ARES 2008

The Third International Conference on Availability, Security and Reliability

PROCEEDINGS

March 4-7, 2008 Barcelona, Spain

Edited by Stefan Jakoubi, Simon Tjoa, and Edgar R. Weippl

Organised by

[SECURE] Business Austria

DEXA

In cooperation with

UNIVERSITAT POLITÈCNICA DE CATALUNYA

enisa European Network and Information Security Agency

OESTERREICHISCHE COMPUTER GESELLSCHAFT AUSTRIAN COMPUTER SOCIETY

Proceedings of the

The Third International Conference on Availability, Security, and Reliability

March 4-7, 2008, Barcelona, Spain

CPS

Los Alamitos, California

Washington • Tokyo

IEEE computer society

IEEE Computer Society Order Number P3102
ISBN 0-7695-3102-4
ISBN 978-0-7695-3102-1
Library of Congress Number 2007909935

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
http://computer.org/cspress
csbooks@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
http://shop.ieee.org/store/
customer-service@ieee.org

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
tokyo.ofc@computer.org

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

Editorial production by Bob Werner
Cover art production by Joe Daigle/Studio Productions
Printed in the United States of America by The Printing House

IEEE computer society

CPS Conference Publishing Services

IEEE Computer Society
***Conference Publishing Services* (CPS)**
http://www.computer.org/cps

Table of Contents

The Third International Conference on Availability, Reliability and Security (ARES 2008)

Session 3: Models

Session 4: Database

Session 5: Mobile

Session 6: RBAC and Recommender

Session 7: Risk Management

Session 8: Networks

Session 9: Software

Session 4: Networks

Session 5: Crypto

Session 6: Crypto and Health

Session 7: Models and Networks

Session 8: IDS

Session 9: Hardware

Workshop FARES

Session 1: Miscellaneous

Session 2: Access Control and Algorithms

Session 3: Crypto

Session 4: Risk Management

Workshop PSAI

Workshop APE

Workshop SECSE

Workshop DAWAM

Workshop WAIS

Workshop WSDF

Workshop SREIS

Chair's Message

The Third International Conference on Availability, Reliability and Security (ARES 2008 – The International Dependability Conference) brings together researchers and practitioners in the area of dependability. ARES 2008 highlights the various aspects of dependability, with special focus on the crucial linkage between availability, reliability and security.

ARES aims at a full and detailed discussion of research issues in the field of dependability as an integrative concept that covers amongst others availability, safety, confidentiality, integrity, maintainability and security and their different areas of applications.

This conference emphasizes the interplay between foundations and practical issues of dependability in areas such as information systems, e-government, m-government, location-based services, ubiquitous computing, and autonomous computing.

This years ARES conference is devoted to the critical examination and research challenges of the various aspects of Dependable Computing and the definition of a future road map.

We are very happy to welcome three well-known keynote speakers:

- Prof. Ravi Sandhu (Executive Director, Institute for Cyber-Security Research (ICSR) and Lutcher Brown Endowed Chair in Cyber-Security, University of Texas, San Antonio)
- Prof. Vijay Atluri (Management Science and Information Systems Department, Rutgers University)
- Prof. Günther Pernul (Department of Information Systems, University of Regensburg)

From over 200 submissions we have selected the 44 best for a presentation as full paper. In addition, several workshops and short papers show ongoing research projects and offer interesting starting points for discussions.

We wish all participants an enjoyable conference and interesting discussions.

Edgar R. Weippl, Secure Business Austria, Vienna University of Technology

Gerald Quirchmayr, University of Vienna and University of South Australia

Jill Slay, University of South Australia

Conference Officers

Honorary Co-Chairs
Roland Wagner, University of Linz, Austria

General Co-Chairs
Guenther Pernul, University of Regensburg, Germany
Makoto Takizawa, Tokyo Denki University, Japan

Program Co-Chairs
Gerald Quirchmayr, University of South Australia, Australia
Jill Slay, University of South Australia, Australia
Edgar Weippl, Vienna University of Technology / Secure Business Austria, Austria

Workshops Co-Chairs
Leonard Barolli, Fukuoka Institute of Technology (FIT), Japan
A Min Tjoa, Vienna University of Technology, Austria

Organizing Chair
Fatos Xhafa, Technical University of Catalonia, Spain

International Liaison Co-Chairs
Maria Wimmer, University of Koblenz-Landau, Germany
Charles Shoniregun, University of East London, United Kingdom

Publicity Chair
Vladimir Marik, Czech Technical University, Czech Republic

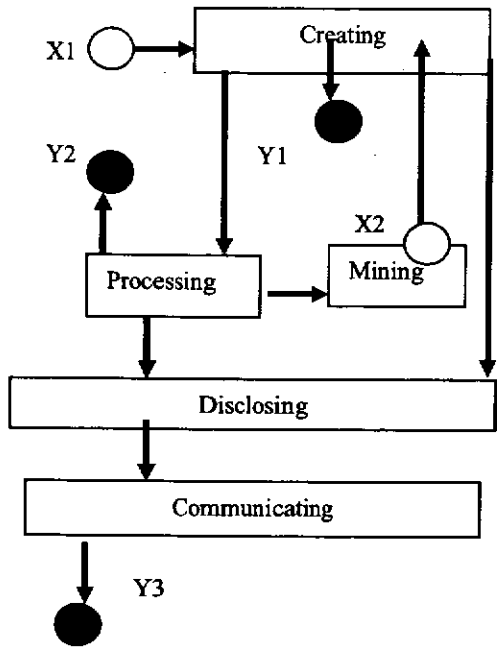

Figure 12. Entry and exit points for internally generated data.

VIII. CONCONCLUSION

One of the important aspects of security organization is to establish a framework to identify significant security points where policies and procedures are declared. The (information) security infrastructure comprises entities, processes, and technology. All are participants in handling information, which is the item that needs to be protected. Our approach is to identify information stream as the principal focus of security consideration. It is based on identifying points of transformation in the flow of information as the pivots around which security organization is built.

Currently, there is an implicit understanding of such a method. For example, in a computer system critical security posts are built around a core set of functionalities such as access control of stored information, internal processes, logon accounts, external communication connections, etc. Inside a single system (e.g., database security) security functions are unsystematically allocated according to information type, operation type, output type (e.g., mining), etc. The information stream model introduces a general coherent blueprint of security-significant posts that complements identification of security-related controls.

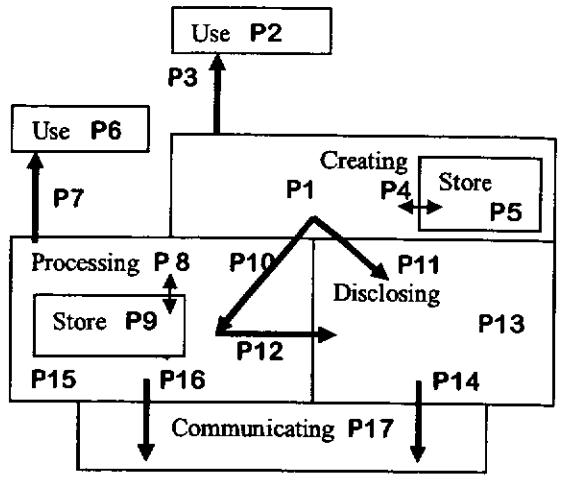

Figure 13. Security agents for internally generated

REFERENCES

[1] Al-Fedaghi, S. Aspects of Personal Information Theory, 7th, The Seventh Annual IEEE Information Assurance Workshop (IEEE-IAW), West Point, NY: United States Military Academy, June 20-23.2006.

[2] Al-Fedaghi, S., Personal Information Model for P3P, W3C Workshop on Languages for Privacy Policy Negotiation and Semantics-Driven Enforcement, 17 and 18 October 2006, Ispra/Italy.

[3] Beauregard, J. MODELING INFORMATION ASSURANCE, Thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, March 2001. http://www.iwar.org.uk/iwar/resources/usaf/maxwell/students/2001/afit-gor-ens-01m-03.pdf

[4] Blain, B. The Story of the Information Chain Theory, 2001 The Edwardsville Journal of Sociology, Volume 1. http://www.siue.edu/SOCIOLOGY/journal/blain.htm

[5] Commission of European Communities. Information technology security evaluation criteria, version 1.2, 1991.

[6] Hafiz M., and Johnson, R. Security Patterns and their Classification Schemes, https://netfiles.uiuc.edu/mhafiz/www/ResearchandPublications/secpatclassify.pdf

[7] Maconachy, W., Schou, C., Ragsdale D. and Welch, D. A Model for Information Assurance: An Integrated Approach, Proceedings of the 2001 IEEE Workshop on Information Assurance and Security United States Military Academy, West Point, NY, 5-6 June.

[8] McCumber, John. "Information Systems Security: A Comprehensive Model". Proceedings 14th National Computer Security Conference. National Institute of Standards and Technology. Baltimore, MD. October 1991.

[9] McCumber, J. "Application of the comprehensive INFOSEC Model: Mapping the Canadian Criteria for Systems Certification, Unpublished Manuscript, February 1993.

[10] Tomhave, B. Alphabet Soup: Making Sense of Models, Frameworks, and Methodologies, 8/16/2005. http://falcon.secureconsulting.net/professional/papers/Alphabet_Soup.pdf

Security Requirements Variability for Software Product Lines

Daniel Mellado
Ministry of Work and Social Affairs; Social Security IT Department, Madrid, Spain
Daniel.Mellado@alu.uclm.es

Eduardo Fernández-Medina
University of Castilla La-Mancha, Alarcos Research Group, Information Systems and Technologies Department, Spain.
Eduardo.FdezMedina@uclm.es

Mario Piattini
University of Castilla La-Mancha, Alarcos Research Group, Information Systems and Technologies Department, Spain.
Mario.Piattini@uclm.es

Abstract

Software product line engineering has proven to be one of the most successful paradigms for developing a diversity of similar software applications and software-intensive systems at low costs, in short time, and with high quality, by exploiting commonalities and variabilities among products to achieve high levels of reuse. At the same time, due to the complexity and extensive nature of product line development, security and requirements engineering are critical success factors in the development of a software product line. However, most of the current product line practices in requirements engineering do not adequately address the security requirements engineering. Therefore, in this paper we will propose a security requirements decision model driven by security standards along with a security variability model to manage the variability of the security requirements related artefacts. The aim of this approach is to deal with security requirements from the early stages of the product line development in a systematic way, in order to facilitate the conformance to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408.

1. Introduction

In search for improved software quality and high productivity, software product line (SPL) based development has become the most successful approach in the reuse field, because it can help us significantly reduce time-to-market as well as development costs [2, 4], by maximizing software reuse and managing variability, which also assists in achieving software of high quality and maintaining a desired quality level of products [24].

Security is a cross-cutting concern in software intensive systems and should consequently be subject to careful requirements analysis and decision making. Moreover, in SPL engineering, security is one of the most important attributes concerning quality, given that a weakness in security can cause problems throughout all the products of a product line. Specifying requirements for a SPL is a challenging task [24], and specifying security quality requirements for a SPL is even more challenging due to the varying security properties required in different products. Therefore, the discipline known as Security Requirements Engineering is a very important part of the SPL development process for the achievement of secure SPL and products, because it provides techniques, methods, standards and systematic and repeatable procedures for tackling SPL security requirement issues throughout the SPL development lifecycle both to ensure the definition of security quality requirements and to manage variability of security properties. Nevertheless, software engineering methodologies and standard proposals of SPL engineering have traditionally ignored security requirements and security variability issues.

In the last few years, it has been a spectacular growing of security standards and security requirements related proposals, such as [3, 6-8, 14, 18, 20, 21, 27]. Recently, several attempts have also been made for defining SPL architectures for security, as those of [1, 5, 9], although they are more oriented to the software solution than to security requirements. Thus, neither of the proposals provides a systematic approach to manage and trace security requirements, with the aim of including security requirements variability in the models of a SPL. In addition, after

0-7695-3102-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ARES.2008.165

analyzing most of them in [23], we also concluded that neither of them facilitates the SPL products security certification against the most relevant international security standards with respect to the management of security requirements (such as mainly ISO/IEC 15408 [11], ISO/IEC 27001 [13] and ISO/IEC 17799 [12]).

In this paper, we will present a proposal for the management of the security requirements variability as an extension of our already presented Security Requirements Engineering Process for Software Product Lines (SREPPLine) [22], where it was only described the process and its workflows without explaining the security variability management nor the model that is supported by the Security Resources Repository. Therefore the aim of this approach is to deal with the security requirements artefacts variability from the early stages of the product line development in a systematic and intuitive way, in order to facilitate the conformance of SPL products to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408 (Common Criteria). To this end, we will propose a security requirements decision model driven by security standards in order to assist in the SPL products security certification as well as a security variability model to manage the variability and traceability of the security requirements related artefacts of the SPL and its products.

The rest of the paper is structured as follows. In section 2 we will present our Security Requirements Engineering Process for software Product Lines (SREPPLine). Next, in section 3, we will explain our proposal for the security requirements variability management in SREPPLine. Finally, in section 4, we will discuss our contributions and future work.

2. Overview of SREPPLine: security requirements engineering process for software product lines

The Security Requirements Engineering Process for software Product Lines [22] (SREPPLine) is an add-in of activities, which can be incorporated into a SPL development process model of an organization providing it with a security requirements engineering approach.

SREPPLine is a security features or security goals based process which is driven by risk and security standards (concretely ISO/IEC 27001 and Common Criteria) that deals with security requirements and their related artefacts from the early stages of SPL development in a systematic and intuitive way especially adapted for SPL based development. It is based on the use of the latest security requirements techniques, such as security use cases [6] or misuse cases [27], as well as the integration of the Common Criteria (CC) components and ISO/IEC 27001 controls into the SPL lifecycle in order to facilitate SPL products security certification. Moreover, our proposed process suggests using a method to carry out the risk assessment which conforms to ISO/IEC 13335, specifically it uses Magerit [19] for both SPL risk assessment and SPL products risk assessment. Furthermore, SREPPLine has the aim of minimizing the necessary security standards knowledge as well as security expert participation during SPL products development. To this end, it provides a Security Core Assets Repository to facilitate the security artefacts reuse and to implement the security variability model and the security requirement decision model, that assist in the management of the variability and traceability of the security requirements related artefacts of the SPL and its products.

Our process is composed of two subprocesses with their respective activities: Product Line Security Domain Requirements Engineering subprocess and Product Line Security Application Requirements Engineering subprocess. These subprocesses cover the four basic phases of requirements engineering according to [15]: requirements elicitation; requirements analysis and negotiation; requirements documentation; and requirements validation and verification. At least, they have to be performed for each iteration of the Domain or Application Requirements Engineering Process of the SPL respectively.

3. Security requirements variability management

As the existing variability management approaches address functional variability instead of security requirements variability, in SREPPLine we will propose a Security Core Assets Repository to support the security variability model and the security requirement decision model described below in order to assist in security requirements and their related artefacts reuse, as well as in the management of commonalities and variabilities of security requirements artefacts and their traceability links in the SPL lifecycle. In essence, it is a knowledge repository with a structure to support holistic security requirements reasoning in SPL.

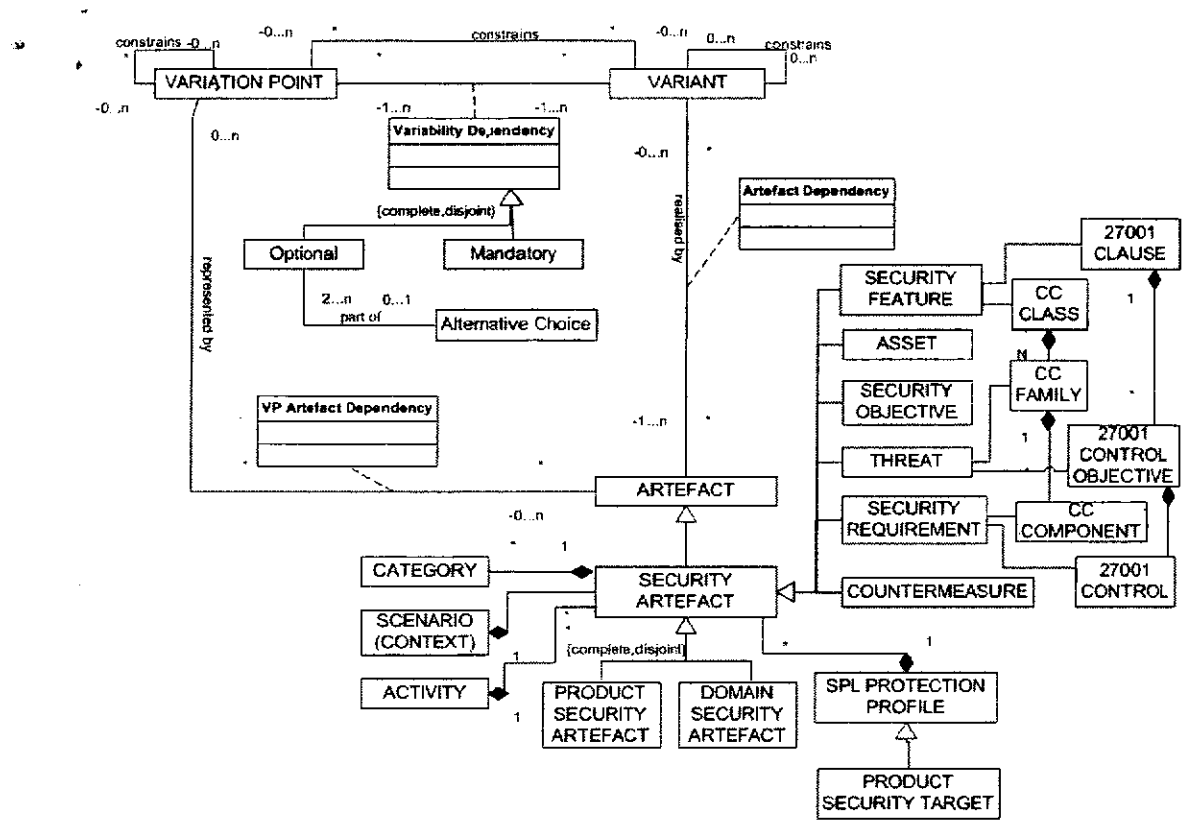

Fig. 1 Security variability meta-model

3.1. Security variability model

Our proposed Security Variability Model, which will be shown in Fig. 1 is based on the Reusable Assets Specification (RAS), adopted as OMG standard [25] and extends the orthogonal variability model of Pohl et al.[26] as well. Besides, it is part of the Security Requirement Decision Model. This variability model relates the defined variability to other software development models such as feature models, use case models, design models and test models. Thus, it is easier to adopt it in the SPL development methodology of an IT-department. Furthermore, it provides a cross-cutting view of the security variability across all security development artefacts and assists in keeping the different views of variable security requirements artefacts consistent.

The basic elements of our security variability model are defined in the meta model using UML 2 as shown in Fig. 1. The three main elements of the orthogonal variability model [26] are 'variation point' and 'variant' and 'artefact' classes.

The association class 'variability dependency' relates a variation point to at least one variant and vice versa, and it could be either a mandatory or an optional relationship. The mandatory variability dependency states that a variant is required for a variation point to which it is related. This does not imply that this variant has to be included in all SPL applications. A mandatory variant is part of an application only if the related variation point is part of it. The optional variability dependency states that a variant related to the variation point can be part of a particular SPL application but does not need to be part of it. The alternative choice groups a set of variants which are related through an optional variability dependency to the same variation point and defines the range for the amount of optional variants to be selected for this group.

The 'variant to variation point constrains dependency' describes a relationship between a variant and a variation point where the selection of a variant requires or excludes the consideration of a variation point. A 'variant constraint dependency' describes a relationship between two variants, so that the selection of a variant requires or excludes the selection of another variant independent of the variation points the variants are associated with. Similarly, a 'variation

point constraint dependency' describes a relationship between two variation points, where a variation point requires or excludes the consideration of another variation point in order to be realised.

In addition, it is also important to relate the variability defined in the variability model to software artefacts specified in other models. Thereby, the meta-model depicted in Fig. 1 contains the class 'artefact' that represents any kind of development artefact. Particular development artefacts are sub-classes of the 'artefact' class, such as 'security artefact' which is a specialization of an artefact. A development artefact can but does not have to be related to one or several variants, but a variant must be related to at least one development artefact. Moreover, a development artefact can but does not have to be related to one or several variation points and vice versa.

Therefore, the Security Core Assets Repository, proposed in SREPPLine [22], must be integrated into the SPL core assets repository to facilitate these traceability links between the SPL variability model and the different types of security artefacts and the other development artefacts.

As it is depicted in Fig. 1, the 'security artefact' class has a complete and disjoint specialization relationship where any security artefact has to be a product security artefact or a SPL domain security artefact. Each 'security artefact' has to be part of the related artefacts of a development activity, but a development activity can but does not have to be related to one or several security artefacts. Similarly, a security artefact has to be part of a scenario (it has to have a context), although a scenario can but does not have to be related to one or several security artefacts. Furthermore, a security artefact can but does not have to be categorized. The 'category' class helps us avoid semantic problems and assists in reusing security artefacts, even in applying security patterns. It is a key class for the security requirement decision model, because it guides us throughout the categories to identify the security artefacts. Moreover, the class 'security artefact' has as version' as a mandatory attribute in order to facilitate the security artefacts versions traceability and variability, as there could be products with different versions of the same security artefacts.

Finally, in Fig. 1 we have represented the security standards variability, by integrating the Common Criteria (CC) elements, and the ISO/IEC 27001 controls into the security variability model. These security standards elements are related to the categories of some particular security artefacts (security features, threats and security requirements) with the aim of assisting in the SPL or SPL products certification against these standards and making their reasoning easier.

3.2. Security requirement decision model

We treat security artefacts as a natural source of variability among the products or SPL artefacts. In order to capture and manage knowledge related to security requirements we propose a security requirement decision model for SPL engineering, which will be shown in a different figure (Fig. 2) to make its comprehension easier.

This model facilitates the security requirements related artefacts reasoning and the security standards conformance. It supports capturing, specifying and reasoning about security requirements for both SPL and SPL members.

In terms of security requirements, this model has as a prime objective to construct systems (products) that carefully balance the risk with the impact on SPL members, that is, with the economical impact of implementing the countermeasures related to these security requirements. The proposed model is not, however, intended to be a tool for risk assessment, but a validated methodology which conforms ISO/IEC 13335 [10], such as Magerit [19] is proposed to support this process.

As a starting point we used the goals/soft-goals [3] and feature models and their correlations in order to take into consideration functional and non-functional requirements, concretely security requirements. To express the intentions of a system, goal models as well as feature models can be used, and in most cases define similar information [26]. Therefore, the interest of starting from goal/softgoal model comes from the fact that it allows us to decide (if the traceability links are carefully established) what security features are needed to reach the selected security goals and which is the optimal set of security features/goals of a determined priority in the context of the different scenarios of the SPL that provides the rationale of the selection. This supposes a rise in the abstraction level of the variants selection process, making the selection in the requirements level instead of in the design level [17].

In addition, within this model we characterize a SPL as a set of 'variation points' which are represented by 'features' or 'goals', and each goal can be achieved by lots of concrete ways, which are represented as 'scenarios'.

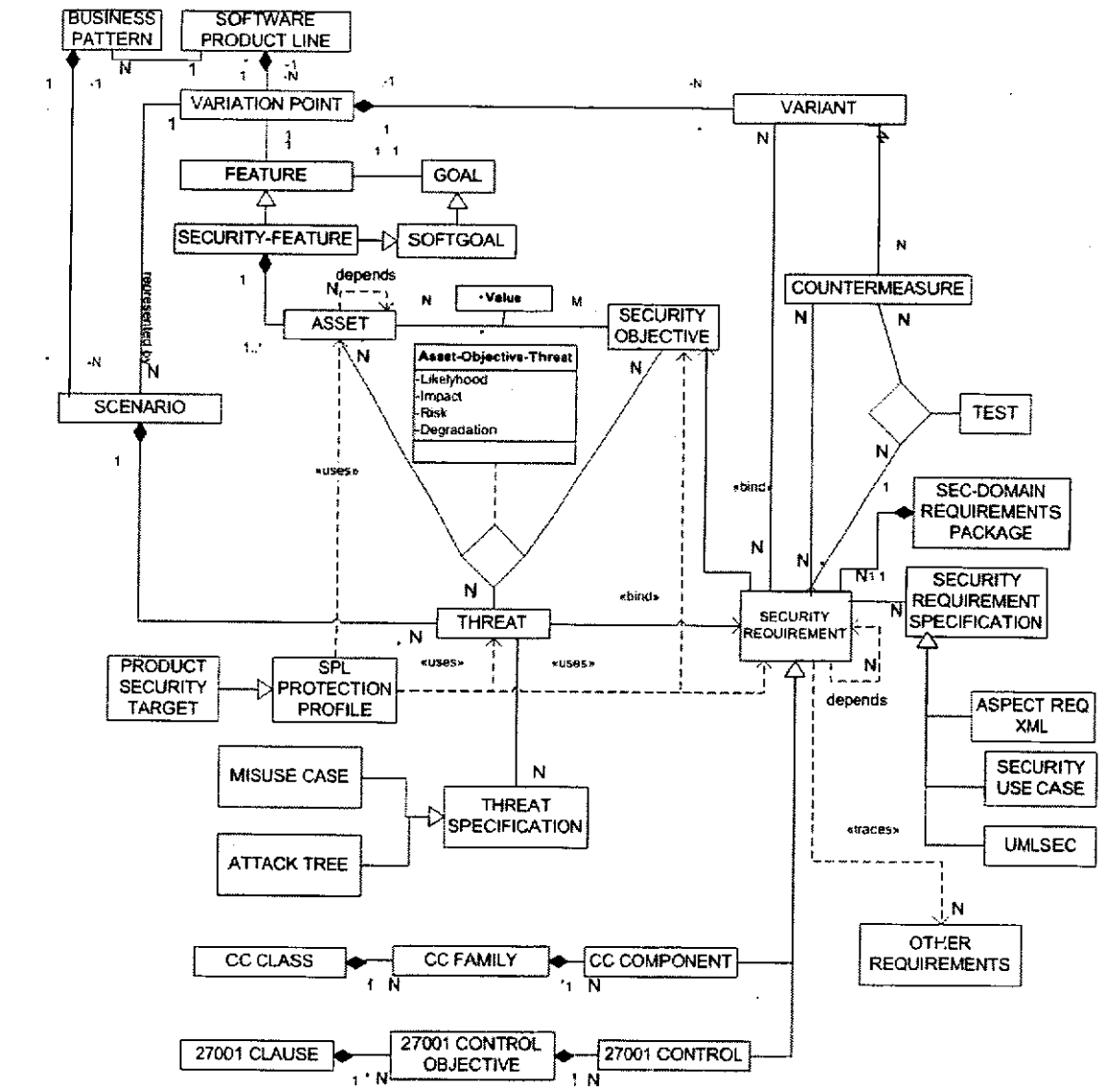

Fig. 2 Security requirement decision meta-model

The security core concepts that this model deals with are as follows: security feature, asset, security objective, threat, security requirement and SPL protection profile. Other supplementary security concepts include: risk, impact, degradation, likelihood, countermeasures and security standards concepts (CC and ISO/IEC 27001 concepts).

Security features are those features that describe security characteristics of the system which correspond with the security goals that the system under consideration should achieve. Thereby, a group of assets will be involved in the achievement of each security feature.

These assets are the resources in the information systems of the SPL or related to them that are necessary for the organization to operate correctly and achieve their goals. And there will be different categories or types of assets (such as the environment, information systems, services, components and information or data). Moreover, there could be dependencies between assets. Also, an asset, as it is shown in Fig. 1, is a class which inherits from the 'security artefact' class, so it can be a 'variation point'. Each asset will have different related security objectives (or security dimensions) with the corresponding assigned value (following a standardized scale from 0 to 10 according to the risk methodology called Magerit [19]) which is agreed by

the stakeholders, who have also to reach an agreement about the common and optional assets. The valuation of each asset is given in each security objective and it is propagated through the dependencies tree assets, therefore only the higher assets in the dependency tree have to be explicitly valued.

The security objectives or security dimensions are the objectives which must be achieved in order to protect the organization business goals. Following Magerit [19], the security objectives/dimensions managed by the model can only be the following ones: integrity, confidentiality, availability, authenticity of service users, authenticity of data origin, accountability of service use and accountability of data access. Throughout the selected category/ies of the asset, this model could propose security objectives related to these categories to assist in the security objectives identification and valuation for each asset.

Furthermore, the assets are exposed to threats which can prevent the security objective from being achieved. Not all threats affect all assets nor all their security objectives so the common and optional ones have to be identified. In addition, there is a certain relationship between the category of the asset and what could happen to it. Thus, throughout the selected category/ies of the asset this model could propose threats or categories of threats related to these categories of assets to assist in the common and optional threats identification and valuation. To calculate the impact of each threat, the value of the assets of each security objective along with the degradation caused by the threat are taken into account. To estimate the risk, the impact and the likelihood of occurrence of the threat are taken into account. Then, the risk is classified in a range from 0 (neglible) to 5 (very high) (according to Magerit [19] scale).

Each type (category) of asset and depending on their associated categories of threats will have related a category or categories of security requirements that could mitigate the impact or reduce the likelihood of these threats. This mechanism facilitates the elicitation of the common and optional security requirements of the SPL as well as the security requirements instantiation in the products. Moreover, there could be dependencies between security requirements, so there could be security requirements packages structured by the security dimension of the requirements, that is, they are a group of security requirements that work together in order to mitigate the same threats and satisfy similar security objectives of the assets. However, there still could be groups of requirements, which differ one from another in the level of detail they describe and in the testability they support. Therefore, a hierarchy of security requirements could be defined.

A security goal might be satisfied by multiple security requirements (different variants), and a mapping of the security requirements to countermeasures is carried out to give the best possible effect for the assets associated to the security feature. Thus, a variant is realised by one or more security requirements as well as it is supported by one or more countermeasures, which are procedures or technical mechanisms that reduce the risk and that are identified at the design stage. Countermeasures are architectural decisions that are used to achieve a security goal.

Furthermore, SPL Protection Profile is an implementation independent statement of security requirements and their related security artefacts that is shown to address threats that exist in a SPL environment; it has the aim of assisting in the SPL certification against the CC. There could be SPL Protection Profiles associated with Business Patterns. Similarly, the Product Security Target is an implementation dependent statement of security requirements and their related security artefacts for a specific identified product of the SPL; it has the purpose of facilitating the SPL product certification against the CC. In addition, there are security standards elements integrated into our proposed Security Requirement Decision model with the objective of assisting in the SPL or SPL products certification against these standards and making their reasoning easier. These security standards elements (CC elements and ISO/IEC 27001 controls) are related to the categories of some particular security artefacts (security features, threats and security requirements) to assist in this task.

We use scenarios to represent variants. All scenarios have an environment, a context that may include aspects such as assets, actors, misusers or misactors, use cases, misuse cases [27] or threats and security use cases [6]. Feature models contain variability themselves. However defining the variability of a feature tree with the orthogonal variability model enhances the expressive capabilities and helps us to document in a common way across different models without modifying the existing notations. We model threats as misuse case templates or attack trees in order to document the variability of them. Security requirements can be documented as security use case templates, UMLsec [14] with additional stereotypes, or as textual requirements by using aspect-XML specification [16].

The orthogonal variability model, in which our approach is based on, allows us to relate the different places at which the variability is defined to each other. In fact, starting from a changed artefact, other artefact affected by the change can be found by following the relationship with the associated variant and from the variant with the other associated artefact. Thereby, the variability of the security artefacts of the security decision model is clearly and unambiguously documented throughout the artefact dependencies of the security variability model.

4. Conclusions

Security requirements issues are extremely important in SPL because a weakness in security can cause problems throughout the lifecycle of a line. Although there have been several attempts to fill the gap between requirements engineering and SPL requirements engineering, there is not a systematic approach available for defining security quality requirements and managing the variability of them and their related security artefacts to the models of a SPL.

In [22] we only described the SREPPLine process and its workflows but it was not explained the management of the security variability nor the model that is supported by the Security Resources Repository. Therefore, the contribution of this work is that of providing, as an extension to SREPPLine [22], an holistic approach for the systematic management of the security requirements variability from the early stages of the product line development, in order to facilitate the conformance of the SPL products to the most relevant security standards with regard to the management of security requirements, such as ISO/IEC 27001 and ISO/IEC 15408 (Common Criteria). Our proposal defines a security requirements decision model driven by security standards in order to assist in the SPL security requirements definition and to facilitate products security certification. Moreover, it is proposed a security variability model to manage the variability and traceability of the security requirements related artefacts of the SPL and its products.

SREPPLine [22] together with the extension proposed in this paper is a suitable approach especially for SPL where security is a key quality issue. This is due to the broader impact of the existence or non-existence of specific security goals on all SPL members, as well as to the level of management of variable security features required for the diversity of market segments.

Finally, further work is also required to develop a CARE (Computer Aided Requirements Engineering) tool to support SREPPLine and the Security Resources Repository, and to assist in the complex management and maintainability of the variability and traceability relations. Furthermore, we will carry out a refinement of our approach by proving it with a real case study to validate and deeply illustrate SREPPLine, with the aim of providing an holistic framework for security requirements engineering in SPL.

5. Acknowledgments

This paper is part of the ESFINGE (TIN2006-15175-C05-05), DSMD (TIN2005-25866-E) and ELEPES (TIN2006-27690-E) projects of the Ministry of Education and Science (Spain), and of the MISTICO (PBC-06-0082) and DIMENSIONS (PBC-05-012-2) projects of the Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla- La Mancha and the FEDER.

6. References

[1] J. L. Arciniegas, J. C. Dueñas, J. L. Ruiz, R. Cerón, J. Bermejo, and M. A. Oltra, "Architecture Reasoning for Supporting Product Line Evolution: An Example on Security," in *Software Product Lines: Research Issues in Engineering and Management*, T. Käkölä and J. C. Dueñas, Eds.: Springer, 2006.

[2] J. Bosh, *Design & Use of Software Architectures*: Pearson Education Limited, 2000.

[3] L. Chung, B. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*: Kluwer Academic Publishers, 2000.

[4] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*: Addison-Wesley, 2002.

[5] T. E. Faegri and S. Hallsteinsen, "A Software Product Line Reference Architecture for Security," in *Software Product Lines: Research Issues in Engineering and Management*, T. Käkölä and J. C. Dueñas, Eds.: Springer, 2006.

[6] D. G. Firesmith, "Engineering Security Requirements," *Journal of Object Technology*, vol. 2, pp. 53-68, 2003.

[7] P. Giorgini, H. Mouratidis, and N. Zannone, "Modelling Security and Trust with Secure Tropos," in *Integrating Security and Software*

Engineering: Advances and Future Visions, H. Mouratidis and P. Giorgini, Eds.: Idea Group Publishing, 2007, pp. 160-189.

[8] C. B. Haley, R. Laney, J. D. Moffett, and B. Nuseibeh, "Arguing Satisfaction of Security Requirements," in *Integrating Security and Software Engineering: Advances and Future Visions*: Idea Group Publishing, 2007.

[9] A. Immonen, "A Method for Predicting Reliability and Availability at the Architecture Level," in *Software Product Lines: Research Issues in Engineering and Management*, T. Käkölä and J. C. Dueñas, Eds.: Springer, 2006.

[10] ISO/IEC, "ISO/IEC 13335 Information technology - Security techniques - Management of information and communications technology security - Part 1: Concepts and models for information and communications technology security management," 2004.

[11] ISO/IEC, "ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0)," 2005.

[12] ISO/IEC, "ISO/IEC 17799 Information technology - Security techniques - Code of practice for information security management," 2005.

[13] ISO/IEC, "ISO/IEC 27001 Information technology -- Security techniques -- Information security management systems -- Requirements.," 2006.

[14] J. Jürjens, "UMLsec: extending UML for secure systems development.," *UML 2002 - The Unified Modeling Language. Model Engineering, Languages, Concepts, and Tools. 5th International Conference.*, vol. LNCS 2460, pp. 412-425, 2002.

[15] G. Kotonya and I. Sommerville, *Requirements Engineering Process and Techniques*: John Willey & Sons, 2000.

[16] C. Kuloor and A. Eberlein, "Aspect-Oriented Requirements Engineering for Software Product Lines," presented at Proceedings of the 10 th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS'03), 2003.

[17] M. A. Laguna and B. Gonzalez-Baixauli, "Goals and MDA in Product Line Requirements Engineering," Departmente of Computer Science, University of Valladolid, Valladolid (Spain) GIRO-2005-01, 2005.

[18] L. Liu, E. Yu, and J. Mylopoulus, "Security and Privacy Requirements Analysis within Social Setting." 11th IEEE International Requirements Engineering Conference, 2003.

[19] F. López, M. A. Amutio, J. Candau, and J. A. Mañas, *Methodology for Information Systems Risk Analysis and Management*: Ministry of Public Administration, 2005.

[20] J. McDermott and C. Fox, "Using Abuse Case Models for Security Requirements Analysis," presented at Annual Computer Security Applications Conference, Phoenix, Arizona, 1999.

[21] N. R. Mead, "Identifying Security Requirements Using the Security Quality Requirements Engineering (SQUARE) Method," in *Integrating Security and Software Engineering: Advances and Future Visions*, H. Mouratidis and P. Giorgini, Eds.: Idea Group Publishing, 2007.

[22] D. Mellado, E. Fernandez-Medina, and M. Piattini, "SREPPLine: Towards a Security Requirements Engineering Process for Software Product Lines," *9th International Conference on Enterprise Information Systems (ICEIS 2007). 5th International Workshop on Security In Information Systems (WOSIS-2007)*, pp. 220-232, 2007.

[23] D. Mellado, E. Fernández-Medina, and M. Piattini, "A Comparison of the Common Criteria with Proposals of Information Systems Security Requirements," *"First International Conference on Availability, Reliability and Security" (ARES'06)*, pp. 654-661, 2006.

[24] E. Niemelä and A. Immonen, "Capturing quality requirements of product family architecture," in *Information & Software Technology*, vol. 49, 2007, pp. 1107-1120.

[25] OMG_(Object_Management_Group), "Reusable Assets Specification (RAS)," ptc/04-06-06, 2004.

[26] K. Pohl, G. Böckle, and F. v. d. Linden, *Software Product Line Engineering. Foundations, Principles and Techniques*. Berlin Heidelberg: Springer, 2005.

[27] G. Sindre and A. L. Opdahl, "Eliciting security requirements with misuse cases," *Requirements Engineering 10*, vol. 1, pp. 34-44, 2005.

Author Index

The Third International Conference on Availability, Reliability and Security (ARES 2008)

Notes

IEEE Computer Society Conference Publications Operations Committee

IEEE Computer Society Publications

The world-renowned IEEE Computer Society publishes, promotes, and distributes a wide variety of authoritative computer science and engineering texts. These books are available from most retail outlets. Visit the CS Store at *http://www.computer.org/portal/site/store/index.jsp* for a list of products.

IEEE Computer Society *Conference Publishing Services* (CPS)

The IEEE Computer Society produces conference publications for more than 250 acclaimed international conferences each year in a variety of formats, including books, CD-ROMs, USB Drives, and on-line publications. For information about the IEEE Computer Society's *Conference Publishing Services* (CPS), please e-mail: cps@computer.org or telephone +1-714-821-8380. Fax +1-714-761-1784. Additional information about *Conference Publishing Services* (CPS) can be accessed from our web site at: *http://www.computer.org/cps*

IEEE Computer Society / Wiley Partnership

The IEEE Computer Society and Wiley partnership allows the CS Press *Authored Book* program to produce a number of exciting new titles in areas of computer science and engineering with a special focus on software engineering. IEEE Computer Society members continue to receive a 15% discount on these titles when purchased through Wiley or at: *http://wiley.com/ieeecs*. To submit questions about the program or send proposals, please e-mail jwilson@computer.org or telephone +1-714-816-2112. Additional information regarding the Computer Society's authored book program can also be accessed from our web site at:
http://www.computer.org/portal/pages/ieeecs/publications/books/about.html

Revised: 21 January 2008

CPS Online™

IEEE Online Collaborative Publishing Environment

CPS Online is our innovative online collaborative conference publishing system designed to speed the delivery of price quotations and provide conferences with real-time access to all of a project's publication materials during production, including the final papers. The ***CPS Online*** workspace gives a conference the opportunity to upload files through any Web browser, check status and scheduling on their project, make changes to the Table of Contents and Front Matter, approve editorial changes and proofs, and communicate with their CPS editor through discussion forums, chat tools, commenting tools and e-mail.

The following is the URL link to the ***CPS Online*** Publishing Inquiry Form:
http://www.ieeeconfpublishing.org/cpir/inquiry/cps_inquiry.html

Published by the IEEE Computer Society
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

IEEE Computer Society Order Number P3102
Library of Congress Number 2007909935
ISBN 0-7695-3102-4