Lecture Notes in Computer Science

The LNCS series reports state-of-the-art results in computer science research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNCS has grown into the most comprehensive computer science research forum available.

The scope of LNCS, including its subseries LNAI and LNBI, spans the whole range of computer science and information technology including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on outstanding PhD work, research projects, technical reports, etc.)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS Online.

Detailed information on LNCS can be found at
http://www.springeronline.com

Proposals for publication should be sent to
LNCS Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany
E-mail: lncs@springer.de

ISSN 0302-9743

Lecture Notes in Computer Science
LNCS LNAI LNBI

springeronline.com

Atzeni et al. (Eds.)

LNCS 3288

Conceptual Modeling – ER 2004

ER 2004

- Paolo Atzeni
- Wesley Chu
- Hongjun Lu
- Shuigeng Zhou
- Tok Wang Ling (Eds.)

Conceptual Modeling – ER 2004

23rd International Conference on Conceptual Modeling
Shanghai, China, November 2004
Proceedings

Springer

Lecture Notes in Computer Science 3288

Commenced Publication in 1973

Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Paolo Atzeni Wesley Chu
Hongjun Lu Shuigeng Zhou
Tok Wang Ling (Eds.)

Conceptual Modeling – ER 2004

23rd International Conference on Conceptual Modeling
Shanghai, China, November 8-12, 2004
Proceedings

Springer

Volume Editors

Paolo Atzeni
Universtità Roma Tre, Dipart Informatica e Automazione
Via Vasca Navale, 79 00146 Roma, Italy
E-mail: atzeni@dia.uniroma3.it

Wesley Chu
University of California, Computer Science Department
3731 Boelter Hall, Los Angeles, CA, 90095, USA
E-mail: wwc@cs.ucla.edu

Hongjun Lu
Hong Kong University of Science and Technology, Department of Computer Science
Clear Water Bay, Kowloon, Hong Kong, China
E-mail: luhj@cs.ust.hk

Shuigeng Zhou
Fudan University, Department of Computer Science and Engineering
220 Handan Road, Shanghai, 200433, China
E-mail: sgzhou@fudan.edu.cn

Tok Wang Ling
National University of Singapore, School of Computing
3 Science Drive 2, Singapore 117543
E-mail: lingtw@comp.nus.edu.sg

Library of Congress Control Number: 2004114106

CR Subject Classification (1998): H.2, H.4, F.4.1, I.2.4, H.1, J.1, D.2, C.2

ISSN 0302-9743
ISBN 3-540-23723-2 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11335221 06/3142 5 4 3 2 1 0

Foreword

On behalf of the Organizing Committee, we would like to welcome you to the proceedings of the 23rd International Conference on Conceptual Modeling (ER 2004). This conference provided an international forum for technical discussion on conceptual modeling of information systems among researchers, developers and users. This was the third time that this conference was held in Asia; the first time was in Singapore in 1998 and the second time was in Yokohama, Japan in 2001. China is the third largest nation with the largest population in the world. Shanghai, the largest city in China and a great metropolis, famous in Asia and throughout the world, is therefore a most appropriate location to host this conference.

This volume contains papers selected for presentation and includes the two keynote talks by Prof. Hector Garcia-Molina and Prof. Gerhard Weikum, and an invited talk by Dr. Xiao Ji.

This volume also contains industrial papers and demo/poster papers. An additional volume contains papers from 6 workshops.

The conference also featured three tutorials: (1) Web Change Management and Delta Mining: Opportunities and Solutions, by Sanjay Madria, (2) A Survey of Data Quality Issues in Cooperative Information Systems, by Carlo Batini, and (3) Visual SQL – An ER-Based Introduction to Database Programming, by Bernhard Thalheim.

The technical program of the conference was selected by a distinguished program committee consisting of three PC Co-chairs, Hongjun Lu, Wesley Chu, and Paolo Atzeni, and more than 70 members. They faced a difficult task in selecting 57 papers from many very good contributions. This year the number of submissions, 293, was a record high for ER conferences. We wish to express our thanks to the program committee members, external reviewers, and all authors for submitting their papers to this conference.

We would also like to thank: the Honorary Conference Chairs, Peter P. Chen and Ruqian Lu; the Coordinators, Zhongzhi Shi, Yoshifumi Masunaga, Elisa Bertino, and Carlo Zaniolo; Workshop Co-chairs, Shan Wang and Katsumi Tanaka; Tutorial Co-chairs, Jianzhong Li and Stefano Spaccapietra; Panel Co-chairs, Chin-Chen Chang and Erich Neuhold; Industrial Co-chairs, Philip S. Yu, Jian Pei, and Jiansheng Feng; Demos and Posters Co-chair, Mong-Li Lee and Gillian Dobbie; Publicity Chair, Qing Li; Publication Chair cum Local Arrangements Chair, Shuigeng Zhou; Treasurer, Xueqing Gong; Registration Chair, Xiaoling Wang; Steering Committee Liaison, Arne Solvberg; and Webmasters, Kun Yue, Yizhong Wu, Zhimao Guo, and Keping Zhao.

We wish to extend our thanks to the Natural Science Foundation of China, the ER Institute (ER Steering Committee), the K.C. Wong Education Foundation in Hong Kong, the Database Society of the China Computer Federation, ACM SIGMOD, ACM SIGMIS, IBM China Co., Ltd., Shanghai Baosight Soft-

ware Co., Ltd., and the Digital Policy Management Association of Korea for their sponsorships and support.

At this juncture, we wish to remember the late Prof. Yahiko Kambayashi who passed away on February 5, 2004 at age 60 and was then a workshop co-chair of the conference. Many of us will remember him as a friend, a mentor, a leader, an educator, and our source of inspiration. We express our heartfelt condolence and our deepest sympathy to his family.

We hope that the attendees found the technical program of ER 2004 to be interesting and beneficial to their research. We trust they enjoyed this beautiful city, including the night scene along the Huangpujiang River and the postconference tours to the nearby cities, leaving a beautiful and memorable experience for all.

November 2004

Tok Wang Ling
Aoying Zhou

Preface

The 23rd International Conference on Conceptual Modeling (ER 2004) was held in Shanghai, China, November 8–12, 2004. Conceptual modeling is a fundamental technique used in analysis and design as a real-world abstraction and as the basis for communication between technology experts and their clients and users. It has become a fundamental mechanism for understanding and representing organizations, including new e-worlds, and the information systems that support them.

The International Conference on Conceptual Modeling provides a major forum for presenting and discussing current research and applications in which conceptual modeling is the major emphasis. Since the first edition in 1979, the ER conference has evolved into the most prestigious one in the areas of conceptual modeling research and applications. Its purpose is to identify challenging problems facing high-level modeling of future information systems and to shape future directions of research by soliciting and reviewing high-quality applied and theoretical research findings. ER 2004 encompassed the entire spectrum of conceptual modeling. It addressed research and practice in areas such as theories of concepts and ontologies underlying conceptual modeling, methods and tools for developing and communicating conceptual models, and techniques for transforming conceptual models into effective information system implementations.

We solicited forward-looking and innovative contributions that identify promising areas for future conceptual modeling research as well as traditional approaches to analysis and design theory for information systems development.

The Call for Papers attracted 295 exceptionally strong submissions of research papers from 36 countries/regions. Due to limited space, we were only able to accept 57 papers from 21 countries/regions, for an acceptance rate of 19.3%. Inevitably, many good papers had to be rejected. The accepted papers covered topics such as ontologies, patterns, workflows, metamodeling and methodology, innovative approaches to conceptual modeling, foundations of conceptual modeling, advanced database applications, systems integration, requirements and evolution, queries and languages, Web application modeling and development, schemas and ontologies, and data mining.

We are proud of the quality of this year's program, from the keynote speeches to the research papers, with the workshops, panels, tutorials, and industrial papers. We were honored to host the outstanding keynote addresses by Hector Garcia-Molina and Gerhard Weikum. We appreciate the hard work of the organizing committee, with interactions around the clock with colleagues all over the world. Most of all, we are extremely grateful to the program committee members of ER 2004 who generously spent their time and energy reviewing submitted papers. We also thank the many external referees who helped with the review process. Last but not least, we thank the authors who wrote high-quality

research papers and submitted them to ER 2004, without whom the conference would not have existed.

November 2004 Paolo Atzeni, Wesley Chu, and Hongjun Lu

ER 2004 Conference Organization

Honorary Conference Chairs

Peter P. Chen	Louisiana State University, USA
Ruqian Lu	Fudan University, China

Conference Co-chairs

Aoying Zhou	Fudan University, China
Tok Wang Ling	National University of Singapore, Singapore

Program Committee Co-chairs

Paolo Atzeni	Università Roma Tre, Italy
Wesley Chu	University of California at Los Angeles, USA
Hongjun Lu	Univ. of Science and Technology of Hong Kong, China

Workshop Co-chairs

Shan Wang	Renmin University of China, China
Katsumi Tanaka	Kyoto University, Japan
Yahiko Kambayashi[1]	Kyoto University, Japan

Tutorial Co-chairs

Jianzhong Li	Harbin Institute of Technology, China
Stefano Spaccapietra	EPFL Lausanne, Switzerland

Panel Co-chairs

Chin-Chen Chang	Chung Cheng University, Taiwan, China
Erich Neuhold	IPSI, Fraunhofer, Germany

Industrial Co-chairs

Philip S. Yu	IBM T.J. Watson Research Center, USA
Jian Pei	Simon Fraser University, Canada
Jiansheng Feng	Shanghai Baosight Software Co., Ltd., China

[1] Prof. Yahiko Kambayashi died on February 5, 2004.

Demos and Posters Chair

Mong-Li Lee	National University of Singapore, Singapore
Gillian Dobbie	University of Auckland, New Zealand

Publicity Chair

Qing Li	City University of Hong Kong, China

Publication Chair

Shuigeng Zhou	Fudan University, China

Coordinators

Zhongzhi Shi	ICT, Chinese Academy of Science, China
Yoshifumi Masunaga	Ochanomizu University, Japan
Elisa Bertino	Purdue University, USA
Carlo Zaniolo	University of California at Los Angeles, USA

Steering Committee Liaison

Arne Solvberg	Norwegian University of Sci. and Tech., Norway

Local Arrangements Chair

Shuigeng Zhou	Fudan University, China

Treasurer

Xueqing Gong	Fudan University, China

Registration

Xiaoling Wang	Fudan University, China

Webmasters

Kun Yue	Fudan University, China
Yizhong Wu	Fudan University, China
Zhimao Guo	Fudan University, China
Keping Zhao	Fudan University, China

Program Committee

Jacky Akoka	CNAM & INT, France
Hiroshi Arisawa	Yokohama National University, Japan
Sonia Bergamaschi	Università di Modena e Reggio Emilia, Italy
Mokrane Bouzeghoub	Université de Versailles, France
Diego Calvanese	Università di Roma La Sapienza, Italy
Cindy Chen	University of Massachusetts at Lowell, USA
Shing-Chi Cheung	Univ. of Science and Technology of Hong Kong, China
Roger Chiang	University of Cincinnati, USA
Stefan Conrad	Heinrich-Heine-Universität Düsseldorf, Germany
Bogdan Czejdo	Loyola University, New Orleans, USA
Lois Delcambre	Oregon Health Science University, USA
Debabrata Dey	University of Washington, USA
Johann Eder	Universität Klagenfurt, Austria
Ramez Elmasri	University of Texas at Arlington, USA
David W. Embley	Brigham Young University, USA
Johann-Christoph Freytag	Humboldt-Universität zu Berlin, Germany
Antonio L. Furtado	PUC Rio de Janeiro, Brazil
Andreas Geppert	Credit Suisse, Switzerland
Shigeichi Hirasawa	Waseda University, Japan
Arthur ter Hofstede	Queensland University of Technology, Australia
Matthias Jarke	Technische Hochschule Aachen, Germany
Christian S. Jensen	Aalborg Universitet, Denmark
Manfred Jeusfeld	Universiteit van Tilburg, Netherlands
Yahiko Kambayashi	Kyoto University, Japan
Hannu Kangassalo	University of Tampere, Finland
Kamalakar Karlapalem	Intl. Institute of Information Technology, India
Vijay Khatri	Indiana University at Bloomington, USA
Dongwon Lee	Pennsylvania State University, USA
Mong-Li Lee	National University of Singapore, Singapore
Wen Lei Mao	University of California at Los Angeles, USA
Jianzhong Li	Harbin Institute of Technology, China
Qing Li	City University of Hong Kong, Hong Kong, China
Stephen W. Liddle	Brigham Young University, USA
Ee-Peng Lim	Nanyang Technological University, Singapore
Mengchi Liu	Carleton University, Canada
Victor Zhenyu Liu	University of California at Los Angeles, USA
Ray Liuzzi	Air Force Research Laboratory, USA
Bertram Ludäscher	San Diego Supercomputer Center, USA
Ashok Malhotra	Microsoft, USA
Murali Mani	Worcester Polytechnic Institute, USA
Fabio Massacci	Università di Trento, Italy
Sergey Melnik	Universität Leipzig, Germany

Xiaofeng Meng	Renmin University of China, China
Renate Motschnig	Universität Wien, Austria
John Mylopoulos	University of Toronto, Canada
Sham Navathe	Georgia Institute of Technology, USA
Jyrki Nummenmaa	University of Tampere, Finland
Maria E. Orlowska	University of Queensland, Australia
Oscar Pastor	Universidad Politécnica de Valencia, Spain
Jian Pei	Simon Fraser University, Canada
Zhiyong Peng	Wuhan University, China
Barbara Pernici	Politecnico di Milano, Italy
Dimitris Plexousakis	FORTH-ICS, Greece
Sandeep Purao	Pennsylvania State University, USA
Sudha Ram	University of Arizona, USA
Colette Rolland	Univ. Paris 1 Panthéon-Sorbonne, France
Elke Rundensteiner	Worcester Polytechnic Institute, USA
Peter Scheuermann	Northwestern University, USA
Keng Siau	University of Nebraska-Lincoln, USA
Janice C. Sipior	Villanova University, USA
Il-Yeol Song	Drexel University, USA
Nicolas Spyratos	Université de Paris-Sud, France
Veda C. Storey	Georgia State University, USA
Ernest Teniente	Universitat Politècnica de Catalunya, Spain
Juan C. Trujillo	Universidad de Alicante, Spain
Michalis Vazirgiannis	Athens Univ. of Economics and Business, Greece
Dongqing Yang	Peking University, China
Jian Yang	Tilburg University, Netherlands
Ge Yu	Northeastern University, China
Lizhu Zhou	Tsinghua University, China
Longxiang Zhou	Chinese Academy of Science, China
Shuigeng Zhou	Fudan University, China

External Referees

A. Analyti	Wynne Hsu	Noel Novelli
Michael Adams	Stamatis Karvounarakis	Alexandros Ntoulas
Alessandro Artale	Ioanna Koffina	Phillipa Oaks
Enrico Blanzieri	George Kokkinidis	Seog-Chan Oh
Shawn Bowers	Hristo Koshutanski	Justin O'Sullivan
Paolo Bresciani	Kyriakos Kritikos	Manos Papaggelis
Linas Bukauskas	Lotfi Lakhal	V. Phan-Luong
Ugo Buy	Domenico Lembo	Pierluigi Plebani
Luca Cabibbo	Shaorong Liu	Philippe Rigaux
Andrea Calì	Stéphane Lopes	Nick Russell
Cinzia Cappiello	Bertram Ludaescher	Ulrike Sattler
Alain Casali	Chang Luo	Monica Scannapieco
Yu Chen	Gianni Mecca	Ka Cheung Sia
V. Christophidis	Massimo Mecella	Riccardo Torlone
Fang Chu	Carlo Meghini	Goce Trajcevski
Valter Crescenzi	Paolo Merialdo	Nikos Tsatsakis
Michael Derntl	Antonis Misargopoulos	Haixun Wang
Arnaud Giacometti	Paolo Missier	Moe Wynn
Paolo Giorgini	Stefano Modafferi	Yi Xia
Cristina Gómez	Wai Yin Mok	Yirong Yang
Daniela Grigori	Enrico Mussi	Fan Ye

Co-organized by

Fudan University of China
National University of Singapore

In Cooperation with

Database Society of the China Computer Federation
ACM SIGMOD
ACM SIGMIS

Sponsored by

National Natural Science Foundation of China (NSFC)
ER Institute (ER Steering Committee)
K.C. Wong Education Foundation, Hong Kong

Supported by

IBM China Co., Ltd.
Shanghai Baosight Software Co., Ltd.
Digital Policy Management Association of Korea

Table of Contents

Conceptual Modeling IV

Data Warehouse

Schema Integration I

Schema Integration II

Data Classification and Mining I

Data Classification and Mining II

Web-Based Information Systems

Query Processing I

Query Processing II

Web Services I

Web Services II

Schema Evolution

Conceptual Modeling Applications I

Conceptual Modeling Applications II

UML

XML Modeling

Industrial Presentations I: Applications

Industrial Presentations II: Ontology in Applications

Demos and Posters

Panel

5. Cockburn, A. : Structuring use cases with goals. Technical report. Human and Technology, 7691 Dell Rd, Salt Lake City, UT 84121, HaT.TR.95.1 (1995).
6. Dano, B., Briand, H., Barbier, F.: A use case driven requirements engineering process. Third IEEE International Symposium On Requirements Engineering RE'97, Antapolis, Maryland, IEEE Computer Society Press (1997).
7. Rilson, F., Freire, J.: DWARF: AN Approach for Requirements Definition and Management of Data Warehouse Systems. Proceeding of the 11th IEEE International Requirements Engineering Conference, September 08 - 12 (2003), 1090-1099.
8. Golfarelli, M., Maio, D., Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schemes. Proceedings of the 31 st HICSS, IEEE Press (1998).
9. Inmon, W.H. : Building the Data Warehouse. John Wiley and Sons, (1996).
10. Jacobson, I. : The use case construct in object-oriented software Engineering. In *Scenario-based design: envisioning work and technology in system development*, J. M. Carroll (ed.), John Wiley and Sons, (1995) 309–336.
11. Jarke, M., Jeusfeld, A., Quix, C., Vassiliadis, P.: Architecture and Quality in Data Warehouses. Proceedings 10th CAiSE Conference (1998) 93–113.
12. Poe, V. : Building a Data Warehouse for Decision Support. Prentice Hall (1996).
13. Potts, C., Takahashi, K., Anton, A. I. : Inquiry-based requirements analysis. IEEE Software **11(2)**, (1994) 21-23.
14. Prakash, N., Gosain, A.: Requirements Engineering for Data warehouse Development. Proceedings of CAiSE03 Forum (2003).
15. Bruckner, R. M., List, B. : Developing requirements for data warehouses using use cases. Seventh Americas Conference on Information Systems (2003).
16. Rolland, C., Souveyet, C., Achour, C. B.: Guiding goal modelling using scenarios. IEEE Transactions on Software Engineering, Special Issue on Scenario Management. **24(12)** (1998).
17. Rolland, C., Grosz, G., Kla, R. : A proposal for a scenario classification framework. Journal of Requirements Engineering (RE'98), (1998).
18. Rubin, K. S., Golberg, A.: Object behavior analysis. Communications of the ACM. **35(9)** (1992) 48–62.
19. Winter, R., Strauch, B.: A Method for demand driven information requirements analysis in data warehouse projects. Proceeding of the Hawai International conference on system sciences. January 6-9, (2003).

Extending UML for Designing Secure Data Warehouses

Eduardo Fernández-Medina[1], Juan Trujillo[2], Rodolfo Villarroel[3], and Mario Piattini[1]

[1] Dep. Informática, Univ. Castilla-La Mancha, Spain
{Eduardo.FdezMedina,Mario.Piattini}@uclm.es
[2] Dept. Lenguajes y Sistemas Informáticos, Univ.Alicante, Spain
jtrujillo@dlsi.ua.es
[3] Dept. Comput. e Informática, Univ. Católica del Maule, Chile
rvillarr@spock.ucm.cl

Abstract. Data Warehouses (DW), Multidimensional (MD) Databases, and On-Line Analytical Processing Applications are used as a very powerful mechanism for discovering crucial business information. Considering the extreme importance of the information managed by these kinds of applications, it is essential to specify security measures from early stages of the DW design in the MD modeling process, and enforce them. In the past years, there have been some proposals for representing main MD modeling properties at the conceptual level. Nevertheless, none of these proposals considers security measures as an important element in their models, so they do not allow us to specify confidentiality constraints to be enforced by the applications that will use these MD models. In this paper, we discuss the confidentiality problems regarding DW's and we present an extension of the Unified Modeling Language (UML) that allows us to specify main security aspects in the conceptual MD modeling, thereby allowing us to design secure DW's. Then, we show the benefit of our approach by applying this extension to a case study. Finally, we also sketch how to implement the security aspects considered in our conceptual modeling approach in a commercial DBMS.

Keywords: Secure data warehouses, UML extension, multidimensional modeling, OCL

1 Introduction

Multidimensional (MD) modeling is the foundation of Data Warehouses (DW), MD Databases and On Line Analytical Processing Applications (OLAP). These systems are used as a very powerful mechanism for discovering crucial business information in strategic decision making processes. Considering the extreme importance of the information that a user can discover by using these kinds of applications, it is crucial to specify confidentiality measures in the MD modeling process, and enforce them.

On the other hand, information security is a serious requirement which must be carefully considered, not as an isolated aspect, but as an element presented in all stages of the development lifecycle, from the requirement analysis to implementation and maintenance[4, 6]. To achieve this goal, different ideas for integrating security in the system development process are proposed [2, 8], but they only considered information security from a cryptographic point of view, and without considering database and DW specific issues.

There are some proposals that try to integrate security into conceptual modeling. UMLSec [9], where UML is extended to develop secure systems, is probably the most

P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 217–230, 2004.

relevant one. This approach is very interesting, but it only deals with information systems (IS) in general, whilst conceptual database and DW design are not considered. A methodology and a set of models have recently been proposed [5] in order to design secure databases to be implemented with Oracle9i Label Security (OLS) [11]. This approach, based on the UML, is important because it considers security aspects in all stages of the development process, from requirement gathering to implementation. Together with the previous methodology, the proposed Object Security Constraint Language (OSCL) [14], based on the Object Constraint Language (OCL) [19] of UML, allows us to specify security constraints in the conceptual and logical database design process, and to implement these constraints in a concrete database management system (DBMS) such as OLS. Nevertheless, the previous methodology and models do not consider the design of secure MD models for DW's.

In the literature, we can find several initiatives to include security in DW [15, 16]. Many of them are focused on interesting aspects related to access control, multilevel security, its applications to federated databases, applications using commercial tools and so on. These initiatives refer to specific aspects that allow us to improve DW security in acquisition, storage, and access aspects. However, neither of them considers the security aspects comprising all stages of the system development cycle nor considers security in the MD conceptual modeling.

Regarding the conceptual modeling of DW's, various approaches have proposed to represent main MD properties at the conceptual level (due to space constraints, we refer the reader to [1] for a detail comparison between the most relevant ones). These proposals provide their own non-standard graphical notations, and none of them has been widely accepted as a standard conceptual model for MD modeling. Recently, another approach [12, 18] has been proposed as an object-oriented conceptual MD modeling approach. This proposal is a profile of the UML [13], which uses its standard extension mechanisms (stereotypes, tagged values and constraints). However, none of these approaches considers security as an important issue in their conceptual models, so they do not solve the problem of security in DW's.

In this paper, we present an extension of the UML (profile) that allows us to represent main security information of data and their constraints in the MD modeling at the conceptual level. The proposed extension is based on the profile presented in [12] for the conceptual MD modeling because it allows us to consider main MD modeling properties as well as it is based on the UML (designers avoid learning a new specific notation or language). We consider the multilevel security model [17], but focusing on considering aspects regarding *read* operations because this is the most common operation for final user applications. This model allows us to classify both information and users into security classes, and enforce the mandatory access control [17]. By using this approach, we are able to implement secure MD models with any commercial DBMS that is able to implement multilevel databases, such as OLS [11] or DB2 Universal Database (UDB) [3].

The remainder of this paper is structured as follows: Section 2 briefly summarizes the conceptual approach for MD modeling in which we based on. Section 3 proposes the new UML extension for secure MD modeling. Section 4 presents a case study and apply our UML extension for secure MD modeling, Section 5 sketches some further implementation issues. Finally, Section 6 presents the main conclusions and introduces immediate our future work.

2 Object-Oriented Multidimensional Modeling

In this section, we outline our approach, based on the UML [12, 18], for DW conceptual modeling. This approach has been specified by means of a UML profile that contains the necessary stereotypes to represent all main features of MD modeling at the conceptual level [7]. In this approach, structural properties are specified by a UML class diagram in which information is organized into facts and dimensions.

Facts and dimensions are represented by means of fact classes and dimension classes respectively. Fact classes are defined as composite classes in shared aggregation relationships of n dimension classes. The *many-to-many* relations between a fact and a specific dimension are specified by means of the multiplicity 1..* on the role of the corresponding dimension class. In our example in Fig. 1, we can see how the *Sales* fact class has a *many-to-many* relationship with the *Product* dimension.

A fact is composed of measures or fact attributes. By default, all measures are considered to be additive. For non-additive measures, additive rules are defined as constrains. Moreover, derived measures can also be explicitly represented (by /) and their derivation rules are placed between braces near the fact class. Our approach also allows the definition of identifying attributes in the fact class (stereotype OID). In this way *degenerated dimensions* can be considered [10], thereby representing other fact features in addition to the measures for analysis. For example, we could store the ticket number (*ticket_number*) as degenerated dimensions, as reflected in Fig. 1.

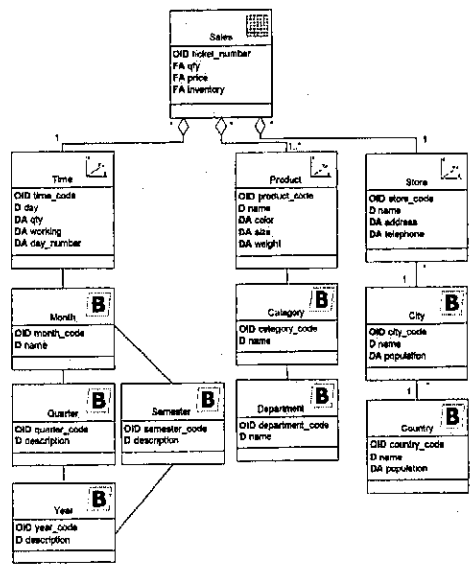

Fig. 1. Multidimensional modeling using the UML

Regarding dimensions, each level of a classification hierarchy is specified by a base class (stereotype Base). An association of base classes specifies the relationship between two levels of a classification hierarchy. These classes must define a Directed Acyclic Graph (DAG) rooted in the dimension class (DAG constraint). The DAG structure can represent both multiple and alternative path hierarchies. Every base class must also contain an identifying attribute (OID) and a descriptor attribute[1] (stereotype D). These attributes are necessary for an automatic generation process into commercial OLAP tools, as these tools store this information on their metadata.

[1] A descriptor attribute will be used as the default label in the data analysis in OLAP tools.

We can also consider non-strict hierarchies (an object at a hierarchy's lower level belongs to more than one higher-level object) and complete hierarchies (all members belong to one higher-class object and that object consists of those members only). These characteristics are specified by means of the multiplicity of the roles of the associations and defining the constraint {completeness} in the target associated class role respectively. See *Store* dimension in Fig. 1 for an example of all kinds of classification hierarchies. Lastly, the categorization of dimensions is considered by means of the generalization / specialization relationships of UML.

3 UML Extension for Secure Multidimensional Modeling

The goal of this UML extension is to allow us to design MD conceptual models, but classifying the information in order to define which properties users must own to be entitled to access the information. Therefore, we have to consider three main stages:

1. Defining precisely the organization of the users that will have access to the MD system. We can define a precise level of granularity considering three ways of organizing the users: Security hierarchy levels (which indicates the clearance level of the user), user Compartments (which indicates a horizontal classification of users), and user Roles (which indicates a hierarchical organization of users according to their roles or responsibilities into the organization).
2. Classifying the information into the MD model. We can define the security information for each element of the model (fact class, dimension class, etc.) by using a tuple composed of a sequence of security levels, a set of user compartments, and a set of user roles. We can also specify security constraints considering this security information. This security information and constraints indicate the security properties that users must own to be able to access the information.
3. Enforcing the mandatory access control (AC). The typical operations executed by final users in this type of systems are query operations. So, the mandatory access control has to be enforced for the *read* operations, whose access control rule is as follows: A user can access to an information only if, a) the security level of the user is greater or equal than the security level of the information, b) all the user compartments that have been defined for the information is owned by the user, and, c) at least one of the user roles defined for the information, is played by the user.

In this paper, we will only focus on the second stage by defining a UML extension that allows us to classify the security elements in a conceptual MD model and to specify security constraints. Furthermore, in Section 5, we sketch a prominent work to deal with the third stage by generating the needed structures in the target DBMS to consider all security aspects represented in the conceptual MD model. Finally, let us point out that the first stage concerns with security policies defined in the organization by managers, and it is out of the scope of this paper.

We define our UML extension for secure conceptual MD modeling following the schema composed of these elements: description, prerequisite extensions, stereotypes/tagged values, well-formedness rules, and comments. For the definition of the stereotypes, we consider an structure that is composed of a name, the base metaclase, the description, the tagged values and a list of constraints defined by means of OCL. For the definition of tagged values, the type of the tagged values, the multiplicity, the description, and the default value are defined.

3.1 Description

This UML extension reuses a set of stereotypes previously defined in [12], and defines new tagged values, stereotypes, and constraints, which enables us to define secure MD models. The 20 tagged values we have defined are applied to certain components that are specially particular to MD modeling, allowing us to represent them in the same model and in the same diagrams that describe the rest of the system. These tagged values will represent the sensitive information for the different elements of the MD modeling (fact class, dimension class, etc.), and they will allow us to specify security constraints depending on this security information and on the value of certain attributes of the model. The stereotypes will help us identify a special class that will define the profile of the system users. A set of inherent constraints are specified in order to define well-formedness rules. The correct use of our extension is assured by the definition of constraints in both natural language and OCL [19].

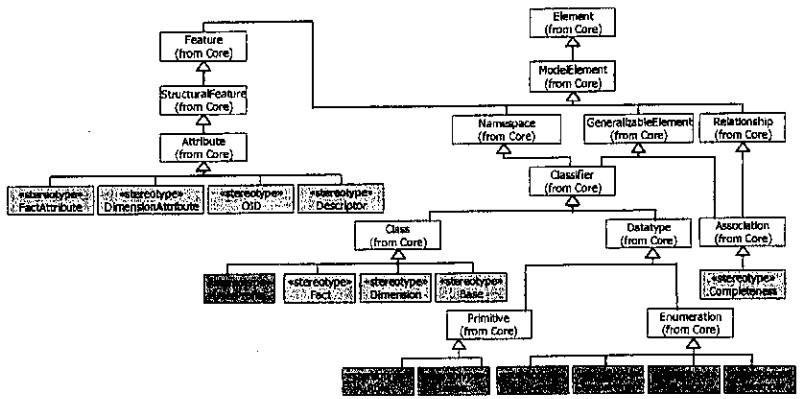

Fig. 2. Extension of the UML with stereotypes

Thus, we have defined 7 new stereotypes: one specializes in the Class model element, two specializes in the Primitive model element and four specialize in the Enumeration model element. In Fig. 2, we have represented portions of the UML metamodel[2] to show where our stereotypes fit. We have only represented the specialization hierarchies, as the most important fact about a stereotype is the base class that the stereotype specializes. In these figures, new stereotypes are colored in a dark grey, whereas stercotypes we reuse from our previous profile [27] are in a light grey and classes from the UML metamodel remain white.

3.2 Prerequisite Extensions

This UML profile reuses stereotypes previously defined in another UML profile [12]. This profile provided the needed stereotypes, tagged values, constraints to accomplish

[2] All the metaclasses come from the *Core Package*, a subpackage of the *Foundation Package*. We based our extension on the UML 1.5 as this is the current accepted standard. To the best of our knowledge, the current UML 2.0 is not the final accepted standard yet.

the MD modeling properly, allowing us to represent main MD properties at the conceptual level. To facilitate the comprehension of the UML profile we present and use in this paper, we provide a brief description of the of these stereotypes in Table 1.

Table 1. Stereotype from the UML profile for conceptual MD modeling [12].

Name	Base Class	Description
Fact	Class	Classes of this stereotype represent facts in a MD model
Dimension	Class	Classes of this stereotype represent dimensions in a MD model
Base	Class	Classes of this stereotype represent dimension hierarchy levels in a MD model
OID	Attribute	Attributes of this stereotype represent OID attributes of Facts, Dimensions or Base classes in a MD model
Fact Attributes	Attribute	Attributes of this stereotype represent attributes of Fact classes in a MD model
Descriptor	Attribute	Attributes of this stereotype represent descriptor attributes of Dimension or Base classes in a MD model
Dimension-Attribute	Attribute	Attributes of this stereotype represent attributes of Dimension or Base classes in a MD model
Completeness	Association	Associations of this stereotype represent the completeness of an association between a Dimension class and a Base class or between two Base classes

3.3 Datatypes

First of all, we need the definition of some new data types to be used in our tagged values definitions. The type Level (Fig. 3 (a)) will be the ordered enumeration composed by all security levels that have been considered (these values, tipically are unclassified, confidential, secret and top secret, but they colud be different). The type Levels (Fig. 3 (b)) will be an interval of levels composed by a lower level and an upper level. The type Role (Fig. 3 (c)) will represent the hierarchy of user roles that can be defined for the organization. The type Roles is a set of role trees or subtrees. The type Compartment (Fig. 3 (d)) is the enumeration composed by all user compartments that have been considered for the organization. The type compartments is a set of user compartments. The type Privilege (Fig. 3 (e)) will be an ordered enumeration composed by all different privileges that have been considered (these values, typically are read, inserte, delete, update, and all). The type Attempt Fig. 3 (f) wille be an ordered enumeration composed by all different access attempt that have been considered (these values arc typically none, all, frustratedAttempt, sucessfullAccess, but they could be different.

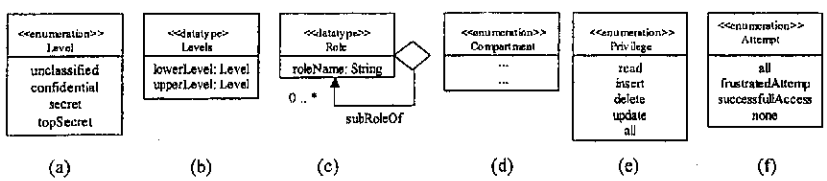

Fig. 3. New Data types

In Fig. 2 we can see the base classes these new stereotypes are specialized from. All the information surrounded in these new stereotypes has to be defined for each

MD model depending on its confidentiality properties, and on the number of users and complexity of the organization in which the MD model will be operative. Finally, we need some syntactic definitions that are not considered in the standard OCL. Particularly, we need the new collection type *Tree* with its typical operations.

3.4 Tagged Values

In this section, we provide the definition of several tagged values for the model, classes, attributes, instances and constraints.

Table 2. Stereotypes of the new data types.

Tagged Values of the Model			
Name	Type	M*	Description
classes	Set(OclType)	1..*	It specifies all classes of the model. This new tagged value is useful in order to navigate through all classes of the model
securityLevels	Sequence (Levels)	1..*	It specifies all security levels (ordered from less to more restrictive) that can be used by the model elements
securityRoles	Role	0..*	It specifies the hierarchical role structure that has been definedfor the organization. This type will be managed as a tree
security-Compartments	Set (Compartment)	0..*	It specifies the set of compartments that have been defined for the organization
Tagged Values of the Class			
Name	Type	M	Description
SecurityLevels	Levels	1..*	It specifies the interval of possible security level values, that aninstance of this class can receive. If the upper and lower security levels are the same, all instances will have the same security level. Otherwise, the concrete instance security level will be defined according to a security constraint
SecurityRoles	Set(Role)	0..*	It specifies a set of user roles. Each role is the root of a subtree of the general user role hierarchy defined for the organization. All instances of this class can have the same user roles, or maybe subtrees of the roles that have been defined for the class. A security constraint can decide the user roles for each instance according to the value of some attributes of the instance
Security-Compartments	Set (Compartment)	0..*	It specifies a set of compartments. All instances of this class can have the same user compartments, or a subset of them. A security constraint can decide the user compartments for each instance according to the value of some attribute of the instance
LogType	Attempt	0..1	It specifies whether the access has to be recorded: none, all access, only frustrated accesses, or only successful accesses
LogCond	OCLExpression	0..1	It specifies whether the access has to be recorded
Involved-Classes	Set(OclType)	1..*	It specifies the classes that have to be involved in a query to be enforced in an exception
ExceptSign	{+,-}	0..1	It specifies if an exception permit (+) or deny (-) the access to instances of this class to a user or a group of users
Except-Privilege	Set(Privilege)	1..*	It specifies the privileges the user can receive or remove
ExceptCond	OCLExpression	0..*	It specifies the condition that users have to fulfill to be affectedby this exception
Tagged Values of the Attribute			
Name	Type	M	Description
SecurityLevels	Levels	1..*	Due to space constraints, we do not include the descriptions of the tagged values of attributes as they are similar to their counterpart tagged values of classes.
SecurityRoles	Set(Role)	0..*	
Security-Compartments	Set (Compartment)	0..*	

Table 2. (Continued)

Tagged Values of the Instance			
Name	Type	M[a]	Description
SecurityLevel	Level	1..*	It specifies the security level of an instance
SecurityRoles	Set(Role)	0..*	It specifies a set of user roles for this instance. Each role is a subtree of the user role hierarchy defined for the organization.
Security-Compartments	Set (Compartment)	0..*	It specifies the set compartments for an instance
Tagged Values of the Constraint			
Name	Type	M	Description
Involved-Classes	Set(OCLType)	0..1	It specifies the classes, that are involved in a query, to be enforced in the constraint

[a] M stands for Multiplicity

Table 2 shows the tagged values of all elements in this extension. All default values of security tagged values of the model are empty collections. On the other hand, the default value of security tagged values for each class is the less restrictive (the lower security level, the security role hierarchy that has been defined for the model and the empty set of compartments). The default value of the security tagged values for attributes is inherited from the class they belong.

If we need to specify the situation in which accesses to the information of a class have to be recorded in a log file for future audit, we should use *LogType* and *LogCond* tagged values together in that class. By default, the value of *LogType* is *none*, so audit is not necessary by default. On the other hand, if we need to specify a security constraint, we can use OCL and the *InvolvedClasses* tagged value to specify in which situation the constraint has to be enforced. By default, the value of this tagged value is the class to which the constraint is associated. Finally, if we need to specify a special security constraint in which a user/s (depending on a condition) can or cannot access to the corresponding class, independently of the security information of that class, we should use *exceptions* together with the following tagged values: *InvolvedClasses, ExceptSign, ExceptPrivilege* and *ExceptCond*. The default value of *InvolvedClasses* is the own class. The default value for *ExceptSign* is +, and for *ExceptPrivilege* is Read.

3.5 Stereotypes

By using all these tagged values, we can specify security constraints on a MD model depending on the values of attributes and tagged values. In this extension, we need to define one stereotype in order to specify other types of security constraints (Table 3). The stereotype *UserProfile* can be necessary to specify constraints depending on a particular information of a user or a group of users, e.g., depending on citizenship, age, etc. Then, the previously-defined data types and tagged values will be used on the *fact*, *dimension* and *base* stereotypes in order to consider other security aspects.

3.6 Well-Formedness Rules

We can identify and specify in both natural language and OCL constraints some well-formedness rules. These rules are grouped in Table 4.

Table 3. Stereotype *UserProfile* of our extension.

Name	UserProfile
Base class	Class
Description	Classes of this stereotype contain all the properties that the systems manage from users
Constraints	– This class has no associations to another classes Self.AssociationsEnd.size()=0 – There is no more than one class of this type **Context** Model Inv self.classes->forAll(ocllsTypeOf(UserProfile))->size()<=1 – The name of a class of this stereotype will be *UserProfile* self.className=*UserProfile*
Tagged Values	None
Icon	

Table 4. Well-Formedness constraints.

– Correct value of the tagged values:
The security levels defined for each class of the model (fact, dimension, and base classes) and for each attribute of each class (OID, FactAttribute, Descriptor, and DimensionAttribute) has to belong to the sequence of security levels that has been defined for the model.
context Model **Inv self.classes->** forAll(c | self.securityLevels -> includesAll(subSequence [a] (c.securityLevels.lowerLevel, c.securityLevels.upperLevel))) **inv self.classes->** forAll(c | c.attributes-> forAll(a | self.securityLevels-> includesAll(subSequence(a.securityLevels.lowerLevel, a.securityLevels.upperLevel)))
The set of user roles defined for each class and attribute of the model has to be a subtree of the roles tree that has been defined for the model.
context Model inv self.classes-> forAll(c | c.Roles-> forAll(r | self.Role->includesAll(r))) Inv self.classes-> forAll(c | c.attributes-> forAll(a | a.Roles-> forAll (r | self.Role-> includesAll(r))))
The set of user compartments defined for each class and attribute of the model has to be a subset of the compartments that have been defined for the model.
context Model inv self.classes-> forAll(c | c.Compartments-> forAll(comp | self.Compartments->includes(comp))) Inv self.classes-> forAll(c | c.attributes-> forAll(a | a.Compartments-> forAll (comp | self.Compartments-> includes(comp))))
– The security information of instances:
The security level of the instance of a class has to be included in the ranking of security levels that has been defined for the class. The same rule is applicable for the instances of attributes.
context Model inv self.classes-> forAll(c | c.allInstances -> forAll (i | self.securityLevels-> subSequence(c.securityLevels.lowerLevel, c.securityLevels.upperLevel)-> includes(i.securityLevel)))
The user roles of an instance of a class, has to be subtress of the roles trees that have been defined for the class. The same rule is applicable for the instance of attributes.
context Model Inv self.classes-> forAll(c | c.allInstances -> forAll (i | c.securityRoles-> includesAll(i.securityRoles)))
The user compartments of an instance of a class, has to be a subset of the compartments that have been defined for the class. The same rule is applicable for the instance of attributes.
context Model Inv self.classes-> forAll(c | c.allInstances -> forAll (i | i.securityCompartments-> includesAll(i.securityCompartments)))
– Relationship between the security information of classes and its attributes:
The security levels defined for an attribute have to be equal or more restricted that the security levels defined for its class. The same rule is applicable for the role hierarchies and user compartments.
context Model inv self.classes-> forAll(c | c.attributes-> forAll(a | self.securityLevels-> subSequence(c.securityLevels.lowerLevel, a.securityLevels.upperLevel)-> includesAll(self.securityLevels-> subSequence(a.securityLevels.lowerLevel,a.securityLevels.upperLevel))) Inv self.classes-> forAll(c | c.attributes-> forAll (a | c.securityRoles-> includesAll(a.securityRoles))) Inv self.classes-> forAll (c | c.attributes-> forAll (a | c.securityCompartments-> includesAll(a.securityCompartments)))

[a] The type of the arguments of *subSequence* collection is integer, but for the sake of readability, we consider that the arguments can be elements of the subSequence. The correct expression should be subSequence(self.securityLevels->indexOf(c.securityLevels.lowerLevel),self.securityLevels ->indexOf (c.securityLevels.upperLevel). We consider this simplification in all uses of *subSequence* operation.

Table 4. (Continued)

– Categorization of dimensions:
When a dimension class is specialized in several base classes, the security levels of the subclasses have to be equal or more restrictive that the security levels of the superclass. The same rule is applicable for role hierarchies and user compartments.
context Model **inv** self.classes-> forAll(c | c.subClasses-> forAll(s | self.securityLevels-> subSequence(c.securityLevels.lowerLevel, s.securityLevels.upperLevel)-> includesAll(self.securityLevels-> subSequence (s.securityLevels.lowerLevel,s.securityLevels.upperLevel)) **inv** self.classes-> forAll(c | c.subClasses-> forAll (s | c.securityRoles-> includesAll(s.securityRoles)) **inv** self.classes-> forAll (c | c.subClasses-> forAll (s | c.securityCompartments-> includesAll(s.securityCompartments)))
– Classification hierarchies. As a general rule, we can consider that the more specific the information is, the more restricted its access is:
If the class A has a 1..* association with the class B, means that information of A groups information of B, so B is more specific than A. The security level defined for the class B has to be more restrictive than the security level defined for the class A. This rule is also applicable for user roles and compartments.
context Model **inv** self.classes-> forAll(c | c.associationEnd-> forAll (a | c.a.multiplicity>1 implies self.securityLevels-> subsequence(c.securityLevels.lowerLevel, a.securityLevels.upperLevel)-> includesAll(self.securityLevels-> subSequence(a.securityLevels.lowerLevel, a.securityLevels.upperLevel)) **inv** self.classes-> forAll(c | c.associationEnd-> forAll (a | c.a.multiplicity>1 implies c.securityRoles-> includesAll(a.securityRoles)) **inv** self.classes-> forAll(c | c.associationEnd-> forAll (a | c.a.multiplicity>1 implies c-securityCompartments-> includesAll (a.securityCompartments)))
If the class A has a *..* association with the class B, the designer has to decide which class contains the most specific information. This well-formedness rule cannot be specified because it depends of design decisions.
– Derived attributes:
The security levels of a derived attribute have to be equal or more restricted than the attributes which it is based on. The same rule is applicable for user roles and compartments. By default, the derived attributed inherit the security information of the attribute it is based on.
context Model **inv** self.classes-> forAll(c | c.attributes -> forAll(a | a.derived implies a.derivedFrom-> forAll (d | self.securityLevels-> subsequence(a.securityLevels.lowerLevel, d.securityLevels.upperLevel)-> includesAll(self.securityLevels-> subSequence(d.securityLevels.lowerLevel, d.securityLevels.upperLevel)) **inv** self.classes-> forAll(c | c.attributes -> forAll(a | a.derived implies a.derivedFrom-> forAll (d | d.securityRoles -> includesAll (a.securityRoles) **inv** self.classes-> forAll(c | c.attributes -> forAll(a | a.derived implies a.derivedFrom-> forAll (d | d.securityCompartments -> includesAll (a.securityCompartments)
– Combination of dimensions:
A query on the fact class has to consider the security information that has been defined for that class.
A query that involves the combination of a dimension class (or maybe a base class) and a fact class has to consider the combination of the security information on the dimension (or base) class and on the fact class. The security levels of the combination will be the most restrictive from the security levels of the dimension (or base) class and the fact class. The same rule is applicable for the user roles and compartments.
A query that involves the combination of several dimension classes, and the fact class, has to consider the combination of the security information of all classes. The security levels of the combination will be the most restrictive one from the security levels of all classes. The same rule is applicable for the user roles and compartments.

3.7 Comments

Many of the previous constraints are very intuitive, although we have to ensure its fulfillment, otherwise the system can be inconsistent. Moreover, the designer can specify security constraints with OCL. If the security information of a class or an attribute depends on the value of an instance attribute, it can be specified as an OCL expression (Fig. 4). Normally, security constraints defined for stereotypes of classes (fact, dimension and base) will be defined by using a UML note attached to the corresponding class instance. We do not impose any restriction on the content of these notes in order to allow the designer the greatest flexibility, only those imposed by the tagged values definitions. The connection between a note and the element it applies to is shown by a dashed line without an arrowhead as this is not a dependency [13].

4 A Case Study Applying Our Extension for Secure MD Modeling

In this section, we apply our extension to the conceptual design of a secure DW in the context of a reduced health-care system. The simplified hierarchy of the system user roles is as follows: *HospitalEmployees* are classified into *health* and *non-health* users, *health* users can be *Doctors* or *Nurses* and *non-health* users can be *Maintenance* or *Administrative*. The defined security levels are *unclassified*, *secret* and *topSecret*.

1. Fig. 4 shows an MD model that includes a fact class (*Admission*), two dimensions (*Diagnosis* and *Patient*), two base classes (*Diagnosis_group* and *City*), and a class (*UserProfile*). *UserProfile* class (stereotype *UserProfile*) contains the information of all users who will have access to this MD model. *Admission* fact class -stereotype *Fact*- contains all individual admissions of patients in one or more hospitals, and can be accessed by all users who have *secret* or *top secret* security levels -tagged value *SecurityLevels (SL) of classes*-, and play *health* or *administrative* roles -tagged value *SecurityRoles (SR) of classes*-. Note that the *cost* attribute can only be accessed by users who play *administrative* role -tagged value *SR of attributes- Patient* dimension contains the information of hospital patients, and can be accessed by all users who have *secret* security level –tagged value *SL*-, and play *health* or *administrative* roles –tagged value *SR*-. The *Address* attribute can only be accessed by users who play *administrative* role –tagged value *SR* of attributes-. *City* base class contains the information of cities, and it allows us to group patients by cities. Cities can be accessed by all users who have *confidential* security level –tagged value *SL*-. *Diagnosis* dimension contains the information of each diagnosis, and can be accessed by users who play *health* role –tagged value *SR*-, and have *secret* security level –tagged value *SL*-. Finally, *Diagnosis_group* contains a set of general groups of diagnosis. Diagnosis groups can be accessed by all users who have *confidential* security level –tagged value SLs-.

Several security constraints have been specified by using the previously defined constraints, stereotypes and tagged values (the number of each numbered paragraph corresponds to the number of each note in Fig. 4):

2. The security level of each instance of *Admission* is defined by a security constraint specified in the model. If the value of the *description* attribute of the *Diagnosis_group* which belongs to the *diagnosis* that is related to the *Admission* is *cancer* or *AID*, the security level –tagged value *SL*- of this admission will be *top secret*, otherwise *secret*. This constraint is only applied if the user makes a query whose the information comes from the *Diagnosis* dimension or *Diagnosis_ group* base classes together with the *Patient* dimension –tagged value *involvedClasses*-.
3. The security level –tagged value *SL*- of each instance of *Admission* can also depend on the value of the *cost* attribute, which indicates the price of the admission service. In this case, the constraint is only applicable for queries that contain information of the *Patient* dimension –tagged value *involvedClasses*-.
4. The tagged value *logType* has been defined for the *Admission* class, specifying the value *frustratedAttempts*. This tagged value specifies that the system has to record, for future audit, the situation in which a user tries to access to information of this fact class, and the system denies it because of lack of permissions.

5. For confidentiality reasons, we could deny access to admission information to users whose working area is different than the area of a particular admission instance. This is specified by another exception in *Admission* fact class, considering tagged values *involvedClasses, exceptSign* and *exceptCond.*

If patients are special users of the system, they could access to their own information as patients (e.g., for querying their personal data). This constraint is specified by using the *excepSign* and *exceptCond* tagged values in the *Patient* class.

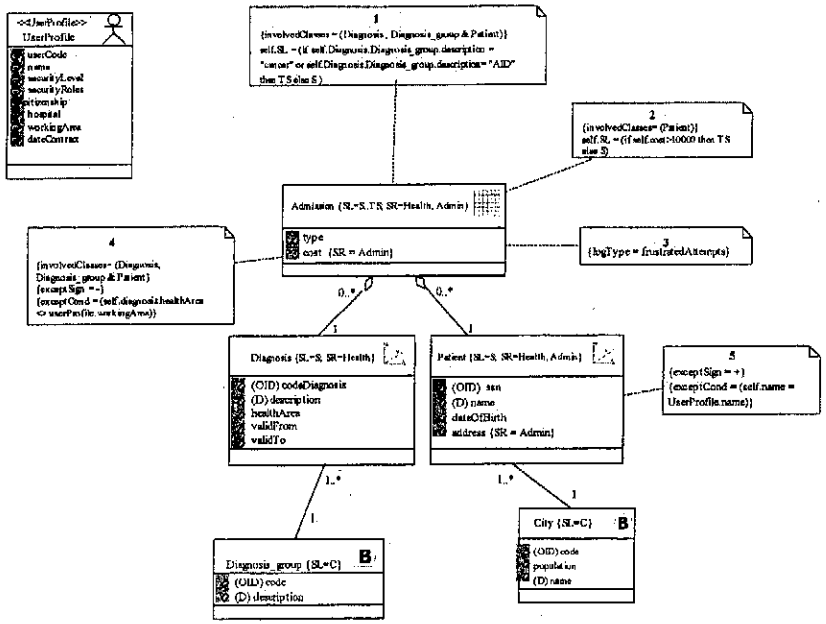

Fig. 4. Example of multidimensional model with security information and constraints[3]

5 Implementation

Oracle*9i* Label Security [11] allows us to implement multilevel databases. It defines labels that are assigned to the rows and users of a database that contain confidentiality information and authorization information for rows and users respectively. Moreover, OLS allows us to specify labeling functions and predicates that are triggered when an operation is executed, and which define the value of security labels.

A secure MD model can be implemented by OLS. The two main security elements that we include in this UML extension are confidentiality information of data, and security constraints. The basic concepts of a MD model (facts, dimension and base classes) are implemented as tables in a relational database. The security information

[3] Version 2 of OCL considers a special syntaxis for enumerations (EnumTypeName::Enum Literal-Value), but in this example, for the sake of readability, we consider only EnumLiteralValue.

of the MD model can be implemented by the security labels of OLS, and the security constraints can be implemented by labeling functions and predicates of OLS.

For instance, we could consider the table *Admission* with *CodeAdmission, Type, Cost, CodeDiagnosis* and *PatientSSN* columns. This table will have a special column to store the security label for each instance. For each instance, this label will contain the security information that has been specified in Fig. 4 (Security *Level=Secret.. TopSecret; SecurityRoles=Health, Admin*). But this security information depends on several security constraints that can be implemented by labeling functions. Table 5 (1) shows an example in which we implement the security constraints labeled with number 2 in Fig. 4. If the value of *Cost* column is greater than 10000 then the security label will be composed of *TopSecret* security level and *Health* and *Admin* user roles, else the security label will be composed of *Secret* security level and the same user roles. Table 5 (2) shows how to link this labeling function with *Admission* table.

Table 5. Example of labeling function in OLS.

```
(1) CREATE FUNCTION Which_Cost (Cost: Integer) Return LBACSYS.LBAC_LABEL
    As MyLabel varchar2(80);
    Begin
      If Cost>10000 then MyLabel := 'TS::Health,Admin'; else MyLabel := 'S::Health,Admin'; end if;
      Return TO_LBAC_DATA_LABEL('MyPolicy', MyLabel);
    End;
(2) APPLY_TABLE_POLICY ('MyPolicy', 'Admission', 'Scheme',, 'Which_Cost')
```

6 Conclusions and Future Work

In this paper, we have presented an extension of the UML that allows us to represent main security aspects in the conceptual modeling of Data Warehouses. This extension contains the needed stereotypes, tagged values and constraints for a complete and powerful secure MD modeling. These new elements allow us to specify security aspects such as security levels on data, compartments and user roles on the main elements of a MD modeling such as facts, dimensions and classification hierarchies. We have used the OCL to specify the constraints attached to these new defined elements, thereby avoiding an arbitrary use of them. We have also sketched how to implement a secure MD model designed with our approach in a commercial DBMS. The main relevant advantage of this approach is that it uses the UML, a widely-accepted object-oriented modeling language, which saves developers from learning a new model and its corresponding notations for specific MD modeling. Furthermore, the UML allows us to represent some MD properties that are hardly considered by other conceptual MD proposals.

Our immediate future work is to extend the implementation issues presented in this paper to allow us to use the considered security aspects when querying a MD model from OLAP tools. Moreover, we also plan to extend the set of privileges considered in this paper to allow us to specify security aspects in the ETL processes for DWs.

Acknowledgements

This research is part of the CALIPO and RETISTIC projects, supported by the Dirección General de Investigación of the Ministerio de Ciencia y Tecnología.

References

1. Abelló, A., Samos, J., and Saltor, F., *A Framework for the Classification and Description of Multidimensional Data Models.* 12th International Conference on Database and Expert Systems Aplications. LNCS 2113., 2001: pp. 668-677.
2. Chung, L., Nixon, B., Yu, E., and Mylopoulos, J., *Non-functional requirements in software engineering.* 2000, Boston/Dordrecht/London: Kluwer Academic Publishers.
3. Cota, S., *For Certain Eyes Only.* DB2 Magazine, 2004. **9**(1): pp. 40-45.
4. Devanbu, P. and Stubblebine, S., *Software engineering for security: a roadmap*, in *The Future of Software Engineering*, Finkelstein, A., Editor. 2000, ACM Press. pp. 227-239.
5. Fernández-Medina, E. and Piattini, M., *Designing Secure Database for OLS*, in *Database and Expert Systems Applications: 14th International Conference (DEXA 2003)*, Marik, V., Retschitzegger, W., and Stepankova, O., Editors. 2003, Springer. LNCS 2736: Prague, Czech Republic. pp. 886-895.
6. Ferrari, E. and Thuraisingham, B., *Secure Database Systems*, in *Advanced Databases: Technology Design*, Piattini, M. and Díaz, O., Editors. 2000, Artech House: London.
7. Gogolla, M. and Henderson-Sellers, B. *Analysis of UML Stereotypes within the UML Metamodel.* in UML'02. Springer, LNCS 2460. pp. 84-99. Dresden, Germany.
8. Hall, A. and Chapman, R., *Correctness by Construction: Developing a Commercial Secure System.* IEEE Software, 2002. **19**(1): pp. 18-25.
9. Jürjens, J., *UMLsec: Extending UML for secure systems development*, in *UML'02* Springer. LNCS 2460.: Dresden, Germany. pp. 412-425.
10. Kimball, R., *The data warehousing toolkit. 2 edn.* 1996: John Wiley.
11. Levinger, J., *Oracle label security. Administrator's guide. Release 2 (9.2).* 2002: http://www.csis.gvsu.edu/GeneralInfo/Oracle/network.920/a96578.pdf.
12. Luján-Mora, S., Trujillo, J., and Song, I.Y. *Extending the UML for Multidimensional Modeling.* in *UML'02.* Springer, LNCS 2460. pp. 290-304. Dresden, Germany.
13. OMG, *Object Management Group: Unified Modeling Language Specification 1.5.* 2004.
14. Piattini, M. and Fernández-Medina, E. *Specification of Security Constraint in UML.* in *35th Annual 2001 IEEE Intl. Carnahan Conf.on Security Technology.* London pp. 163-171
15. Priebe, T. and Pernul, G. *Towards OLAP Security Design - Survey and Research Issues.* in *3rd ACM International Workshop on Data Warehousing and OLAP (DOLAP'00).* Washington DC, USA. pp. 33-40
16. Rosenthal, A. and Sciore, E. *View Security as the Basic for Data Warehouse Security.* in *2nd International Workshop on Design and Management of Data Warehouse (DMDW'00).* Sweden. pp. 8.1-8.8
17. Samarati, P. and De Capitani di Vimercati, S., *Access control: Policies, models, and mechanisms*, in *Foundations of Security Analysis and Design*, Focardi, R. and Gorrieri, R., Editors. 2000, Springer: Bertinoro, Italy. pp. 137-196.
18. Trujillo, J., Palomar, M., Gómez, J., and Song, I.Y., *Designing Data Warehouses with OO Conceptual Models.* IEEE Computer, special issue on DWs, 2001(34): pp. 66-75.
19. Warmer, J. and Kleppe, A., *The Object Constraint Language Second Edition. Getting Your Models Ready for MDA*, 2003: Addison Wesley.

Data Integration with Preferences Among Sources*

Gianluigi Greco[1] and Domenico Lembo[2]

[1] Dip. di Matematica, Università della Calabria, Italy
ggreco@mat.unical.it
[2] Dip. di Informatica e Sistemistica, Università di Roma "La Sapienza", Italy
lembo@dis.uniroma1.it

Abstract. Data integration systems represent today a key technological infrastructure for managing the enormous amount of information even more and more distributed over many data sources, often stored in different heterogeneous formats. Several different approaches providing transparent access to the data by means of suitable query answering strategies have been proposed in the literature. These approaches often assume that all the sources have the same level of reliability and that there is no need for preferring values "extracted" from a given source. This is mainly due to the difficulties of properly translating and reformulating source preferences in terms of properties expressed over the global view supplied by the data integration system. Nonetheless preferences are very important auxiliary information that can be profitably exploited for refining the way in which integration is carried out. In this paper we tackle the above difficulties and we propose a formal framework for both specifying and reasoning with preferences among the sources. The semantics of the system is restated in terms of preferred answers to user queries, and the computational complexity of identifying these answers is investigated as well.

1 Introduction

The enormous amount of information even more and more distributed over many data sources, often stored in different heterogeneous formats, had boosted in recent years the interest for data integration systems. Roughly speaking, a data integration system offers transparent access to the data by providing users with the so-called global schema, which they can query in order to extract data relevant for their aims. Then, the system is in charge of accessing each source separately, and combining local results into the global answer. The means that the system exploit to answer users' queries is the mapping specifying the relationship between the sources and the global schema [16].

However, data at the sources, may result mutually inconsistent, because of the presence of integrity constraints specified on the global schema in order to

* This work has been partially be supported by the European Commission FET Programme Project IST-2002-33570 INFOMIX.

P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 231–244, 2004.