A Review of Security Requirements Engineering Methods with Respect to Risk Analysis and Model-Driven Engineering

Denisse Muñante, Vanea Chiprianov, Laurent Gallon, and Philippe Aniorté

LIUPPA University of Pau, France
{denisseyessica.munantearzapalo,laurent.gallon,
vanea.chiprianov,philippe.aniorte}@univ-pau.fr

Abstract. One of the most important aspects that help improve the quality and cost of secure information systems in their early stages of the development lifecycle is Security Requirements Engineering (SRE). However, obtaining such requirements is non-trivial. One domain dealing also with eliciting security requirements is Risk Analysis (RA). Therefore, we perform a review of SRE methods in order to analyse which ones are compatible with RA processes. Moreover, the transition from these early security requirements to security policies at later stages in the lifecycle is generally non-automatic, informal and incomplete. To deal with such issues, model-driven engineering (MDE) uses formal models and automatic model transformations. Therefore, we also review which SRE methods are compatible with MDE approaches. Consequently, our review is based on criteria derived partially from existing survey works, further enriched and specialized in order to evaluate the compatibility of SRE methods with the disciplines of RA and MDE. It summarizes the evidence regarding this issue so as to improve understanding and facilitate evaluating and selecting SRE methods.

Keywords: Security requirements engineering, risk analysis, model-driven engineering, review.

1 Introduction

Millions of dollars in losses are the result of attacks on unsecured systems. Many security breaches occur in software because errors and misspecifications in analysis, design and implementation [1]. Hence, information security is gaining more and more emphasis in recent years. In this sense, security requirements engineering (SRE) is an appropriate means to elucidate and model security requirements in the analysis stage in software development. Moreover, some works, such as UMLsec [7], SecureUML [8], MODELO [22] ..., allow us to define security aspects (policies) in the design and the software architecture stages. However, the transition from security requirements to security policies is generally non-automatic, unstructured and informal causing information loss, and thus the generation of incorrect security policies. To avoid these negative consequences, an automatic generation of

S. Teufel et al. (Eds.): CD-ARES 2014, LNCS 8708, pp. 79–93, 2014.

security policies starting from KAOS [11] is proposed in [25]. Despite of the fact that this work helps obtaining security policies of access control in a more formal way, it does not formalise its process of generation of security policies.

Model-driven engineering (MDE) encourages efficient use of models in several domains. Model-driven architecture (MDA) [20] uses model in the software development process and proposes three levels of abstraction: computation independent model (CIM), platform independent model (PIM) and platform specific model (PSM). A CIM presents what the system is expected to do (i.e. requirements), a PIM represents how the system reach its requirements out technical details (i.e. design and architecture) and a PSM combines the specification in PIMs with details required to stipulate how a system uses a particular type of platform. To build a software system, a series of transformations is performed: transformation from CIM to PIM, transformation from PIM to PSM, and transformation from PSM to code. To benefit from MDE advantages, some works have already been done to model Requirements Engineering (RE) as CIM. Tao et al. [21] review such approaches with respect to MDE principles, in particular on the possibility of (semi)automatic transformation from RE definition to PIM. In contrast, in this paper, we focus our study on SRE, i.e. regarding specific security concerns in requirement engineering, in order to analyse the compatibility of SRE methods with MDE approaches.

SRE methods often offer in practice just a general list of security features, which are implementation mechanisms rather than security requirements [26]. On the other hand, risk analysis (RA) is the activity of analyzing threat, vulnerability and impact on each component of the system. Therefore, RA could be used to elicit a more complete list of security requirements. So it is necessary to combine SRE methodologies with RA methodologies. Several works have already been proposed this: KAOS [25], Secure Tropos [24] [27], CORAS [15] [28].

To sum up, we investigate three disciplines. RA comprises processes that can help identify security requirements early in the development lifecycle of information systems. The definition of these security requirements is dealt with by the discipline of SRE. To enable automatic, formal transition from early stage requirements to later-stage security policies, we investigate the field of MDE. Therefore, the paper's contribution is a summary and a comparison of state of the art security requirements engineering methods according to risk analysis demands and model-driven exigencies. Consequently, this survey helps to improve understanding and facilitates the evaluation and selection of SRE methods as part of an MDE approach based on a RA process.

The remainder of this paper is organized as follows. Section 2 presents method of review. An analysis and discussion is presented in Section 3. Finally, Section 4 concludes this paper and gives perspectives.

2 Method of Review

Our method of review is based on two steps. Firstly, we identify and select criteria to classify SRE methods, related to MDE and RA points of view. Secondly, we identify and select SRE methods which we classify using our criteria.

As basis, we start by selecting from Karpati paper [2] criteria which are related to MDE. Karpati paper is a tertiary study (i.e. review of review papers) which defines groups of criteria, called dimensions, to categorize SRE methods. These criteria are extracted from all of those SRE review papers, up to 2010, they find in their study. We enrich this set of criteria with new criteria selected from more recent papers, in particular Salini paper [5] (2012). Salini is a review paper that analyses and compares SRE methods in order to guide developers to adopt SRE methods for software systems. None of these papers are MDE-oriented. Despite of this, after an analysis, we succeded to identify some criteria which are related to MDE principles (5 criteria). They are detailled in section 3.3.

We also indentify and select RA criteria mainly based on Fabian review paper [3]. Fabian paper defines a conceptual framework (CF) in order to categorize SRE methods. This CF establishes a vocabulary and the interrelations between the different notions used in security engineering. Among these different security notions, we select those which are related to a RA process (8 criteria). They are detailled in section 3.2.

Then, we select the list of SRE methods that we review starting from three SRE methods review papers: Fabian [3], Mellado [4] and Salini [5]. Fabian and Mellado are the most recent review papers taken into account by Karpati. Salini is a review paper more recent than Karpati paper. Fabian paper present a comparison of 18 fully developed SRE methods, classified using its conceptual framework. Mellado performs a systematic review on the SRE litterature of the period 2004 - 2009. It identifies 22 initiatives, and compares them using an analytical framework. Finaly, Salini classifies 11 different methods. Therefore, we are confident that our list contains most of SRE methods.

From these 32 distinct methods, we considered only the SRE methods that focus strictly on requirements. Although, some works considered UMLsec [7] and SecureUML [8] as SRE methods, we did not consider them because they are used in the system design. We employ the same reasoning for all the methods focus on later stages of the system development. In the same way, we did not study SRE methods that do not generate any analysis artifact such as agile methods, capability maturity model (CMM) methods, etc. Finally, the result of this selection is represented by 13 SRE methods. We evaluate these selected SRE methods and present an analysis of them in the next section.

3 Analysis and Discussion

In this section, we present our analysis and discussion of SRE methods. This analysis is divided in three parts. The first part introduces *the reviewed SRE methods*. The second one presents an evaluation of these SRE methods with respect to criteria corresponding to *risk analysis*. And, the third one presents an evaluation of these SRE methods with respect to criteria corresponding to *model-driven engineering*. For each part of our analysis, we present the criteria we used, a comparative table between SRE methods according to these criteria and a discussion of this comparison.

Notice that a dash symbol (-) in cells of the comparative tables implies that the method does not consider the related criterion. In contrast, the mark symbol (x) in cells indicates that the related criterion is considered by the method. Some criteria are described in a textual manner. Moreover, a table entry labelled with $\supseteq$ means that the notion defined in the considered method is used in a narrower sense than the related criteria.

3.1 The Security Requirements Engineering Methods Reviewed

In this section, we introduce the reviewed SRE methods. For this, we give a brief introduction of each SRE method. Then, we summarize these methods focussing on the main characteristics that are relevant to our study.

As we mentioned, we study 13 SRE methods:

(a) *Security quality requirements engineering methodology (SQUARE) [9]:* is a comprehensive methodology, which consists of 9 steps. Its aim is to integrate security requirements engineering into software development processes.
(b) *Misuse cases [10]:* extends use cases to represent behaviour not wanted in the system. Ordinary use cases represent requirements, security cases represent security requirements, and misuse cases represent security threats.
(c) *Keep all objectives satisfied (KAOS) [11] with anti-models:* extends KAOS to include the elaboration of security requirements using anti-models. An anti-model is constructed using obstacles. An obstacle negates existing goals of the system.
(d) *Secure Tropos [12]:* extends Tropos, which is a software development methodology, with new concepts to cover security modelling, such as the *security features* of the system-to-be.
(e) *Secure i* [13]:* extends i*-modeling framework with modeling and analysis of security trade-offs. Secure i* focuses on the alignment of security requirements with other requirements.
(f) *Goal-based req. analysis method (GBRAM) [14]:* allows to use goal- and scenario-driven req. engineering methods to formulate privacy and security policies.
(g) *CORAS [15]:* is a model-based method for security risk analysis. CORAS consists of eight steps, provides a customized language for threat and risk modelling, and comes with detailed guidelines explaining how the language should be used.
(h) *Tropos goal-risk framework [16]:* extends Tropos methodology to assess risk based on trust relations among actors. Risk analysis is used to evaluate alternative goals and to assess countermeasures to mitigate risks.
(i) *Model-based information system security risk management (ISSRM) [17]:* proposes a risk analysis process that consists of four steps.
(j) *Abuse Frames [29]:* is based on problem frame to define anti requirements (i.e. requirements for malicious users) and abuse frame to analyse security threats.

(k) *Security engineering process using patterns (SEPP) [30] [31]:* is a security engineering process based on security problem frames and associated solution approaches.They are defined using patterns.
(l) *Security requirements engineering framework (SREF) [32]:* is based on constructing a context for the system using a problem-oriented notation to represent security requirements as constraints, and to develop and evaluate satisfaction arguments for the security requirements.
(m) *Security requirements engineering process (SREP) [18]:* is an iterative and incremental process. Furthermore, SREP is asset-based, risk driven, and, following the Common Criteria (CC) supports the reuse of security requirements, as well as the reuse of knowledge on assets, threats, and countermeasures.

Following Fabian et al. [3], we classify the reviewed SRE methods using:

- *Type of method:* indicates the global/general type of a SRE method or a SRE process.
- *Method/Process:* corresponds to the name of a SRE method or a SRE process (hereinafter SRE method).

Following Fabian et al. [3], Mellado et al. [4] and Salini et al. [5], we give the main characteristics of these methods:

- *Contribution:* indicates the purpose of each selected SRE method. This characteristic is divided in *integration of standards* and *main contributions* [4].
- *Security Properties:* indicates the security properties accomplish by a SRE method [3]. This characteristic is divided in: *CIA* corresponding to confidentiality, integrity and availability security properties, and *Other* related to other security properties such as non-repudiation, authentication, and others.

In Table 1, the first, second and fourth columns summarize the reviewed SRE methods. According to the third column (integration standards criterion), three methods only consider the integration of a standard. CORAS considers the ISO 31000 standard, which is related to risk management. ISRRM uses the ISO 27001 standard, which is related to information security management. Abuse Frames considers the ISO 13335. And, SEPP and SREP include Common Criteria to propose a software lifecycle model.

According to this criterion, CORAS is only the method which are close to our analysis based on a risk analysis method.

Moreover, SQUARE is reported to be used by a few organizations [9]. It means that SQUARE is validated in the both academic and industrial context. Although we did not found a similar report for the other SRE methods, it does not imply that they are not employed for any organization (i.e. within an industrial context).

According to the fifth and sixth columns, almost all the SRE methods address security properties (8 SRE methods). SEPP and SREP address partially security properties, i.e. SEPP only addresses confidentiality and integrity security

Table 1. Relevant characteristics of security requirements engineering methods for our study

Type of method	Method / Process	Contribution		Secu Props	
		Integration Standards	Main Contributions	CIA	Other
Multilateral approaches	SQUARE	-	SQUARE: 9-step process for eliciting, categorizing and prioritizing security requirements.	x	x
UML-based approaches	Misuse cases	-	Executable misuse cases (UML extension for modeling threats in use case diagrams).	x	x
Goal-oriented approaches	KAOS	-	Use of antimodels to elaborate security requirements.	x	x
	Secure Tropos	-	Extension of Tropos methodology. Secure dependencies.	x	x
	Secure i*	-	i* framework for alignment of security requirements in organizations.	x	x
	GBRAM	-	Formulate privacy and security policies using heuristic activities.	-	-
Risk analysis-based approaches	CORAS	ISO 31000	Three artefacts (language, tool and process) to support a risk analysis activity.	-	-
	Tropos goal-risk	-	To assess risk based on trust relations among actors.	x	x
	ISSRM	ISO 27001	Security RE process: 4-step. It uses i* RE techniques.	x	x
Problem frame-based approaches	Abuse frames	ISO 13335	To define anti-requirements and abuse frames.	-	-
	SEPP	CC	To define Security Problem Frames (security requirements) and Concretized Security Proble Frames (security problem solutions).	-	x
	SREF	-	To analyse security goals, argumentation and software evolution.	x	x
Common Criteria	SREP	CC	Sw lifecycle model with multiple stages based on CC.	x	-

properties, whilst SREP only addresses confidentiality, integrity and availability (CIA) security properties. And, GBRAM, CORAS and Abuse Frames do not address security properties. Therefore, in terms of security properties, these 8 methods are the most compatible.

3.2 SRE Methods and Risk Analysis

In this section, we analyse the reviewed SRE methods with respect to criteria corresponding to a risk analysis process. For this, we present a set of criteria

Table 2. Correspondence between ISO 27005 terms and terms of the CF

ISO 27005 terms	**CF Fabian et al.**
Security objectives	Security goal
Security mechanism control	Security requirement
Interested parties	Stakeholder
Criteria definition (risk evaluation, impact and risk acceptation criteria)	Domain Knowledge
Asset	Asset
Threat	Threat
Vulnerability	Vulnerability
Risk	Risk

related to risk analysis, which are used to elaborate a comparative table. Finally, an analysis of this table is presented.

1) Criteria of comparison
As mentioned, our objective is to evaluate current SRE methods according to risk analysis. Among existing risk analysis approaches, we focus on the ISO 27005 [19] standard. ISO31000 talks about Risk Management covering concepts, definitions and methodology for a Risk Management process to be applied to any industry or activity. It is broad enough to be used by any activity touching the management of risks. ISO27005 talks about IT Risk Management. It uses the same framework described in 31000 and applies it to IT needs. It supports the general concepts specified in ISO/IEC 27001. It also revised and superseded ISO 13335. Common criteria defines concepts and principles of IT security evaluation. Security requirements can be defined, but mainly for an evaluation purpose. ISO 27005 takes into account Common Criteria. To sum up, ISO 27005 is the most recent IT Risk Management standard.

To choose our criteria, we employ the following method: Fristly, we select ISO 27005 terms as criteria using the list of terms resulting from the metrics analysis proposed by Mayer et al. [6]. This list of terms populates the first column of Table 2. Secondly, to reuse the analysis of SRE methods proposed by Fabian, we define a mapping between the previous selected criteria and the Fabian conceptual framework (CF). The terms of Fabian CF populates the second column of Table 2.

2) Comparison
Table 3 gives the result of the evaluation of SRE methods related to risk analysis criteria. Whe consider that a SRE method is totally compatible (or related) to the ISO 27005 risk analysis analysis process if it addresses all its concepts (i.e. all the criteria of Table 3). According to this table, the most compatible methods are *GBRAM* and *ISSRM* . So, these methods are compatible with ISO 27005 process.

Table 3. Evaluation of SRE methods related to Risk Analysis criteria

ISO 27005 criteria								
Method/ Process	**Security goal**	**Security requirement**	**Stakeholder**	**Domain Knowledge**	**Asset**	**Threat**	**Vulnerability**	**Risk**
SQUARE	x	System req.	⊇ Client	-	-	x	-	x
Misuse cases	x	-	⊇ Actor	-	x	x	-	x
KAOS	x	x	⊇ Agent	Domain properties, expectation	⊇ Object	x	x	-
Secure Tropos	Softgoal	cf. Security goal	⊇ Actor	-	-	x	x	x
Secure i*	Softgoal	cf. Security goal	⊇ Actor	-	x	x	x	-
GBRAM	x	x	x	-	Information	x	x	x
CORAS	x	-	-	Assumption	x	x	x	x
Tropos goal-risk	Softgoal	cf. Security goal	⊇ Actor	-	-	Event	-	x
ISSRM	Softgoal	cf. Security goal	⊇ Actor	Context	x	x	x	x
Abuse Frames	Sec. objective	Negated anti-req.	⊇ Biddable domain	-	x	x	x	-
SEPP	-	x	⊇ Biddable domain	Fact, assumption	⊇ Lexical domain, Phenomenon	x	-	-
SREF	x	x	⊇ Biddable domain	⊇ Fact, trust assumption	x	x	-	x
SREP	Sec. objective	x	-	Sec. objective	x	x	x	x

Moreover, the almost compatible methods are *KAOS*, *Secure Tropos*, *Secure i**, *Abuse Frames*, *SREF* and *SREP*. Secure i* includes almost all the terms, risk is not considered as entity but this term is considered as an evaluated value of *risk level*, so we consider that Secure i* is compatible with ISO 27005. KAOS is similar to Secure i* because it does not consider the *risk* term. In contrast to Secure i*, KAOS and Abuse Frames do not quantify the *risk level* of a system. So, they are a little bit less compatible than Secure i*. However, KAOS can improve its compatibility by including a quantification method that adds information related to risks (such as impact and exploitability) in contrast to Abuse Frames. On the other hand, SREF does not include the *vulnerability* term, but this term can be included to improve this SRE method.

On the other hand, Secure Tropos does not specify assets, thus security goals are related only to system goals. In ISO 27005, the asset identification is an important activity because the later stages use the valuable assets in order to evaluate and to protect them. Consequently, Secure Tropos is not clearly compatible with ISO 27005.

Despite of stakeholders are interested parties to perform a risk analysis in ISO 27005 (i.e. they establish the context composed of risk analysis objectives, criteria of estimation risk, ..), the conceptual representation of stakeholders is not essential in ISO 27005. Therefore, SREP is compatible with ISO 27005 in spite of it does not consider stakeholders as a conceptual entity.

In our evaluation, we find that SQUARE, misuse cases, Tropos Goal-Risk and SEPP do not fulfill enough criteria to be compatible with ISO 27005. Notice that, for misuse cases and CORAS security requirement criterion is not included according to the Fabian analysis [3]. This criterion is important because it corresponds to the target resulting from a requirement engineering process. Finally, from our comparison, KAOS, Secure i*, GBRAM, ISSRM, SREF and SREP are compatible with risk analysis approaches.

3.3 SRE Methods and Model-Driven Engineering

In this section, we analyse the reviewed SRE methods with respect to criteria corresponding to model-driven engineering. For this, we present a set of criteria related to a model-driven approach, these criteria are used to elaborate a comparative table. Finally, an analysis of this table is presented.

1) Criteria of comparison
As mentioned, our objective is to evaluate current SRE methods according to model-driven concepts. For this, we identify and select some model-driven criteria with the aim of analysing the possibility of the selected SRE methods to be part of a model-based approach. Therefore, we selected the criteria proposed in [3], [4] and [5]:

- *Security RE tools (language, profile, technique, etc.):* describes the means to identify or elucidate security requirements [4]. Criterion included to analyse if Security RE tools can be formalized.
- *Model/Standard of Development:* indicates if there is a formal language as the basis of the method [4]. This criterion is used to analyse if there is a model or standard used by a SRE method.
- *Support for other development stages:* indicates which later stages in software development are supported by security requirements [4]. This criterion allows to analyse if it is possible to define a transition (e.g. an MDE automatic model transformation) between artifacts in different stages of software development.
- *Formality:* corresponds to formal validation of the method, i.e. if a method can evaluate its outputs [3].

Table 4. Evaluation of SRE methods related to Model-Driven Engineering criteria

MDE criteria					
Method / Process	**Security RE tool (language, profile, technique, etc.)**	**Model/Standard of Development**	**Support for other development stages**	**Formality**	**Prototype**
SQUARE	Use/misuse cases, etc. (selection of elicitation technique)	-	Design, Testing	-	P-SQUARE tool (web application)
Misuse cases	Use/misuse cases	UML	Design	-	-
KAOS	KAOS + anti-models	Goal-oriented requirements	Not used	x	Objectiver (XML as the output format)
Secure Tropos	Tropos language, Secure Tropos	Agent oriented software development	Later requirements	x	SecTro (export diagram to XML)
Secure i*	Strategic Dependency and Strategic Rationale models	Agent oriented software development	Not used	x	Sistar and Serenity
GBRAM	Identify, elaborate, refine operationalized goals	Goal and scenario-driven requirements	Not used	-	SMaRT
CORAS	CORAS method	Modeling QFTP [1]	Not used	-	Coras
Tropos goal-risk	Risk analysis is used to evaluate alternative goals	Goal model	Later requirements	-	GR-tool
ISSRM	Risk analysis method	ISSRM domain model	Not used	-	-
Abuse Frames	Anti-requirements and abuse frames	-	Not used	-	-
SEPP	Security problem frames and concretized security problem frames	UML profile	Not used	-	UML4PF support tool
SREF	System context and satisfaction arguments	ESR (Evolving Security Requirements) metamodel	Not used	-	OpenPF / Decreasoner
SREP	Use/misuse cases	-	Not used	-	-

[1] Quality Of Service and Fault Tolerance Characteristics And Mechanisms

– *Prototype:* indicates if there is a prototype (tool) that implements the corresponding SRE method [5]. Criterion used to know if the SRE method is not only conceptual.

We elaborate a comparative table, which is presented and analysed in the next section.

2) Comparison

In Table 4 gives the result of the evaluation of SRE methods related to a model-driven approach. We consider that a SRE method is totally compatible to a MDE approach if the requirement it produces can be described as a model (CIM). It means that the SRE method addresses all the criteria of Table 4. The first column shows the security requirement engineering tool, which considers languages, profiles, techniques or others, used by a SRE method to elucidate security requirements. The second column shows if a formal model/standard of development is proposed by SRE methods. These two first columns allow to analyse if it is possible to represent the SRE method's elements through formal models. As we can see, SQUARE, Abuse Frames and SREP do not have their own model or standard because they are processes that use existing techniques such as use/misuse cases or problem frames. Consequently, they can be compatible with MDE whether the used technique is compatible with MDE.

Moreover, the third column in Table 4 shows which later stages in systems development are supported by SRE methods. For a model-driven approach, this criterion allows to analyse if there is a way to derive from security requirements to another concepts corresponding to later stages, i.e. if a model transformation process is feasible. As we can see, security requirements derived from SQUARE, misuse cases, Secure Tropos or Tropos goal-risk can be used by the later stages such as design, testing and later requirements. Therefore, they are good candidates for being compatible with MDE approaches. However, this does not mean that the other methods have to be excluded. There are works that define a transition between development stages, for example, a correspondence between KAOS and SecureUML is proposed in [23] to define security policies related of access control. And, in [24] a mapping between Secure Tropos and UMLsec is proposed.

In an object-oriented environment, later requirements stage presents the system-to-be, as a part of the design stage. Hence, Secure Tropos and Tropos goal-risk are not the most appropriate but they propose some evidences to define model transformations. In contrast, SQUARE and misuse cases give means to define model transformations, whilst, for the other methods, we have to find a correspondence between security requirements and others artifacts from later stages of an object-oriented system development.

Moreover, we use the *formality* criterion to determine if an SRE method is validated formally. KAOS, Secure Tropos and Secure i* are validated formally. In a model-driven approach, we can say that these SRE methods are appropriate thanks to their level of formality. Additionally, the majority of methods implement a prototype (tool). It implies that they have been implemented and tested except misuse cases, ISSRM, Abuse Frames and SREP.

Finally, we can conclude, KAOS, Secure Tropos, Secure i* and Tropos goal-risk are the most compatible methods with MDE approaches. That is because, Secure Tropos accomplishes all the MDE criteria, and KAOS, Secure i* and Tropos goal-risk consider all but one criterion, the *support for other development stages* criterion for KAOS and Secure i* and the *formality* criterion for Tropos goal-risk. However, these SRE methods can be extended to included elements in order to improve their compatibility with MDE.

3.4 General Discussion

Returning to previous analysis, CORAS is close to our analysis because it is based on ISO 31000. However, CORAS, GBRAM and Abuse Frames do not address security properties while SEPP and SREP address partially security properties. Hence, the 8 others methods are the most compatible in terms od security properties. On the other hand, KAOS, Secure i*, GBRAM, ISSRM and SREP are compatible with risk analysis approaches, whilst KAOS, Secure Tropos, Secure i* and Tropos goal-risk are compatible with model-driven approaches. As we can see, KAOS and Secure i* are compatible with security properties, RA and MDE approaches. Therefore, we conclude that they can be integrated into a model-driven approach based on a risk analysis. Notice that, this result does not imply limiting the selection of SRE methods to KAOS or Secure i*. KAOS and Secure i* are the most compatible, but some other SRE methods can be adapted to improve their compatibility. For example, Secure Tropos or Tropos goal-risk could be extended to include elements related to ISO 27005. Similarly, GBRAM, ISSRM, SREF or SREP could be extended to include elements related to MDE approaches. Moreover, notice that, GBRAM does not address security properties. This could be a real drawback if we want to be able to model the security policy of a system.

Despite KAOS and Secure i* being the most compatible SRE methods for our study, these methods do not consider any technique to derive from security requirements at an early stage of the system development to security policies at later stages (e.g. design, architecture and implementation stages). Hence, in order to reach a totally compatibility, it is necessary to study this derivation technique and define a (semi)automatic model transformation. Such a model transformation will allow to prevent an incorrect, incomplete or informal definition of security policies.

4 Conclusions

We compared and discussed various types of Security Requirements Engineering (SRE) methods and processes. Our perspective of comparison and evaluation uses some criteria defined by previous works found in the literature. We have selected criteria according to a risk analysis process (namely, we use the ISO 27005 standard) and a model-driven approach. One objective of these criteria is to analyse which SRE methods can be used to (semi)automatically derive from

security requirements at an early stage of system development lifecycle to later stages security policies. Another objective is to analyse which SRE methods can be used to evaluate/quantify the security/protection level of a system against attacks. Then, our analysis shows which SRE methods are suitable to be part of a model-based approach based on a risk analysis.

We conclude that *KAOS* and *Secure i** are the most compatible SRE methods with a model-driven approach because they use a model/standard of development and they are validated formally. Despite these methods not presenting all the risk analysis terms, we consider that extending the method to these concepts is feasible.

For future works, we will use this review to choose an SRE method, which will be part of a model-driven approach based on the risk analysis ISO 27005. Namely, we want to integrate this SRE method with MODELO [22], which is a UML profile that we proposed to build access control policies (i.e. OrBAC) at an abstract level. More precisely, we want to derive (semi)automatically the access control policy from the definition of these security requirements.

References

1. Anderson, R.: Security Engineering: A Guide to Building Dependable Distributed Systems. John Wiley and Sons (2001)
2. Karpati, P., Sindre, G., Opdahl, A.L.: Characterising and analysing security requirements modelling initiatives. In: Proceedings of the International Conference on Availability, Reliability and Security (ARES), pp. 710–715. IEEE Computer Society (2011)
3. Fabian, B., Gürses, S., Heisel, M., Santen, T., Schmidt, H.: A comparison of security requirements engineering methods. Requir. Eng. 15(1), 7–40 (2010)
4. Mellado, D., Blanco, C., Sánchez, L.E., Fernández-Medina, E.: A systematic review of security requirements engineering. Computer Standards & Interfaces 32(4), 153–165 (2010)
5. Salini, P., Kanmani, S.: Survey and analysis on Security Requirements Engineering. Computers & Electrical Engineering 38(6), 1785–1797 (2012)
6. Mayer, N., Dubois, E., Matulevicius, R., Heymans, P.: Towards a Measurement Framework for Security Risk Management. In: Modeling Security Workshop (MODSEC 2008), in conjunction with the 11th International Conference on Model Driven Engineering Languages and Systems (MODELS 2008), Toulouse, France (September 2008)
7. Jurjens, J.: UMLsec: Extending UML for secure systems development. In: Fifth International Conference on the Unified Modeling Language, Model Engineering, Languages Concepts and Tools (2002)
8. Lodderstedt, T., Basin, D., Doser, J.: SecureUML: A UML-Based Modeling Language for Model-Driven Security. In: Fifth International Conference on the Unified Modeling Language, Model Engineering, Languages Concepts and Tools (2002)
9. N. Mead, E. Houg, T. Stehney: Security quality requirements engineering (SQUARE) Methodology. Technical report CMU/SEI-2005-TR-009. Software Eng. Inst., Carnegie Mellon Univ. (2005)
10. Sindre, G., Opdahl, A.L.: Capturing security requirements by misuse cases. Presented at 14th Norwegian Informatics Conference (NIK 2001), Tromsø, Norway (2001)

11. van Lamsweerde, A.: Elaborating security requirements by construction of intentional anti-models. In: Proceedings of the 26th International Conference on Software Engineering, May 23-28, pp. 148–157 (2004)
12. Mouratidis, H., Giorgini, P.: Secure tropos: A security-oriented extension of the tropos methodology. Int. J. Softw. Eng. Knowl. Eng. 17(2), 285–309 (2007)
13. Elahi, G., Yu, E.: A goal oriented approach for modeling and analyzing security trade-offs. University of Toronto, Department of Computer Science. Technical report (2007)
14. Anton, A.I., Earp, J.B.: Strategies for developing policies and requirements for secure electronic commerce systems. Department of Computer Science, North Carolina State University. Technical report (2000)
15. Braber, F., Hogganvik, I., Lund, M.S., Stolen, K., Vraalsen, F.: Model-based security analysis in seven steps-a guided tour to the CORAS method. BT Technol. J. 25(1), 101–117 (2007)
16. Asnar, Y., Giorgini, P., Massacci, F., Zannon, N.: From trust to dependability through risk analysis. In: Proceedings of the International Conference on Availability, Reliability and Security (AReS), pp. 19–26. IEEE Computer Society (2007)
17. Mayer, N., Rifaut, A., Dubois, E.: Towards a risk-based security requirements engineering framework. In: Proceedings of the 11th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2005), in conjunction with the 17th Conference on Advanced Information Systems Engineering, CAiSE 2005 (2005)
18. Mellado, D., Fernandez-Medina, E., Piattini, M.: Applying a security requirements engineering process. In: Proceedings of the 11th European Conference on Research in Computer Security, Hamburg, Germany, September 18-20, pp. 192–206 (2006)
19. Hervé Schauer Consultants. ISO/IEC 27005:2011 Information technology – Security techniques – Information security risk management (2010)
20. Kleppe, A., Warmer, J., Bast, W.: MDA explained the model driven architecture: Practice and promise. Addison-Wesley, Boston (2003)
21. Yue, T., Briand, L.C., Labiche, Y.: A systematic review of transformation approaches between user requirements and analysis models. Requirements Engineering 16(2), 75–99 (2011)
22. Muñante, D., Gallon, L., Aniorté, P.: An approach based on Model-driven Engineering to define Security Policies using the access control model OrBAC. In: The Eight International Workshop on Frontiers in Availability, Reliability and Security (FARES 2013), in conjonction with the 8th ARES Conference (ARES 2013), September 2-6. University of Regensburg, Germany (2013)
23. Ledru, Y., Richier, J., Idani, A., Labiadh, M.: From KAOS to RBAC: A Case Study in Designing Access Control Rules from a Requirements Analysis. In: 6 me Conference sur la Scurit des Architectures Rseaux et des Systmes d'Information (SARSSI 2011). La Rochelle, France (2011)
24. Mouratidis, H., Jürjens, J., Fox, J.: Towards a comprehensive framework for secure systems development. In: Martinez, F.H., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 48–62. Springer, Heidelberg (2006)
25. Graa, M., Cuppens-Boulahia, N., Autrel, F., Azkia, H., Cuppens, F., Coatrieux, G., Cavalli, A., Mammar, A.: Using Requirements Engineering in an Automatic Security Policy Derivation Process. In: Garcia-Alfaro, J., Navarro-Arribas, G., Cuppens-Boulahia, N., de Capitani di Vimercati, S. (eds.) DPM 2011 and SETOP 2011. LNCS, vol. 7122, pp. 155–172. Springer, Heidelberg (2012)

26. Mead, N.R., Allen, J.H., Barnum, S.J., Ellison, R.J., McGraw, G.: Software Security Engineering: A Guide for Project Managers. Addison-Wesley Professional (2004)
27. Matulevicius, R., Mayer, N., Mouratidis, H., Dubois, E., Heymans, P., Genon, N.: Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Development. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 541–555. Springer, Heidelberg (2008)
28. Braber, F., Dimitrakos, T., Gran, B.A., Lund, M.S., Stolen, K., Aagedal, J.O.: The CORAS methodology: Model-based risk assessment using UML and UP. In: UML and the Unified Process, pp. 332–357. IGI Publishing (2003)
29. Lin, L., Nuseibeh, B., Ince, D., Jackson, M.: Using Abuse Frames to Bound the Scope of Security Problems. In: Proceedings of the 12th IEEE International Conference on Requirements Engineering (RE 2004), pp. 354–355. IEEE Computer Society (2004)
30. Hatebur, D., Heisel, M., Schmidt, H.: A security engineering process based on patterns. In: Proceedings of the International Workshop on Secure Systems Methodologies Using Patterns (SPatterns), pp. 734–738. IEEE Computer Society (2007)
31. Beckers, K., Hatebur, D., Heisel, M.: A problem-based threat analysis in compliance with Common Criteria. In: Proceedings of the International Conference on Availability, Reliability and Security (ARES 2013), pp. 111–120 (2013)
32. Haley, C.B., Laney, R., Moffett, J., Nuseibeh, B.: Security requirements engineering: A framework for representation and analysis. IEEE Trans. Softw. Eng. 34(1), 133–153 (2008)