Principle Advancements in Database Management Technologies:
New Applications and Frameworks

Keng Siau
*University of Nebraska–Lincoln, USA*

John Erickson
*University of Nebraska–Omaha, USA*
Preface

Chapter 1
A Multiple-Bits Watermark for Relational Data

Yingjiu Li, Singapore Management University, Singapore
Huiping Guo, California State University at Los Angeles, USA
Shuhong Wang, University of Wollongong, Australia

At the heart of the information economy, commercially and publicly useful databases must be sufficiently protected from pirated copying. Complementary to the Database Protection Act, database watermarking techniques are designed to thwart pirated copying by embedding owner-specific information into databases so that the ownership of pirated copies of protected databases can be claimed if the embedded information is detected. This chapter presents a robust watermarking scheme for embedding a multiple-bits watermark to numerical attributes in database relations. The scheme is robust in the sense that it provides an upper bound for the probability that a valid watermark is detected from unmarked data, or a fictitious secret key is discovered from pirated data. This upper bound is independent of the size of the data. The scheme is extended to database relations without primary-key attributes to thwart attribute-related attacks. The scheme is also extended to multiple watermarks for defending additive attacks and for proving joint ownership.

Chapter 2
BROOD: Business Rules-Driven Object Oriented Design

Pericles Loucopoulos, Loughborough University, UK
Wan M.N. Wan Kadir, Universiti Teknologi Malaysia, Malaysia

A critical success factor for information systems is their ability to evolve as their environment changes. There is compelling evidence that the management of change in business policy can have a profound effect on an information system’s ability to evolve effectively and efficiently. For this to be successful, there is a need to represent business rules from the early requirements stage, expressed in user-understandable terms, to downstream system design components and maintain these throughout the lifecycle of the system. Any user-oriented changes could then be traced and if necessary propagated from requirements to design specifications and evaluated by both end-users and developers about their impact on the system. The BROOD approach, discussed in this chapter, aims to provide seamless traceability between require-
ments and system designs through the modelling of business rules and the successive transformations, using UML as the modelling framework.

Chapter 3
Bug Fixing Practices within Free/Libre Open Source Software Development Teams ........................................... 51
Kevin Crowston, Syracuse University, USA
Barbara Scozzi, Politecnico di Bari, Italy

Free/Libre open source software (FLOSS, e.g., Linux or Apache) is primarily developed by distributed teams. Developers contribute from around the world and coordinate their activity almost exclusively by means of email and bulletin boards, yet somehow profit from the advantages and evade the challenges of distributed software development. This chapter investigates the structure and the coordination practices adopted by development teams during the bug-fixing process, which is considered one of main areas of FLOSS project success. In particular, based on a codification of the messages recorded in the bug tracking system of four projects, this chapter identifies the accomplished tasks, the adopted coordination mechanisms, and the role undertaken by both the FLOSS development team and the FLOSS community. The chapter concludes with suggestions for further research.

Chapter 4
Conflicts, Compromises, and Political Decisions: Methodological Challenges of Enterprise-Wide E-Business Architecture Creation ................................................................. 82
Kari Smolander, Lappeenranta University of Technology, Finland
Matti Rossi, Helsinki School of Economics, Finland

This chapter describes the architecture development process in an international ICT company, which is building a comprehensive e-business system for its customers. The implementation includes the integration of data and legacy systems from independent business units and the construction of a uniform Web-based customer interface. The authors followed the early process of architecture analysis and definition over a year. The research focuses on the creation of e-business architecture and observes that instead of guided by a prescribed method, the architecture emerges through somewhat non-deliberate actions obliged by the situation and its constraints, conflicts, compromises, and political decisions. The interview-based qualitative data is analyzed using grounded theory and a coherent story explaining the situation and its forces is extracted. Conclusions are drawn from the observations and possibilities and weaknesses of the support that UML and RUP provide for the process are pointed out.

Chapter 5
Evaluation of MDE Tools from a Metamodeling Perspective ................................................................. 105
João de Sousa Saraiva, INESC-ID/Instituto Superior Técnico, Portugal
Alberto Rodrigues da Silva, INESC-ID/Instituto Superior Técnico, Portugal

Ever since the introduction of computers into society, researchers have been trying to raise the abstraction level at which software programs are built. Currently an abstraction level based on graphical models instead of source code is being adopted: MDE. MDE is the driving force for some recent modeling languages and approaches, such as OMG’s UML or Domain-Specific Modeling. All these approaches
are founded on metamodeling: defining languages that represent a problem-domain. A key factor for the
success of any approach is appropriate tool support. However, only recently have tool creators started
considering metamodeling as an important issue in their list of concerns. This chapter evaluates a small
set of MDE tools from the perspective of the metamodeling activity, focusing on both architectural and
practical aspects. Then, using the results of this evaluation, the authors discuss open research issues for
MDE-based software development tools.

Chapter 6
Exploring the Effects of Process Characteristics on Product Quality in Open Source
Software Development

Stefan Koch, Vienna University of Economics and Business Administration, Austria
Christian Neumann, Vienna University of Economics and Business Administration, Austria

There has been considerable discussion on the possible impacts of open source software development
practices, especially in regard to the quality of the resulting software product. Recent studies have shown
that analyzing data from source code repositories is an efficient way to gather information about proj¬
ect characteristics and programmers, showing that OSS projects are very heterogeneous in their team
structures and software processes. However, one problem is that the resulting process metrics measuring
attributes of the development process and of the development environment do not give any hints about
the quality, complexity, or structure of the resulting software. Therefore, this chapter expands the analysis
by calculating several product metrics, most of them specifically tailored to object-oriented software.
The authors then analyzed the relationship between these product metrics and process metrics derived
from a CVS repository. The aim was to establish whether different variants of open source development
processes have a significant impact on the resulting software products. In particular, the authors analyzed
the impact on quality and design associated with the numbers of contributors and the amount of their
work, using the GINI coefficient as a measure of inequality within the developer group.

Chapter 7
The Impact of Ideology on the Organizational Adoption of Open Source Software

Kris Ven, University of Antwerp, Belgium
Jan Verelst, University of Antwerp, Belgium

Previous research has shown that the open source movement shares a common ideology. Employees
belonging to the open source movement often advocate the use of open source software within their
organization. Hence, their belief in the underlying open source software ideology may influence the
decision making on the adoption of open source software. This may result in an ideological—rather
than pragmatic—decision. A recent study has shown that American organizations are quite pragmatic
in their adoption decision. This chapter argues that there may be circumstances in which there is more
opportunity for ideological behavior. The authors therefore investigated the organizational adoption de-
cision in Belgian organizations. Results indicate that most organizations are pragmatic in their decision
making. However, the authors have found evidence that suggests that the influence of ideology should
not be completely disregarded in small organizations.
Chapter 8
Web Services, Service-Oriented Computing, and Service-Oriented Architecture: Separating Hype from Reality

John Erickson, University of Nebraska - Omaha, USA
Keng Siaw, University of Nebraska - Lincoln, USA

Service-oriented architecture (SOA), Web services, and service-oriented computing (SOC) have become the buzz words of the day for many in the business world. It seems that virtually every company has implemented, is in the midst of implementing, or is seriously considering SOA projects, Web services projects, or service-oriented computing. A problem many organizations face when entering the SOA world is that there are nearly as many definitions of SOA as there are organizations adopting it. Further complicating the issue is an unclear picture of the value added from adopting the SOA or Web services paradigm. This chapter attempts to shed some light on the definition of SOA and the difficulties of assessing the value of SOA or Web services via return on investment (ROI) or nontraditional approaches, examines the scant body of evidence empirical that exists on the topic of SOA, and highlights potential research directions in the area.

Chapter 9
Approximate Query Answering with Knowledge Hierarchy

Wookey Lee, Inha University, Korea
Myung-Keun Shin, Telecom Business Division, SK C&C, Korea
Soon Young Huh, Korea Advanced Institute of Science and Technology, South Korea
Donghyun Park, Inha University, South Korea
Jumi Kim, Small Business Institute, Korea

Approximate Query Answering is important for incorporating knowledge abstraction and query relaxation in terms of the categorical and the numerical data. By exploiting the knowledge hierarchy, a novel method is addressed to quantify the semantic distances between the categorical information as well as the numerical data. Regarding that, an efficient query relaxation algorithm is devised to modify the approximate queries to ordinary queries based on the knowledge hierarchy. Then the ranking measures work very efficiently to cope with various combinations of complex queries with respect to the number of nodes in the hierarchy as well as the corresponding cost model.

Chapter 10
Abstract DTD Graph from an XML Document: A Reverse Engineering Approach

Joseph Fong, City University of Hong Kong, China
Herbert Shiu, City University of Hong Kong, China

Extensible Markup Language (XML) has become a standard for persistent storage and data interchange via the Internet due to its openness, self-descriptiveness and flexibility. This chapter proposes a systematic approach to reverse engineer arbitrary XML documents to their conceptual schema — Extended DTD Graphs — which is a DTD Graph with data semantics. The proposed approach not only determines the structure of the XML document, but also derives candidate data semantics from the XML element instances by treating each XML element instance as a record in a table of a relational database. One
application of the determined data semantics is to verify the linkages among elements. Implicit and explicit referential linkages are among XML elements modeled by the parent-children structure and ID/IDREF(S) respectively. As a result, an arbitrary XML document can be reverse engineered into its conceptual schema in an Extended DTD Graph format.

Chapter 11
A Dynamic Model of Adoption and Improvement for Open Source Business Applications

Michael Brydon, Simon Fraser University, Canada
Aidan R. Vining, Simon Fraser University, Canada

This chapter develops a model of open source disruption in enterprise software markets. It addresses the question: Is free and open source software (FOSS) likely to disrupt markets for enterprise business applications? The conventional wisdom is that open source provision works best for low-level system-oriented technologies while large, complex enterprise business applications are best provided by commercial software vendors. The authors challenge the conventional wisdom by developing a two-stage model of open source disruption in business application markets that emphasizes a virtuous cycle of adoption and lead-user improvement of the software. The two stages are an initial incubation stage (the I-Stage) and a subsequent snowball stage (the S-Stage). Case studies of several FOSS projects demonstrate the model's ex post predictive value. The authors then apply the model to SugarCRM, an emerging open source CRM application, to make ex ante predictions regarding its potential to disrupt commercial CRM incumbents.

Chapter 12
Aiding the Development of Active Applications: A Decoupled Rule Management Solution

Florian Daniel, University of Trento, Italy
Giuseppe Pozzi, Politecnico di Milano, Italy

Active applications are characterized by the need for expressing, evaluating, and maintaining a set of rules that implement the application's active behavior. Typically, rules follow the Event-Condition-Action (ECA) paradigm, yet oftentimes their actual implementation is buried in the application code, as their enactment requires a tight integration with the concepts and modules of the application. This chapter proposes a rule management system that allows developers to easily expand its rule processing logic with such concepts and modules and, hence, to decouple the management of their active rules from the application code. This system derives from an exception manager that has previously been developed in the context of an industry-scale workflow management system and effectively allows developers to separate active and non-active design concerns.

Chapter 13
Dimensions of UML Diagram Use: Practitioner Survey and Research Agenda

Brian Dobing, University of Lethbridge, Canada
Jeffrey Parsons, Memorial University of Newfoundland, Canada

The UML is an industry standard for object-oriented software engineering. However, there is little empirical evidence on how UML is used. This chapter reports results of a survey of UML practitioners.
The authors found differences in several dimensions of UML diagram usage on software development projects, including frequency, the purposes for which they were used, and the roles of clients/users in their creation and approval. System developers are often ignoring the “Use Case-driven” prescription that permeates much of the UML literature, making limited or no use of either Use Case Diagrams or textual Use Case descriptions. Implications and areas requiring further investigation are discussed.

Chapter 14
A 360-Degree Perspective of Education in 3-D Virtual Worlds
Brenda Eschenbrenner, University of Nebraska-Lincoln, USA
Fiona Fui-Hoon Nah, University of Nebraska-Lincoln, USA
Keng Siau, University of Nebraska-Lincoln, USA

Three-dimensional virtual world environments are providing new opportunities to develop engaging, immersive experiences in education. These virtual worlds are unique in that they allow individuals to interact with others through their avatars and with objects in the environment, and can create experiences that are not necessarily possible in the real world. Hence, virtual worlds are presenting opportunities for students to engage in both constructivist and collaborative learning. To assess the impact of the use of virtual worlds on education, a literature review is conducted to identify current applications, benefits being realized, as well as issues faced. Based on the review, educational opportunities in virtual worlds and gaps in meeting pedagogical objectives are discussed. Practical and research implications are also addressed. Virtual worlds are proving to provide unique educational experiences, with its potential only at the cusp of being explored.

Chapter 15
Using Graphics to Improve Understanding of Conceptual Models
Kamal Masri, Simon Fraser University, Canada
Drew Parker, Simon Fraser University, Canada
Andrew Gemino, Simon Fraser University, Canada

Making Entity-Relationship diagrams easier to understand for novices has been a topic of previous research. This study provides experimental evidence that suggests using small representative graphics (iconic graphics) to replace standard entity boxes in an ER diagram can have a positive effect on domain understanding for novice users. Cognitive Load Theory and the Cognitive Theory of Multimedia Learning are used to hypothesize that iconic graphics reduce extraneous cognitive load of model viewers leading to more complete mental models and consequently improved understanding. Domain understanding was measured using comprehension and transfer (problem solving) tasks. Results confirm the main hypothesis. In addition, iconic graphics were found to be less effective in improving domain understanding with English as second language (ESL) participants. ESL results are shown to be consistent with predictions based on the Cognitive Load Theory. The importance of this work for systems analysts and designers comes from two considerations. First, the use of iconic graphics seems to reduce the extraneous cognitive load associated with these complex systems. Secondly, the reduction in extraneous load enables users to apply more germane load which relates directly with levels of domain understanding. Thus iconic graphics may provide a simple tool that facilitates better understanding of ER diagrams and the data structure for proposed information systems.
Chapter 16
Beyond Open Source: The Business of ‘Whole’ Software Solutions

Joseph Feller, University College Cork, Ireland
Patrick Finnegan, University of New South Wales, Australia
Jeremy Hayes, University College Cork, Ireland

Researchers have argued that competitive necessities will require open source software companies to participate in cooperative business networks in order to offer the complete product/service (whole product) demanded by customers. It is envisaged that these business networks will enhance the business models of participant firms by supplementing their value adding activities and increasing responsiveness to customers. However, while such propositions have intuitive appeal, there is a paucity of empirical research on such networks. This study examines Zea Partners, a network of small open source companies cooperating to deliver the ‘whole product’ in the area of Content Management Systems (CMS). It investigates how network participation augments the business models of the participant companies, and identifies the agility challenges faced by the business network. The chapter concludes that reconciling the coordination needs of OSS networks with the operational practices of participant firms is of crucial importance if such networks are to achieve adaptive efficiency to deliver whole products in a ‘bazaar-friendly’ manner.

Chapter 17
The Application-Based Domain Modeling Approach: Principles and Evaluation

Iris Reinhartz-Berger, University of Haifa, Israel
Arnon Sturm, Ben-Gurion University of the Negev, Israel

Domain analysis provides guidelines and validation aids for specifying families of applications and capturing their terminology. Thus, domain analysis can be considered as an important type of reuse, validation, and knowledge representation. Metamodeling techniques, feature-oriented approaches, and architectural-based methods are used for analyzing domains and creating application artifacts in these domains. These works mainly focus on representing the domain knowledge and creating applications. However, they provide insufficient guidelines (if any) for creating complete application artifacts that satisfy the application requirements on one hand and the domain rules and constraints on the other hand. This chapter claims that domain artifacts may assist in creating complete and valid application artifacts and present a general approach, called Application-based Do main Modeling (ADOM), for this purpose. ADOM enables specifying domains and applications similarly, (re)using domain knowledge in applications, and validating applications against the relevant domain models and artifacts. The authors demonstrate the approach, which is supported by a CASE tool, on the standard modeling language, UML, and report experimental results which advocate that the availability of domain models may help achieve more complete application models without reducing the comprehension of these models.

Chapter 18
The Use of Ontology for Data Mining with Incomplete Data

Hai Wang, Saint Mary’s University, Canada
Shouhong Wang, University of Massachusetts Dartmouth, USA
Ontology has recently received considerable attention. Based on a domain analysis of knowledge representations in data mining, this chapter presents a structure of ontology for data mining as well as the unique resources for data mining with incomplete data. This chapter demonstrates the effectiveness of ontology for data mining with incomplete data through an experiment.

Compilation of References ................................................................. 389

About the Contributors ...................................................................... 428

Index ................................................................................................. 436