ENC04

Fifth Mexican International Conference on

Computer Science

Colima, Mexico
20-24 September 2004

2004

IEEE COMPUTER SOCIETY

IEEE

PROCEEDINGS OF THE

FIFTH MEXICAN INTERNATIONAL CONFERENCE IN COMPUTER SCIENCE

ENC 2004

September 20-24, 2004
Colima, México

Sponsored by

Sociedad Mexicana de Ciencias de la Computación
Consejo Nacional de Ciencia y Tecnología
National Science Foundation
Universidad de Colima
Universidad Michoacana de San Nicolás de Hidalgo
Centro de Investigación Científica y de Educación Superior de Ensenada
Centro de Investigación en Matemáticas
DCC, Universidad de Chile
Universidad de las Américas, Puebla
Instituto Tecnológico Autónomo de México
Instituto Tecnológico y de Estudios Superiores de Monterrey
Sun Microsystems
Microsoft

Edited by

Ricardo Baeza-Yates, José Luis Marroquín and Edgar Chávez

IEEE
COMPUTER
SOCIETY

Los Alamitos, California
Washington • Brussels • Tokyo

IEEE Computer Society Order Number P2160
ISBN 0-7695-2160-6
Library of Congress 2004106722

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: + 1 800 272 6657
Fax: + 1 714 821 4641

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN
Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553

Individual paper REPRINTS may be ordered at: reprints@computer.org

Editorial production by Frances M. Titsworth

Cover art production by Alex Torres

Printed in the United States of America by The Printing House

IEEE COMPUTER SOCIETY

IEEE

Table of Contents

ENC 2004

Software Engineering

Evolutionary Computing

Computer Graphics and Image Processing

Computer Vision

Pattern Recognition and Machine Learning

Human-Computer Interaction

Vision and Robotics

Optimization

FOREWORD

It is with great pleasure that we welcome you to the Fifth Mexican International Conference in Computer Science ENC 2004. The SMCC (Sociedad Mexicana de Ciencias de la Computación) has organized these academic meetings to bring together scientists and students working in all fields of Computer Science from the major research institutions and universities in Mexico, as well as guest institutions from around the world, to share the latest advances in computer science research. This event seeks to promote research exchanges that will benefit our academic community. We are proud to have a high quality technical program, as exemplified by the papers collected in these proceedings, which provides evidence of a maturing computer science community in Mexico. This year we have co-located two events, the '3rd International Conference on Parallel Computing Systems' (PCS'04) and the 'International Conference on Reconfigurable Computing and FPGAs'. The proceedings of the former event are included in this issue. The proceedings of the latter event are published in a separate issue.

In putting together this conference, we have had the pleasure of working with an outstanding group of volunteers who dedicated substantial portions of their already busy schedules to create a superb conference. We thank them all for their hard work on behalf of this conference and applaud the program they have helped create. In particular we would like to thank the hard working group of local sponsors at the Universidad de Colima, led by Juan Contreras-Castillo.

We extend our thanks to our sponsoring SMCC organization, its executive committee for the confidence it has shown on us and their constant advice, and its staff who supported us in innumerable ways.

We extend a special acknowledgment to our sponsoring organizations: Consejo Nacional de Ciencia y Tecnología (CONACyT) and the National Science Foundation (NSF), as well as our corporate sponsors: Sun Microsystems and Microsoft. Their support has helped make this conference possible.

We sincerely hope you find these conference proceedings to be enriching.

Edgar Chávez
ENC'04 General Chair

Jesus Favela
President SMCC

PREFACE

ENC'04 is a symposium in computer science, organized by The Mexican Society of Computer Science (SMCC), in collaboration with several educational and research institutions across the country. The ENC started as a biannual conference gathering researchers, educators and scholars from around the world to present and discuss progresses, challenges and opportunities in Computer Science. This year we shortened the period and began an annual meeting.

The venue for ENC'04 was the city of Colima. Our host was the Universidad de Colima. With the enthusiastic participation of the computer science community, we hope ENC'04 has surpassed the achievements of previous editions of the conference, which were held in Queretaro (1997), Pachuca (1999), Aguascaliente (2001) and Tlaxcala (2003)

With this fifth edition, ENC aimed at continuing to provide a forum in Mexico for presentation and discussion of advances and new research in computer science; strengthening the sense of community among computer science researchers and practitioners in Mexico; as well as reinforcing our academic community's links with the global community through a high-quality international technical program.

The selection of papers followed a strict refereeing process by a renowned international committee. In addition to research papers in all areas of computer science, ENC featured a rich mix of discussion panels, plenary lectures and hands-on tutorials given by international lecturers and instructors invited by each of the Special Interest Groups of SMCC. We received over 200 contributions, and 47 were selected to be presented and published in this proceedings volume.

We also had six internationally renowned scholars as keynote speakers

Ricardo Baeza-Yates, José Luis Marroquín and Edgar Chávez
The editors
Colima, México, September 21 2003.

Organizing Committee

SMCC

Jesus Favela, *Presidente, CICESE*
Marcelo Mejia, *Vicepresidente, ITAM*

Conference Chair

Edgar Chávez, *U. Michoacana*

Program Committee

José Luis Marroquín, *CIMAT*
Ricardo Baeza-Yates, *U. Chile, Chile*

Organizing Committee

Juan Contreras-Castillo, *U. de Colima*
Lourdes Feria, *U. de Colima*

Publicity

Cuauhtémoc Rivera Loaiza, *U. Michoacana and U. de Chile, Chile*
Luis Villaseñor, *INAOEP*
Diana Caballero, *UDLAP*

Workshops

Alexander Gelbuck, *CIC*
Miguel Arias, *INAOE*

Invited Speakers

Sergio Rajsbaum, *UNAM*
Oscar Mayora, *ITESM*

Panels

Alfredo Sanchez, *UDLAP*

Posters

Maria Osorio, *BUAP*
Héctor Pérez, *UASLP Felipe Rolando Menchaca, IPN*

Doctoral Consortium

César Osuna, *IMP*

Finances

David Sol, *UDLAP*

Tutorials

Rogelio Davila, *ITESM*
Juan Flores, *U. Michoacana*

Sponsors

Sociedad Mexicana de Ciencias de la Computación

Consejo Nacional de Ciencia y Tecnología

National Science Foundation

Universidad de Colima

Universidad Michoacana de San Nicolás de Hidalgo

Centro de Investigación Científica y de Educación Superior de Ensenada

Centro de Investigación en Matemáticas

DCC, Universidad de Chile

Universidad de las Américas, Puebla

Instituto Tecnológico Autónomo de México

Instituto Tecnológico y de Estudios Superiores de Monterrey

Sun Microsystems

Microsoft

Program Committee

Ricardo Baeza-Yates, *U. Chile, Chile (co-chair)*
Karin Becker, *PUC-RS, Brazil*
Jose Blakeley, *Microsoft*
Alex Buchmann, *Technical University of Darmstadt, Germany*
Francisco Cervantes, *ITAM, Mexico*
Edgar Chavez, *U. Michoacana, Mexico*
Paolo Ciaccia, *U. Bologna, Italy*
Carlos Coello, *CINVESTAV, Mexico*
Ulises Cortes, *UPC, Spain*
Dominique Decouchant, *IMAG, France*
Vladimir Estivil Castro, *U. Griffith, Australia*
Klaus Ecker, *TU-Clausthal, Germany*
Martin Farach-Colton, *Rutgers, USA*
Rodolfo Garcia, UANL, *Mexico*
Alexander Gelbukh, *CIC-IPN, Mexico*
Antonio F. Gomez Skarmeta, *U. Murcia, Spain*
Jose Luis Gordillo, *ITESM, Mexico*
Adolfo Guzman Arenas, *CIC-IPN, Mexico*
Arturo Hernandez, *CIMAT, Mexico*
Ela Hunt, *Glasgow, UK*
Alejandro Jaimes, *FujiXerox Nakai Research Center, Japan*
Evangelos Kranakis, *Carleton, Canada*
Aurelio Lopez, *INAOE, Mexico*
Jose Luis Marroquin, *CIMAT, Mexico (co-chair)*
Hanna Oktaba, *UNAM, Mexico*
Sergio Ochoa, *U. Chile, Chile*
Eugenio Oñate, *Spain*
Mario Piatini, U. *Castilla La Mancha, Spain*
Christian Lemaitre, *LANIA, Mexico*
Marcelo Mejia, *ITAM, Mexico*
Carlos Montes de Oca, *CIMAT, Mexico*
Gonzalo Navarro, *U. Chile, Chile*
Sergio Rajsbaum, *UNAM, Mexico*
Jesus Rico Melgoza, *U. Michoacana, Mexico*
Isaac Rudomin, *ITESM, Mexico*
Alfredo Sanchez, *UDLA, Mexico*
Humberto Sossa, *CIC-IPN, Mexico*
Enrique Sucar. *ITESM Cuernavaca. Mexico*
Andrei Tchernykh, *CICESE, Mexico*
Denys Tristram, *IMAG, France*
Jorge Urrutia, *UNAM, Mexico*
Gabriel Valiente, *UPC, Spain*
Santosh Vempala, *MIT, USA*
Alfredo Viola, *U. de la Republica, Uruguay*
Alan Watt, *U. Sheffield, UK*
Roger Z. Rios, *UANL, Mexico*

List of Reviewers

Adlemo Anders
Baillot Patrick
Barrera Edmundo
Blois Marcelo
Buckles Bill
Calderón Felix
Calvo Francisco
Collazos César
Cuayáhuitl Heriberto
Diaz Marissa
Fernandez Eduardo
Flores Juan
Galesi Nicola
Garcia Felix
Gilbert Anna
Gokcen Ibrahim
Gonzalez Marco
Hernandez Benjamin
Hidalgo Hugo
Incera Jose
Lemus Cuauhtémoc
Littman Michael
Lozano Antoni
Martínez Gregorio
Martínez Francisco
Millán Erik
Moreira Flavio
Ormeo Emilio
Padilla Gerardo
Romero Leonardo
Ruiz Francisco
Ruiz Hector
Ruiz Pedro
Shokoufandeh Ali
Villarreal César
Villaseñor Luis
Vizcaino Aurora
Von-Borstel Fernando
Zapata Sergio

PCS Program Committee

Jacek Blazewicz, *Politechnika Poznanska, Poznan, Poland*
Klaus Ecker, *T. U., Clausthal, Germany*
Jesus Favela, *CICESE, Mexico*
Jean-Luc Gaudiot, *U. California, USA*
Andrei Tchernykh, *CICESE, Mexico*
Denis Trystram, *IMAG, France*
Roman Wyrzykowski, *IMCS, Poland*
Alfredo Goldman, *U. Sao Paulo, Brazil*
Lonnie Welch, *Ohio University, USA*
Gerhard Fohler, *Mardalen University, Sweden*
Claudia Roncancio, *LSR-IMAG , France*
Bruno Raffin, *INRIA Grenoble, France*
Frederic Guinand, *Universite Le Havre, France*
Christophe Cérin, *LARIA, Amiens, France*
Victor Ivannikov, *Institute for System Programming, Russia*

PCS Additional Reviewers

Avetisyan, Arutyun
Lazarev, Alexander
Salas, Alfredo
Rodríguez, Antonio
Hirales, Adán

Incorporating Security Issues in the Information Systems Design

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

(1) Departamento de Computación e Informática. Universidad Católica del Maule (Chile)
rvillarr@spock.ucm.cl
(2) Departamento de Informática. Universidad de Castilla-La Mancha (Spain)
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Abstract

Nowadays, security solutions are mainly focused on providing security defences (such as firewalls, routers, configuration server, password and encryption) instead of solving one of the main reasons of security problems that refers to an appropriate software design. Fortunately, there have been developed new methodologies incorporating security into their development processes. This paper allows to make a comparison of six secure systems design methodologies and our proposal for the segure multidimensional conceptual modeling. The analysed methodologies fulfil criteria partially and in this paper, we make it clear that security aspects cannot be completely specified by these methodologies since they have a series of limitations that we have to take into account. At the same time, each one of these methodologies comprises very important aspects concerning security that can be used as a basis for new methodologies or extensions that may be developed.

1. Introduction

Security is a "horizontal" aspect of software development that affects very closely each component of an application and, its integration into the process of software development is not appropriately understood. In the field of security, we can distinguish between the concept of security (capability of a system to manage, protect and distribute sensitive information) and the concept of safety (absence of catastrophic consequences to the environment). The issue of security will be dealt with in this paper.

According to ISO/IEC 15408-1 [8], the concept of security refers to the capability of a software product to protect data and information in order to avoid that unauthorized individuals or systems are able to read and modify them and not to deny access to authorized staff.

Castano et al. [2] refer to computing security as the protection of information against unauthorized queries, inappropriate modifications or the lack of availability of a service in a given moment.

We can see that both definitions of security are basically similar according to the following components: confidentiality (to prevent, to detect, to avoid the improper revelation of information), integrity (to prevent, to detect, to avoid the undue modification of information) and availability (to prevent, to detect, to avoid the denial of access to the services provided by the system). Sometimes, databases and data warehouses also store information regarding private or personal aspects of individuals, such as identification data, medical data or even religious beliefs, ideologies, or sexual tendencies. In this case, confidentiality is redefined as privacy. At present, ensuring appropriate information privacy is a pressing concern for many companies since there is privace legislation such as the United States'HIPAA (Health Insurance Portability and Accountability Act), Gramm-Leach-Bliley Act, Sarbanes-Oxley Act, and the EU's Safe Harbour Law.

The above mentioned components (confidentiality, integrity and availability) should be taken into account in every information systems (IS) development. However, ISs security is considered once the system is developed. This approach is known as "Penetrate and Patch" [15], and it has been proved to have bad results. It is less common that developers take this aspect into consideration in earlier stages such as analysis and design. Solutions are mainly focused on providing security defences (such as firewalls, routers, configuration server, password and encryption) instead of solving one of the main reasons of security problems that refers to an appropriate software design [5]. In simple economic terms, to find and eliminate mistakes in a software system before it is finished is cheaper and more effective than to try to correct systems after having been finished [1].

0-7695-2160-6/04 $20.00 © 2004 IEEE

Several papers deal with the importance of security in the software development process. Ghosh et al. [5] state that security must influence all aspects of design, implementation and software tests. Hall and Chapman [6] put forward ideas about how to build correct systems that fulfil not only the normal requirements but also the security ones. These ideas are based on the use of several formal techniques of requirement representation and a strong correction analysis of each stage. Nevertheless, databases and data warehouses security in relation to design is not paid enough attention since it is only focused on the security of data in their encryptions.

As a result of technological changes, such as access to databases via web, development of electronic commerce, advances in data warehouses and even the use of data mining techniques [17], data security problems have increased. This fact justifies the use of methodologies incorporating security into the stages of ISs development.

The rest of the paper is organized as follows: In section 2, we will shortly describe each one of the seven proposals that incorporate security into the stages of systems development; in section 3, we will show the comparison framework that we have used and we will justify why we have chosen it. In section 4, we will make the comparison, and finally, in section 5, we will explain our conclusions.

2. Proposal of Methodologies incorporating Security

The proposals that will be analysed in our comparison are listed below:

- MOMT: Multilevel Object Modeling Technique, by Donald Marks, Peter Sell and Bhavani Thuraisingham.
- Business Process-Driven Framework for Security Engineering, by José L. Vivas, José A. Montenegro and Javier López.
- UMLSec: Secure Systems Development Methodology using UML, by Jan Jürgens.
- Secure Database Design Methodology, by Eduardo Fernández-Medina and Mario Piattini.
- Security and Privacy Requirements Analysis Methodology within a Social Setting, by Lin Liu, Eric Yu and John Mylopoulos.
- A New Paradigm for Adding Security into IS Development Method, by Mikko Siponen.
- An UML extension for Secure Multidimensional Conceptual Modeling, our proposal.

We have chosen these seven methodologies because the majority of them try to solve the problem of security (mainly confidentiality) from the earliest stages of the ISs development, emphasize security modeling aspects and use modeling languages that make it easier the security design process.

2.1 Multilevel Object Modeling Technique

Marks et al. define MOMT (Multilevel Object Modeling Technique) [14] as a methodology to develop secure databases by extending OMT in order to be able to design multilevel databases providing the elements with a security level and establishing interaction rules among the elements of the model.

MOMT is mainly composed of three stages:

- Analysis Stage: It allows us to analyse the requirements to detect potential system vulnerabilities. This stage consists of three models whose aim is to collect system information from several perspectives: multilevel object model (to represent static features), multilevel dynamic model (to represent dynamic features) and multilevel functional model (to represent system transformation features).
- System Design Stage. It allows us to design multilevel databases. To do so, it defines, at a high level, systems structure and multilevel database.
- Object Design Stage: It allows us to design the modules of the automated system in a more detailed way.

2.2 Business Process-Driven Framework for Security Engineering

Vivas et al. [18] propose a business process-driven system development method where technology decisions are guided by the business model. Expressing security requirements at the business model level is motivated by the fact that the applications like e-commerce transactions are conceptually similar to traditional non-automated business transactions. Notions such as non-repudiation, confidentiality, integrity, access control and authentication have played a role in business transactions long before the appearance of automated systems.

This framework is based on the UML and integrates security requirements into a business process model of the system. The UML is extended in order to express security notions. With the aim of facilitating its adoption by system developers, the framework intends to integrate security requirements into standard system development methodologies which, currently, are often UML-based and use case-driven.

Use cases and the corresponding scenarios are used as the basic tools to build threats models and elicit security requirements. The latter are originally stated at the high level of abstraction within a functional representation of the system, thus yielding a security-enriched specification. Thereafter, a machine readable XMI-representation of the system is produced and the security requirements are integrated into the functional description by means of the pattern-based analysis and design process yielding a new specification of the system which the security requirements have been into. The resulting representation is translated into a formal notation for testing, validation and verification. This procedure is iterated as many times as required. The result is used as an input to the following stages of system development.

2.3 UMLSec

Jürgens states a methodology [10] to specify requirements regarding confidentiality and integrity in analysis models based on UML. Multilevel security and Mandatory Access Control are the security models highlighted in this proposal. This approach considers an UML extension to develop secure systems. In order to analyse security of a subsystem specification, the behaviour of the potential attacker is modeled; hence, specific types of attackers, that can attack different parts of the system in a specific way, are modelled. This proposal uses the majority of UML diagrams to model security aspects, mainly those referred to confidentiality and integrity. Besides, this methodology incorporates the translation of UMLSec models defined for the introduction of patterns into the design process.

2.4 Secure Databases Design

Fernández-Medina and Piattini propose a methodology to design multilevel databases [3] by integrating security into each one of the stages of the databases life cycle.

This methodology includes the following aspects:

- A specification language of multilevel security constraints about the conceptual and logical models.
- A technique to the early gathering of multilevel security requirements.
- A technique to represent multilevel databases conceptual models.
- A logical model to specify the different multilevel relationships, the metainformation of databases and constraints.
- A methodology based upon the Unified Process [9], with different stages that allow us to design multilevel databases.
- A CASE tool that helps us to automate multilevel databases analysis and design process.

2.5 Security and Privacy Requirement Analysis

In [13], it is stated a methodological framework to deal with security and privacy requirements based on i*, which is an agent-oriented requirements modeling language.

This framework is formed by a set of analysis techniques:

- Attacker analysis: It helps us to identify system potential attackers and their malicious intents.
- Dependency vulnerability analysis: It helps us to detect vulnerabilities in terms of organizational relationships among stakeholders.
- Countermeasure analysis: The necessary factors for a successful attack are the attacker motivation, the system vulnerabilities and the attackers' capabilities to carry out the attack.
- Access control analysis: It establishes a link between security requirements models and security implementation models. To do so, it uses i* models to polish a proposed solution and to generate a system design.

The concepts provided by i* language enable us to analyse security aspects within their social settings, giving place to a systematic way to find vulnerabilities and threats.

2.6 A paradigm for adding security into IS development methods.

Siponen proposes as new paradigm a meta-notation to address the following four problems [16]:

- Diferent secure IS design approaches cover the diferent levels of IS, but lack the comprehensiveness needed.
- Most of the approaches for designing secure IS are difficult to integrate into IS development methods.
- Existing secure IS design approaches do not assist the autonomy of developers.
- New IS development methods spring up very ocasionally

Rather than presenting another security approach with its own novel security features, this paradigm proposes that secure IS design approaches must be elevated a level of abstraction above the barriers. Moving a level away from methodology takes us into the realm of meta-methodology, a new paradigm for secure IS which helps developers to use and modify their existing methods as needed.

The meta-notation includes six dimensions: security subjects, security objects, security constraints, security classifications, abuse scenarios, and security policy.

2.7 An UML extension for secure multi-dimensional conceptual modeling

We propose an extension of the UML that allows us to represent the main security information of data and their constraints in the multidimensional modeling (MD) at the conceptual level [4]. The proposed extension is an UML profile that allows us to consider the main MD modeling properties and it is based on the UML (designers can avoid learning a new specific notation or language). We consider the multilevel security model, but we focus on taking into consideration aspects regarding read operations because this is the most common operation for final user application. This model allows us to classify both information and user into security classes, and enforce the mandatory access control. By using this approach, we make it possible to implement secure MD models with any of the DBMS that are able to implement multilevel databases, such as Oracle Label Security [12] and DB2 Universal Database (UDB) [7]. An extension to the UML begins with a brief description and then lists and describes all of the stereotypes, tagged values, and constraints of the extension. In addition to these elements, an extension contains a set of well-formedness rules. These rules are used to determine whether a model is semantically consistent with itself. According to this fact, we define our UML extension for secure conceptual MD modeling following a schema composed of these elements: *description* (a little description of the extension in natural language), *prerequisite extensions* (it indicates whether the current extension needs the existence of previous extensions), *stereotypes / tagged values* (the definition of the stereotypes and / or tagged values), *well-formedness rules* (the static semantics of the metaclasses are defined both in natural language and as a set of invariants expressed by means of OCL expressions), and *comments* (any additional comment, decision or example, usually written in natural language). For the definition of the stereotypes, we follow a structure which is composed of a name, the base metaclass, 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 expressed.

3. Comparison Framework

The comparison framework that we have used is that proposed by Khwaja and Urban [11]. We have chosen this framework since it establishes a clear differentiation between the concepts of specification and specification techniques. There are other comparison frameworks but this is one of the most recent and it solves the problem that many authors intermingle the concepts of specification and specification technique. The criteria used for one of these concepts should not be applied to the other, since it can influence the establishment/adaptation and suitable use of a methodology that considers information security aspects. For instance, a specification can be complete and consistent regardless of the way used to represent it, the process used in its construction, the degree/extent of tools and automation used or whether it is formal or informal. However, it is significant to indicate that a technique can be used to produce consistent or complete specifications. The criteria should be separated but it should exist a mapping between them, which means that the specification technique features help us to achieve certain features in a specification.

In the context of software engineering, specification is a description of externally known features, a complete behaviour, in other words, input/output, description of several systems interfaces, etc. The concept of specification is, thus, a precise sentence of the requirements that a system must satisfy. A software specification technique is a method to achieve the desired purpose or product.

The fulfilment of a technical criterion should carry out the fulfilment of the specification criteria related to that technical criterion as well. For example, if the technical criterion is the formality level, then, a high level of formality in a specification technique can help us to achieve a precise, unambiguous, consistent, complete definition and verifiable specifications.

The specification criteria, with their terms and phrases that describe the same criterion, are the following: Understandable (a system specification must be a cognitive model, comprehensibility), Appropriate (separate functionality from implementation), Unambiguous (precision, lack of ambiguity), Complete, Consistent, Correct, Verifiable (analysability), Validateable (testability), Modifiable (maintainability, adaptability), Traceable, Minimal (economy of expression).

The specification technique criteria and their meanings are:

- Expressive adequacy: Technique supports conciseness of representation. The expressive capability of a technique may enhance specification comprehensibility, apropriateness, and minimality.
- Constructibility: It refers to the ease of construction of a specification using the technique.
- Scope of specifications: Scope deals with both functional and performance specifications. Indeed, a specification for a system should consist of functional, as well as non-functional requirements specifications.

- Level of formality: High level of formality in a specification technique may help us to define precise, unambiguous, consistent, complete, and verifiable specifications.
- Formal foundation: High formal foundation in a specification technique may help us to define precise, unambiguous, consistent, complete, and verifiable specifications.
- Extent of applicability: It deals with the range of domains that can be specified by a technique.
- Easy of use: It deals with the ease that a tecnique may be used without much knowledge or special training.
- Help support: It deals with aspects such as the procedures, guidelines, standards, and case studies available for a technique. This criterion may help us to use a technique and construct specifications within the technique.
- Integrated environment & tool support: It deals with the tools available in an integrated fashion for a technique. This criterion may help us to use a technique, construct specifications within a technique, and make an automatic analysis of specifications.
- Specification organizational support: Technique supports good organizational principles to control complexity. Good specification organization helps us to control complexity and enhance understandability.
- Support for maintainability: The technique should facilitate specification modifications. Maintainable specifications are easily modifiable and traceable.
- Executable: A specification must be operational. An operational model of specifications may help us to increase understandability, reduce ambiguity, improve consistency, ensure completeness and correctness, and make specifications more verifiable and validateable.
- Tolerance for incompleteness: The system specification must be tolerant of incompleteness and augmentable. Execution of incomplete specifications may help testability at several stages of the specification development.
- Multiple views: Proper use of a technique should enhance understandability for non-computer specialists. Multiple views of a specification may enhance its understandability.
- Notational simplicity & flexibility: Technique supports conceptual clarity to the client. It may improve specification understandability.
- Internal verification support: A technique should provide means for specification consistency checks. Automatic internal verification supported by a technique may improve the reduction of ambiguity, ensure completeness, improve consistency, and hence make specifications more verifiable.
- External validation support: It may ensure correctness of a specification by validating against requirements and/or implementation. This criterion may also improve validation by generating test cases and using the same test for specification, as well as the implementation validation.
- Support for other development phases: Automatic design and implementation generation from specification may improve traceability across the development phases.
- Support for documentation generation: Automatic documentation generation from specifications may help us to increase understandability of specifications.

4. Comparison

Table 1 allows us to relate specification and specification technique criteria. We can see, for instance, that the fulfilment of a technical criterion must generate the fulfilment of all specification criteria related to that criterion. The fulfilment of a specification criterion (for example, unambiguous) can partially help to achieve the fulfilment of several technical criteria such as the level of formality, formal foundation, maintainability and internal verification support. The degree of fulfilment will be "X" for *Yes*, " " for *No* and "(x)" for *Partial*. As each specification technical criterion can be associated to one or more specification criteria, the answer of each methodology will be related to the technical fulfilment with respect to a specification criterion. For example, we can look up if there is an "expressive adequacy" that allows an "understandable" specification. To know if this criterion is completely fulfilled, the specification must be "understandable", "appropriate" and "minimal".

Table 1. Evaluation criteria for software specifications and specification techniques

Technique criterion	Specification Criteria	Methodologies						
		MOMT	Vivas	UMLSec	Fdez	Liu&Yu	Siponen	Our Proposal
Expressive adequacy	Understandable	X	X	X	X	X	X	X
	Appropriate	(x)	X	(x)	X	(x)	X	X
	Minimal	X	X	X	X	(x)	(x)	X
Constructibility	-	X	X	X	X	(x)	(x)	X
Scope of specifications	Complete	(x)	X	(x)	(x)	(x)		X
Level of formality	Unambiguous	X	(x)	X	X	X	(x)	X
	Consistent	X	(x)	X	X	X	(x)	X
	Complete	X	(x)	X	(x)	X	(x)	X
	Verifiable	X	(x)	X	X	X	(x)	X
	Validateable	X	(x)	X	X	X	(x)	X
Formal foundation	Unambiguous	X	(x)	X	X	X	X	X
	Consistent	X	X	X	X	X	X	X
	Complete	X	(x)	X	(x)	X	X	X
	Verifiable	X	X	X	X	X	X	X
	Validateable	X	X	X	X	X	X	X
Extent of applicability	-	(x)	X	(x)	X	(x)	X	X
Easy to use	-	X	X	X	X	(x)	X	X
Help support	-		(x)	(x)	(x)			(x)
Integrated environment & tool support	-		(x)	(x)	(x)			
Specification organization support	Understandable	X	(x)	X	X	X	(x)	(x)
	Modifiable	X	X	X	X	X	(x)	(x)
Support for maintainability	Modifiable	X	(x)	X	X	X		
	Traceable	(x)	(x)	X	(x)	X		
Executable	Understandable	(x)	(x)		X	(x)	(x)	(x)
	Unambiguous	(x)	(x)		X	X	(x)	(x)
	Consistent	(x)	(x)		X	X	(x)	(x)
	Complete	(x)	(x)		(x)	(x)	(x)	(x)
	Correct	(x)	(x)		X	X	(x)	(x)
	Verifiable	(x)	(x)		X	X	(x)	(x)
	Validateable	(x)	(x)		X	(x)	(x)	(x)
Tolerance for incompleteness	Verifiable	X	X	X	(x)	X	X	
	Validateable	X	X	X	(x)	X	X	
Multiple views	Understandable	X	(x)	X	X	X		
Flexibility & notational simplicity	Understandable	X	X	X	X	(x)	(x)	X
Internal verification support	Unambiguous							
	Complete							
	Consistent							
	Verifiable							

Technique criterion	Specification Criteria	Methodologies						
		MOMT	Vivas	UMLSec	Fdez	Liu&Yu	Siponen	Our Proposal
External validation Support	Correct	X	(x)	(x)	(x)	(x)		
	Validateable	(x)	(x)	(x)	(x)	(x)		
Support for other development phases	Traceable			(x)	(x)			
Support for documentation generation	Understandable	(x)		(x)	(x)			

5. Conclusions

All the above proposed ideas are very interesting and provide us with important contributions to solve the security problem in a methodological way. We can conclude that, at a general level, all of them fulfil the criteria associated to formal aspects; they are serious proposals, very well based and supported by a modeling language. The deficiency is observed in the automated support that each one of these methodologies needs, specifically, it can be mentioned the lack of an automatic instrument of internal verification. Moreover, each proposal has several weaknesses.

The multilevel databases design methodology called MOMT was not too much taken into consideration, in spite of the fact that it was an OMT extension, a well-known and consolidated methodology, due to its complexity caused by the integration of security constraints in the ISs development. MOMT was not completely developed due to the little success that it had when it was proposed. Furthermore, this methodology does not consider databases design nor propose valid solutions for current situations in which used technologies and security needs have changed, as we can see in the specification criterion "appropriate".

The proposal made by Vivas et al. is an ongoing work intended to establish a use case-driven software development framework based on the UML, as well as to integrate security requirements into a business process model of the system. The proposal is tentative and exploratory, and focuses on a discussion and identification of the problems rather than on providing solutions. This fact can be seen in the partial fulfillment of most of the criteria

The proposal made by Liu et al. is mainly associated to the security requirement analysis process from a top-down or bottom-up perspective. Moreover, the used techniques allow us to check the model and can be applied in several stages of the requirements process. The weakness of this methodology is the fact that it mentions neither the database processing nor the remaining stages of the information systems development. In addition to this, it does not consider tools that support the kind of reasoning regarding security. This fact can be visualized in the nonfulfilment of most of the necessary supports, except for the external validation support, where goal-reasoning techniques such as qualitative goal labeling algorithms and quantitative techniques can be used to identify the best design solutions.

UMLSec security proposal takes into account security requirements related to confidentiality and integrity aspects. However, it does not comprise aspects associated to databases security design. As this methodology tries to make a broader study, it does not consider secure databases design according to conceptual, logical and physical aspects, which is essential in systems security.

The proposal stated by Fernández-Medina and Piattini only studies use cases diagrams, class diagrams and OCL (*Object Constraint Language*) to model security. This proposal offers us an extension of involved languages and techniques to comprise confidentiality aspects in databases. Nevertheless, this methodology is not adequate for developing secure ISs.

The proposal made by Siponen is a meta-notation (six dimensions) which ensures that security issues are properly and easily addressed in IS secure design. According to the author, this proposal satisfies the requirements of autonomy better than any existing approach for designing secure IS/software. Practitioners can continue to use their favoured IS secure design methods as a basis for the development and management of secure IS. Our critic is associated to a deficiency of the graphical models and languages used to support this proposal or to provide it with greater formality. The proposal will be successful depending on the ability of the user who must incorporate elements of security into the methodologies that are being used in a given moment, because this proposal, in spite of being very interesting, is dealt with in a very general way. Furthermore, doubts can be casted upon the applicability of the idea of the meta-notation in the different IS development methods.

Our proposal solves an important problem, because there are many proposals for developing MD models, but none of them include confidentiality aspects. On the other hand, there are many proposals for including security aspects in the IS development process, but none of them considers MD models. Therefore, the problem of ensuring that information is only accesed by the authorized users, in the context of data warehouse applications, has not yet been solved. The proposal, in spite of being very interesting and solid in terms of conceptual modeling, has deficiencies associated to the remaining stages of the development process. In addition, it does not have, until the moment, an automatic support that allows us to work with it, for example, the implementation of a CASE tool based on UML, incorporated to the multidimensional modeling.

It is very difficult to develop a methodology that fulfils all criteria and comprises all security constraints in terms of confidentiality, integrity and availability. If that methodology was developed, its complexity would avoid its success. Therefore, the solution would be a more complete approach in which techniques and models defined by the most accepted model standards were used. And, if these techniques and models cannot be directly applied, they must be extended by integrating the necessary security aspects that, at present, are not covered by the analysed methodologies.

Acknowledgements

This research is part of the CALIPO (TIC2003-07804-C05-03) and RETISTIC (TIC2002-12487-E) projects, supported by the Dirección General de Investigación of the Ministerio de Ciencia y Tecnología, and the network VII-J.RITOS2 financed by CYTED.

References

[1] F. Brooks, *The Mythical Man Month:Essays on Software Engineering*. 1995: Addison-Wesley.

[2] S. Castano, M. Fugini, G. Martella, and P. Samarati, *Database Security*. 1995: Addison-Wesley.

[3] E. Fernández-Medina and M. Piattini. *Designing Secure Database for OLS*. in *Database and Expert Systems Applications: 14th International Conference (DEXA 2003)*. 2003. Prague, Czech Republic: Springer.

[4] E. Fernández-Medina, J.C. Trujillo, R. Villarroel, and M. Piattini. *Extending the UML for Designing Secure Data Warehouses*. in *23rd International Conference on Conceptual Modeling ER2004 (under revision)*. 2004. Shanghai, China: Springer-Verlag.

[5] A. Ghosh, C. Howell, and J. Whittaker, *Building Software Securely from the Ground Up*. IEEE Software, 2002. 19(1): p. 14-16.

[6] A. Hall and R. Chapman, *Correctness by Construction: Developing a Commercial Secure System*. IEEE Software, 2002. 19(1): p. 18-25.

[7] IBM, *Security: IBM to provide multilevel security on the zSeries mainframe*. 2004.

[8] ISO/IEC, *ISO/IEC 15408-1. Information Technology. Security Techniques. Evaluation Criteria for TI Security. Part I: Introduction and General Model*. 1999: Switzerland.

[9] I. Jacobson, G. Booch, and J. Rumbaugh, *The unified software development process*. 1999: Addison Wesley.

[10] J. Jürjens, *UMLsec: Extending UML for secure systems development*, in *UML 2002 - The Unified Modeling Language, Model engineering, concepts and tools*, J. Jézéquel, H. Hussmann, and S. Cook, Editors. 2002, Springer: Dresden, Germany. p. 412-425.

[11] A. Khwaja and J. Urban, *A Synthesis of Evaluation Criteria for Software Specifications and Specification Techniques*. International Journal of Software Engineering and Knowledge Engineering, 2002. 12, Nº 5: p. 581-599.

[12] J. Levinger, *Oracle label security. Administrator's guide. Release 2 (9.2)*. 2002: http://www.csis.gvsu.edu/General Info/Oracle/network.920/a96578.pdf

[13] L. Liu, E. Yu, and J. Mylopoulos. *Security and Privacy Requirements Analysis within a Social Setting*. in *11th International Requirements Engineering Conference*. 2003: IEEE Computer Society.

[14] D. Marks, P. Sell, and B. Thuraisingham, *MOMT: A multi-level object modeling technique for designing secure database applications*. Journal of Object-Oriented Programming, 1996. 9(4): p. 22-29.

[15] G. McGraw, *Penetrate and Patch is Bad*. IEEE Software, 2002: p. 15-15.

[16] M. Siponen, *Designing secure information systems and software (Academic Dissertation)*, in *Department of Information Processing Science*. 2002, University of Oulo: OuLu, Finland.

[17] B. Thuraisingham, L. Schlipper, L. Samarati, T.Y. Lin, S. Jajodia, and C. Clifton, *Security Issues in data warehousing and data mining: panel discussion*. Database Security XI: Status and prospects, 1998: p. 3-16.

[18] J.L. Vivas, J. Montenegro, and J. López. *Towards a Business Process-Driven Framework for Security Engineering with the UML*. in *Information Security, 6th International Conference (ISC 2003)*. 2003. Bristol, UK: Springer-Verlag.