Preface

Databases and database systems continually assume a more critical place at the center of the information systems architecture for many companies and organizations. Coupled with data warehouses and advanced data mining techniques, an increasing number of organizations now have powerful analytic and predictive tools available to help them gain and maintain competitive advantage. In addition, connecting back office databases and data warehouses with the Web is becoming vital for a growing number of organizations. The preceding developments and events in the practical business world provide the backdrop for research into the creation of ever more sophisticated means to the ends regarding information systems.

In the current environment, research investigating the entire discipline of database should be at the core of teaching as well as extending research in all related areas of database. Database lines of research include business intelligence, query languages, query optimization, data warehouse design, data mining algorithms, XML tool development, and tools for the modeling, design, and development of information systems. Some of the more recent techniques involve design and deployment of object-relational databases that include support for object-oriented systems. Other research and development streams involve Web Services, Service Oriented Architectures, and Open Source Systems. As the complexity of database systems increases, modeling databases and database systems has assumed increased importance in database research. Future databases or data warehouses are likely to include real-time analysis using advanced statistical methods, with increasing immediacy and connection to the Web, Supply Chain Management, Customer Relationship Management, and Knowledge Management systems.

Over the past forty years, IS and database researchers have conducted empirical investigations that have resulted in a better understanding of the impacts and values of advanced database principles in business on a global basis. Past database research has focused primarily on technical and organizational issues, and less on social issues. Issues such as text mining and opinion mining that depend on state of the art database systems and can be used to infer meaning and emotional content are also likely to garner more attention in future research.

In accordance with the high standard of previous volumes in the Advances in Database Research Series, we edited this volume by including only the best research in the field. A majority of the chapters included in this volume are conducted by internationally renowned scholars. We believe this volume will provide a convenient store of valuable knowledge on the topic of database, systems analysis and design, design science, and software engineering. This volume can serve as a starting point for references and citation pieces for researchers, graduate students and practitioners in the field. This volume consists of eighteen chapters; three are focused on database, three on systems analysis and design, four on modeling, two on architecture, five on open systems development, and one on educational efforts. A brief description of each chapter is presented below.
Chapter 1, “A Multiple-Bits Watermark for Relational Data,” by Yingjiu Li, Huiping Guo, and Shuhong Wang, presents a technique to mark data in databases protected by copyright. The technique is robust enough that it can estimate the probability regarding whether the watermark itself can be detected. The technique can also work on databases that do not use primary key attributes and it can prevent attribute related attacks. Finally, the technique supports multiple watermarks so that joint owners can each place their own security measure, or to detect multiple (additive) attacks.

Chapter 2, “BROOD: Business Rules-Driven Object Oriented Design,” by Pericles Loucopoulos and Wan Kadir, identifies a critical success factor for information systems as their ability to change with environmental changes. The authors go on to explicate their approach to deriving business rules that include means to evolve or change information systems from an object-oriented perspective. They propose the use of modelling techniques, in particular UML as the basis for modelling business rules that allow or encourage changes in the depicted information systems.

Chapter 3, “Bug Fixing Practices within Free/Libre Open Source Software Development Teams,” by Kevin Crowston and Barbara Scozzi, examine the processes and practices of distributed development teams working on open source projects. They approach the issues involved by analyzing messages recorded in the error tracking system of four projects. By doing this the authors were able to identify common tasks, coordination efforts, and roles of the development teams. The results can be compared with those of non open source teams and other open source teams as well to provide insight into improving development efforts.

Chapter 4, "Conflicts, Compromises and Political Decisions: Methodological Challenges of Enterprise-Wide E-Business Architecture Creation," by Kari Smolander and Matti Rossi, examines how an international ICT company developed its architecture. The authors monitored the early architectural phases of the development effort as part of the research project. Results indicate that the final architecture often derives from the conditions and environment present at the time of its creation. According to the authors, other elements affecting the architecture can include political compromises and constraints.

Chapter 5, “Evaluation of MDE Tools from a Metamodeling Perspective,” by João de Sousa Saraiva and Alberto Rodrigues, explores and enhances the ideas of Model Driven Architecture (MDA) by creating an additional abstraction layer that they call the graphical model layer. The paper goes on to describe the evaluation of tools supporting metamodels from the MDA perspective. Based on the evaluation, the chapter closes with a possible research agenda for MDA development tools.

Chapter 6, “Exploring the Effects of Process Characteristics on Products Quality in Open Source Software Development,” by Stefan Koch and Christian Neumann, proposes metrics that purport to measure open system development processes. In particular, the metrics are aimed at object-oriented processes. A problem the authors note is that the existing metrics do not measure quality, complexity or structure. The goal of the research is to determine whether metrics can be used to assess the aforementioned issues.

Chapter 7, “The Impact of Ideology on the Organizational Adoption of Open Source Software,” by Kris Ven and Jan Verelst, examines the ideology underlying the open source community of developers. Other studies have indicated that US organizations are more interested in the practical uses of open source. The authors propose that other opportunities might exist that allow more of the underlying ideologies to emerge. The findings indicate that most organizations favor the practical over the ideological, but that, in small organizations, ideological influences might yet play a role in the adoption of open source.

Chapter 8, “Web Services, Service-Oriented Computing, and Service-Oriented Architecture: Separating Hype from Reality,” by John Erickson and Keng Siau, provides an overview of the Service Oriented Architecture (SOA), Web services, and Service Oriented Computing (SOC) areas of software
and systems development. The authors note that the definitions of the system types are not agreed upon by business or researchers, and provide a framework for understanding the components of SOA. The authors provide some evidence suggesting that the areas are understudied in terms of research, and suggest future directions or gaps in the current research for investigators.

Chapter 9, “Approximate Query Answering with Knowledge Hierarchy,” by Wookey Lee, Myung-Keun Shin, Soon Young Huh, Donghyun Park, and Jumi Kim, creates an efficiency relaxation algorithm to change approximation queries into ordinary queries. The approach uses the knowledge hierarchy as a means to enable this transformation. Then the authors apply ranking measures to help deal with the many complex nodes generated by using the knowledge hierarchy to simplify the approximation query.

Chapter 10, “Abstract DTD Graph from an XML Document: A Reverse Engineering Approach,” by Joseph Fong and Herbert Shiu, proposes a means to reverse engineer XML documents back into their “conceptual schema,” which they call Extended DTD graphs. The authors argue that their approach can do two tasks; first to determine the structure of XML documents, and second to extract the data schemas from the XML elements. They accomplish these tasks by considering the XML element instances as records in a relational database.

Chapter 11, “A Dynamic Model of Adoption and Improvement for Open Source Business Applications,” by Michael Brydon and Aidan R. Vining, proposes a way to model open source disruption in software markets. Their two stage model includes an incubation stage, where the initial adoption and development are nurtured, followed by a snowball stage, where momentum is gathered. The authors then apply their model to a Customer Relationship Management application named SugarCRM as a test case.

Chapter 12, “Aiding the Development of Active Applications: A Decoupled Rule Management Solution,” by Florian Daniel and Giuseppe Pozzi, examines the set of rules that commonly describe what they call active applications. They use the Event-Condition-Action paradigm as the starting point for their explanatory vehicle, and the rules management system they derive allows developers to separate active and non-active design issues.

Chapter 13, “Dimensions of UML Diagram Use: Practitioner Survey and Research Agenda,” by Brian Dobing and Jeffrey Parsons, examines field use of UML. The research was executed by means of a survey to UML practitioners. Results indicate that practitioners generally tend not to use UML Use Case diagrams. They either do not utilize Use Cases at all or instead make use of textual based Use Case descriptions. This finding is directly at odds with much of the literature on UML, and is also counter to how the OMG (Object Management Group) prescribes best practices for UML.

Chapter 14, “A 360-Degree Perspective of Education in 3-D Virtual Worlds,” by Brenda Eschenbrenner, Fiona Fui-Hoon Nah, and Keng Siau, examines education from the perspective of 3D virtual worlds, such as Second Life. The research assesses the impact of such virtual worlds on education via a review of current literature on the subject. Based on the literature, pedagogical, practice, and research objectives are discussed. The literature suggests that research into the impacts of virtual worlds on education is at a very early stage, and many opportunities for education and research remain unexplored.

Chapter 15, “Using Graphics to Improve Understanding of Conceptual Models,” by Kamal Masri, Drew Parker, and Andrew Gemino, provides the results of an experiment involving the replacement of standard identity boxes in ERDs (Entity Relationship Diagrams) with iconic graphics (small representative graphics). The primary problem under investigation was how to enhance novice understanding of ERDs. Findings indicate that a reduction in “extraneous” cognitive load for those using the iconic
graphics was possible, further allowing an increase in “germane” cognitive load. This implies better understanding of the diagrams.

Chapter 16, “Beyond Open Source: The Business of ‘Whole’ Software Solutions,” by Joseph Feller, Patrick Finnegan, and Jeremy Hayes, examines a common research notion that open source developers will be forced (by competitive pressures) to join cooperative type networks so that a complete product can be provided to customers. The chapter uses a case study at Zea Partners, an open source content management application developer, to conclude that if such networks are to succeed, then the participant organizations must reconcile the coordination concerns with the operational concerns.

Chapter 17, “The Application-Based Domain Modeling Approach: Principles and Evaluation,” by Iris Reinhartz-Berger and Arnon Sturm, investigates the area of domain analysis with the goal of developing an approach that can overcome some of the shortcomings of modeling the domain using metamodeling techniques. The authors propose that domain artifacts can be used to assemble relatively complete and valid artifacts in their approach called Application based Domain Modeling. They demonstrate the viability of their approach using a CASE tool created for UML.

Chapter 18, “The Use of Ontology for Data Mining with Incomplete Data,” by Hai Wang and Shouhong Wang, demonstrates how a domain analysis of knowledge representations in a data warehouse or other data set, can be used in combination with a formal ontology, developed specifically for data mining, to extract relatively complete results with incomplete data. They provide experimental evidence supporting their claim.