

# Towards Security Risk-oriented Misuse Cases

Inam Soomro and Naved Ahmed

Institute of Computer Science, University of Tartu  
J. Liivi 2, 50409 Tartu, Estonia  
{inam, naved}@ut.ee

**Abstract.** Security has turned out to be a necessity of information systems (ISs) and information *per se*. Nevertheless, existing practices report on numerous cases when security aspects were considered only at the end of the development process, thus, missing the systematic security analysis. Misuse case diagrams help identify security concerns at early stages of the IS development. Despite this fundamental advantage, misuse cases tend to be rather imprecise; they do not comply with security risk management strategies, and, thus, could lead to misinterpretation of the security-related concepts. Such limitations could potentially result in poor security solutions. This paper applies a systematic approach to understand how misuse case diagrams could help model organisational assets, potential risks, and security countermeasures to mitigate these risks. The contribution helps understand how misuse cases could deal with security risk management and support reasoning for security requirements and their implementation in the software system.

**Keywords:** Security risk management, Misuse cases, Security engineering, Information system security

## 1 Introduction

During the last two decades, the line between digital and social life is diminishing, leading to modern society being mainly dependent on information systems (IS) and their security. The demand for IS security is constantly growing. Also, developing and maintaining system security is increasingly gaining attention. Consideration of IS security at the early stages of software development is also acknowledged in [18]. Security breaches in IS can lead to negative consequences. Practitioners of IS security must inspect security threats with a negative perspective from the very beginning of the IS development process. Consideration of security at early development stages assists in analysing and estimating security measures of the IS to be developed.

This paper discusses security risk management at requirement elicitation and analysis stages. We will consider the question “*how security risk management could be addressed using misuse case diagrams?*”. To answer this question, we analyse misuse cases proposed by Sindre and Opdahl [18]. Misuse case diagrams [17, 18] are one of the possible techniques to relate security analysis and functional requirements of software systems. The main goal is to model negative scenarios with respect to func-

tional requirements. The misuse cases are already proved to be useful in industry [15]. Existing misuse cases is relatively a simple language, since it contains few constructs to model security concerns. However the previous analysis [9] showed several limitations of misuse cases; for example, misuse cases do not comply with security risk management strategies, because they lack several concrete constructs to address secure assets, security risks and their countermeasures; misuse cases lack distinct constructs for representing security risk concepts These limitations could result in misinterpretation of the security-related concepts leading to poor security solutions. In this paper we tend to propose few improvement to the misuse cases diagrams.

We apply a systematic approach to understand how misuse case diagrams could help to model organisational assets, potential system risks, and security requirements to mitigate these risks. More specifically we introduce new constructs to extend the misuse cases in order to align their constructs with the concepts of Information Systems Security Risk Management (ISSRM) domain model [11, 12]. The benefit of syntactical and semantic extensions is that they introduce the missing semantics in to the language. The domain model is a touchstone to verify if the concepts presented are acceptable and appropriate for the security risk management.

The structure of the paper is organised as follows: in Section 2 we provide background knowledge needed for our study. In Section 3, we describe our research method and introduce Security Risk-oriented Misuse Cases (SRMUC) through an online banking example [1, 8]. Next we discuss alignment of SRMUC to ISSRM. In Section 4 we review the related work, discuss our results and conclude our study.

## 2 Background

### 2.1 Information System Security Risk Management (ISSRM)

Information System Security Risk Management (ISSRM) [11, 12] is a systematic approach, which addresses the security related issues in an IS domain. The model is defined after a survey of risk management and security related standards, security risk management methods and software engineering frameworks [12]. The domain model (see Fig. 1) supports the alignment of security modelling languages. It improves the IS security and security modelling languages as it conforms to the security risk management of organizations. The model describes three different conceptual categories:

**Asset-related concepts** describe the organization's assets grouped as *business asset* and *IS asset*. It also defines the *security criterion* as a constraint of a business asset expressed as *integrity*, *confidentiality* and *availability*.

**Risk-related concepts** define *risk*, potential harm to business, it is composed of a threat that contains one or more vulnerabilities, if executed successfully, harms the system assets which has negative consequences on assets defined as an *impact*. They negate the security criterion imposed by the business asset. An *event* is an abstraction aggregated as a threat and vulnerability where *vulnerability* is a weakness in a system that can be exploited by threat agent. A *threat* is a way to inflict an attack. It harms IS and business asset carried out by a threat agent and an attack method to target IS as-

sets. *Threat Agent* is an attacker that initiates a threat to harm the IS asset. *Attack Method* is a mean through which a threat agent executes a threat.

**Risk treatment related concepts** define a risk treatment *decision* to avoid, reduce, retain, or transfer the potential risks. It is refined by the *security requirement*. A *control* implements the security requirement.

The ISSRM process [11,12] is a 6-step process, based on existing risk analysis methodologies and standards. It starts with *context and asset identification* of the organization, proceeding to *determine the security objectives* for identified assets. Next, *risk analysis and assessment* to examine and estimate potential risks and its impacts. In next step, *risk treatment decisions* are taken to identify the security requirements. Finally, *security control* is implemented as security requirement. The process is iterative which may identify new risks and security controls.

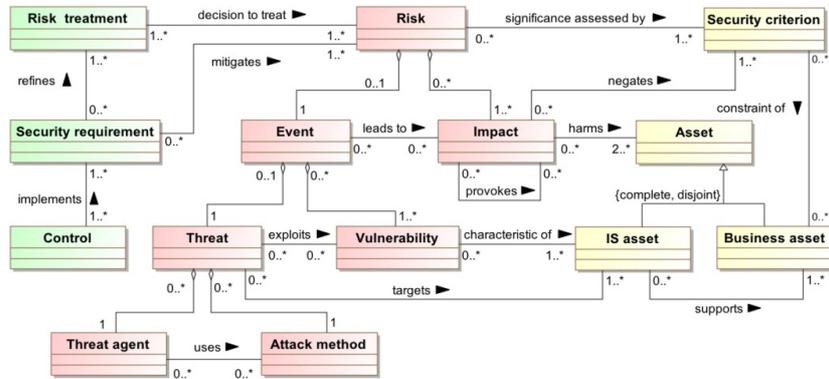


Fig. 1. ISSRM Domain Model [11]

## 2.2 Misuse Cases

Misuse cases are proposed by Sindre and Opdhal in [18]. They have extended the standard UML use cases to model security concerns at the early stages of software system development. The misuse cases include both the graphical notation and textual representation. Sindre and Opdahl define *misuse case* as a list or sequence of steps, if performed by an agent successfully, cause harm to the stakeholder and/or to the system. They define *misuser* as an actor that is willing to use the system with unfavourable intents. Initially, only threats were modelled as misuse cases. Later on, Sindre and Opdahl adapted the concept of security use case discussed by Firesmith [6] where security use cases are defined as a function to protect the system assets from the identified risks. In [16] Røstad has extended the misuse cases with a concept of vulnerability as weakness of the system (see a grey-filled use case in Fig. 3).

## 3 Security Risk-oriented Misuse Cases (SROMUC)

This section describes the research method used to develop SROMUC. We illustrate SROMUC using three different security scenarios on asset *integrity* (see Fig. 2, 3, and 4), *confidentiality* (see Fig. 5), and *availability* (see Fig. 6) in an example of online

banking. This section results in a conceptual alignment between SROMUC and ISSRM domain model.

### 3.1 Research Method

The main research objective of this study is to enable misuse cases to support the security risk management during the IS development. We followed a 3-step research method: firstly, we conduct literature review of security in IS and the ISSRM domain model to identify the security risk concepts. Secondly, we investigate how the misuse case diagrams express the security risk concepts. Hence, we observed the limitations of misuse cases in modelling the ISSRM concepts and executing the risk management process. Lastly, we define misuse case extensions, thus resulting in the Security Risk-oriented Misuse Cases (SROMUC). The extensions are done on all three components of the modelling language, namely concrete syntax, meta-model and semantics.

### 3.2 Scenario 1: SROMUC Modelling for Integrity

We illustrate the application of SROMUC using the online banking example [1, 8]. This scenario is particularly focussed on the IS integrity. To achieve better understandability, we split the scenario to 3 models<sup>1</sup>: one for assets (see Fig. 2), one for security threats (see Fig. 3), and one for security requirements (see Fig. 4).

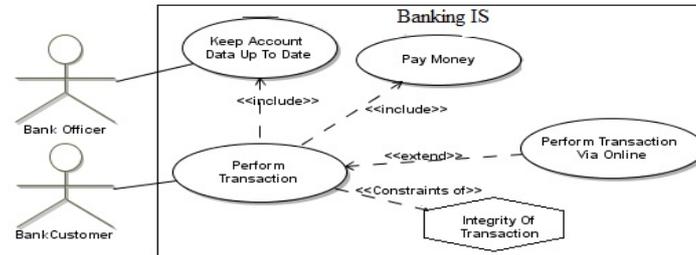


Fig. 2. Asset Modelling

**Asset model.** In Fig. 2, we illustrate the context of an online banking IS in a use case diagram. A *security criterion* is a security constraint imposed on *business use case* (i.e., business asset). The example focusses on the bank customer and bank officer who both communicate with Banking IS. The Bank Customer and Bank Officer are the assets characterising the users of the system in reference to ISSRM domain model. The bank customer seeks to Perform Transaction and bank officer seeks to Keep Account Data Up To Date. The Perform Transaction *includes* two use cases Pay Money and Keep Account Data Up To Date and *extends* Perform Transaction Via Online. Perform Transaction has a *security criterion* Integrity of Transaction represented as a *hexagon* (see Fig. 2) as it characterises a security constraint of a *business use case* (i.e., Perform Transaction). In Fig. 2, a dotted line with stereotype type

<sup>1</sup> To create these models we use the Microsoft Visio tool.

*constraints of* is linked from *business use case* (i.e., Perform Transaction) to *security criterion* (i.e., Integrity of Transaction) shows the relationship between the two. According to ISSRM domain model we identified Perform Transaction as the business asset that has some business value. Hence Perform Transaction Via Online supports the business asset and is considered as an IS asset.

**Risk model.** In Fig. 3, we model the potential security threat scenario. A *misuser* (i.e., Attacker) initiates a *misuse case* (i.e., Intercept Money *includes* Transfer money to another account and Change details of transaction) by exploiting the *vulnerability* (i.e., Unsecure Network Channel) in a *use case* (i.e., IS asset). Following [10] in Fig. 3, this *vulnerability* is represented by filled grey use case. The *misuse case* Intercept Payment *threatens* the *use case* Perform Transaction Via Online (i.e., IS Asset). The threat Intercept Money *leads to* an *impact* (i.e., Money Transferred to Unintended Account) which *harms* the *business use case* (i.e., Perform Transaction) and disaffirms the *security criterion* (i.e., Integrity of Transaction). An *impact* is a state of system that is represented as *rounded rectangle* (see Fig. 3). A *misuse case* is linked to impact using *leads to* relationship. On one hand, an *impact* disaffirms the *security criterion* linked with *negates* relationship. On another hand *impact* *harms* a *business use case* (i.e., Perform Transaction).

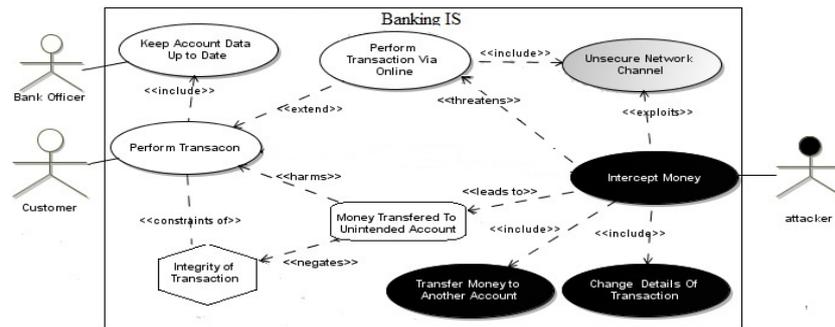


Fig. 3. Threat Modelling

**Risk treatment model.** The ISSRM domain model defines the risk treatment, control and its implementation. However, SROMUC does not support the modelling of these concepts but security requirement is modelled as a *security use case*. The *security use case* is represented as a *use case with a lock inside* (see Fig. 4). In Fig. 4, we present the security requirement for identified threats. The *use case* Perform Transaction Via Online (i.e., IS Asset) *includes* a *security use cases* (i.e., Apply Cryptographic Procedures and Use Secure Communication Protocol). The *security use case* *mitigates* the *misuse case* (i.e., Intercept Money). It ensures *security criterion* (i.e., Integrity of Payment) imposed by *business use case* (i.e., Perform Transaction).

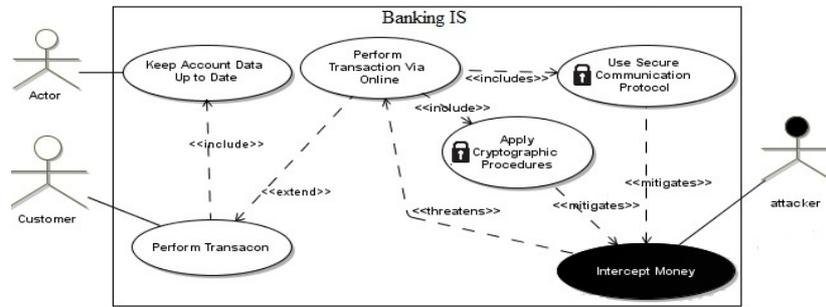


Fig. 4. Security Requirement Modelling

### 3.3 Scenario 2: SROMUC Modelling for Availability

In Fig. 5, we model an online banking IS [1, 8] for Availability of Service. In our example, the *business use case* (i.e., Perform Transaction) has a constraint of *security criterion* (i.e., Availability Of Online Service). The *misuser* (i.e., Attacker) initiates a *misuse case* (i.e., Make Online Service Unavailable includes Initiate Half Opened Connections To Server). It exploits the *vulnerability* (i.e., Allow Unlimited Number Of Connections) included in a *use case* Perform Transaction Via Online (i.e., IS Asset). The *misuse case* Make Online Service Unavailable threatens use case Perform Transaction Via Online (i.e., IS asset) and leads to an *impact* (i.e., Availability Of Service Is Compromised), moreover, it harms the business use case Perform Transaction. The *impact* of the *misuse case* negates the *security criterion*.

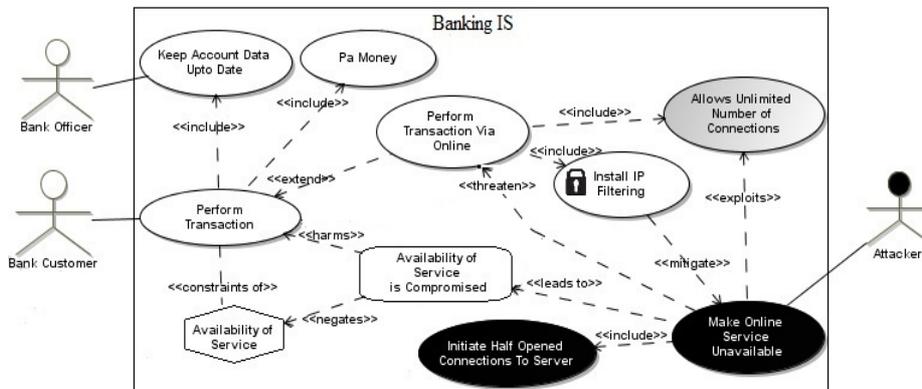


Fig. 5. Modelling for Availability of Service

### 3.4 Scenario 3: SROMUC Modelling for Confidentiality

In Fig. 6, we model the example of an online banking IS [1, 8] for the Confidentiality Of Data. In this example, the *business use case* (i.e., Perform Transaction) has a constraint of *security criterion* (i.e., Confidentiality Of Transaction). The *use case* Perform Transaction Via Online (i.e., IS asset) *includes* another *use case* (i.e., Ensure Account privacy *includes* Enter PIN Code) for securing an online transaction. The *misuser* (i.e., Attacker) initiates a *misuse case* (i.e., Steal Account Data *includes* Retrieve Transaction Data *includes* Disclose Transaction Data) by exploiting the *vulnerability* (i.e., Data Is Not Encrypted and Accept Malicious Data). The *misuse case* (i.e., Steal Account Data) *threatens* the *use case* Perform Transaction Via Online (i.e., IS asset) and *leads to an impact* (i.e., Confidentiality Of Data Is Compromised), moreover, It also *harms* the *business use case* (i.e., Perform Transaction). The *impact* of the *misuse case* *negates* the *security criterion*.

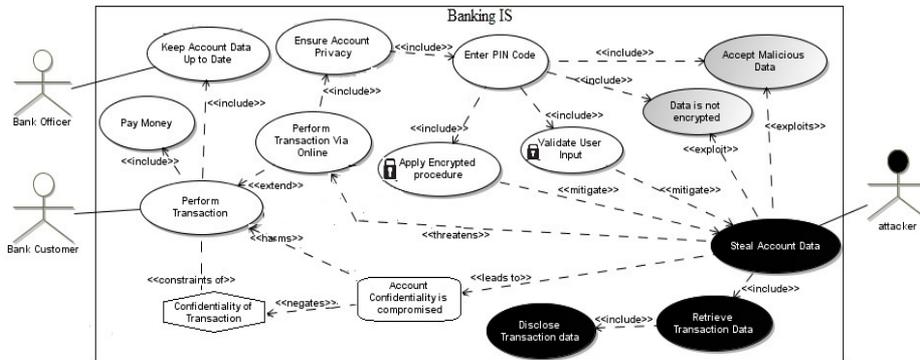


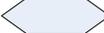
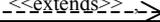
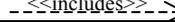
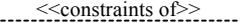
Fig. 6. Modelling for Confidentiality of Data

### 3.5 Concept Alignment of SROMUC and ISSRM

In [9] authors discuss the alignment between the misuse cases and the ISSRM domain model. However it presents only the correspondences, overlaps or/and similarities. In this section we describe the alignment of SROMUC with the concepts found in ISSRM domain model. In Table 1, 2 and 3, first column outlines the ISSRM concepts. The second column expresses their synonyms found in the literature. The third column distinguishes the concepts and relationship. The last column defines the SROMUC visual constructs.

**Alignment of asset-related concepts.** In Table 1, we introduce SROMUC syntax to represent the ISSRM asset-related concepts. In ISSRM domain model, assets correspond to *Actor* and *Use case* in SROMUC. The business asset and the IS asset are modelled as a *use case*. The *supports relationship* in ISSRM between IS asset and business assets is expressed using *extends* and *includes relationships*. We introduce *hexagon* construct in SROMUC to represent the ISSRM *security criterion*. A *security criterion* is the constraint on business asset therefore the *hexagon* is linked to *business use case* through dotted line with *constraint of relationship*.

**Table 1.** Asset Related Concepts (C – Concept, R – Relationships)

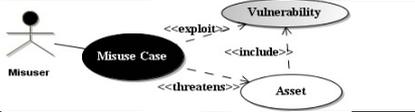
ISSRM Concepts	Synonyms	Type	SROMUC Syntax
Assets		C	 actor
Business Asset	Business Use Case	C	
IS Asset	IS Use Case	C	
Security Criterion	Security Constraint	C	
Supports	-	R	 
Constraints of	Restriction	R	

**Alignment of risk-related concepts.** In Table 2, we introduce the SROMUC syntax to represent the ISSRM risk-related concepts. In SROMUC, a *threat agent* is represented as *misuser*, *attack method* as *misuse case* and *vulnerability* as a *use case* filled in grey. A *threat* is modelled as a combination of *misuser* and *misuse case* (i.e., misuser communicates with misuse case). The ISSRM *targets relationship* is represented as an SROMUC *threatens relationship*. We introduced a *rounded rectangle* to model the *impact* concept of ISSRM.

In order to be compliant with ISSRM domain model, we also introduce the *exploits*, *leads to*, *harms* and *negates* relationships. *Exploits* relationship defines a link between *misuse case* and the *vulnerability* whereas the *leads to* relationship defines a link between the *misuse case* and the *impact*. The *harms* relationship defines the link between an *impact* and a *business use case* whereas a *negates* relationship defines a link between an *impact* and the *security criterion* (see Table 2). We combine the concepts of *threat agent*, *attack method*, *vulnerability*, and *impact* all together to represent an *event*, where a *risk* is understood as a combination of *event* and the *impact*.

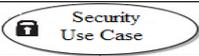
**Alignment of risk treatment-related concepts.** In risk treatment-related concepts, we update the visual syntax of *security use case* by adding a padlock to *security use case*, which represents *security requirement* (see Table 3). The ISSRM *mitigates relationship* is modelled with *mitigates relationship* from *security use cases* (i.e., security requirement) to *misuse case* in SROMUC.

**Table 2.** Alignment of Risk related Concepts(C – Concepts, R – Relationships)

ISSRM Concepts	Synonyms	Type	SROMUC Syntax
Risk	Hazard	C	
Impact	Effect	C	
Event	Incident	C	
Attack Method	Violence	C	

Vulnerability	Weakness	C	
Threat Agent	Attacker	C	
Threat	Hazard	C	
Exploits	-	R	--><<exploits>>
Negates	Denies,	R	--><<negates>>
Harms	-	R	--><<harms>>
Leads to	-	R	--><<leads to>>
Characteristics of	-	R	--><<includes>>      --><<extends>>
Uses	-	R	_____

**Table 3.** Risk Treatment related Concepts (C – Concepts, R – Relationships)

ISSRM Concepts	Synonyms	Type	SROMUC Syntax
Risk Treatment		C	
Security Requirement	Countermeasure	C	
Control		C	-
Refines		R	-
Mitigates	Diminishes	R	--><<mitigates>>
Implements			-

### 3.6 Abstract Syntax of Security Risk-oriented Misuse Cases

In Section 3.1, we presented the SROMUC before abstract syntax due to the simple introduction of the language. However, to illustrate the application of proposed SROMUC, we need to introduce its abstract syntax in Fig. 7. The major elements in the meta-model are an *Actor OR Misuser* and *Use OR Misuse Case*. *Actor OR Misuser* initiates the *communication* to interact with *Use OR Misuse Case*. Their cardinality shows that an *Actor or Misuser* can communicate with one or more *Use or Misuse Case*. *Actor* and *misuser* are the specialisations of an *Actor OR Misuser*. *Use OR Misuse case* can *includes* or *extends* another *Use OR Misuse Case*. The *Use Case*, *Vulnerability* and *Misuse Case* are the specialization of *Use OR Misuse Case*. The *Use Case* includes one or more Vulnerabilities that can be exploited by one or more misuse cases. A *Misuse Case* threatens (i.e., *threatening*) one or more use cases. A *Misuse Case Leads To* one or more *Impact*. An *Impact Harms* one or more use cases (see Fig. 3) by negating one or more *Security Criterion* define as *Constraint Of* on that *use case*. A *Security Use Case* is a specialised *Use Case* that *Mitigates* one or more *Misuse Cases*.



It performs Business-IT-alignment and prioritises the IT requirement. Similarly, ISSRM align these concepts by supporting the definition of security for the key IS constituents and addresses the IS security risk management process at three different conceptual levels (see Section 2.1).

**Misuse cases.** There have been few studies carried out on misuse cases and its extension. In [13, 14] McDermott and Fox have proposed abuse cases to explore how threats and countermeasures could be modelled using standard UML use case but keeping abuse cases in a separate model. Abuse case focusses on security requirements whereas our approach is aligned with ISSRM and focusses on the overall security risk management. It identifies vulnerabilities and threats, and analyses potential risks and their impacts. Therefore, the elicited security requirements are aligned with the functional system requirements. In [2] Alexander has considered how security use cases can be threatened by misuse cases. Matulevičius *et al.* [9] have aligned misuse cases with ISSRM however they leave the misuse case extensions for the future development. In this paper the extensions of the misuse cases are built on the previous work of Matulevičius *et al.* [9] and covers the complete security risk management strategy of an organisation at the early development stage.

## 4.2 Discussion

SROMUC is an approach to elicit security requirements at the early stages of the system development. It will potentially help designers, architects and analysts to understand the potential threats and security attacks. At both the architecture and design stages, risk analysis is a necessity. The SROMUC approach enables the security analysts to discover the architectural flaws so that their mitigation could begin early in the system development. Otherwise disregarding the risk analysis at this level leads to costly problems later. In practice, system stakeholders are not motivated to invest on security concerns, as it does not add direct value to the systems' functionality. The proposed SROMUC strengthens the misuse case diagrams by extending their syntax and semantics. The proposed graphical extensions are not intuitive and they related to the security concerns supported by the ISSRM domain model. However the primary idea is to keep it comprehensible and to compliant with the original definition of (mis)use cases. We differentiate the construct for impact and security criterion from the standard UML use case constructs. The security use case construct has been enhanced to differentiate security requirements from the functional requirements. In [9] Matulevičius *et al.* have suggested to differentiate the concepts of the IS asset and the business asset. But here, we did not differentiate the assets as it changes the definition of original use case construct. We make an exception regarding the security use because it addresses the system functionality in terms of security countermeasures. Regarding the completeness of alignment between SROMUC and ISSRM domain model, SROMUC does not address the risk treatment and control implementation.

SROMUC is not the only approach that has been aligned to ISSRM domain model. Currently ISSRM is becoming a common model [11] to understand security risk modelling using different modelling languages, like BPMN [3], Secure Tropos [10],

KAOS extensions to security [11], and Mal-activities [4]. Finally, this may lead to interoperability between different security languages.

Although in the online banking example we have illustrated the applicability and performance of our proposal, we acknowledge the importance of the industrial case study to validate the SROMUC in the practice. As a future work, we also plan to experiment the language in a case study to validate its usefulness and effectiveness.

## References

1. Ahmed, N., Matulevičius, R., Mouratidis, H.: A Model Transformation from Misuse Cases to Secure Tropos. In: Proc of the CAiSE'12 Forum at the 24th International Conference (CAiSE). pp. 7–14. CEUR-WS (2012)
2. Alexander, I.: Misuse cases: Use cases with Hostile Intent. *IEEE Soft.* 20(1), 58–66 (2003)
3. Althohova, O., Matulevičius, R., Ahmed, N.: Towards Definition of Secure Business Processes. In: CAiSE Workshops. vol. 112, pp. 1–15. Springer (2012)
4. Chowdhury, M., Matulevičius, R., Sindre, G., Karpati, P.: Aligning Mal-activity Diagrams and Security Risk Management for Security Requirements Definitions. In: REFSQ, vol. 7195, pp. 132–139. Springer Berlin / Heidelberg (2012)
5. Ekelhart, A., Fenz, S., Neubauer, T.: AURUM: A Framework for Information Security Risk Management. In: HICSS '09. pp. 1–10. IEEE Computer Society (2009)
6. Firesmith, D.: Security Use Cases. *Journal of Object Technology* 2(3), 53–64 (2003)
7. Herrmann, A., Morali, A., Etalle, S., Wieringa, R.J.: RiskREP: Risk-based Security Requirements Elicitation and Prioritization. In: Perspectives in Business Informatics Research, Riga, Latvia. pp. 155–162. Riga Technical University (2011)
8. van Lamsweerde, A.: Elaborating Security Requirements by Construction of Intentional Anti-Models. In: Proceedings of the 26th International Conference on Software Engineering. pp. 148–157. ICSE '04, IEEE Computer Society (2004)
9. Matulevičius, R., Mayer, N., Heymans, P.: Alignment of Misuse Cases with Security Risk Management. In: Proceedings of 3rd International Conf. on Availability, Reliability and Security. pp. 1397–1404. IEEE Computer Society (2008)
10. Matulevičius, R., Mouratidis, H., Mayer, N., Dubois, E., Heymans, P.: Syntactic and Semantic Extensions to Secure Tropos to Support Security Risk Management. *J. UCS* 18(6), 816–844 (2012)
11. Mayer, N.: Model-based Management of Information System Security Risk. Ph.D. thesis, University of Namur (2009)
12. Mayer, N., Heymans, P., Matulevičius, R.: Design of a Modelling Language for Information System Security Risk Management. In: Proceedings of the First International Conference on Research Challenges in Information Science, RCIS 2007. pp. 121–132 (2007)
13. McDermott, J.: Abuse-Case-Based Assurance Arguments. In: Proc. of the 17th Annual Comp. Security Applications Conf. pp. 366–. ACSAC '01, IEEE Computer Society (2001)
14. McDermott, J., Fox, C.: Using Abuse Case Models for Security Requirements Analysis. In: Proceedings of ACSAC'99. pp. 55–, IEEE Computer Society (1999)
15. Pauli, J.J., Xu, D.: Trade-off Analysis of Misuse Case-based Secure Software Architectures: A Case Study. In: Proc. of MSVVEIS Workshop. pp. 89–95. INSTICC Press (2005)
16. Røstad, L.: An Extended Misuse Case Notation: Including Vulnerabilities and The Insider Threat. In: Proc. 12th Working Conf. REFSQ'06 (2006)
17. Sindre, G., Opdahl, A. L.: Templates for Misuse Case Description. In: Proc. of the 7th International Workshop on REFSQ'01 (2001)
18. Sindre, G., Opdahl, A. L.: Eliciting Security Requirements with Misuse Cases. *Requir. Eng.* 10(1), 34–44 (2005)