Alessandra Bagnato (Ed.)

cea list

SEC-MDA 2010

Second International Workshop on Security in Model Driven Architecture

—

2010 ECMFA

European Conference on Modelling Foundations and Applications

A CEA LIST Publication

Preface

This volume contains the proceedings of the Second International Workshop on Security in Model Driven Architecture (SEC-MDA 2010) held on 16th June 2010 in Paris, France, in conjunction with Sixth European Conference on Modelling Foundations and Applications (ECMFA 2010).

The SEC-MDA 2010 workshop aimed at helping the convergence of the academia and industry, from all the different areas that wanted to/might play an active role in domain of security solutions by focusing on how software security can be improved through the MDA approach.

The main discussion topics were:

- How security specialists can capture their security expertise in form of reusable models
- How the security requirements and goals can be traced all along the development process
- How security models and profiles can be merged with system models in different abstraction levels
- How security models can be shared and reused
- How security testing can be improved through security models.
- Which are the requirements on tools to support the creation, transformation and use of security models?

We would like to take this opportunity to thank the people who have contributed to the SEC-MDA 2010 workshop. We wish to thank all authors of papers included in this volume and all reviewers for their valuable contributions, and we wish them a successful continuation of their work in this area. We wish to thank our invited speaker Prof. Nora Cuppens-Boulahia from Institut Telecom / Telecom Bretagne for her talk on 'Yet Another Security Meta Model Or An Enforcement Security Meta Model?'.

Finally, we thank the organization of the ECMFA 2010 conference in which this workshop has been embedded.

May 2010

The Organizational Committee

Alessandra Bagnato, TXT e-solutions
Amel Mammar, Telecom SudParis
Per Håkon Meland, SINTEF
Txus Sánchez, European Software Institute

Organisation

Workshop Chairs

Alessandra Bagnato	TXT e-solutions
Amel Mammar	Telecom SudParis
Txus Sánchez	European Software Institute

Programme Committee

Habtamu Abie (Norwegian Computing Center, Norway)
Alessandra Bagnato (TXT e-solutions, Corporate Research Division, Italy)
Ruth Breu (University of Insbruck, Austria)
Ana Cavalli (GET/INT, France)
Jorge Cuellar (Siemens CERT Siemens AG, Germany)
Violeta Damjanovic (Salzburg Research, Austria)
Marina Egea Gonzalez (ETH Zürich, Swiss)
Jan Jurjens (TU Dortmund and Fraunhofer ISST, Germany)
Filippo Lanubile (Università degli Studi di Bari, Italia)
Xabier Larrucea, (European Software Institute, Spain)
Amel Mammar (Telecom SudParis, France)
Jason Xabier Mansell, (European Software Insitute, Spain)
Per Håkon Meland (SINTEF, Norway)
Matteo Meucci (OWASP-Italy Chair, OWASP Testing Guide lead, Italy)
Charles Bastos Rodriguez, (Atos Research & Innovation Security Unit, Spain)
Bernhard Rumpe (RWTH Aachen University, Germany)
Nahid Shahmehri (Linkoping University, Sweden)
Txus Sánchez (European Software Institute, Spain)
Ståle Walderhaug, (SINTEF, Norway)

Supporting Organisations

University of Pierre & Marie Curie, Paris, France
SHIELDS Project http://shields-project.eu/

Table of Contents

Security Requirements Models: A Survey and Comparison

Abel E. Fornaris[1], Eduardo Fernández-Medina[1]

[1] Grupo GSyA. Dep. de Tecnologías y Sistemas de Información.
Universidad de Castilla-La Mancha. Paseo de la Universidad 4, Ciudad Real, España.
(AbelEnrique.Fornaris, Eduardo.FdezMedina)@uclm.es

Abstract. Security requirements engineering has become one of the key fields in secure software development. Languages, techniques, artifacts, diagrams, resumed as models in general, are used in order to represent the elicitation of security requirements from the early phases of the development. Although there is an increasing interest in this area, reflected in the great number of studies covering new models and the surveys and comparisons between them, there is a lack of systematic, thorough and unbiased studies. In this paper, the results, comparison and discussion obtained from a systematic literature review of the literature dealing with this area are presented in order to identify and provide a background to existing security requirements models and to state possible new research activities in this field with the ultimate aim of discovering an MDD approach which integrates these models.

Keywords: Security requirements; Security Engineering; Security requirements techniques, models, languages, diagrams, artifacts; Survey, comparison; Systematic Literature Review, Model-driven Security; MDD.

Introduction

The increased use of computers has meant that valuable business and mission critical assets are increasingly stored and manipulated by computer-based systems. The incidence of misuse of those assets has also increased because of the worldwide accessibility of the Internet and the automation of systems [1]. Given this situation, information and information system security has become a major problem and the target of many researches. In this scenario, the security engineering community has developed a set of strategies in order to secure applications. However, these strategies have primarily been focused on design and implementation stages when it is absolutely vital that Information Systems (IS) are properly assured from the very beginning [2]. The cost of dealing with security issues in the latter phases of development grows in comparison to dealing with them in the requirements phase.

When referring to MDD this problem is not tackled efficiently either. Security, as part of the system's non-functional requirements, must be integrated into the system model from the very beginning if full advantage of MDD is to be taken. Still, existing approaches using MDD typically ignore security at the modeling stage, rather adding security aspects later in an ad-hoc manner. This can lead to deficiencies in security and a less cost-effective development process [3].

However, some initiatives to integrate security in MDD have been developed. For instance, Model-Driven Security (MDS) [4] adds three new aspects: (i) the system models are enriched with primitives and rules for integrating security into the development process, (ii) the model transformation techniques are extended to ensure that these security details are also transformed, and (iii) the system is obtained, including the security properties and the corresponding security mechanisms [5].

The importance of secure development from these early stages signifies that security requirements (SR) engineering has become a highly studied field through which to produce methodologies, processes and frameworks that could integrate these non-functional requirements with the rest [6].

Languages, techniques, artifacts and diagrams, which will be termed as *models* in general from here on, are defined in order to represent these SR and provide the first step with which to propagate them throughout the stages of the system development.

During the last few years, a number of papers have focused on security requirements and some have carried out systematic literature reviews of this subject. However, most of these reviews are focused on the processes used to elicit security requirements and do not take an inside look at the models they propose or use [7]. After performing preliminary searches aimed at both identifying existing systematic reviews and assessing the volume of potentially relevant studies, we can highlight several works that contain summaries of and/or comparisons between models, such as [8-17]. However, none of them carry out a review focused on security requirements models in a systematic manner, that is, none of them carry out a systematic review of the security requirements models used in security requirements engineering. They are not, therefore, a sufficiently good context in which to operate in this area. Besides this, the particular proposal by Mellado et al. [13], which is not centred solely on security requirements models, can be improved by including several new models and comparison criteria aiming to achieve a different conclusion.

In this study we shall present the results of a systematic literature review (SLR) on these security requirements models which will summarize the initiatives in this field, compare them and ultimately arrive at conclusions.

The ultimate objective is to identify the need for a unified set of models, or to find links between them in order to propose a MDD compatible approach which could be propagated throughout all stages of the development itself.

In contrast to an unsystematically conducted literature review, an SLR is developed in a formal and systematic manner. This means that the research process of a systematic type of review follows a very well defined and strict sequence of methodological steps according to an aprioristically developed protocol. The methodological steps, the strategies to retrieve the evidence, and the focus on the question are explicitly defined, so that other professionals can reproduce the same protocol as well as judge about the adequacy of the chosen standards for the case given [18].

In this case we have developed the SLR by using the guidelines on systematic reviews provided by Kitchenham et al. [19] and it is implemented with the template described by Biolchini et al. [18] which facilitates the planning and execution of the SLR through a series of defined steps.

The remainder of this paper is organized as follows: in Section 2 we shall present the review protocol used to carry out the SLR and the results obtained, namely the different proposals for new models to elicit security requirements. Section 3 will contain an analysis of our results, a comparison and a discussion. Lastly, the conclusions obtained from this analysis will be shown in Section 4.

Review Planning and Execution

As stated previously, the SLR was planned and carried out by following a template. A summary of both phases, the review protocol and the results, is presented below.

Review Planning

The *question focus* was to identify the existing languages, techniques, artifacts, diagrams, defined as *models* in general, used to elicit security requirements from the early stages of software development. Studies related to the comparison and/or surveying of these models were also identified and described.

The *problem* was that existing security requirements engineering methodologies, processes and frameworks tend to use different models to represent the elicitation of security requirements which leads to the necessity of learning a great number of them. Furthermore, they define different security requirements depending on the definition they use, and some of them sometimes lack some of these requirements since they may be centred on a particular SR. An approach with which to discover or define a set of common models or verifiable links between them in order to compose a standard model in this area might therefore be a solution to this.

Only publications proposing and describing new models or new approaches, used in security requirements engineering or those which perform surveys or comparisons between them were considered as *inclusion criteria* for the SLR. The *expected result* at the conclusion of this SLR was to identify, describe and compare the different models used in SR engineering, including their surveys and comparisons.

The *main application* areas that will benefit from the SLR results are secure software development and software engineering, specifically security requirements engineering, along with security experts and requirements engineers, as well as MDD researchers and developers. A fair background was expected to be provided in order to understand the extent of models currently used in SR engineering.

The criterion used to select *sources* from which to obtain these studies was based on the availability of articles on the Internet or in the digital library of the University of Castilla-La Mancha. Digital libraries, such as Scopus Digital Library, Springer Digital Library, Science@Direct Digital Library, ACM Digital Library and IEEE Computer Digital Library, were searched. Valid *search strings* were defined for each one in order to obtain the best results, taking as basis the following one: "Security AND Requirement AND (Engineer OR Engineering) AND (Model OR Diagram OR Artifact OR Method OR Language OR Technique)" and also including MDD related terms.

Review Execution

After performing the search in each of the digital libraries previously mentioned, the following *primary studies* were indentified, each one around a model proposal. Table 1 and Table 2 contain a summary of these proposals.

Table 1. Summary of proposals to elicit security requirements

Author(s)	Definition and Artifacts	Proposal
Firesmith [20]	Standard UML use cases which complement misuse cases in order to represent SR.	Security Use Cases
Jürjens [21]	Extensions to several UML artifacts with stereotypes, tags and constraints to represent SR.	UMLsec
Lodderstedt, Basin et al. [22]	Stereotyped UML class diagram to represent role based SR.	SecureUML
Mouratidis and Giorgini [23]	Tropos methodology extension to represent SR as constraints. It is based on i*[24] and uses its concepts.	Secure Tropos
Rosado et al. [25]	UML profile using stereotyped secure use cases for grid.	GridUCSec
Zannone [26], Massacci, Mylopoulos et al. [27]	Modeling language based on social modeling language i* [24] to capture SR from the organizational point of view.	SI*
Jackson [28], Hatebur, Heisel et al. [29]	Kinds of patterns to represent threat models and SR from them.	Security Problem Frames
Jennex [30]	Graphical method of identifying and documenting SR by pointing out threats combined with defense in depth concept.	Barrier analysis diagrams
Haley, Laney et al. [31]	SR considered in terms of constraints which operationalize as security goals. Assets involved in functional requirements must be identified.	Constraints

Table 2. Summary of proposals to represent security threats & their possible countermeasures

Author(s)	Definition and Artifacts	Proposal
Sindre and Opdahl [32]	Extended UML use case to represent undesirable behavior of the software.	Misuse Cases
Sindre [33]	Standard UML activity diagram to capture malicious activities and actors using the same syntax and semantics.	Mal Activity Diagrams
McDermott and Fox [2]	Standard UML use case representing complete harmful actions in the system.	Abuse Cases
Hussein and Zulkernine [34]	UML profile to deal with attacks using several stereotyped, tagged artifacts.	UMLintr
Zulkernine, Graves et al. [35]	UML state charts to specify attack. Ability to represent complex multiple step attacks which it is not possible to represent in other ways.	UML State Charts for Security
Schneier [36]	Model to calculate the risk and cost of potential attacks and their countermeasures, using a tree structure.	Attack Trees
Lin et.al [1]	Represents security threats and derives SR by defining anti-requirements as action to subvert these SR.	Abuse Frames
Peeters [37]	Plain text stories of how attackers abuse the system. These are agile counterparts to abuse cases.	Abuser Stories
Graves and Zulkernine [38], Raihan and Zulkernine [39]	AsmL is an extended finite state machine-based executable software specification language, also used to specify attack scenarios in extensions such as AsmLSec	AsmL, AsmLSec
Eckman, Vigna et al. [40]	Domain-independent attack description language that could be extended in a well defined way to match different operating environments.	STATL

These results show two large areas in SR modeling, one of which refers to the representation, documentation and direct elicitation of the SR by means of the models. The other one refers to specify attacks, threats and vulnerabilities, in order to discover the SR that are necessary to prevent them. Each of these areas using language extensions (mostly UML) or describing a new model instead.

Results Analysis and Discussion

A thorough comparison of the different proposals is presented in Table 3 following the criteria shown below:

- **HLA** stands for "High Level Alignment" and may be "M" if the model is presented as part of a *methodology*, "P" if it is part of a *process* and "F" if a *framework* using it is defined in a paper contained within the SLR.
- **FD** refers to the degree of "Formal Definition" of the model. "X" stands for no formal definition at all, "P" means it has a partial definition and "*" is used for models that have a strong formal definition, including UML metamodels.
- **CTR** stands for the availability of "Constraint" specification by the model. "*" for Yes, and "X" for No.
- **ET** signifies "Empirical Testing" and refers to surveys, cases study or experiments used in order to validate the proposal. "X" stands for no validation found in the literature.
- **SSA** ("Security Standard Aligment") describes the compliance of the model with some international standard like ISO/IEC or others. "X" signifies that no compliance has been found in the relevant literature.
- **AT** refers to "Automated Tool" found for the proposal. "X" means no tool found in the relevant literature.
- **ScA** signifies "Scientific Awareness" and measures the degree of "popularity" of each proposal in the scientific community. Each proposal can be found in the relevant literature as part of a "C" for *comparison* or "S" for *survey* as part of the results obtained from the SLR.

In Table 4 another comparison is performed in terms of the different security quality sub-characteristics defined in ISO/IEC 25010 [41] still under development. It also includes other criteria which might be interesting to understand the capacity of each model to represent all possible SR currently considered by the standards and the scientific community in general. The "*" signifies that the ability to represent that SR or countermeasures (mitigating actions) in order to guarantee it with the model was found in the relevant literature. The "X" signifies that no information regarding that ability was found. Lastly, "P" signifies that the model could potentially be used to represent or elicit that SR, although this is not stated clearly in the relevant literature.

After performing a high level analysis on Table 3, a number of considerations can be drawn. The ScA column allows us to identify how "popular" a model is in the scientific community through the amount of times that it is referred to in comparisons or surveys. Several models emerge as being popular for several reasons, and could be the basis for other analyses. However, new proposals not mentioned could still be valid.

The HLA column shows that most of the proposals have been applied, many of them as part of methodologies, which are the highest step in integrating the model into the whole software development cycle. However, many of these "strong" proposals lack a complete formal definition, which could be a problem if an attempt is made to integrate them in a single model or create links between them. Still, some of them

Table 3. Comparison of proposals

Proposal	HLA	FD	CTR	ET	SSA	AT	ScA
Security Use Cases	M [42]	*	X	X	ISO 9126	X	C [13]
UMLsec	P [21]	P	*	Industrial Case of Study: Biometric Authentication [21]	X	UMLSec tool.	C [9, 10] [13]
SecureUML	M [43]	*	*	Case of Study: J2EE "Pet Store" application [43]	X	SecureUML Template.	C [9, 10]
Secure Tropos	M [23]	*	*	Case of Study: Single Assessment Process (eSAP) system [23]	ISO 17799	SI* tool.	C [8-10], S [17]
GridUCSec-Profile	M [25]	*	*	GREDIA European project [25]	X	X	X
SI*	F [26]	*	X	X	X	Eclipse plugin as case tool.	C [8, 9], S [17]
Security Problem Frames	X	P	X	Case of Study: PDA bluetooth authentication [29]	X	X	C [8]
Barrier analysis diagrams	M [30]	X	X	X	X	X	C [13]
Constraints	F [31]	X	*	X	X	X	X
Misuse Cases	M [42]	*	X	Experiment: Comparison between misuse cases and attack trees [15]	X	Scenario Plus toolkit.	C [8, 10, 12, 15], S

							[11, 14, 16, 17]
Mal Activity Diagrams	X	*	X	Small Case of Study: "The Art of Deception" [33]	X	X	X
Abuse Cases	X	*	X	X	X	UML tool by Rational Rose.	C [10], S [17]
UMLintr	F [34]	P	X	X	X	Prototype tool	C [10]
UML State Charts for Security	X	X	X	Experiment: AsmL and UML state charts translated to Snort rules [35]	X	UML tool by Rational Rose	C [10]
Attack Trees	M [44]	P	X	Experiment: Comparison between misuse cases and attack trees [15]	X	SecureITree	C [8, 12, 15], S [11, 16]
Abuse Frames	X	P	X	X	X	X	C [9]
Abuser Stories	X	X	X	X	X	X	C [8, 13], S [11]
AsmL, AsmLSec	X	X	X	Experiment: Evaluate AsmL and AsmLSec [38] [39]	X	X	C [10]
STATL	X	X	X	X	X	X	C [10]

Table 4. Comparison of proposals based on well-known security requirements

Proposal	Authenticity	Confidentiality	Compliance	Attack Detection	Availability	Integrity	Non-Repudiation	Accountability	Safety
Security Use Cases	*	*	*	X	*	*	*	P	P
UMLsec	*	*	X	P	P	*	*	P	X
SecureUML	*	X	X	X	X	X	X	X	X
Secure Tropos	*	*	*	X	*	*	*	P	P
GridUCSec-Profile	*	*	X	X	*	*	*	*	X
SI*	*	*	X	X	*	P	X	P	P
Security Problem Frames	*	*	X	*	X	*	P	X	P
Barrier analysis diagrams	*	*	X	*	X	*	P	X	*
Constraints	*	P	X	X	*	*	X	*	P
Misuse Cases	P	P	*	*	P	P	P	X	*
Mal Activity Diagrams	P	P	X	*	P	P	P	P	P
Abuse Cases	X	X	X	*	X	X	X	X	P
UMLintr	X	X	X	*	X	X	X	X	P
UML State Charts for Security	X	X	X	*	X	X	X	X	X
Attack Trees	P	P	X	*	P	P	P	P	*
Abuse Frames	P	P	X	*	X	P	X	X	P
Abuser Stories	P	P	P	*	P	P	P	P	P
AsmL, AsmLSec	X	X	X	*	X	X	X	X	X
STATL	X	X	X	*	X	X	X	X	X

have metamodels defined, so there is hope that this goal can be achieved by means of transformations following the MDD paradigm. On the other hand, it is not generally possible to specify constraints, which could prevent from getting strong SR modeling.

There is a general lack of empirical testing on "soft" or "unpopular" models, although some cases of study and even experiments have been carried out on others. This signifies that their effectiveness has been proved in real world applications. Most of these also have automated tools available to work with, which is an important step in facilitating the use of their capabilities.

Most of the models are not standard compliant, or have been developed by following a particular standard, which leads to the problems found in Table 4, where some SR are able to be specified by the models and others are not.

As is generally accepted, the triad formed by *confidentiality, integrity, and availability* are the most important factors to observe when developing models to elicit SR. As is shown in the table, most of the models include support for them or have the potential to include this. Also, depending on its purpose, each model has higher or lower support for *attack detection*. The SR elicitation models tend to include support for *authenticity*, while the rest may vary from one to another. *Non-repudiation* and *accountability* are SR that should be taken into account more often in models, since they are not so widely supported. Finally, *safety* requirements can be potentially elicited by many models, taking into account that the main difference between security and safety is *intention* [31].

Conclusions

This survey and comparison, which is the result of a systematic literature review, could allow us to conclude that a great number of models are used to elicit SR, either by representing the SR themselves or by specifying attacks, threats or vulnerabilities models and countermeasures to elicit them. Each model has its own characteristics and covers certain aspects of security; while in general, none of them is a complete solution to all the possible scenarios.

The ultimate objective of this paper was to state new research possibilities with regard to security requirements engineering modeling and to identify these “holes”, in order to show the need for a unified model, whilst still attempting to reuse key features in existing successful models. This objective has been accomplished by showing the absence of an integrated approach that could guarantee the elicitation of all the SR accepted by the standards and the scientific community in general. This, while being part of a model-driven development process to transform these SR into effective security mechanisms in the system to be.

The task can be achieved by means of transformations between existing successful models which have formal definitions in order to obtain a new MDD-compatible model which could take advantage of this new paradigm in the development process. This is part of the future work derived from this paper.

Acknowledgments

This research has been carried out in the framework of the following projects: QUASIMODO (PAC08-0157-0668), SISTEMAS (PII2I09-0150-3135) and SEGMENT (HITO-09-138) financed by the “Viceconsejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha” and the FEDER, BUSINESS (PET2008-0136) financed by the "Ministerio de Ciencia e Innovación” (Spain) and MEDUSAS (IDI-20090557) project financed by the "Centro para el Desarrollo Tecnológico Industrial. Ministerio de Ciencia e Innovación"(CDTI).

References

1. Lin, L. et al., Introducing Abuse Frames for Analysing Security Requirements, in Proceedings of the 11th IEEE International Conference on Requirements Engineering. 2003, IEEE Computer Society. p. 371.
2. McDermott, J. and C. Fox, Using Abuse Case Models for Security Requirements Analysis, in Proceedings of the 15th Annual Computer Security Applications Conference. 1999, IEEE Computer Society. p. 55.
3. Xiao, L., An adaptive security model using agent-oriented MDA. Information and Software Technology, 2009. **51**(5): p. 933-955.
4. Basin, D., J. Doser, and T. Lodderstedt, Model driven security for process-oriented systems, in Proceedings of the eighth ACM symposium on Access control models and technologies. 2003, ACM: Como, Italy. p. 100-109.
5. Fernández-Medina, E. et al., Model-Driven Development for secure information systems. Information and Software Technology, 2009. **51**(5): p. 809-814.
6. Mellado, D., E. Fernández-Medina, and M. Piattini, A common criteria based security requirements engineering process for the development of secure information systems. Computer Standards & Interfaces, 2007. **29**(2): p. 244-253.
7. Mellado, D. et al., A systematic review of security requirements engineering. Computer Standards & Interfaces, 2010. **32**(4): p. 153-165.
8. Romero-Mariona, J., H. Ziv, and D.J. Richardson, Later stages support for security requirements, in The Fifth Richard Tapia Celebration of Diversity in Computing Conference: Intellect, Initiatives, Insight, and Innovations. 2009, ACM: Portland, Oregon. p. 103-107.
9. Fabian, B. et al., A comparison of security requirements engineering methods. Requirements Engineering, 2009.
10. Khan, M.U.A. and M. Zulkernine, On Selecting Appropriate Development Processes and Requirements Engineering Methods for Secure Software, in 33rd Annual IEEE International Computer Software and Applications Conference. 2009, Computer Software and Applications Conference, Annual International. p. pp. 353-358.
11. Tondel, I.A., M.G. Jaatun, and P.H. Meland, Security requirements for the rest of us: A survey. IEEE Software, 2008. **25**(1): p. 20-27.
12. Diallo, M.H. et al., A Comparative Evaluation of Three Approaches to Specifying Security Requirements, in REFSQ. 12th International Working Conference on Requirements Engineering: Foundation for Software Quality. 2006: Luxembourg.

13. Mellado, D., E. Fernández-Medina, and M. Piattini, A comparative study of proposals for establishing security requirements for the development of secure information systems. 2006: Glasgow. p. 1044-1053.
14. Mead, N.R., Experiences in eliciting security requirements. CrossTalk, 2006. **19**(12): p. 14-19.
15. Opdahl, A.L. and G. Sindre, Experimental comparison of attack trees and misuse cases for security threat identification. Information and Software Technology, 2009. **51**(5): p. 916-932.
16. Mead, N.R., How To Compare the Security Quality Requirements Engineering (SQUARE) Method with Other Methods. Technical Report CMU/SEI-2007-TN-021, C.M.U. Software Engineering Institute, Editor. 2007.
17. Hadavi, M.A., V.S. Hamishagi, and H.M. Sangchi. Security Requirements Engineering; State of the Art and Research Challenges. in Proceedings of the International MultiConference of Engineers and Computer Scientists 2008 Vol I IMECS 2008. 2008. Hong Kong.
18. Biolchini, J. et al., Systematic review in software engineering. Rio de Janeiro Brazil, Systems Engineering and Computer Science Department, UFRJ. 2005.
19. Kitchenham, B.A. and S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE-2007-01, Keele University. EBSE Technical Report, 2007.
20. Firesmith, D.G., Security use cases. Journal of Object Technology, 2003. **2**(3): p. 53-64.
21. Jürjens, J., Model-Based Security Engineering with UML. 2005. p. 42-77.
22. Lodderstedt, T. et al., SecureUML: A UML-Based Modeling Language for Model-Driven Security, in Proceedings of the 5th International Conference on The Unified Modeling Language. 2002, Springer-Verlag. p. 426-441.
23. Mouratidis, H. and P. Giorgini, Secure Tropos: A security-oriented extension of the Tropos methodology. International Journal of Software Engineering and Knowledge Engineering, 2007. **17**(2): p. 285-309.
24. Yu, E.S.K. Towards modelling and reasoning support for early-phase requirements engineering. 1997. Annapolis, MD, USA: IEEE.
25. Rosado, D.G. et al., Analysis of Secure Mobile Grid Systems: A Systematic Approach. Information and Software Technology, 2010. **52**: p. p. 517-536.
26. Zannone, N., The si* modeling framework: Metamodel and applications. International Journal of Software Engineering and Knowledge Engineering, 2009. **19**(5): p. 727-746.
27. Massacci, F., J. Mylopoulos, and N. Zannone, Security Requirements Engineering: The SI* Modeling Language and the Secure Tropos Methodology. 2010. p. 147-174.
28. Jackson, M., Problem frames: analyzing and structuring software development problems. 2001: Addison-Wesley Longman Publishing Co., Inc. 390.
29. Hatebur, D., M. Heisel, and H. Schmidt, Security engineering using problem frames. 2006. p. 238-253.
30. Jennex, M.E., Modeling security requirements for information systems development. SREIS 2005, 2005.
31. Haley, C.B. et al., Security requirements engineering: A framework for representation and analysis. IEEE Transactions on Software Engineering, 2008. **34**(1): p. 133-153.
32. Sindre, G. and A.L. Opdahl, Eliciting security requirements with misuse cases. Requirements Engineering, 2005. **10**(1): p. 34-44.
33. Sindre, G., Mal-activity diagrams for capturing attacks on business processes. 2007: Trondheim. p. 355-366.
34. Hussein, M. and M. Zulkernine UMLintr: A UML Profile for Specifying Intrusions, in Engineering of Computer-Based Systems, IEEE International Conference on the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06). 2006. p. 279-288.
35. Zulkernine, M., M. Graves, and M.U.A. Khan, Integrating software specifications into intrusion detection. International Journal of Information Security, 2007. **6**(5): p. 345-357.
36. Schneier, B., Attack trees. Secrets & Lies: Digital Security in a Networked World, 2000: p. 318-333.
37. Peeters, J., Agile security requirements engineering. SREIS 2005, 2005.
38. Graves, M. and M. Zulkernine, Bridging the gap: Software specification meets intrusion detector. Proceedings of the Fourth Annual Conference on Privacy, Security and Trust (PST), 2006: p. 265-274.
39. Raihan, M. and M. Zulkernine. AsmLSec: An extension of abstract state machine language for attack scenario specification. 2007. Vienna.
40. Eckmann, S.T., G. Vigna, and R.A. Kemmerer, STATL: An attack language for state-based intrusion detection. Journal of Computer Security, 2002. **10**(1-2): p. 71-103.
41. ISO/IEC, ISO/IEC FCD 25010 Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Quality model and guide (DRAFT).
42. Sindre, G., D. Firesmith, and A. Opdahl, A Reuse-Based Approach to Determining Security Requirements. 9th International Workshop on Requirements Engineering: Fundation for Software Quality, 2003.
43. Basin, D., J. Doser, and T. Lodderstedt, Model driven security: From UML models to access control infrastructures. ACM Transactions on Software Engineering and Methodology, 2006. **15**(1): p. 39-91.
44. Saini, V., Q. Duan, and V. Paruchuri, Threat Modeling using Attack Trees. Journal of Computing Sciences in Colleges , Consortium for Computing Sciences in Colleges, USA., 2008. **vol. 23, no. 4.**

SEC-MDA 2010

The workshop addresses problems and solutions for Security in MDA.
The topics of interest include, but are not restricted to:

- Security Modelling
- Security requirements tracking in MDA
- Model-based security testing
- Transformation of model-based security knowledge
- Interoperability between security models
- Platform dependent and platform independent models for security solutions
- Model-based behavior analysis
- Security Tools using security models
- Security design patterns in MDA
- Abuse and Misuse cases
- Standards for modeling and sharing vulnerabilities and security issue knowledge
- Standards for storing and querying vulnerabilities and security issue knowledge bases
- Requirements for new security improved tools