Published by
Faculty of Computer Science and Information Technology
Riga Technical University
1 Kalku, Riga
LV-1658, Latvia

ISBN 9984-9767-1-8

Grundspenkis, Kirikova (Eds.) CAiSE'04 WORKSHOPS Volume 1

Knowledge and Model Driven Information Systems Engineering for Networked Organisations

CAiSE'04 WORKSHOPS

in connection with
The 16th Conference on
Advanced Information Systems Engineering
Riga, Latvia, 7-11 June, 2004

Proceedings Volume 1

Edited by
Janis Grundspenkis
Marite Kirikova

Janis Grundspenkis
Marite Kirikova (Eds.)

CAISE'04 WORKSHOPS

Knowledge and Model Driven Information Systems Engineering for Networked Organisations

The 16th Conference on
Advanced Information Systems Engineering
Riga, Latvia, 7 – 11 June 2004

Workshop proceedings
Volume 1

© The authors mentioned in the Table of Contents.

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission from the authors of the papers.

ISBN 9984-9767-1-8

CONFERENCE on Advanced Information Systems Engineering

Knowledge and Model Driven Information Systems Engineering for Networked Organisations: workshop proceedings Volume 1/ CAiSE'04 Workshops = the 16th Conference on Advanced Information Systems Engineering, Riga, Latvia, 7 – 11 June, 2004; Janis Grundspenkis, Marite Kirikova (Eds.). – Riga: Faculty of Computer Science and Information Technology, Riga Technical University, 2004.

Publisher:
Faculty of Computer Science and Information Technology
Riga Technical University
1 Kalku, Riga
LV-1658, Latvia

Editors:

Janis Grundspenkis
Department of Systems Theory and Design
Riga Technical University, 1 Kalku,
Riga, LV-1658, Latvia
e-mail: jgrun@egle.cs.rtu.lv

Marite Kirikova
Department of Systems Theory and Design
Riga Technical University, 1 Kalku,
Riga, LV-1658, Latvia
e-mail: marite@egle.cs.rtu.lv

Section editors (Volume 1 – 3)

Peter Barna	The Netherlands	John Krogstie	Norway	
Zohra Bellahsène	France	Graham Low	Australia	
Ilia Bider	Sweden	Peter McBrien	UK	
Paolo Bresciani	Italy	Michele Missikoff	Italy	
Martin J. Eppler	Switzerland	Barbara Pernici	Italy	
Flavius Frasincar	The Netherlands	Gil Regev	Switzerland	
Terry Halpin	USA	Keng Siau	USA	
Markus Helfert	Ireland	Pnina Soffer	Israel	
B. Henderson-Sellers	Australia	Richard Vdovjak	The	Netherlands
Geert-Jan Houben	The Netherlands			

Printed by SIA "Zalktis ZB", Prūšu ielā 3/5, Rīga, LV-1057, Latvia

Preface

Holding workshops that precede the main conference is a tradition of CAiSE, the international Conference of Advanced Information Systems. This tradition was followed also by the 16th instance of the conference - CAiSE'04, held in Riga, Latvia, June 7-11, 2004.

The following ten workshops were organised in connection with CAiSE'04:

- The sixth International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS@CAiSE'04)
- The fifth Workshop on Business Process Modelling, Development, and Support (BPMDS'04)
- Doctoral Consortium (DC'04)
- International Workshop on Data and Information Quality (DIQ'04)
- The third International Workshop on Data Integration over the Web (DIWeb'04)
- The ninth CAiSE/IFIP8.1/EUNO International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'04)
- Enterprise Modeling and Ontologies for Interoperability (EMOI'04)
- The tenth Anniversary International Workshop of Requirements Engineering: Foundations for Software Quality (REFSQ'04)
- Ubiquitous Mobile Information and Collaboration Systems (UMICS 2004)
- Web Information Systems Modeling (WISM'04)

All workshops had their own international program committees that selected papers for the presentations and proceedings. Several workshops, such as DC'04, REFSQ'04, and UMICS 2004 have published their own volumes of proceeding. The proceedings of other workshops (AOIS@CAiSE'04, BPMDS'04, DIQ'04, DIWeb'04, EMMSAD'04, EMOI'04, and WISM'04) are amalgamated in three volumes of the CAiSE'04 Workshop Proceedings published by Faculty of Computer Science and Information Technology, Riga Technical University, Latvia.

Volume 1 of CAiSE'04 Workshop Proceedings consists of two sections. Editors of each section are listed at the title pages of the sections. Proceedings of **WISM'04** are published in the first section. The papers of this section address modeling issues relevant to Web information systems development where, due to necessity to integrate heterogeneous data sources and platforms, a large number of requirements are to be understood and fulfilled. The second section consists of research papers chosen by program committee of **EMMSAD'04**. The papers of EMMSAD'04 address capabilities and shortages of different modeling methods and methodologies in systems analysis and design.

Proceedings of DIQ'04, BPMDS'04, and AOIS@CAISE'04 are published in **Volume 2**. Proceedings of DIWeb'04 and EMOI'04 are available in **Volume 3** of CAiSE'04 *Workshop* Proceedings.

We acknowledge time and effort contributed by members of organizing and program committees of all workshops as well as hard work of CAiSE'04 Workshop Proceedings section editors. We are thankful also for their kind cooperation, patience, promptness and friendliness that continuously encouraged us during the half year long process of CAiSE'04 workshop organization.

Janis Grundspenkis, General CAiSE'04 Chair
Marite Kirikova, Workshop Chair
Viktorija Vinogradova, Technical Editor

Riga
June 2004

Brief contents

John Krogstie
Keng Siau
Terry Halpin (Eds.)

EMMSAD'04

Evaluating Modeling Methods for Systems Analysis and Design

Workshop at

CAiSE'04

The 16th Conference on
Advanced Information Systems Engineering
Riga, Latvia, 7–11 June, 2004

EMMSAD'04 Workshop Organization

Program committee

Richard Baskerville	USA	Vesper Owei	USA
Dinesh Batra	USA	Jeffrey Parsons	Canada
Shirley Becker	USA	Barbara Pernici	Italy
Guiseppe Berio	Italy	Michael Petit	Belgium
Ilia Bider	Sweden	Sudha Ram	USA
Paul Bowen	Australia	Colette Rolland	France
Hock Chan	Singapore	Matti Rossi	Finland
David Chen	France	Michael Rosemann	Australia
Robert Chiang	USA	Kurt Sandkuhl	Sweden
Roger Chiang	USA	Peretz Shoval	Israel
Jan Dietz	Netherlands	Riyaz Sikora	USA
Jan Goossenaerts	Netherlands	Guttorm Sindre	Norway
Peter Green	Australia	Veda Storey	USA
Reimgijus Gustas	Sweden	Ramesh Venkataramen	USA
Paul Johannesson	Sweden	Gerd Wagner	Netherlands
Håvard D. Jørgensen	Norway	Christian Wagner	China
Chuck Kacmar	USA	Tewei Wang	USA
Graham McLeod	South Africa	Benkt Wangler	Sweden
Kalle Lyytinen	USA	Roel Wieringa	Netherlands
Fiona Fui Hoon Nah	USA	Carson Woo	Canada
Jim Nelson	USA	Martin Zelm	Germany
Wee Keong Ng	Singapore	Daniel Dajun Zeng	USA
Andreas L. Opdahl	Norway		

Workshop organizers

John Krogstie	Norway
Terry Halpin	USA
Keng Siau	USA

Preface

Welcome to the Ninth International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'04) held in conjunction with CAiSE'04. The EMMSAD workshop series started in 1996. Over the years, EMMSAD has matured and is now recognized by researchers worldwide as a premier workshop focusing on the evaluation of modeling methods and methodologies. Similar to previous years, we had many good submissions this year. After an intensive reviewing process, we accepted 11 papers for full presentations, and 6 papers for short presentations. The submissions came from every corner of the globe. We have submissions from Israel, USA, Norway, Sweden, Finland, England, Germany, France, Spain, the Netherlands, Lithuania, Latvia and Chile. The International Program Committee consists of a group of well-known and highly qualified researchers. The success of EMMSAD is largely due to their generous contribution of time and effort.

Continuing with our very successful collaboration and cooperation with IFIP WG 8.1 that started in 1997, this years workshop is again a joint activity between CAiSE and IFIP WG 8.1. This year EUNO, which stands for European University Network on Enterprise Modelling and Enterprise Architecture, is another group that participates in organizing the workshop. To assist the authors in fining outlet for their papers, we have been recommending top papers from the workshop to journals. For EMMSAD'96, EMMSAD'97 and EMMSAD'99, we recommended top papers to the Australian Journal of Information Systems. For EMMSAD'98 and EMMSAD'00 top papers were recommended to the Journal of Database Management (JDM). Authors of best papers from EMMSAD'01, EMMSAD'02, and EMMSAD'03 were invited to submit chapters to a book that will be published by Idea Group Publishing in 2004.

Enjoy the workshop and Riga! We look forward to your continuing support for EMMSAD.

John Krogstie, Keng Siau, Terry Halpin
EMMSAD'04 Organizers

For more information on the workshop, contact
Dr. John Krogstie
SINTEF
Forskningsveien 1
P.O.Box 124 Blindern
N-0314 Oslo, Norway
Phone +47 93417551
Fax: +47 22067350
Email: John.Krogstie@sintef.no
URL: http://www.idi.ntnu.no/~krogstie

of constraints left in the HDM that have no corresponding target language construct. When there are no constraints left, the resulting target schema should be equivalent to the source schema. Otherwise semantic information is lost in the conversion, and the unmatchable constraints will tell us precisely what information has been lost.

We believe that the framework we have presented in this paper will be of use in formally comparing modelling languages and their expressibility, and that the proposed algorithm development will be of use in data integration.

References

1. M. Andersson. Extracting an entity relationship schema from a relational database through reverse engineering. In *Proc. ER'94*, LNCS, pages 403–419. Springer, 1994.
2. M. Boyd, S. Kittivoravitkul, C. Lazanitis, P.J. McBrien, and N. Rizopoulos. AutoMed: A BAV data integration system for heterogeneous data sources. In *Proc. CAiSE2004*, 2004.
3. S. Cluet, C. Delobel, J. Siméon, and K. Smaga. Your mediators need data conversion! *SIGMOD Record*, 27(2):177–188, 1998.
4. C.J. Date. Object indentifiers vs. relational keys. In *Relational Database: Selected Writings 1994–1997* [5].
5. C.J. Date, H. Darwen, and D. McGoveran. *Relational Database: Selected Writings 1994–1997*. Addison-Wesley, 1998.
6. T. Halpin. *Information Modeling and Relational Databases*. Academic Press, 2001.
7. E. Jasper, A. Poulovassilis, and L. Zamboulis. Processing IQL queries and migrating data in the AutoMed toolkit. Technical Report No. 20, AutoMed, 2003.
8. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. PODS'02*, pages 233–246. ACM, 2002.
9. P.J. McBrien and A. Poulovassilis. A uniform approach to inter-model transformations. In *Proc. CAiSE'99*, volume 1626 of *LNCS*, pages 333–348. Springer, 1999.
10. P.J. McBrien and A. Poulovassilis. Data integration by bi-directional schema transformation rules. In *Proc. ICDE'03*, pages 227–238. IEEE, 2003.
11. R.J. Miller, Y.E. Ioannidis, and R. Ramakrishnan. Schema equivalence in heterogeneous systems: Bridging theory and practice. *Information Systems*, 19(1):3–31, 1994.
12. J-M. Petit, F. Toumani, J-F. Boulicaut, and J. Kouloumdjian. Towards the reverse engineering of denormalized relational databases. In *Proc. ICDE'96*, pages 218–227, 1996.
13. A. Poulovassilis and M. Levene. A nested-graph model for the representation and manipulation of complex objects. *ACM Trans. on Information Systems*, 12(1):35–68, 1994.
14. A. Poulovassilis and P.J. McBrien. A general formal framework for schema transformation. *Data and Knowledge Engineering*, 28(1):47–71, 1998.
15. K. Schewe. Design theory for advanced datamodels. In *Proc. 12th Australasian Conf. on Database Technologies*, pages 3–9, 2001.
16. R. Wieringa. A survey of structured and object-oriented software specification methods and techniques. *ACM Computing Surveys*, 30(4):459–527, 1998.
17. C. Zaniolo and M. Melkanoff. A formal approach to the definition and the design of conceptual schemata for database systems. *ACM TODS*, 1982.

A Comparison of Secure Information Systems Design Methodologies

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

[1] Departamento de Computación e Informática,
Universidad Católica del Maule
Avenida San Miguel 3605. Talca (Chile)
Phone: 56 71 203525, Fax: 56 71 260278
rvillarr@spock.ucm.cl

[2] Departamento de Informática, Universidad de Castilla-La Mancha
Paseo de la Universidad, 4. 13071 Ciudad Real (Spain)
Phone: +34 926 29 53 Ext.: 3744, Fax: +34 926 29 53 54
{Eduardo.FdezMedina, Mario.Piattini}@uclm.es

Abstract. Generally, Information Systems security is taken into consideration once the system has been built, it is in operation and security problems have already arisen. This kind of approach, called "penetrate and patch" is being shifted by methodologies that introduce security in the systems development process. This paper allows us to make a comparison of secure information systems design methodologies, whose main purpose is to safeguard the confidentiality of the information. This comparison is carried out by analysing the specification of four methodologies that consider security, as a part of their designs. The used criteria allow us to distinguish between technical criteria and specification criteria. We notice that the analysed methodologies fulfil criteria partially. In this analysis, we make it clear that security aspects cannot be completely specified in these methodologies, generating a question about the possible consequences that could be generated by the lack of security in the systems.

1 Introduction

According to ISO/IEC 15408-1, security is defined as 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 [6]. Castano et al. [2] define 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 in 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 access denial to the services provided by the system). The above mentioned components 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" [11], which 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 [4]. 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].

Several papers deal with the importance of security in the software development process. Ghosh et al. [4] state that security must influence all aspects of design, implementation and software proofs. Hall and Chapman [5] 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 [13], data security problems have increased. This fact justifies the use of methodologies incorporating security in the stages of ISs development.

The rest of the paper is organized as follows: In section 2, we will shortly describe each of the four proposals that incorporate security in 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 Proposals of Methodologies incorporating Security

The proposals that will be analyzed in our comparison are as follows:

- MOMT: Multilevel Object Modeling Technique by Donald Marks, Peter Sell and Bhavani Thuraisingham.
- 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.

We have chosen these four methodologies because all 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) [12] 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 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 to design multilevel databases. To do so, it defines, at a high level, systems structure and multilevel database.
- Object Design Stage: It allows to design the modules of the automated system in a more detailed way.

2.2 UMLSec

Jürgens states a methodology [8] 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 modelled; 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 refer to confidentiality and integrity. Besides, it incorporates the translation of UMLSec models defined for the introduction of patterns in the design process.

2.3 Secure Databases Design

Fernández-Medina and Piattini propose a methodology to design multilevel databases [3] by integrating security in 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 [7], with different stages that allow us to design multilevel databases.
- A CASE tool that helps to automate multilevel databases analysis and design process.

2.4 Security and Privacy Requirements Analysis

Authors state a methodological framework to deal with security and privacy requirements based on i* [10], 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.

3 Comparison framework

The comparison framework that we have used is that proposed by Khwaja and Urban [9]. We have chosen this framework since it established 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 others, which can influence in the establishment/adaptation and suitable use of a methodology that considers aspects of security of information. For instance, a specification can be complete and consistent regardless of the way used to represent the specification, 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 fulfil 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 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 (analyzability), Validateable (testability), Modifiable (maintainability, adaptability), Traceable, Minimal (economy of expression).

The specification technique criteria and their meanings are:

- Expressive adequacy: The expressive capability of a technique may enhance specification comprehensibility, apropiateness, and minimality.
- Constructibility: It addresses the ease with which a specification may be constructed using the technique.
- Scope of specifications: Scope deals with both functional and performance specifications. In reality, specification for a system should consist of both functional, as well as non-functional requirements specifications.
- Level of formality: High level of formality in a specification technique may help defining precise, unambiguous, consistent, complete, and verifiable specifications.
- Formal foundation: High formal foundation in a specification technique may help defining 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 in using a technique and constructing 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 in using a technique, constructing specifications within a technique, and automatic analysis of specifications.
- Specification organizational support: Good specification organization helps in controling complexity and enhancing understandability.

- Support for maintainability: Maintainable specifications are easily modifiable and traceable.
- Executable: An operational model of specifications may help increase understandability, reduce ambiguity, improve consistency, ensure completeness and correctness, and make them more verifiable and validateable.
- Tolerance for incompleteness: Execution of incomplete specifications may help testability at various stages of specification development.
- Multiple views: Multiple views of a specification may enhance its understandability.
- Notational simplicity & flexibility: It may improve specification understandability.
- Internal verification support: Automatic internal verification support by a technique may improve reducing 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 generating test cases and using same test for specification, as well as implementation validation
- Support for other development phases: Automatic design and implementation generation from specification may improve traceability across development phases.
- Support for documentation generation: Automatic documentation generation from specifications may help increasing understandability of specifications.

4 Comparison

Table 1 allows 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 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 a "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	UMLSec	Fdez.	Liu & Yu
Expressive adequacy	Understandable	X	X	X	X
	Appropriate	(x)	(x)	X	(x)
	Minimal	X	X	X	(x)
Constructibility	-	X	X	X	(x)
Scope of Specifications	Complete	(x)	(x)	(x)	(x)
Level of Formality	Unambiguous	X	X	X	X
	Consistent	X	X	X	X
	Complete	X	X	(x)	X
	Verifiable	X	X	X	X
	Validateable	X	X	X	X
Formal Foundation	Unambiguous	X	X	X	X
	Consistent	X	X	X	X
	Complete	X	X	(x)	X
	Verifiable	X	X	X	X
	Validateable	X	X	X	X
Extent of Applicability	-	(x)	(x)	X	(x)
Easy to Use	-	X	X	X	(x)
Help Support	-		(x)	(x)	
Integrated Environment & Tool Support	-		(x)	(x)	
Specification Organization Support	Understandable	X	X	X	X
	Modifiable	X	X	X	X
Support for Maintainability	Modifiable	X	X	X	X
	Traceable	(x)	X	(x)	X
Executable	Understandable	(x)		X	(x)
	Unambiguous	(x)		X	X
	Consistent	(x)		X	X
	Complete	(x)		(x)	(x)
	Correct	(x)		X	X
	Verifiable	(x)		X	X
	Validateable	(x)		X	(x)
Tolerance for Incompleteness	Verifiable	X	X	(x)	X
	Validateable	X	X	(x)	X
Multiple views	Understandable	X	X	X	X

Technique criterion	Specification Criteria	Methodologies			
		MOMT	UMLSec	Pdez.	Liu & Yu
Flexibility & Notational Simplicity	Understandable	X	X	X	(x)
Internal Verification Support	Unambiguous				
	Complete				
	Consistent				
	Verifiable				
External validation Support	Correct	X	(x)	(x)	X
	Validateable	(x)	(x)	(x)	X
Support for other Development Phases	Traceable		(x)	(x)	
Support for Documentation Generation	Understandable	(x)	(x)	(x)	

5 Conclusions

All the proposed ideas are very interesting and they provide 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 in a modeling language. The deficiency is observed in the automated support that each of these methodologies need, specifically, it can be mentioned the lack of an automatic instrument of internal verification.Nevertheless, 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 Liu et al. is mainly associated to 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 does not mention the database processing, nor considers tools that support the kind of reasoning regarding security and it is mainly thought to counteract intruders' attacks.

UMLSec security proposal takes into account security requirements related to confidentiality and integrity aspects. It does not comprise aspects associated to databases security design. As this methodology tries to make a broader study, it does

not take into account 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 languages and techniques involved to comprise confidentiality aspects in databases. As it is a methodology to design secure databases, it could be extended or be used as a basis for a methodology to design secure data warehouses. Nevertheless, this methodology is not adequate for developing secure ISs.

It is very difficult to develop a methodology that fulfils all the 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.

References

1. Brooks, F.: The Mythical Man-Month: Essays on Software Engineering, 2nd.ed., Addison-Wesley, Reading Mass. (1995).
2. Castano, S., Fugini, M., Martella, M. and Samarati, P.: Database Security. Addison Wesley. (1995).
3. Fernández-Medina, E., Piattini, M.: Designing Secure Databases for OLS. Database and Expert Systems Applications: 14th International Conference DEXA, Praga. Lecture Notes in Computer Science, Vol 2736, Springer-Verlag, Berlin Heidelberg (2003) 886-899
4. Ghosh, A., Howel, C., Whittaker, J.: Building Software Securely from the Ground Up. IEEE Software, January/February (2002) 14-16
5. Hall, A., Chapman, R.: Correctness by Construction: Developing a Commercial Secure System. IEEE Software, January/february (2002) 18-25
6. ISO/IEC 15408-1: Information Technology. Security Techniques Evaluation Criteria for TI Security. Part I: Introduction and General Model.
7. Jacobson, I., Booch, G., and Rumbaugh, J. (1999). The Unified software development process. Addison Wesley.
8. Jürgens, J. : Towards development of Secure Systems using UML. Proceedings of the International Conference on the Fundamental Approaches to Software Engineering

(FASE/TAPS), Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg (2001)
9. Khwaja, A., Urban, J.: A Synthesis of Evaluation Criteria for Software Specifications and Specification Techniques. International Journal of Software Engineering and Knowledge Engineering. World Scientific Publishing Company, Vol 12 N° 5 (2002) 581-599
10. Liu, L., Yu, E., Mylopoulos, J.: Security and Privacy Requirements Analysis within a Social Setting. Proceedings of the 11th IEEE International Requirements Engineering Conference, IEEE Computer Society (2003)
11. McGraw, G.: Penetrate and Patch is Bad. IEEE Software, January/February (2002) 15-15
12. Marks, D., Sell, P. y Thuraisingham, B.: MOMT: A multilevel object modeling technique for designing secure database applications. Journal of Object-Oriented Programming. Vol 9. N° 4, (1996) pp. 22-29
13. Thuraisingham, B., Schlipper, L., Samarati, P., Lin, T. Y., Jajodia, S. and Clifton, C. Security issues in data warehousing and data mining : panel discussion. Database Security XI: Status and prospects. T. Y. Lin and S. Qian (eds.), Chapman &Hall, (1998) pp. 3-16

MC Sandbox – Tool Support for Method Configuration

Fredrik Karlsson[1] and Kai Wistrand[1]

[1] Örebro University, MELAB, Dept. of Informatics (ESI),
SE-701 82 Örebro, Sweden
{fkn, kwd}@esi.oru.se
www.oru.se/esi/melab

Abstract. Method configuration (MC) has been presented as a particular kind of method engineering (ME). ME in general is often a tedious and time consuming task and method configuration is no exception. Consequently, tool support is often required. In this paper we present an operationalization of the Method for Method Configuration into the tool support MC Sandbox. The functionality in MC Sandbox emphasizes reusable method assets and MC based on methods' rationale. The latter is usually not considered as an explicit focal area within ME. Furthermore, we present the experiences from the first workshop where the MC Sandbox has been successfully used during configuration work, and the repercussions it has on the design of the tool and the meta-method.

1 Introduction

Situational methods [1] or methods-in-action [2] are terms that are commonly used today. As stated by Fitzgerald et al. [3] 'it is now widely accepted that methods should be tailored to the actual needs of the development context', while 'there is very little by way of practical guidance to inform developers as to what steps of the method to modify or omit.' Odell [4] and Ralyté et al. [5] have surveyed the range of existing research in the field of Method Engineering (ME) and divided it into different categories. From these articles one can conclude that the modular method construction seems to be the main track. However, in days where 'off-the-shelf' methods, such as the Rational Unified Process (RUP) or Microsoft Solution Framework (MSF), increase in popularity a different point of departure is often needed. In these situations it is a question of tailoring one single method. Ralyté et al. [5] identify an extension strategy for methods and discuss a cancellation operator. But whether it should be treated as part of the extension strategy or not, is unclear. A combination of the cancellation and extension operators has been proposed, labeled method configuration (MC) [6]. Accordingly, MC is a particular form of ME defined as:

Definition 1. Method Configuration is the planned and systematic adaptation of a specific method generating reusable method assets.