based product adoption process. *Expert Systems with Applications*, *38*(6), 7270–7276. doi:10.1016/j.eswa.2010.12.024

Kingston, J. (1987). Rule-Based Expert Systems and Beyond: An Overview. Project Report, Artificial Intelligence Applications Institute-TR-25.

Kinny, D., Georgeff, M., & Rao, A. (1996). A methodology and modelling technique for systems of bdi agents. In *European workshop on modelling autonomous agents in a multi-agent world* (pp. 56–71). 10.1007/BFb0031846

Kinsman, M., McCready, M., & Krishnamurthy, V. (2016). *Stream Analytics Query Language Reference*. Retrieved from https://docs.microsoft.com/en-us/stream-analytics-query/stream-analytics-query-language-reference

Koh, S. L., & Lim, Y. S. (2016). Evaluating the economic benefits of peak load shifting for building owners and grid operator. Proceedings - 2015 International Conference on Smart Grid and Clean Energy Technologies, ICSGCE 2015, 30–34. 10.1109/ICSGCE.2015.7454265

Komosinski, M., & Adamatzky, A. (2009). *Artificial life models in software*. Springer Science & Business Media. doi:10.1007/978-1-84882-285-6

Kortuem, G., Kawsar, F., Sundramoorthy, V., & Fitton, D. (2010). Smart objects as building blocks for the internet of things. *IEEE Internet Computing*, *14*(1), 44–51. doi:10.1109/MIC.2009.143

Kossmann, D. (2000). The state of the art in distributed query processing. *ACM Computing Surveys*, *32*(4), 422–469. doi:10.1145/371578.371598

Kotidis, Y., & Roussopoulos, N. (1999). Dynamat: a dynamic view management system for data warehouses. In *ACM SIGMOD Record* (Vol. 28, pp. 371–382). ACM. doi:10.1145/304182.304215

Kraus, S. (1997). Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, *94*(1-2), 79–97. doi:10.1016/S0004-3702(97)00025-8

Kuhn, D. R., Coyne, E. J., & Weil, T. R. (2010). Adding Attributes to Role-Based Access Control. *Computer*, *43*(6), 79–81. doi:10.1109/MC.2010.155

Kuhnle, H. (2010). *Distributed manufacturing: Paradigm, concepts, solutions and examples*. London, UK: Springer. doi:10.1007/978-1-84882-707-3

Kumar, T. V., & Ghoshal, A. (2009). A reduced lattice greedy algorithm for selecting materialized views. Information Systems, Technology and Management, 31, 6-18.

Kumar, T. V., & Haider, M. (2012a). Materialized view selection for answering queries. In Data Engineering and Management. Springer Berlin Heidelberg.

Kumar, T. V., & Kumar, S. (2012b). Materialized view selection using simulated annealing. In Big Data Analytics. Springer Berlin Heidelberg.

Kumar, T.V., & Haider, M. (2010). *A Query Answering Greedy Algorithm for Selecting Materialized Views*. Springer-Verlag Berlin Heidelberg.

Kumar, M., Gosain, A., & Singh, Y. (2010). Stakeholders driven requirements engineering approach for data warehouse development. *Journal of Information Processing Systems*, *6*(3), 385–402. doi:10.3745/JIPS.2010.6.3.385

Kumar, T. V. (2015). Materialized View Selection using Improvement based Bee Colony Optimization. Iin. *International Journal of Software Science and Computational Intelligence*, *7*(4), 35–61. doi:10.4018/IJSSCI.2015100103

Kumar, T. V., & Kumar, S. (2014). Materialized view selection using differential evolution. Iin. *International Journal of Innovative Computing and Applications*, *6*(2), 102–113. doi:10.1504/IJICA.2014.066499

L'opez, F. L., Luck, M., & d'Inverno, M. (2006). A normative framework for agent-based systems. *Computational & Mathematical Organization Theory*, *12*(2-3), 227–250. doi:10.100710588-006-9545-7

Laird, J. E. (2008). Extending the soar cognitive architecture. *Frontiers in Artificial Intelligence and Applications*, *171*, 224.

Laird, J. E. (2012). *The soar cognitive architecture*. MIT Press. doi:10.7551/mitpress/7688.001.0001

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., P'erolat, J., . . . Graepel, T. (2017). A unified game-theoretic approach to multiagent reinforcement learning. In Advances in neural information processing systems (pp. 4190–4203). Academic Press.

Langton, C. G. (1997). *Artificial life: An overview*. MIT Press.

Lassila, O., & Swick, R. R. (1998). *Resource description framework (RDF) model and syntax specification.* Retrieved September 11, 2019, from https://www.w3.org/TR/REC-rdf-syntax/

Lawrence, M. (2006). Multiobjective genetic algorithms for materialized view selection in OLAP data warehouses. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 699-706. 10.1145/1143997.1144120

Lawrence, M., & Rau-Chaplin, A. (2006). Dynamic view selection for OLAP. *International Conference on Data Warehousing and Knowledge Discovery*, 33–44. 10.1007/11823728_4

Law, Y. N., Wang, H., & Zaniolo, C. (2011). Relational languages and data models for continuous queries on sequences and data streams. *ACM Transactions on Database Systems*, *36*(2), 8. doi:10.1145/1966385.1966386

Le Goaer, O. (2009). *Styles d'évolution dans les architectures logicielles, Université de Nantes; Ecole Centrale de Nantes (ECN)(ECN)(ECN)*. ECN.

Le Goaer, O., Oussalah, M., Tamzalit, D., & Seriai, A.-D. (2007). *Evolution dirigée par les styles. Atelier RIMEL*. Rétro-Ingénierie, Maintenance et Evolution des Logiciels.

Lee, M., & Hammer, J. (2001). Speeding up materialized viewselection in data warehouses using a randomized algorithm. *International Journal of Cooperative Information Systems*, *10*(3), 327–353. doi:10.1142/S0218843001000370

Lehmann, C., Schock, C., & Kufner, T. (2017). Requirements for blockchain applications in manufacturing small and medium sized enterprizes. *International Conference on Production Research*.

Lerner, A., & Shasha, D. (2003). AQuery: query language for ordered data, optimization techniques, and experiments. In *Proceedings 2003 VLDB Conference* (pp. 345-356). Morgan Kaufmann. 10.1016/B978-012722442-8/50038-0

Li, C., Yu, X., Yu, W., Chen, G., & Wang, J. (2017). Efficient Computation for Sparse Load Shifting in Demand Side Management. *IEEE Transactions on Smart Grid*, *8*(1), 250–261. doi:10.1109/TSG.2016.2521377

Li, J., Li, X., & Juntao, L. V. (2012). Selecting Materialized Views Based on Top-k Query Algorithm for Lineage Tracing. In *Third Global Congress on Intelligent Systems*. IEEE. 10.1109/GCIS.2012.28

Limam, N., & Boutaba, R. (2010). Assessing software service quality and trustworthiness at selection time. *IEEE Transactions on Software Engineering*, *36*(4), 559–574. doi:10.1109/TSE.2010.2

Lin, C. C., Deng, D. J., Liu, W. Y., & Chen, L. (2017). Peak Load Shifting in the Internet of Energy With Energy Trading Among End-Users. *IEEE Access : Practical Innovations, Open Solutions*, *5*(c), 1967–1976. doi:10.1109/ACCESS.2017.2668143

Lin, W. Y., & Kuo, I. C. (2004). A genetic selection algorithm for OLAP data cubes. *Knowledge and Information Systems*, *6*(1), 83–102. doi:10.100710115-003-0093-x

List, B., Schiefer, J., & Min Tjoa, A. (2000). Process-Oriented Requirement Analysis Supporting the Data Warehouse Design Process A Use Case Driven Approach. LNCS, 1873, 593–603.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In Machine learning proceedings 1994 (pp. 157–163). doi:10.1016/B978-1-55860-335-6.50027-1

Liu, J., Jiang, P., & Leng, J. (2017). A framework of credit assurance mechanism for manufacturing services under social manufacturing context. *IEEE Conference on Automation Science and Engineering*. 10.1109/COASE.2017.8256072

Lloyd, J. (1987). Foundations of Logic Programming (2nd ed.). Springer-Verlag. doi:10.1007/978-3-642-83189-8

Lockett, H., Johnson, M., Evans, S., & Bastl, M. (2011). Product service systems and supply network relationships: An explanatory case study. *Journal of Manufacturing Technology Management*, *22*(3), 293–313. doi:10.1108/17410381111112684

Louati, A. (2015). *Une approche multi-agents pour la composition de services Web fondée sur la confiance et les réseaux sociaux* (Thèse de doctorat). Université Paris Dauphine.

Luján-Mora, S., Trujillo Vitae, J., & Song, I. Y. (2006). A UML profile for multidimensional modeling in data warehouses. *Data & Knowledge Engineering*, *59*(3), 725–769. doi:10.1016/j.datak.2005.11.004

Lu, Y., Morris, K., & Frechette, S. (2016). *Current standards landscape for manufacturing systems*. National Institute of Standards and Technology. doi:10.6028/NIST.IR.8107

Lynch, R. L., & Cross, K. F. (1991). *Measure Up!: Yardsticks for Continuous Improvement*. Wiley, Blackwell.

Macal, C. M., & North, M. J. (2005). Tutorial on agent-based modeling and simulation. In *Proceedings of the winter simulation conference*, *2005* (pp. 14–pp). Academic Press.

Madhu Bhan, Kumar, & Rajanikanth. (2013). Materialized view size estimation using sampling. *International Conference on Computational Intelligence and Computing Research*.

Mahmoud, H. Q. (2005). *Getting Started With the Java Rule Engine API (JSR 94): Toward Rule-Based Applications*. Retrieved from https://www.oracle.com/technetwork/articles/java/javarule-139829.html

Mami, I., Coletta, R., & Bellahsene, Z. (2011). Modeling view selection as a constraint satisfaction problem. In *Database and Expert Systems Applications* (Vol. 6861, pp. 396–410). Springer Berlin Heidelberg. doi:10.1007/978-3-642-23091-2_33

Manson, S. M. (2006). Bounded rationality in agent-based models: Experiments with evolutionary programs. *International Journal of Geographical Information Science*, *20*(9), 991–1012. doi:10.1080/13658810600830566

Maps. (2019). Google maps is no longer free. Retrieved from https://www.shoredigitalinc.com/google-maps-is-no-longer-free/

Marginas, I. (2006). *DEMO and Security, an Investigation of the Connections Between Security and the Design & Engineering Methodology for Organizations (Demo)* (Master's thesis). Delft: Delft University of Technology, Faculty of Electrical Engineering, Mathematics and, Computer Science, Information Architecture Group.

Mármol, F. G., & Kuhnen, M. Q. (2015). Reputation-based Web service orchestration in cloud computing: A survey. *Concurrency and Computation*, *27*(9), 2390–2412. doi:10.1002/cpe.3177

Maturana, H. R., & Varela, F. J. (1991). *Autopoiesis and cognition: The realization of the living* (Vol. 42). Springer Science & Business Media.

McAllester, D. A. (1990). *Truth maintenance* (Vol. 90). AAAI.

McMullin, B. (2004). Thirty years of computational autopoiesis: A review. *Artificial Life*, *10*(3), 277–295. doi:10.1162/1064546041255548 PMID:15245628

Mei, S., Zarrabi, N., Lees, M., & Sloot, P. M. (2015). Complex agent networks: An emerging approach for modeling complex systems. *Applied Soft Computing*, *37*, 311–321. doi:10.1016/j.asoc.2015.08.010

Meneguzzi, F., & Luck, M. (2009). Norm-based behaviour modification in bdi agents. In *Proceedings of the 8th international conference on autonomous agents and multiagent systems-volume 1* (pp. 177–184). Academic Press.

Meng, X., & Chen, T. (2013). Event based agreement protocols for multi-agent networks. *Automatica*, *49*(7), 2125–2132. doi:10.1016/j.automatica.2013.03.002

Merkle, D., Middendorf, M., & Schmeck, H. (2002). Ant colony optimization for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation*, *6*(4), 333–346. doi:10.1109/TEVC.2002.802450

Merrick, J., Ye, Y., & Entriken, R. (2017). Assessing the System Value of Optimal Load Shifting. *IEEE Transactions on Smart Grid*, 1–10.

Miller, M. S., & Drexler, E. K. (1988). Markets and computation: Agoric open systems. *The Ecology of Computation,* 133-176.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61. doi:10.1016/j.advengsoft.2013.12.007

Mistry, H., Roy, P., Sudarshan, S., & Ramamritham, K. (2001). Materialized view selection and maintenance using multi-query optimization. *SIGMOD Conference*, 307–318. 10.1145/375663.375703

Monakova, G., & Schaad, A. (2011). Visualizing Security in Business Processes. In *Proceedings of the 16th Acm Symposium on Access Control Models and Technologies*. New York, NY: ACM. doi:10.1145/1998441.1998465

Moreno, A., & Etxeberria, A. (2005). Agency in natural and artificial systems. *Artificial Life*, *11*(1-2), 161–175. doi:10.1162/1064546053278919 PMID:15811225

Morse, S., & Isaac, D. (1998). *Parallel Systems in the Data Warehouse*. Upper Saddle River, NJ: Prentice Hall.

Mukherjee, S., Srinivasa, S., & D, S. C. (2008). Validating for liveness in hidden adversary systems. *Electronic Notes in Theoretical Computer Science*, *203*(3), 53–67. doi:10.1016/j.entcs.2008.04.086

Munshi, A. A., & Mohamed, Y. A. R. I. (2017). Big data framework for analytics in smart grids. *Electric Power Systems Research*, *151*, 369–380. doi:10.1016/j.epsr.2017.06.006

Nadeua, T. P., & Teorey, T. J. (2002). Achieving Scalability in OLAP Materialized View Selection. *DOLAP*, *02*, 28–34. doi:10.1145/583890.583895

Nakamoto, S. (2008). *Bitcoin: A peer to peer electronic cash system.* Retrieved from https://bitcoin.org/bitcoin.pdf

National Academy of Sciences. (2016). Platform Industrie 4.0: Progress report. Berlin: Author.

National Science and Technology Council. (2012). *A national plan for advanced manufacturing*. Washington, DC: Author.

Nedic, A., Ozdaglar, A., & Parrilo, P. A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, *55*(4), 922–938. doi:10.1109/TAC.2010.2041686

NIST. (2019). *2010 Annual Report, Computer Security Division. National Institute of Standards and Technology (Nist)*. Retrieved from http://csrc.nist.gov/publications/nistir/ir7751/nistir-7751_2010-csd-annual-report.pdf

Noghabi, S. A., Paramasivam, K., Pan, Y., & Ramesh, N., B. J., Gupta, I., & C. R. (2017). Samza: stateful scalable stream processing at LinkedIn. *Proceedings of the VLDB Endowment*, *10(12)*, 1634-1645. 10.14778/3137765.3137770

Noor, T. H., Sheng, Q. Z., Zeadally, S., & Yu, J. (2013). Trust management of services in cloud environments: Obstacles and solutions. *ACM Computing Surveys*, *46*(1), 12. doi:10.1145/2522968.2522980

Norvig, P. (1991). *Paradigms of AI Programming: Case Studies in Common Lisp*. San Francisco: Morgan Kaufmann Publishers.

O'Neil, O'Neil, & Chen. (2007). *The star schema benchmark (SSB)' Pat*. Academic Press.

OMG. (2003). *Model Driven Architecture.* Retrieved from http://www.omg.org/mda/specs.htm

OMG. (2008). *Semantics of Business Vocabulary and Business Rules (SBVR) vol. 1.0*. Retrieved from http:// www.omg.org /spec/SBVR/1.0/PDF

OMG. (2011). *Business Motivation Model.* Retrieved from http://www.omg.org/spec/BMM/1.1/

OMG. (2019). *Object Management Group*. Retrieved from Http://Www.omg.com

Organization Structure Metamodel (OSM), 2nd Initial Submission, OMG Document bmi/2006-11-02 (2006)

Ostrom, E. (1990). *Governing the common*. Boston, MA: Cambridge University Press. doi:10.1017/CBO9780511807763

Ostrom, E. (2010). Beyond markets and states: Polycentric governance of complex economic systems. *The American Economic Review*, *100*(3), 641–672. doi:10.1257/aer.100.3.641

Ostrom, E., Gardner, R., & Walker, J. (1992). Covenants with and without a sword: Self governance is possible. *The American Political Science Review*, *86*(2), 404–417. doi:10.2307/1964229

Ostrom, E., Gardner, R., & Walker, J. (1994). *Rules, games and common pool resources*. University of Michigan Press. doi:10.3998/mpub.9739

Oubelli, L.A., & Oussalah, M. (2016). Design and development of business rules management system (BRMS) using AtlandEclipse SIRIUS. *International Journal of Computer Science & Information Technology*, *8*(5), 93-105.

Ozadowicz, A. (2017). A new concept of active demand side management for energy efficient prosumer microgrids with smart building technologies. *Energies*, *10*(11), 1771. doi:10.3390/en10111771

Padhy, N. P. (2002). Demand Side Management and Distribution System Automation : A Case Study on Indian Utility. *National Power Systems Conference (NPSC)*, 308–311.

Pan, X., Han, C. S., & Law, K. H. (2005). A multi-agent based simulation framework for the study of human and social behavior in egress analysis. In Computing in civil engineering (2005) (pp. 1–12). doi:10.1061/40794(179)92

Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, *11*(3), 387–434. doi:10.100710458-005-2631-2

Panigati, E., Schreiber, F. A., & Zaniolo, C. (2015). Data streams and data stream management systems and languages. Data Management in Pervasive Systems, 93-111.

Pan, X., Han, C. S., Dauber, K., & Law, K. H. (2007). A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *AI & Society*, *22*(2), 113–132. doi:10.100700146-007-0126-1

Parker, D. C., Manson, S. M., Janssen, M. A., Hoffmann, M. J., & Deadman, P. (2003). Multi-agent systems for the simulation of land-use and land-cover change: A review. *Annals of the Association of American Geographers*, *93*(2), 314–337. doi:10.1111/1467-8306.9302004

Parkin, S. E., van Moorsel, A., & Coles, R. (2009). An Information Security Ontology Incorporating Human-Behavioural Implications. In *Proceedings of the 2nd International Conference on Security of Information and Networks*, (pp. 46–55). New York, NY: ACM.

Park, J. S., Costello, K. P., Neven, T. M., & Diosomito, J. A. (2004). A Composite Rbac Approach for Large, Complex Organizations. In *SACMAT '04: Proceedings of the Ninth Acm Symposium on Access Control Models and Technologies*. New York, NY: ACM. doi:10.1145/990036.990063

Parsons, S., & Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, *5*(3), 243–254. doi:10.1023/A:1015575522401

Patil, S., Srinivasa, S., & Venkatasubramanian, V. (2009). *Classes of optimal network topologies under multiple efficiency and robustness constraints. In 2009 ieee international conference on systems, man and cybernetics* (pp. 4940–4945). doi:10.1109/ICSMC.2009.5346272

Patil, S., Srinivas, S., Mukherjee, S., Rachakonda, A. R., & Venkatasubramanian, V. (2009). Breeding diameter-optimal topologies for distributed indexes. *Complex Systems*, *18*(2), 175.

Pau, R. (2010). *Business Rules Approach.* Retrieved from, https://sites.google.com/site/raquelpau/Home/businessrules

Perez, L. L., & Jermaine, C. M. (2014). History-aware Query Optimization with Materialized Intermediate Views. *Proceedings of International conference on Data Engineering*. 10.1109/ICDE.2014.6816678

Pfeiffer, S. (2017). The vision of Industrie 4.0 in the making - A case of future told, tamed and traded. *NanoEthics*, *11*(1), 107–121. doi:10.100711569-016-0280-3 PMID:28435474

Pirotte, J. (2014). *Les quatre défis à surmonter pour l'Internet des Objets*. Academic Press.

Poslad, S. (2007, November). Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems*, *2*(4), 15. doi:10.1145/1293731.1293735

Powell, W. W. (1990). Neither market nor hierarchy: Network forms of organization. *Research in Organizational Behavior*, *12*, 295–326.

Prakash, D., & Prakash, N. (2018). A multifactor approach for elicitation of information requirements of data warehouses. *REJ, 24*(1).

Prakash, N., & Bhardwaj, H. (2012). Early Information Requirements Engineering for Target Driven Data Warehouse Development. LNBIP, 134, 188-202.

Prakash, N., & Gosain, A. (2003, June). Requirements Driven Data Warehouse Development. In CAiSE Short Paper Proceedings (Vol. 252). Academic Press.

Prakash, N., Sharma, D. K., & Singh, D. (2014). Business rules for Business Governance. In *16th International conference on Enterprise Information system* (*vol. 3*,pp. 360-367). Proceeding in LNBIP Springer.

Prakash, N., Sharma, D. K., Deepika, & Singh, D. (2013). A Framework for Business Rules. *Proceeding in Springer LNCS*.

Prakash, N., Singh, Y., & Gosain, A. (2004). Informational Scenarios for Data Warehouse Requirements Specification. LNCS, 3288, 205-216.

Prakash, D. (2016). Eliciting Information Requirements for DW Systems. In *CAiSE*. Doctoral Consortium.

Prakash, D. (2018, September). Direct Conversion of Early Information to Multi-dimensional Model. In *International Conference on Database and Expert Systems Applications* (pp. 119 126). Springer. 10.1007/978-3-319-98812-2_8

Prakash, D., & Gupta, D. (2014). Eliciting Data Warehouse Contents for Policy Enforcement Rules. *International Journal of Information System Modeling and Design*, *5*(2), 41–69. doi:10.4018/ijismd.2014040103

Prakash, D., & Prakash, N. (2019). A multifactor approach for elicitation of Information requirements of data warehouses. *Requirements Engineering*, *24*(1), 103–117. doi:10.100700766-017-0283-9

Prakash, N., & Gosain, A. (2008). *An Approach to Engineering the Requirements of Data Warehouses. Requirements Engineering Journal, 13*(1), 49–72.

Prakash, N., Prakash, D., & Sharma, Y. K. (2009). Towards Better Fitting Data Warehouse Systems. In A. Persson & J. Stirna (Eds.), *PoEM 2009. LNBIP* (Vol. 39, pp. 130–144). Heidelberg, Germany: Springer.

Prakash, N., Singh, Y., & Gosain, A. (2004, November). Informational scenarios for data warehouse requirements elicitation. In *International Conference on Conceptual Modeling* (pp. 205-216). Springer. 10.1007/978-3-540-30464-7_17

Prat, N., Akoka, J., & Comyn-Attiau, I. (2006). A UML-based data warehouse design method. *Decision Support Systems*, *42*(3), 1449–1473. doi:10.1016/j.dss.2005.12.001

Prodanuks, T., & Blumberga, D. (2015). Methodology of demand side management Study course. experience of case studies. *Agronomy Research (Tartu)*, *13*(2), 520–525.

Prud'hommeaux, E., & Seaborne, A. (2017). *SPARQL query language for RDF.* W3C Recommendation.

Rackov, M., Kaplar, S., Filipovic, M., & Milosavljevic, M. (2016). Java code generation based on OCL rules. In *6th International Conference on Information Society and Technology* (pp. 191-196). Academic Press.

Raghavan, P., & Henzinger, M. (1999). Computing on data streams. *DIMACS Workshop External Memory and Visualization*, 50, 107.

Rahman, A., Islam, R., Sharif, K. F., & Aziz, T. (2016). Developing Demand Side Management Program for Commercial Customers : A Case Study. 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 1–6. 10.1109/CEEICT.2016.7873061

Rai, V., & Anantaram, C. (2003). Business Rules Approach to Information System Development. *The Pacific Asia Conference on Information Systems*, 123.

Rao, A. S., & Georgeff, M. P. (1991). Modeling rational agents within a bdiarchitecture. *KR*, *91*, 473–484.

Rao, A. S., & Georgeff, M. P. (1995). *Bdi agents: from theory to practice* (Vol. 95). ICMAS.

Ravi, Edward, Hal, & Charles. (1996). *Role-Based Access Control Models*. IEEE Press. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.30.3867

Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 291–319. doi:10.1016/j.jksuci.2016.10.003

Rosenow, J., & Bayer, E. (2016). *Costs and Benefits of Energy Efficiency Obligation Schemes. The Regulatory Assistance Project*. RAP.

Roshan, C., Kinley, W., & Sastry, M. K. S. (2014), Home Energy Audit: A Case Study of Phuentsholing, *Bhutan. IEEE International Conference on Communication Systems and Network Technologies (CSNT)*. DOI: 10.1109/CSNT.2014.208

Ross, R. G. (Ed.). (2003). *Business Rules Manifesto: The Principles of Rule Independence, Version 2.0*. Retrieved from www.BusinessRulesGroup.org

Santos, F. P., Santos, F. C., Melo, F. S., Paiva, A., & Pacheco, J. M. (2016). Dynamics of fairness in groups of autonomous learning agents. In *International conference on autonomous agents and multiagent systems* (pp. 107–126). 10.1007/978-3-319-46882-2_7

Santos, M. Y., Martinho, B., & Costa, C. (2017). Modelling and implementing big data warehouses for decision support. *Journal of Management Analytics*, *4*(2), 111–129. doi:10.1080/23270012.2017.1304292

Sastry, M. K. S. (2007). Integrated Outage Management System: An Effective Solution for Power Utilities to Address Customer Grievances. *International Journal of Electronic Customer Management*, *1*(1), 30–40.

Schaat, S., Wendt, A., Jakubec, M., Gelbard, F., Herret, L., & Dietrich, D. (2014). Ars: An agi agent architecture. AGI.

Schilit, B., Adams, N., & Want, R. (1994). Context-aware computing applications. In *Mobile Computing Systems and Applications, 1994. WMCSA 1994. First Workshop on*. IEEE. 10.1109/WMCSA.1994.16

Schmidhuber, J. (2007). G¨odel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence* (pp. 199–226). Springer. doi:10.1007/978-3-540-68677-4_7

Schreinemachers, P., & Berger, T. (2006). Land use decisions in developing countries and their representation in multi-agent systems. *Journal of Land Use Science*, *1*(1), 29–44. doi:10.1080/17474230600605202

Schuelke-Leech, B. A., Barry, B., Muratori, M., & Yurkovich, B. J. (2015). Big Data issues and opportunities for electric utilities. *Renewable & Sustainable Energy Reviews*, *52*, 937–947. doi:10.1016/j.rser.2015.07.128

Seidl, D. (2004). Luhmann's theory of autopoietic social systems. *LudwigMaximilians-Universit¨at Mu¨nchen-Munich School of Management*, 36–37.

Sellami, M. (2011), *Découverte d'annuaires de services Web dans un environnement distribué* (Thèse de doctorat). Télécom Sud Paris et l'Université d'Evry-Val d'Essonne.

Semsar-Kazerooni, E., & Khorasani, K. (2009). Multi-agent team cooperation: A game theory approach. *Automatica*, *45*(10), 2205–2213. doi:10.1016/j.automatica.2009.06.006

Shackelford, S., & Myers, S. (2017). *Block by block: Leveraging the power of blockchain technology to build trust and promote cyber peace*. *Yale Journal of Law and Technology*.

Shanks, G., Sharma, R., Seddon, P., & Reynolds, P. (2010). Business Analytics and Competitive Advantage: A Review and Research Agenda. *21st Australasia Conference on Information Systems*, 1-11.

Sharma, D.K., Prakash, N., Mahajan, M., & Singh, D. (2017). Design and Implementation of a Software System for High Level Business Rules. *International Journal on Advanced Science, Engineering and Information Technology, 7*(6), 2203-2212.

Sharma, V., & Kumar, M. (2011). Web service discovery research: A study of existing approaches. *Int. Journal on Recent Trends in Engineering & Technology*, *1*, 106–109.

Shen & Hong. (2006). An Attribute-Based Access Control Model for Web Services. *Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on*, 74–79. . doi:10.1109/PDCAT.2006.28

Shipman, R., Gillott, M., & Naghiyev, E. (2013). The Mediterranean Green Energy Forum 2013, MGEF-13 SWITCH : Case studies in the demand side management of washing appliances. *Energy Procedia*, *42*, 153–162. doi:10.1016/j.egypro.2013.11.015

Shoham, Y., Powers, R., & Grenager, T. (2003). Multi-agent reinforcement learning: a critical survey. *Web Manuscript*.

Shyam, R., Ganesh, H. B. B., Kumar, S. S., Poornachandran, P., & Soman, K. P. (2015). Apache Spark a Big Data Analytics Platform for Smart Grid. Procedia Technology, 21, 171–178. doi:10.1016/j.protcy.2015.10.085

Smith, C. (1997). A Survey to Determine Federal Agency Needs for a Role-Based Access Control Security Product. *Software Engineering Standards, International Symposium on*, 222.

Smith, P. (2017). *Stop your (business rules) engines!* Retrieved from https://adhocteam.us/2017/10/09/stop-your-business-rules-engines/

Socha, K., & Dorigo, M. (2008). Ant colony optimization for continuous domains. *European Journal of Operational Research*, *185*(3), 1155–1173. doi:10.1016/j.ejor.2006.06.046

Stonebraker, M. (2012). New opportunities for new sql. *Communications of the ACM*, *55*(11), 10–11. doi:10.1145/2366316.2366319

Stötzer, M., Gronstedt, P., & Task, E. T. G. (2011). Demand side management potential a case study for Germany. *21st International Conference on Electricity Distribution*.

Suhothayan, S., Gajasinghe, K., Loku Narangoda, I., Chaturanga, S., Perera, S., & Nanayakkara, V. (2011). Siddhi: A second look at complex event processing architectures. *Proceedings of the 2011 ACM workshop on Gateway computing environments*. 10.1145/2110486.2110493

Sun, X., & Ziqiang, W. (2009). An Efficient Materialized Views Selection Algorithm Based on PSO. *Proceedings of the International Workshop on Intelligent Systems and Applications*. 10.1109/IWISA.2009.5072711

Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 2). MIT Press.

Talebi, Z. A., Chirkova, R., & Fathi, Y. (2013). An integer programming approach for the view and index selection problem. *Data & Knowledge Engineering, Elsevier*, *83*, 111–125. doi:10.1016/j.datak.2012.11.001

Tamiozzo, A. S., & Ale, J. M. (2014). A solution to the materialized view selection problem in data warehousing. *XX Congreso Argentino de Ciencias de la Computación*.

Tang, F., & Ren, A. (2008). Agent-based evacuation model incorporating fire scene and building geometry. *Tsinghua Science and Technology*, *13*(5), 708–714. doi:10.1016/S1007-0214(08)70115-9

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning* (pp. 330–337). 10.1016/B978-1-55860-307-3.50049-6

TDWI. (2018). Retrieved from https://tdwi.org/whitepapers/2018/06/adv-all-cloudian-eight-storage-requirements-for-ai-and-machine-learning.aspx

Tennenholtz, M. (1999). *Electronic commerce: from economic and gametheoretic models to working protocols*. IJCAI.

Terry, D. G. (1992). *Continuous queries over append-only databases*. Academic Press.

Tian, F. (2017). *A supply chain traceability system for food safety based on HACCP, blockchain and the Internet of things*. Vienna: Academic Press.

Tibermacine, O., Tibermacine, C., & Cherif, F. (2015, June). Regression-based bootstrapping of web service reputation measurement. In *2015 IEEE International Conference on Web Services* (pp. 377-384). IEEE. 10.1109/ICWS.2015.57

Toyoda, K., Mathiopolos, T., Sasase, I., & Ohtsuki, T. (2017). A novel blockhain based product ownership management system for anticounterfeit in the supply chains. *IEEE Access: Practical Innovations, Open Solutions*, *5*, 17465–17477. doi:10.1109/ACCESS.2017.2720760

Tu, C., He, X., Shuai, Z., & Jiang, F. (2017). Big data issues in smart grid – A review. *Renewable & Sustainable Energy Reviews*, *79*, 1099–1107. doi:10.1016/j.rser.2017.05.134

Valluri, S. R., Vadapalli, S., & Karlapalem, K. (2002). View relevance driven materialized view selection in data warehousing environment. *Australasian Database Conference*.

Van der Hoek, W., & Wooldridge, M. (2003). Towards a logic of rational agency. *Logic Journal of the IGPL*, *11*(2), 135–159. doi:10.1093/jigpal/11.2.135

Varela, F. G., Maturana, H. R., & Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *Bio Systems*, *5*(4), 187–196. doi:10.1016/0303-2647(74)90031-8 PMID:4407425

Vidal, J. M., & Durfee, E. H. (1995). *Recursive agent modeling using limited rationality*. ICMAS.

Villez, K., Gupta, A., Venkatasubramanian, V., & Rieger, C. (2011). Resilient design of recharging station networks for electric transportation vehicles. In *2011 4th international symposium on resilient control systems* (pp. 55–60). 10.1109/ISRCS.2011.6016089

Vogel-Houser, B., Kegel, G., Bender, K., & Wucherr, K. (2009). Global information architecture for industrial automation. *Standardisierung,* 108-115.

Von Neumann, J., Morgenstern, O., & Kuhn, H. W. (2007). *Theory of games and economic behavior (commemorative edition)*. Princeton University Press. doi:10.1515/9781400829460

Wang, L., Wijesekera, D., & Jajodia, S. (2004). A Logic-Based Framework for Attribute Based Access Control. In *Proceedings of the 2004 Acm Workshop on Formal Methods in Security Engineering*, (pp. 45–55). New York, NY: ACM.

Wang, S., Huang, L., Hsu, C. H., & Yang, F. (2016). Collaboration reputation for trustworthy Web service selection in social networks. *Journal of Computer and System Sciences*, *82*(1), 130–143. doi:10.1016/j.jcss.2015.06.009

Wang, S., Zheng, Z., Sun, Q., Zou, H., & Yang, F. (2011, July). Evaluating feedback ratings for measuring reputation of web services. In *2011 IEEE International Conference on Services Computing* (pp. 192-199). IEEE. 10.1109/SCC.2011.32

Wang, S., Zheng, Z., Wu, Z., Lyu, M. R., & Yang, F. (2014). Reputation measurement and malicious feedback rating prevention in web service recommendation systems. *IEEE Transactions on Services Computing*, *8*(5), 755–767. doi:10.1109/TSC.2014.2320262

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–292. doi:10.1007/BF00992698

Weiser, M. (1991, Sept.). *The computer for the twenty-first century*. *Scientific American*.

Wilson, D., Ostrom, E., & Cox, M. (2013). Generalizing the core design principles for the efficacy of groups. *Journal of Economic Behavior & Organization*, *90*, S21–S32. doi:10.1016/j.jebo.2012.12.010

Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., & Du, H.-Y. (2010). Research on the architecture of Internet of Things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*. IEEE.

Xiangqian, S., & Lin, G. (2010). *An Ant Colony Based Algorithm for Optimal Selection of Materialized View*. IEEE. doi:10.1109/ICISS.2010.5654879

Xu, X., Weber, I., Staples, M., Zhu, L., Bosch, J., Bass, L., ... Rimba, P. (2017). A taxonomy of blockchain based systems for architecture design. *IEEE International Conference on Software Architecture*. 10.1109/ICSA.2017.33

Yang, J., Karlapalem, K., & Li, Q. (1997). Algorithms for materialized view design in data warehousing environment. *VLDB, Proceedings of the 23rd International Conference on Very Large Data Bases*, 97, 136–145.

Yan, Z., Zhang, P., & Vasilakos, A. (2014). A survey on trust management for the Internet of things. *Journal of Network and Computer Applications*, *42*, 120–134. doi:10.1016/j.jnca.2014.01.014

You, A., Be, M. A. Y., & In, I. (2018). Assessment of utility side financial benefits of demand side management considering environmental impacts Assessment of Utility Side Financial Benefits of Demand Side Management Considering Environmental Impacts. doi:10.1063/1.5022897

Young, R. M., & Lewis, R. L. (1999). The soar cognitive architecture and human working memory. *Models of working memory: Mechanisms of active maintenance and executive control*, 224–256.

Yu, D., Dou, W., Zhu, Z., & Wang, J. (2015). Materialized view selection based on adaptive genetic algorithm and its implementation with Apache Hive. International Journal of Computational Intelligence Systems, 8(6), 1091–1102.

Yu, J. X., Yao, X., Choi, C.-H., & Gou, G. (2003). Materialized View Selection as constrained evolutionary optimization. *IEEE Transactions on Systems, Man and Cybernetics. Part C, Applications and Reviews*, *33*(4), 458–467. doi:10.1109/TSMCC.2003.818494

Yu, J. X., Yao, X., Choi, C.-H., & Gou, G. (2003). Materialized view selection as constrained evolutionary optimization. *IEEE Transactions on Systems, Man, and Cybernetics*. *Part C*, *33*(4), 458–467.

Yu, N., Shah, S., Johnson, R., Sherick, R., Hong, M., & Loparo, K. (2015). Big data analytics in power distribution systems. *2015 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference*. doi:10.1109/ISGT.2015.7131868

Zacharia, G., Moukas, A., & Maes, P. (2000). Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, *29*(4), 371–388. doi:10.1016/S0167-9236(00)00084-1

Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., ... Ghodsi, A. (2016). Apache spark: A unified engine for big data processing. *Communications of the ACM*, *59*(11), 56–65. doi:10.1145/2934664

Zhang, F., Liu, M., & Shen, W. (2017). Operation modes of smart factory for high end equipment manufacturing in the internet and big data era. *IEEE International Conference on systems, Man and Cybernetics*. 10.1109/SMC.2017.8122594

Zhang, Y., Huang, T., & Bompard, E. F. (2018). Big data analytics in smart grids: A review. Energy Informatics, 1(1), 1–24. doi:10.1186/s42162-018-0007-5

Zhang, Z. (2008). *Scalable Role & Organization Based Access Control and Its Administration* (PhD thesis). Fairfax, VA: George Mason University.

Zhang, C., & Yang, J. (1999). *Genetic Algorithm for Materialized View Selection in Data Warehouse Environments*. Springer-Verlag Berlin Heidelberg. doi:10.1007/3-540-48298-9_12

Zhang, C., Yao, X., & Yang, J. (2001). An evolutionary approach to materialized views selection in a data warehouse environment. *IEEE Transactions on Systems, Man, and Cybernetics. Part C*, *31*(3), 282–294.

Zhang, L. J., Zhang, J., & Cai, H. (2007). *Services computing*. Beijing: Tsinghua University Press.

Zhixiang, Z., & Dongjin, Y. (2014). Selecting Materialized Views based on Genetic Algorithm. In Advanced Science and Technology Letters: Vol. 45. *CCA 2014* (p. 65). SERSC.

Zhixiong, Z., Zhang, X., & Sandhu, R. (2006). ROBAC: Scalable Role and Organization Based Access Control Models. *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, 50.

Zhou, K., Fu, C., & Yang, S. (2016). Big data driven smart energy management: From big data to big insights. *Renewable & Sustainable Energy Reviews*, *56*, 215–225. doi:10.1016/j.rser.2015.11.050

Zobaa, A. F. (2018). *Big Data Analytics in Future Power Systems* (1st ed.). CRC Press; doi:10.1201/9781315105499.

Zoet, M. (2014). *Methods and Concepts for Business Rules Management* (Ph.D. Thesis). University of Applied Sciences Utrecht.

About the Contributors

Naveen Prakash obtained his doctoral degree from IIT Delhi in 1980. He worked in the Bhabha Atomic Research Centre and at NCSDCT, TIFR before joining the R&D group of CMC Ltd. Thereafter, he worked in the department of Computer Science and Engg., IIT Kanpur, and at NSIT Delhi. He was scientific advisor to the British Council Division, New Delhi. Prof. Prakash took up the directorship of various educational institutes in India. He provided consultancy to Asian and African Development Bank projects in Sri Lanka and Tanzania respectively. He was a UNDP consultant to IGNCA and worked on a World Bank project in Malawi. Professor Praksh has lectured extensively in various universities abroad. He is on the editorial board of the Requirements Engineering Journal, Springer and of IJISMD, IGI Global and has been Chairperson of 5 International Conferences and member of Programme Committees of about 100 others. He has published about 70 research papers and authored two books. Prof. Prakash continues to be an active researcher. Besides Business Intelligence and Data Warehousing, his interests are in the Internet-of-things and NoSQL database. He also lectures in the IIIT, Delhi.

Deepika Prakash obtained her Ph.D. from Delhi Technological University, Delhi in the area of Data Warehouse Requirements Engineering. She has experience in both academics and industrial R&D. As a researcher, she has authored a number of papers in international journals/conferences. She has co-authored a book entitled "Data Warehouse Requirements Engineering - A Decision Based Approach" published by Springer in 2018. She is also on the program committee of international conferences like ENASE. Her responsibilities in industry spanned the complete life cycle, from requirements engineering through conceptual modeling to extract-transform-load (ETL) activities. Her research in industry focused on agile Data Warehouse systems and data mart integration. In academics she is currently faculty in NIIT University. Her current research interests include Data Warehouse systems, NoSQL databases and Health Informatics.

***

Hanu Bhardwaj has more than 18 years of rich experience in academics. She is an M.Tech., M.Phil., and currently pursuing her Ph.D. Her research interests lie in Data Warehousing, Business Intelligence and Agile methodology. She has a good number of publications in national and international journals of repute and has also participated in many credible conferences, FDPs, and Workshops. She was also recently selected among 200 participants out of 1500 applicants across India for AICTE UKIERI Leadership Program, conducted by Dudley College UK. She is the Single Point of Contact for Infosys Campus Connect Program since 2009 and has been awarded with Distinguished Facilitator award from Infosys, for consecutive two years, for innovatively incorporating Agile Methodologies in Programming Courses. She heads the Department of Computer Science and Technology at Manav Rachna University and has been instrumental in establishing BTech Specialization in Data Science & Machine Learning and Digital Transformation Engg in association with Dutch company Xebia at MRU. She played an important role in establishing the first Agile Classroom in higher education in India at MRU campus.

Jayati Deshmukh is a Ph.D. research scholar at Web Science Lab, IIIT-Bangalore. Her research interests broadly lie in the area of artificial intelligence and specifically in multi-agent systems, network science and game theory. She holds M.Tech. in Data Science from IIIT-Bangalore and B.E. in Computer Engineering from Gujarat Technological University. She has been a gold-medalist in both undergraduate and graduate degrees. She has 4+ years of work experience at Accenture Technology Labs, Bangalore where she did research and development in the area of natural language processing, deep learning and knowledge graphs.

Anjana Gosain is working as Professor in University School of Information & Communication Technology, GGSIPU, Delhi, India. She obtained her Ph.D. from GGS Indraprastha University & M.Tech in Information Systems by receiving gold medal from Netaji Subhas Institute of Technology (NSIT) Delhi. Prior to joining University School of Information & Communication Technology, she has worked with Computer Science department of YMCA Institute of Engineering, Faridabad (1994–2002). She has also worked with REC, Kurukshetra. Her technical and research interests include data warehouse, data mining, requirements engineer-ing, databases, software engineering, object orientation and conceptual modeling.

Sérgio Guerreiro is an Assistant Professor at the Department of Computer Science and Engineering, Information Systems scientific area, University of Lisbon, Portugal, since 2017. He is a researcher at INESC-ID, IDSS (Information and Decision Support Systems Lab). Previously, he was a Senior IT Project Manager in Sonae.com and an Assistant Professor at Lusófona University and at Universidade da

Beira Interior, Portugal. He obtained his PhD in Information Systems and Computer Engineering at Instituto Superior Técnico/University of Lisbon (IST/UL). Before that, he concluded the MSc degree and the Engineering degree. Both degrees also in Information Systems and Computer Engineering at IST/UL. His research interests are located in Enterprise Engineering with the goal of finding solutions to control the business transactions operation, identifying and solving run-time misalignments. He uses the specification languages of BPMN, Archimate, DEMO, UML, SysML, WOSL, EPC; the mechanisms of process mining, business analytics; implementing solutions for governance, risk and compliance.

Amira Hakim is doctor in complex systems engineering at the University of Badji Mokhtar Annaba, Algeria She was supervised by Pr. Abdelkrim Amirat from the University of Souk Ahras, Algeria and Pr. Mourad Oussalah from the University of Nantes, France. Her doctoral work explores dynamic reconfiguration of Internet of Things systems. She holds an MA in software engineering and a BA in computer science and artificial intelligence both from the University of Souk Ahras.

Marty Kelley has 10 years professional experience in Manufacturing and Supply Chain in addition to 5 years of experience in systems engineering. He holds a Bachelors degree from the University of Maryland, an M.B.A from George Mason University and a Masters in Systems Engineering from Johns Hopkins University. Marty holds two certifications from APICS (CPIM, CSCP) and is currently a Doctoral candidate with George Washington University.

Toufik Marir is a Lecturer at the Department of Mathematics and Computer Science of The University of Oum El Bouaghi in Algeria. He holds the Magister in Computer Science (Artificial Intelligence) from the University of Khenchela (Algeria) in 2009 and the Doctorate in Science in Computer Science from the University of Annaba (Algeria) in 2015. Currently, he is a member of the team Distributed-Intelligent Systems Engineering (DISE) at the ReLa(CS)2 Laboratory. His main areas of interest include agent-oriented software engineering, software quality and formal methods.

K. S. Sastry Musti has obtained his BTech in Electrical Engineering from JNTU, Kakinada, MTech and PhD from NIT, Warangal., India. He is currently with Namibia University of Science and Technology as Associate Professor with the Department of Electrical and Computer Engineering. He has more than 30 years of professional experience in both academia and industry. His research interests are in the extended areas of Engineering Education, Power Systems, Software Engineering and Systems and Information Systems. Professor Sastry is a senior member IEEE(USA) and a Life member IE (India).

Parimala N. is currently a professor in the School of Computer & Systems Sciences at Jawaharlal Nehru University, India .She has over 35 years of experience in teaching and research. She has supervised a number of post graduate dissertations and Ph.D. theses. Her area of interest include DBMS, Data warehousing, Data mining and Service oriented architecture.

Benaboud Rohallah is a Lecturer at the Department of Mathematics and Computer Science of The University of Oum El Bouaghi in Algeria. He holds the Magister in Computer Science from the University of Oum El Bouaghi (Algeria) in 2005 and the Doctorate in Science in Computer Science from the University of Constantine 2 (Algeria) in 2016. Currently, he is a member of the team Distributed-Intelligent Systems Engineering (DISE) at the ReLa(CS)2 Laboratory. His main areas of interest Web Service, Software Engineering and Multi-Agent Systems.

Kavita Sachdeva is working as an Assistant Professor in SGT University, Gurugram, India. She is pursuing Ph.D. from GGS Indraprastha University, Delhi. She obtained her M.Tech in Computer Science from Banasthali University, Rajasthan and completed her M.Tech dissertation from Biometrics Research lab, IIT Delhi. Prior to joining Shree Guru Gobind Singh Tricentenary University, Gurugram, she has worked with Indira Gandhi Delhi Technical University for Women, Delhi (2012–2015). She has also worked with TCS, Delhi as a Software Engineer. Her technical and research interests include data warehouse, evolutionary optimization, fuzzy logic and software engineering.

Deepak Sharma is a Test Automation Engineer in Venture7 Technology Private Limited at Faridabad, India. He received his Ph.D. (Computer Science and Engineering) from IKG Punjab Technical University, Jalandhar, Punjab, India in 2019. He has more than 8 years of Teaching and 4 years of Industry experience. His research interest is in information system development, business process modeling and business rule modeling. He has published more than 20 research articles in various national and international conferences and journals.

Srinath Srinivasa heads the Web Science lab and is the Dean (R&D) at IIITB. Srinath holds a Ph.D (magna cum laude) from the Berlin Brandenburg Graduate School for Distributed Information Systems (GkVI) Germany, an M.S. (by Research) from IIT-Madras and B.E. in Computer Science and Engineering from The National Institute of Engineering (NIE) Mysore. His research interests are in understanding how the WWW is affecting humanity; and how the web can enable social empowerment and capability building.He has participated in several initiatives for technology enhanced education including the VTU Edusat program, The National Programme for Technology Enhanced Learning (NPTEL), a Switzerland based online MBA school called Educatis, and IIITB's educational outreach program with Upgrad. He is a member of various technical and organizational committees for international conferences like International Conference on Weblogs and Social Media (ICWSM), ACM Hypertext, International Conference on Management of Data and Data Sciencec (COMAD/CoDS), International conference on Ontologies, Databases and Applications of Semantics (ODBASE), International Conference on Big Data Analytics (BDA), etc.As part of academic community outreach, Srinath has served on the Board of Studies of Goa University and as a member of the Academic Council of the National Institute of Engineering, Mysore. He has served as a technical reviewer for various journals like the VLDB journal, IEEE Transactions on Knowledge and Data Engineering, and IEEE Transactions on Cloud Computing. He is currently serving as an Associate Editor of the journal Sadhana from the Indian Academy of Sciences. He is also the recipient of various national and international grants and awards, from foundations and companies like: EU Horizon 2020, UK Royal Academy of Engineering, Research Councils UK, MEITy, DST, Siemens, Intel, Mphasis, EMC and Gooru.

Index

N

O

P

Q

R

S

T

U

W