Alignment of Misuse Cases with Security Risk Management

Raimundas Matulevičius
PReCISE, University of Namur,
rue Grandgagnage 21,
B-5000 Namur, Belgium
rma@info.fundp.ac.be

Nicolas Mayer
CRP Henri Tudor - CITI
29 Av. John F. Kennedy,
L-1855 Luxembourg
nicolas.mayer@tudor.lu

Patrick Heymans
PReCISE, University of Namur,
rue Grandgagnage 21,
B-5000 Namur, Belgium
phe@info.fundp.ac.be

Abstract

It is recognised that security has to be addressed through the whole system development process. However current practices address security only in late stages, i.e., development or maintenance. Due to the success of UML use cases, misuse cases have been accepted by industry as a means to tackle security. However misuse cases, firstly, lack a precise application process, secondly, are too general which results in under-definition or misinterpretation of their concepts. In this paper we examine misuse cases in the light of a reference model for information system security risk management (ISSRM). Using the well-known Meeting Scheduler example we show how misuse cases can be used to follow a security risk management process. Next we check the misuse case ontology according to the concepts found in current risk management standards. The paper suggests improvements for the conceptual appropriateness of misuse cases for the security risk domain.

1 Introduction

The importance of addressing security from the very beginning of system development is now widely acknowledged [7] [13]. Early consideration of security allows envisage threats, their consequences and countermeasures before a system is in place, rather than as a reaction to a possibly disastrous attack. Attacks usually require quick system fixes, which in turn compromise other system qualities such as maintainability and efficiency. In the early phases, examining security risks allows IS modellers to discard design alternatives that do not offer a sufficient security level, or to rescope or even cancel a project if the residual risk is deemed too high for the organisation to bear.

Various modelling languages [11], [14], [17], [25] were suggested in order to cope with security in different development phases. In this work we specifically focus on the risk security management during the early phases of the system development. We analyse misuse cases as proposed by Sindre and Opdahl [21]. Being simple to understand and use misuse cases have proven useful in industrial cases [1] [19]. However, the literature also identifies several limitations. Firstly, misuse cases is considered as an "open-ended method" [23] suggesting no precise guidelines and relying on the modeller's creativity [1]. Secondly, misuse cases are a general rather than a specific technique suggesting a wide range of application possibilities [1] [23]. This is an advantage, but also a problem because it results in under-definions or different interpretations of the language concepts.

In this paper we investigate how misuse cases can be applied to manage security risks. We question *how to improve misuse cases with a better support to analyse problems of security risk management.* We illustrate our analysis using the Meeting Scheduler, a well established exemplar in requirements engineering [9] [12] [26]. To perform security risk analysis we use the reference model for the information system risk management (ISSRM) [15] [16]. This reference model compiles the fundamental concepts of security risk management gathered from security standards and related literature, e.g., [6] [8] [10]. The ISSRM reference model helps checking that the concepts present in misuse cases are adequate and sufficient for security risk management.

The structure of the paper is as follows: in Section 2 we recall a typical risk management process and the ISSRM reference model. Section 3 outlines misuse cases. In Section 4 we apply misuse cases to Meeting Scheduler example. In Section 5 we discuss how misuse cases are aligned with the concepts of the ISSRM reference model. Finally Section 6 presents conclusions and future work.

2 ISSRM and Risk Management Process

The ISSRM reference model [15] [16] presented in Fig. 1, is inspired by, and compliant with the existing security standards, e.g., [6], [8], [10]. Like the Tropos goal-risk framework [2], the ISSRM reference model addresses security risk management at three different levels, combin-

ing together asset, risk, and risk treatment views. However the ISSRM reference model addresses the *information system* security while the Tropos goal-risk framework supports risk in general. This results in the ISSRM reference model being a more focussed and systematic approach motivated and reasoned by the aforementioned standards. In this section we recap some core definitions of ISSRM concepts; for more details see [15] [16].

Asset-related concepts describe what assets are important to protect, and what criteria guarantee asset security. An *asset* is anything that has value to the organisation and is necessary for achieving its objectives. A *business asset* describes information, processes, capabilities and skills inherent to the business, and that has value for it. An *IS asset* is a component of the IS supporting business assets. *Security criterion* characterises a property or constraint on business assets describing their security needs. They are most often confidentiality, integrity and availability, but sometimes, depending on the context, other specific criteria might be added, like non-repudiation or accountability.

Risk-related concepts present how the risk itself is defined. A *risk* is the combination of a threat with one or more vulnerabilities leading to a negative impact harming one or more of the assets. An *impact* describes the potential negative consequence of a risk that may harm assets of, when a threat is accomplished. An *event* is the combination of a threat and one or more vulnerabilities. A *vulnerability* describes a characteristic of an IS asset or group of IS assets and that can constitute a weakness or a flaw in terms of IS security. A *threat* characterises a potential attack or incident, which targets one or more IS assets that may lead to harm the assets. A *threat agent* is an agent that can potentially cause harm to IS assets. An *attack method* is a standard means by which a threat agent carries out a threat.

Risk treatment-related concepts describe what decisions, requirements and controls should be defined and implemented in order to mitigate possible risks. A *risk treatment* is the decision of the way of treating identified risks. They might include risk avoidance, risk reduction, risk transfer, and risk retention. A *security requirement* is the refinement into requirements of a risk treatment decision to mitigate the risk. *Controls* (countermeasures or safeguards) are means designed to improve security, specified by a security requirement, and implemented to comply with it.

Risk management process. The ISSRM activities follow the risk management process described in risk management standards [6] [8] [10]. It can be summarised into six steps briefly recalled bellow. See [15] [16] for more details.

The process begins with a (*a*) definition of the organisation's *context* and the *identification of its assets*. Next, one needs to determine the (*b*) *security objectives*, such as confidentiality, integrity and availability, based on the level of protection required for the assets. During (*c*) *risk analysis* one elicits which risks are harming assets and threatening security objectives. Once risk assessment is performed, decisions about (*d*) *risk treatment* (risk avoidance, risk reduction, risk transfer or risk retention) are taken. *Security requirements* (*e*) on the IS can thus be determined as security solutions to mitigate the risks. Requirements are instantiated into (*f*) *security controls* (countermeasures), which are implemented within the organisation. Finally it should be noted that the risk management process is iterative. After determination of the security controls new risks that overcome or are not addressed by these controls, can emerge.

3 Misuse cases

Misuse cases [23] have two representations: a graphical diagram (see Fig. 3) and a textual template (see Fig. 4). They come with a *security requirements process* [20] [23] which consists of (1) identifying critical assets, (2) defining security goals, (3) identifying threats, (4) identifying and analysing risks, and (5) defining security requirements.

Graphical Misuse Cases. An *actor* (e.g. Initiator or Participant in Fig. 2) specifies a role played by a user or any system that interacts with the subject [18]. A *use case* (e.g. Obtain available dates) is the specification of a set of actions performed by a system, which yields an observable result that is of value for one or more actors, or stakeholders of the system [18]. An *include* relationship defines that a use case contains the behaviour defined in another use case.

Fig. 3 depicts a misuse case diagram. A *misuse case* (e.g. Make agreement unavailable) describes "a sequence of actions, including variants, which a system or other entity can perform, interacting with misusers of the entity and causing harm to some stakeholder if the sequence is allowed to complete" [23]. A *misuser* (e.g., Attacker) is "an actor that initiates misuse cases, either intentionally or inadvertently" [23]. The *threaten* relationship targets a use case (e.g. Obtain agreement) that a misuse case (e.g. Disclose agreement) wants to harm. The *mitigates* relationship (see Fig. 5) characterises how security use cases (e.g., Perform cryptographic procedures) is defined as a countermeasure against a misuse case (e.g., Disclose agreement).

Textual template. The details of *use cases* are usually captured in the associated textual templates. Templates are important because they encourage developers to write clear and simple action sequences. Like ordinary use cases, misuse cases may be described textually using misuse case templates. Two ways of expressing misuse cases textually have been suggested: lightweight descriptions and extensive descriptions. A *lightweight* description is embedded in an ordinary template (such as [3]) and extends it with additional entries for *threat* specification. An *extensive* description supports a detailed analysis of security threats in a dedicated template [22] [23]. See Fig. 4 for an example.

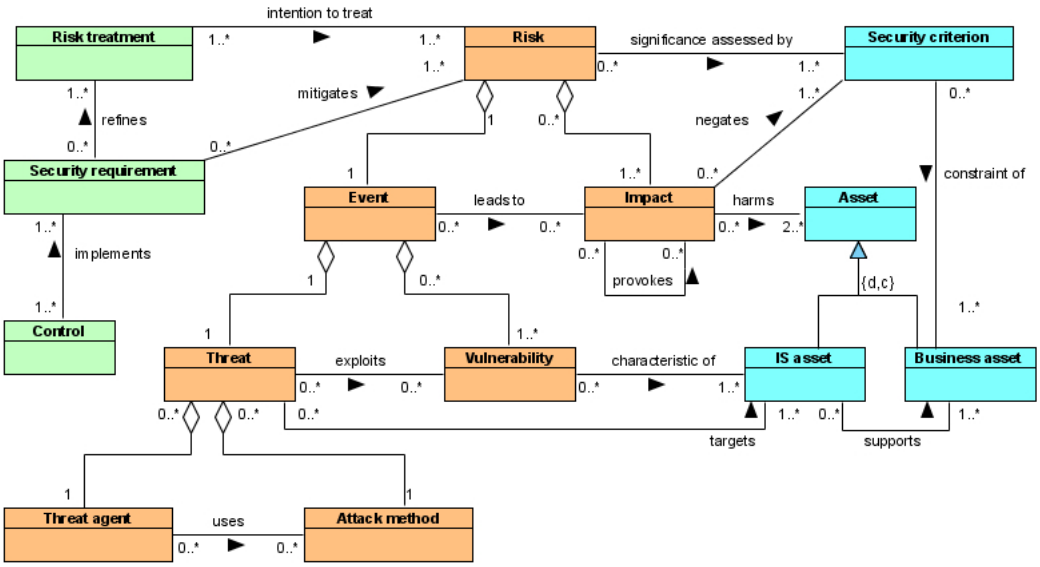

Figure 1. The ISSRM reference model [15] [16]

4 Modelling with Misuse Cases

In this section we illustrate how misuse cases can be used for security risk management in the *Meeting Scheduler* [9].

(*a*) *Context and Asset Identification.* In Fig. 2 we present a context of our example – it is shown in a use case diagram for the Meeting Scheduler. We focus on the Participant and Initiator, who communicate with Scheduler. Participant and Initiator are assets characterising workers (see discussion in Section 5). They seek to Find a date for a meeting. Find a date includes four use cases (described as process assets): Obtain available date, Find agreeable slot, Store agreement, and Obtain agreement.

(*b*) *Security objective determination.* Determination of vulnerable assets and security criteria is not supported by use cases. Use cases can only be used to reason about security criteria without showing them in a diagram. We concentrate on three security objectives: (*i*) *availability* of Agreement, meaning that once obtained the Agreement should be available and accessible both to initiator and participants; (*ii*) *confidentiality* of Agreement, meaning that only the participants who have the right to view the agreement information; and (*iii*) *integrity* of Agreement, meaning that once the agreement is confirmed, it cannot be changed.

(*c*) *Risk analysis and assessment.* In Fig. 3 we identify the misuse cases, which involve a misuser Attacker. The Attacker targets the *availability* with the misuse case Make agreement unavailable. It threatens the use case Store agreement. The same use case is threatened for *integrity* by the misuse case Change the date of agreement. The Attacker also threatens Obtain agreement with the misuse case Disclose agreement. The latter misuse case includes two other misuse cases: Steal date and Reveal stolen date, which describe certain steps in more details.

In Fig. 4 we illustrate the misuse case Disclose agreement with an extensive template. Here the system vulnerabilities are defined in the entries Assumption and Precondition. The way how these vulnerabilities are exploited by the misuse case are described in a Basic path.

(*d*) *Risk treatment.* The misuse cases do not suggest any risk treatment. Following the general security risk management process out of four possible risk treatments we select *risk reduction* by introducing *security use cases*.

(*e*) *Security requirements definition.* The use case Check participant identity (see Fig. 3) can be considered as the *security use case*, which mitigates the identified misuse case Disclose agreement. However the analysis showed that it is not sufficient to ensure the agreement confidentiality. As identified in the template for Disclose agreement (see Mitigation points in Fig. 4) this misuse case is mitigated the security use case Perform cryptographic procedures shown in Fig. 5.

(*f*) *Control selection and implementation.* Misuse cases do not suggest any techniques to select and implement controls. Thus one needs to resort to other means, for example, goal modelling [13] [17] [25] where the concept of a *softgoal* can help select between alternative controls.

5 Misuse cases and ISSRM Reference Model

In this section we analyse how misuse cases are interpreted with respect to the ISSRM concepts. Fig. 6 shows an alignment (we do not mean the exact match, but rather correspondence, similarity or overlap of concepts) between the

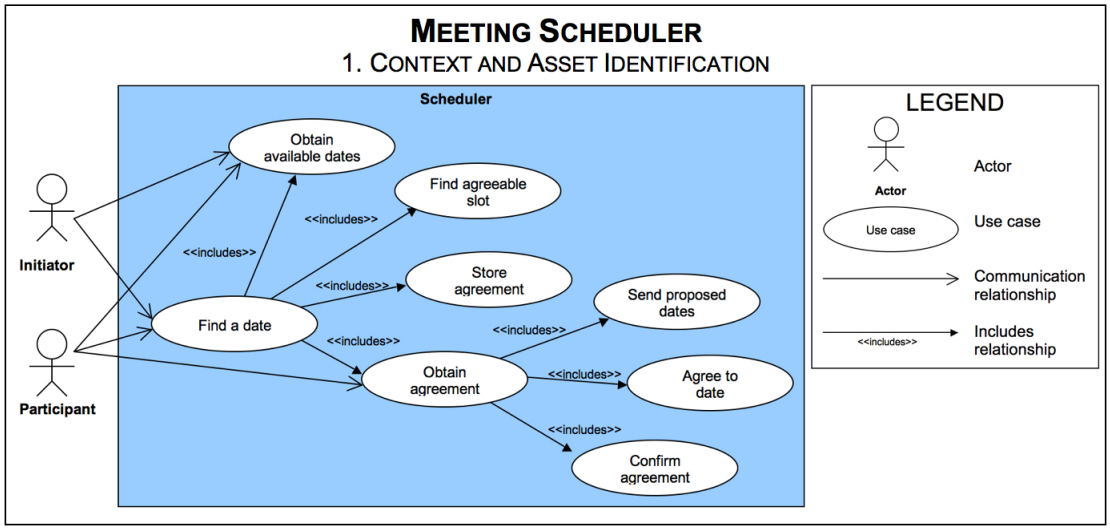

Figure 2. Asset Modelling

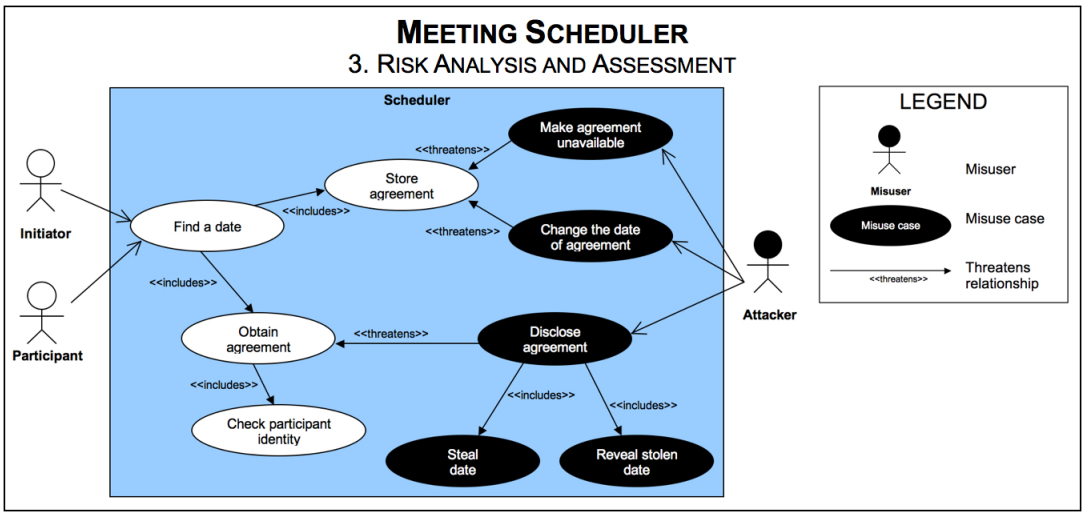

Figure 3. Threat Modelling

ISSRM reference model and misuse cases. In the *Synonyms* column we identify related terms found in the literature [20] [21] [22] [23] [24]. In the *Misuse case diagram* column we identify graphical constructs of misuse cases that correspond to the ISSRM concepts. In the next column we indicate elements of the example presented in Section 4. Finally in the *Misuse case template* column we identify alignment of misuse case template and the ISSRM reference model.

Asset-related Concepts. The most important assets in the organisation are identified as the knowledge and the skills of the workers; but, they are only vulnerable indirectly through the misuse of other more tangible assets [20]. According to [23], a use case "achieves something of value for the system owner". This corresponds to the ISSRM notion of asset. The process guidelines recommend "to concentrate on the normal actors and the main use cases requested by these" [22] and to identify the "critical assets in the system" [23]. Here, the notion of critical assets includes materials, information, locations, activities, knowledge and skills of workers [20], virtual locations, and computerised activities [23]. Thus, in Fig. 2 we consider use cases as ISSRM assets.

The literature provides various definitions for a use case:

- a means to understand and describe business processes, where they are called business use cases [3],
- a means of focusing discussion about requirements of the system to be built. Use cases are transformed into lists of typical functional requirements [3] , and

Name:	**Disclose agreement**
Summary:	Discloses agreement about the date (time and place) of the meeting to other people, who might not be authorised.
Basic path:	**bp1**: Attacker logs to the meeting scheduler system; **bp2**: System accepts login and password; **bp3**: Attacker reads, copies and stores the information about the agreement (extension point ext1); **bp4**: Attacker sends the information about the meeting to other people (extension point ext2).
Mitigation points:	**mp1**: Information is not possible to read because it is encrypted (in bp3), (extension point ext3).
Extension points:	**ext1**: Includes misuse case "Steal date" **ext2**: Includes misuse case "Reveal stolen date" **ext3**: Includes security use case "Perform cryptographic procedures"
Trigger:	Always true, this can happen at any time
Assumption:	**as1**: Agreement is not encrypted
Precondition:	**pr1**: Login information is known publicly, stolen, or a non-trusted participant gave his login information to the attacker.
Worst case threat:	The information of the meeting is distributed to the unauthorised people who might have bad intentions towards the information discussed during the meeting or towards the participants of the meeting.
Mitigation guarantee:	Attacker is not able to read the information (see mp1)
Related business rules:	The information about the meeting should be available only to the concerned meeting participants.
Misuser profile:	Skilled. Knowledge of scheduling systems. Knowledge of database query language.
Stakeholder and risks:	**Participant**: Waste of time by coming to the meeting with the non-relevant people (in the best case). Disclosing the confidential information if the spy was infiltrated during the meeting (in the average case – might be worst, depending on the information content) Being murdered (in the worst case). **Initiator**: Waste of time organising the meeting, loss of reputation, since the meeting was not securely planned. Loss of information and confidence about the potential participants.
Scope:	Business environment
Abstraction level:	Goal of the privacy attacker
Precision level:	Focussed

Figure 4. Example of the misuse case template

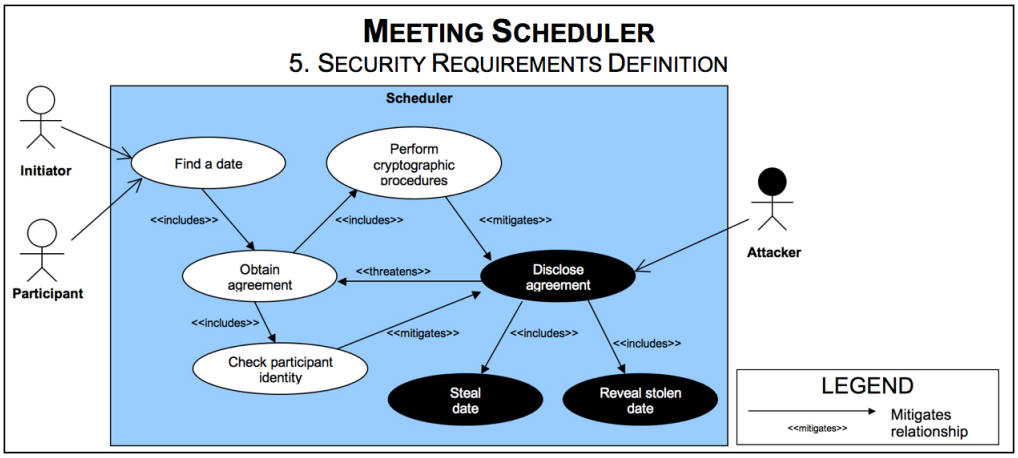

Figure 5. Modelling of Security Requirements

- a part of the functional requirements of the system to be built [3].

The first definition suggests to consider *business use cases* as ISSRM *business assets* (see Find a date, Obtain agreement in Fig. 2), whereas the second and the third definitions suggest to consider use cases as *IS assets*. However the literature does not precisely distinguish *business use cases* from ordinary *use cases*.

In misuse case literature we also find confusion seeking a correspondence for the notion of ISSRM security criteria. In [20], Sindre *et al.* speak about a security goal, which is specified "in terms of (1) who are the potential misusers, (2) the type of security breaches the asset is vulnerable to and (3) the security level necessary for that type of breach". Here, "the security types are violations of" [20] system integrity, availability and confidentiality, and is identified using security taxonomies. Elsewhere [23], the notion of security goal is different: "for each asset preferably aided by a standard typology of security goals" [23]. In both cases no specific graphical construct is suggested, so *security criteria* has to be specified using other modelling means.

Risk-related Concepts. The *risk* is "the estimated likelihood of occurrence and cost of the damage if the threat occurs" [20]. This definition corresponds to definition of *risk* in ISSRM in terms of involved concepts (this definition just put more emphasis on the level of risk than of the components of risk). The notion of *impact* in misuse cases appears as the cost of the damage. It is claimed that relationships *includes*, *extend*, and *generalises*, "identified between misuse cases can aid risk analysis" [23]. This means that

ISSRM domain model		Misuse cases				
		Synonyms	Misuse case diagram	Elements of the example		Misuse case template
Asset	Asset	Vulnerable asset, critical asset, materials, information, (virtual) location, (computerised) activities, knowledge and skills of workers	Actor and Use case	Fig.: 1. Context and asset identification	–	–
	IS asset	–			Obtain available dates, Find agreeable slot, Obtain agreement	–
	Business asset	Business use case			Store agreement, Confirm agreement	Related business rules
	Security criteria	Security goal, type of security breach	–		–	–
Risk	Risk	Risk of various threats	–	Fig.: 3. Risk analysis and assessment	–	Stakeholders and risks
	Impact	Cost of the damage, cost of potential losses	–		–	Worst case threat
	Cause of the risk	–	–		–	–
	Vulnerability	–	–		–	Assumption; Precondition
	Threat	Security threat	Misuser and Misuse case		Attacker and misuse cases Make agreement unavailable, Change the date of agreement, and Disclose agreement	–
	Threat agent	–	Misuser		Attacker	Misuser profile
	Attack method	Action sequence, sequence of both action and interaction, step	Misuse case		Make agreement unavailable, Change the date of agreement attacks, Disclose agreement attacks	Basic path; Alternative path; Extension points
Risk treatment	Risk treatment	–	–	Fig.: 5. Sec. Reqs	–	–
	Security requirements	Security use case, security requirement, countermeasure	Use case		Perform cryptographic procedures	Mitigation points
	Control	–	–		–	–

Figure 6. Concept Alignment between the Misuse Cases and ISSRM Reference Model

misuse cases can be defined at different level of abstraction. If a misuse case is defined at higher level it might refer to a risk. But the literature does not give any example. Thus *risk* remains a concept without a specific graphical notation.

"The security threats identified can be described as misuse cases and misusers" [23]. The statement corresponds to the ISSRM *threat*, which is composed of a threat agent and an attack method. Thus we identify correspondences between the *misuser*, who is the "actor that initiates misuse case" [23] (see Attacker in Fig. 3), and ISSRM *threat agent*. Also we align the misuse case, which is "a sequence of actions [...] interacting with misuser and causing harm to stakeholder" [23] (see Make agreement unavailable, Change the date of agreement, and Disclose agreement in Fig. 3) and the ISSRM *attack method*. Finally the *threatens* relationship, which indicates how a "use case is exploited or hindered by a misuse case" [23], can be seen as the *target* relationship between *threat* and *IS asset*.

Risk Treatment-related Concepts. Sindre *et al.* recommend "for each identified threat and taking its risk into account, [to] determine requirements to mitigate the threat" [20]. This means that "appropriate security requirements must be determined and specified" [20] and that "the use case is a countermeasure against a misuse case" [23]. Further, "security requirements defined are specified [...] as independent security use cases" [23] and the security use case must eventually have a *mitigate* relationship to a misuse case. This concludes that *security use cases* (see Perform cryptographic procedures in Fig. 5) correspond to the ISSRM *security requirements*.

The misuse case *mitigates* link corresponds to the ISSRM *mitigates* relationship. However here, the relationship is used at a lower level indicating how the threat (the misuse case) is mitigated by the means of the security use cases. The misuse cases do not indicate anything that would correspond to the ISSRM notions of *risk treatment* or *controls*.

Alignment of ISSRM and Misuse Case Template. Use case diagrams have to be understood only as a table of content for the textual templates to be filled for each of the use case. The *extensive template* for misuse cases is presented in [22] [23] and an example is given in Fig. 4.

The analysis of the *extensive template* [23] indicates only one *asset-related* entry, called *Related business rules* as a kind of ISSRM *business asset*. The *extensive template* concentrates on *risk-related* concepts. ISSRM *risk* is addressed by the entry *Stakeholders and risks*; ISSRM *vulnerability* – by the entries *Trigger*, *Assumption*, and *Precondition*; ISSRM *impact* – by the *worst case threat*; ISSRM *attack method* – by the *Basic path*, *Alternative path*, and *Extension points*. In the entry *Misuser profile* it is possible to give de-

tails of the misuser; it not only corresponds to the ISSRM *threat agent*, but also allows the modeller to provide more detailed description of the *threat agent*.

The misuse case template depends on the level of detail of the misuse case. If a misuse case is specified at a higher level (e.g., Disclose agreement in Fig. 3) the *Precondition* will correspond to ISSRM vulnerability. But if a misuse case is defined at a lower level of detail (e.g., Steal date, Reveal stolen date in Fig. 3) the *precondition* will define a system state where the misuse begins. Thus, the *precondition* will not have a correspondence in ISSRM. Similar issues arise with other template entries.

An *mitigation points* entry links a misuse case with security use cases. This means correspondence between *mitigation points* and ISSRM *security requirements*. Other details of the risk treatment can be specified in the templates of the security use cases.

Comparison of Security Management Processes. As shown in Fig. 7 the security requirements process [20] [23] does not fully correspond to the steps of the security risk management process. Application of misuse cases covers (*a*) *Context and asset identification*, (*c*) *Risk analysis and assessment* and (*e*) *Security requirements definition*. In [23] Sindre and Opdahl speak about (*b*) *Security objective determination* however the step is not directly supported by misuse cases, but is executed through other means. Finally, two steps – (*d*) *Risk treatment decisions* and (*e*) *Controls selection and implementation* – are not covered by the security requirements process [23].

Security risk management process (Mayer *et al.*, 2007)	**Security requirements process** (Sindre and Opdahl, 2005)	**Correspondence**
(*a*) Context and asset identification	(*1*) Identify critical assets in the system	Misuse cases
(*b*) Security objectives determination	(*2*) Define security goals	Other techniques
(*c*) Risk analysis and assessment	(*3*) Identify threats (*4*) Identify and analyse risks	Misuse cases
(*d*) Risk treatment	–	Not supported
(*e*) Security requirements definition	(*5*) Define security requirements	Misuse cases
(*f*) Control selection and implementation	–	Not supported

Figure 7. Comparison of Security Management Process

6 Conclusion and Future Work

In this paper we analyse the alignment between misuse cases [23] and the ISSRM reference model [15] [16]. The alignment is based on the misuse case meta-model [23] and textual explanations. Our analysis is supported by running example. On the one hand, it illustrates the use of language to address security risk management problems. On the other hand, the example tends to map the language and ISSRM concepts based on the settings of the modelled situation. Fig. 6 represents a clear view of the coverage of misuse cases with respect to the ISSRM reference model. The contribution of the work is twofold: (*i*) it strengthens the process guidelines for the misuse case application; (*ii*) it suggests improvements to misuse cases (both graphical diagrams and textual template) if used for the security risk management. More precisely the major results are:

- Misuse cases do not distinguish some constructs that represent different ISSRM concepts. For example, *IS assets*, *business assets* and *security requirements* are represented using the same visual construct for a use case. The modellers could tag the label to differentiate concepts. For example, in Fig. 2, the *use case* label [**BS**] Obtain available dates would indicate a *business asset*; [**IS**] Store available date would indicate an *IS asset*; in Fig. 5 [**SR**] Perform cryptographic procedures would mean a *security requirement*. However this might not completely solve the problem. For example in Fig. 2 the *use case* Store agreement might be understood as a *business asset* (store the agreement as the contract between participants and initiator) and as an *IS asset* (store the agreement in the database of the scheduler).

- For some concepts (e.g. *Security criteria*, *Risk*, and *Impact*), misuse cases do not provide modelling constructs. For instance ISSRM *risk* is not precisely defined. In [23] [24] risk is said to be represented using the *generalisation/specialisation*, but we did not find sufficient information on this. The only place where *risk* is specified is the template entry called *Stakeholders and risks*. Misuse cases do not cover all ISSRM concepts. For example when using diagrams one needs to decide how to model *security criteria*, *risk*, *impact*, *vulnerability*, *risk treatment decisions*, and *controls*. Some of these concepts can be defined in the misuse case template: for example *impact* in the entry *Worst case threat*, *vulnerability* in the entries *Assumption*, and *Precondition*. Other concepts can be defined by extending template with additional entries. But extending template gives a different level of granularity, thus the misuse case model might become complex.

- We observe partial coverage of some concepts. For example misuse cases allow modelling of assets such as workers who have skills and knowledge about the business, using the *actor* constructs. However the language excludes modelling of the threats to the actors.

The use of the ISSRM reference model can improve the analysis of IS security risks with security modelling language. When more languages are analysed using the same principle this can lead to better the interoperability between those languages: the analysis performed in one language can be completed with one or more other language(s) for the missing issues. For instance, besides misuse cases we have analysed two goal modelling languages, namely KAOS [25]

and Secure Tropos [17]. Goal modelling techniques supports reasoning about security decisions and in this way improve the analysis of security risks *wrt* misuse cases. We found that the Secure Tropos *security constraint* presents the ISSRM *security criteria*; the KAOS *domain property* corresponds to the ISSRM *vulnerability*; in KAOS and Secure Tropos control selection decisions can be argued for or against using *goals* and *softgoal*. This can improve and complement the application of misuse cases. The relationship between goal models and use case diagrams is considered in [4] [5]; however defining precise model traceability links remains a topic for future work.

Acknowledgment. The authors would like to thank E. Dubois and G. Sindre for support and discussions when performing this research.

References

[1] I. Alexander. Misuse Cases: Use Cases with Hostile Intent. *IEEE Software*, pages 58–66, 203.

[2] Y. Asnar and P. Giorgini. Modelling risk and identifying cuntermeasure in organizations. In *Proceedings of the 1st Interational Workshop on Critical Information Intrastructures Security*, pages 55–66. Springer-Verlag Berlin Heidelberg, 2006.

[3] B. Bernardez, A. Duran, and M. Genero. Metrics for Use Cases: A Survey of Current Proposals. In M. Genero, M. Piattini, and C. Colero, editors, *Metrics for Software Conceptual Models*, pages 59–98. Imperial College Press, 2005.

[4] L. Chung and S. Supakkul. Representing NFRs and FRs: a Goal-oriented and Use Case Driven Approach. In *Proceedings of the SERA 2004*, pages 29–41. Springer-Verlag Berlin Heidelberg, 2004.

[5] K. Cooper, S. P.Abraham, R. S. Unnithan, L. ChungChung, and S. Courtney. Integrating visual goal models into the Rational Unified Process. *Journal of Visual Languages and Computing*, 17:551–583, 2006.

[6] DCSSL. EBIOS - Expression of Needs and Identification of Security Objectives, 2004.

[7] P. T. Devanbu and S. Stubblebine. Software Engineering for Security: a Roadmap. In *Proceedings of the Conference on The Future of Software Engineering*. ACM Press, 2000.

[8] ENISA. Inventory of Risk Assessment and Risk Management Methods, 2004.

[9] M. S. Feather, S. Fickas, A. Finkelstein, and A. van Lamsweerde. Requirements and Specification Exemplars. *Automated Software Engineering*, 4:419–438, 1997.

[10] ISO. Information technology - Security techniques - Information security management systems - Requirements, International Organisation for Standardisation, 2005.

[11] J. Jurjens. UMLsec: Extending UML for Secure Systems Development. In *Proceedings of the 5th International Conference on the Unified Modelling Language (UML'02)*, pages 412–425, 2002.

[12] E. Letier. *Reasoning about Agents in Goal-oriented Requirements Engineering*. PhD thesis, Universite Catholique de Louvain, 2001.

[13] L. Liu, E. Yu, and J. Mylopoulos. Security and Privacy Requirements Analysis within a Social Setting. In *Proceedings of the 11th IEEE International Requirements Engineering Conference (RE'03)*, page 151. IEEE Computer Society, 2003.

[14] T. Lodderstedt, D. A. Basin, and J. Doser. SecureUML: A UML-based Modeling Language for Model-driven Security. In *Proceedings of the 5th International Conference on the Unified Modelling Language (UML'02)*, pages 426–441. Springer-Verlag, 2002.

[15] N. Mayer, P. Heymans, and R. Matulevičius. Design of a Modelling Language for Information System Security Risk Management. Technical report, CRP Henri Tudor and University of Namur, 2006.

[16] N. Mayer, P. Heymans, and R. Matulevičius. Design of a Modelling Language for Information System Security Risk Management. In *Proceedings of the 1st International Conference on Research Challenges in Information Science (RCIS 2007)*, pages 121–131, 2007.

[17] H. Mouratidis and P. Giorgini. Enhancing Secure TROPOS to Effectively Deal with Security Requirements in the Development of Multiagent Systems. In *Proceedings of the 1st International Workshop on Safety and Security in Multiagent Systems (AAMAS 2004)*, 2004.

[18] OMG. Unified Modeling Language: Superstructure, version 2.0, 2004.

[19] J. J. Pauli and D. Xu. Trade-off Analysis of Misuse Case-based Secure Software Architectures: A Case Study. In *Proceedings of the 3rd International Workshop on Modeling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS'05)*, pages 89–95. INSTICC Press, 2005.

[20] G. Sindre, D. Firesmith, and A. L. Opdahl. A Reuse-Based Approach to Determining Security Requirements. In *Proceedings of the International Workshop Requirements Engineering: Foundation for Software Quality (REFSQ 2003)*, 2003.

[21] G. Sindre and A. L. Opdahl. Eliciting Security Requirements by Misuse Cases. In *Proceedings of the TOOLS Pacific 2000*, 2000.

[22] G. Sindre and A. L. Opdahl. Template for Misuse Case Description. In *Proceedings of the International Workshop Requirements Engineering: Foundation for Software Quality (REFSQ 2001)*, 2001.

[23] G. Sindre and A. L. Opdahl. Eliciting Security Requirements with Misuse Cases. *Requirements Engineering Journal*, 10(1):34–44, 2005.

[24] G. Sindre, A. L. Opdahl, and G. F. Brevik. Generalization/Specialization as a Structuring Mechanism for Misuse Cases. In *Proceedings of the Symposium on Requirements Engineering for Information Security*, 2002.

[25] A. van Lamsweerde. Elaborating Security Requirements by Construction of Intentional Anti-models. In *Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*, pages 148–157. IEEE Computer Society, 2004.

[26] E. Yu. Towards Modeling and Reasoning Support for Early-phase Requirements Engineering. In *Proceedings of the 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, page 226. IEEE Computer Society, 1997.