UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Institute of Computer Science
Software Engineering

Anastasiia Onchukova

Security Risk Management using Misuse cases and Mal-activities

Master's thesis (30 ECTS)

Supervisor: prof. Raimundas Matulevičius

Author: ……"……"August 2013
Supervisor: ……..…"……"August 2013
Approved for defense
Professor:……....……"……"August 2013

TARTU 2013

Abstract

Security concerns during development of secure information systems (IS) can be addressed at different development stages (e.g. requirements engineering, system and software design, implementation, and other). Security analysis could be performed using different dedicated modeling languages (e.g. Secure Tropos, misuse cases, mal-activity diagrams), what allows developers to express important concerns from the different perspectives. Although each language has its own strengths, combining different perspectives into the coherent and consistent model still remain a challenging activity.

In the thesis we focus on the two modeling languages, called misuse cases (both diagrams and textual templates) and mal-activity diagrams. Although being different in their development perspective, they both could potentially be used at the system and software design stages to elicit, analyze, and document security requirements. In the previous research both these languages were analyzed with respect to the domain model of the information systems security risk management (ISSRM), which determines a systematic process to identify important and valuable assets, determine their security risks and introduce security requirements to mitigate these risks. However in the previous work only the misuse case diagrams (not textual template, nor mal-activities) were completely extended with respect to the ISSRM domain model.

The goal of this work is to define a thorough method, which would facilitate a transformation of the misuse cases to the mal-activities. Based on the aforementioned research in this thesis we extend the misuse case textual template and the mal-activity diagrams, so that they would cover concepts of the ISSRM domain model. Next based on the extensions we introduce set of transformation rules that guides translation of the misuse cases diagrams with the help of the misuse case templates to the mal-activity diagrams.

We validate our proposal in a case study on the analysis of the model quality. We hope that our contribution will help the system and software developers to integrate two modeling viewpoints in order to capture security requirements and systematically to develop and incorporate them into the system design, thus contributing to the secure IS.

Table of Contents

List of Tables

List of Figures

Abbreviations

ISSRM	Information System Security Risk Management
MUC	Misuse Cases
SROMUC	Security Risk-Oriented Misuse Cases
MAD	Mal-Activity Diagrams
BPMN	Business Process Management Notation
UML	Unified Modeling Language
EPC	Event-Driven Process Chain
KAOS	Knowledge Acquisition in automated Specification
KeS	KAOS extension to Security
RBAC	Role-Based Access Control
IS	Information System
ISO/IEC	International Organization for Standardization/International Electro-technical Commission
SQUARE	Security Quality Requirements Engineering
EBIOS	*French*: Expression des Besoins et Identification des Objectifs de Sécurité
AUML	Agent Unified Modeling Language
CNN	Cable News Network
OIS	*Estonian*: Oppeinfosüsteem

1 Introduction

It is crucial to understand security aspects in the process of the Information System (IS) development, since risks affect system components, what consequently leads to big losses of resources (e.g. time, money). Security concerns should be addressed at the early phases of the system development. At later stages it is difficult to deal with them, because changing already implemented system and adding new functionalities requires a lot of time and money. For the moment there is no such modeling language, which will allow analyzing security concerns from several viewpoints during requirements engineering. Thus it is needed to find thorough method for that. Security is understood as a capability of a product to protect information from unauthorized access and destroying or modification of data (Dubois *et al.*, 2010). In this thesis we are going to introduce method, which will help developers to understand possible risks of the system from different viewpoints and to model decisions how to deal with them at the early stages of the IS development.

There is number of existing languages, which aim to model security concerns at the early stages of the IS development. Languages, used the most by developers, are KAOS extension to security (KeS) (Lamsweerde, 2004), misuse cases (MUC) (Sindre & Opdahl, 2005), mal-activity diagrams (MAD) (Sindre, 2007) and Secure Tropos (Mouratidis & Giorgini, 2007). But none of them allows understanding security problems from different points of view (e.g. activity flow, interaction of system components). One of the possible solutions could be combining different languages in the approach, which will allow considering security problems from several points of view and systematically analyze security concerns. In this work we have chosen 2 languages for that: misuse cases and mal-activity diagrams. In order to combine these languages, they should be analyzed with regard to some base. As such base in this work ISSRM domain model is taken.

We investigate how misuse cases can be used in combination with mal-activity diagrams in order to describe risks and their mitigation when developing information systems. To allow this, we are going to revise both languages, add elements to these languages if needed, and elicit transformation rules from one language to another. This will allow understanding security problems from different prospectives and to model effective solutions for risk mitigation. The research questions are:

1. How could misuse case template be extended with respect to the ISSRM domain model?
2. How could graphical representation of the mal-activity diagrams be extended to cover the ISSRM domain model?
3. How misuse case diagrams can be transformed to mal-activity diagrams?

Misuse case diagrams were already extended in the previous research (Soomro & Ahmed, 2012) and thus misuse cases can cover ISSRM domain model. In order to perform transformation, mal-activity diagrams should be extended with respect to the ISSRM domain model as well. After that we elicit step by step set of transformation rules from misuse case diagrams and misuse case templates to mal-activity diagrams.

The thesis consists of 8 chapters. Chapters 2 and 3 provide background information for understanding current status of the research, previously received results and achievements related to this topic. Chapter 2 describes, what is Information System Security Risk Management and chapter 3 gives introduction to the currently existing modeling languages for presenting security concerns. Chapters 4, 5 and 6 are actual contribution of the work. In chapter 4 analysis of misuse case extensive textual template in relation to the ISSRM do-

main model is presented, and suggested extensions to the template. In the chapter 5 we suggest possible extensions to the mal-activity diagrams, with regard to the ISSRM domain model. Chapter 6 shows elicitation of the transformation rules from the misuse cases to mal-activity diagrams, with help of textual templates. In the chapter 7 we validate contribution of the thesis. Chapter 8 contains overview of the results and suggestions for future work. In the Appendices we show explicit example of the Registration to the courses process, presenting textual description of the example, tabular representation, misuse case diagrams and textual templates. Moreover in the Appendices presented data, received during validation.

2 Information System Security Risk Management (ISSRM)

In our work ISSRM domain model will be used as base for analysis and comparison of the languages. Previously security risk-oriented languages (e.g. Secure Tropos, BPMN) were aligned and extended regarding this model. We present ISSRM domain and provide explanation of its main constructs. Also we describe 6-step ISSRM process, following which assets, risks and risk treatment decisions for the system can be identified.

2.1 ISSRM Domain Model

ISSRM domain model (Dubois *et al*, 2010) was created by influence of different security standards (e.g. ISO/IEC, 2002) and methods (e.g. Braber *et al*., 2006). It combines 4 concepts: (1) Security Risk Management Standards, (2) Security-related Standards, (3) Security Risk Management Methods, and (4) Security-oriented Framework. ISSRM domain model (Figure 1) supports definition of security for the main parts of information systems and addresses the IS security risk management process at three different conceptual levels:

- asset-related;
- risk-related;
- risk treatment-related.

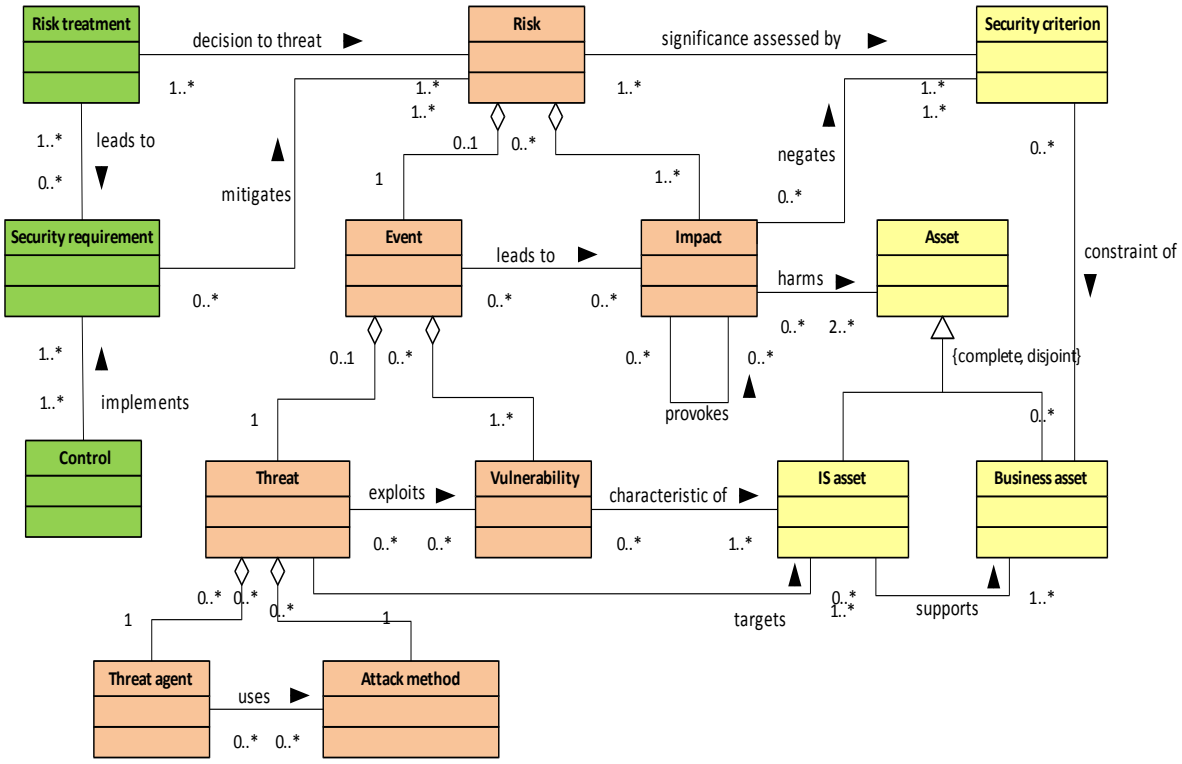


Figure 1: ISSRM domain model (Mayer, 2009)

Asset-related concepts define what are the assets of the system and which of them are the most important ones. *Asset* is anything, that has value for organization and is important in achieving her goals. Two types of assets are differentiated: business assets and IS

assets. *Business assets* are important for achieving objectives of organization, it can be information, people; business assets are always immaterial. *IS assets* are material, can be software, hardware etc. IS assets support business assets in achieving goals of organization. *Security criterion (*e.g. confidentiality, availability, integrity) characterizes the security needs of the business assets. *Confidentiality* is the property that data is not available or disclosed to unauthorized persons or processes. *Availability* is the property, that assets are accurate and complete. *Integrity* is the property, that data can be accessed and used anytime by authorized persons. Definitions of confidentiality, availability and integrity are taken from Dubois *et al.* (2010).

Risk-related concepts present risk itself and its components. *Risk* consists of threat and one or more vulnerabilities, which leads to a negative impact. *Impact* harms business and IS assets and negates security criterion. *Vulnerability* is expressed as the weakness of any flaws of the IS asset or group of IS assets. *Event* includes threat and one or more vulnerabilities. A *threat* is a potential attack, which targets IS assets. An *attack method* characterizes means, using which, threat agent performs threat.

Risk treatment-related concepts present decisions and controls to prevent possible risks. *Risk treatment* is the decision (e.g. avoidance, reduction, retention, transfer) to avoid, minimize or ignore identified risk. *Security requirement* is the improvement of the risk treatment decision to mitigate the risk. *Control* provides means to improve security, defined by implementing the security requirement.

2.2 ISSRM Process

In order to define ISSRM domain model constructs, one should follow ISSRM process. It consists of sixth steps and can be found in any of traditional ISSRM methods (e.g. CORAS (Braber *et al.,* 2006), SQUARE (Mead *et al.,* 2005), and EBIOS (DCSSI, 2010)). Process is illustrated in Figure 2 and described below.

Process steps:

1. *Context and assets identification*. Process starts with learning activities of organization and its environment. Based on the information collected while studying domain, business and IS assets are defined.
2. *Determination of security objectives.* Security needs of the organization, which should be reached, are clarified, based on identified assets. Security objective is considered usually in terms of confidentiality, integrity and availability properties of the assets.
3. *Risk analysis and assessment.* It is the main step of the process, because risks, harming system assets and threatening security objectives, are identified at this step. After that, one should estimate identified risks, quantitatively or qualitatively.
4. *Risk treatment.* Risk treatment decisions are taken after risk assessment. Risk treatment can be of four types:
   - risk avoidance- decision to avoid risk, not to be involved in;
   - risk reduction-lessen probability and negative consequences of the risk;
   - risk transfer- share risk results with another part;
   - risk retention- accept burden of risk.
5. *Security requirements definition.* On this step solutions to mitigate risk are defined, mainly if risk reduction was chosen on the previous step. But security requirements can also emerge from other risk treatment decisions. If, at the end of this step, security requirements considered as unsatisfactory, the risk treatment step should be revised, or even all previous steps should be repeated in order to achieve satisfactory level of risk.

6. *Control selection and implementation.* It is the process of implementation and deciding which countermeasures should be used.

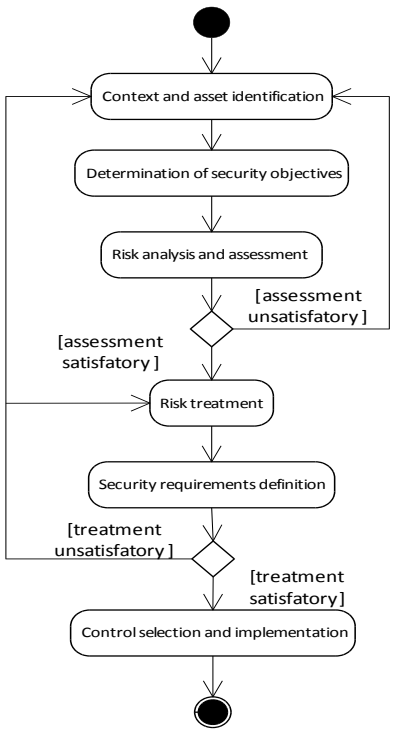


Figure 2: ISSRM process (Mayer, 2009)

2.3 Summary

In this chapter we presented overview of the main ISSRM domain model elements. Also 6-step ISSRM process, which is used for risks elicitation and deciding on risk treatment, is introduced. In the next chapters, ISSRM domain model will be used as a base for languages comparison and ISSRM process - in application for modeling languages in order to perform analysis.

As a conclusion can be said, that ISSRM process is iterative and it should be done until acceptable level for all risks is not reached. Otherwise implementation of the information system can be canceled, in case that dealing with a risk requires a lot of costs, which will be not paid back, or it requires other unaffordable for organization resources. Acceptable level of risk is level when risk will not cause harmful damage for organization or consequences of it can be corrected using minimal resources.

3 Security Risk-Oriented Modeling Languages

In this chapter we present overview of currently existing security risk-oriented modeling languages and extensions to them. Moreover here we provide detailed introduction of languages, which will be considered in the thesis: misuse cases (MUC) and mal-activity diagrams (MAD). Overview is needed in order to understand current situation of existing problems and limitations in the secure software development and to define scope of the thesis.

3.1 Overview of different modeling languages

During IS development, security concerns can be addressed on different software development stages (e.g. requirements engineering, design stage etc.), using one of the many existing security modeling languages (e.g. Secure Tropos, BPMN etc.). Nevertheless, none of the existing languages allows presenting security concerns from different perspectives. Meaning that, if language shows sequence of activities, it doesn't explicitly show how users interact with the system. In this work we are going to provide a solution, which will allow analyzing security issues from several points of view. This solution will be combining 2 languages. In order to choose these languages, we need to make an overview and comparison, which will be presented below.

Nowadays there exists a lot of software development models (e.g. spiral model, iterative development, and agile development). However in research papers, which we use as a base for our work, modeling languages are discussed and analyzed (Sindre & Opdahl, 2005; Mouratidis & Giorgini, 2007; White, 2004) regarding waterfall model (Royce, 1987). Thus in the thesis we present comparison and introduction of approaches based on the waterfall model. The main stages on which can be decomposed development process, according to the waterfall model (Royce, 1987), are:

- Requirements analysis.
- Software design.
- Implementation.
- Testing.
- Installation.
- Maintenance**.**

Secure Tropos. In (Matulevičius *et al.*, 2008A) extension to Secure Tropos methodology (Mouratidis & Giorgini, 2007) are given in order to address security concerns throughout whole development process. Secure Tropos models security using *security constraints* and *attack methods;* initially it contained such elements, as *actor, goal, soft goal, plan, resource,* and *belief.* These elements were complemented with *security constraint* and *threat* constructs to enable modeling of secure entries. *Actor* is an object, which has *goals* and *interests* related to the system. A *goal* is actor's strategic interest. A *plan* is way to satisfy actor's goals. *Resource* – what is required by actor. *Security constraint* is a limitation, which the system must possess. The *threat* is an event, which endangers assets of the system. Above mentioned elements are used to produce actor and goal diagrams (Giorgini *et al.*, 2007). This is an iterative process: diagrams, which are produced at the one phase, are the basis for modeling next diagrams. Secure Tropos is intended for usage at the requirements engineering and design stages of the information system development. Order of activities at the each stage is following.

Requirements analysis stage. On this stage 'system-to-be' is modeled. Process starts with defining actors and list of goals for each actor, together with definition dependencies between actors. The dependency modeling leads us to model dependencies between actor and the system. These allow eliciting system requirements. Then modeling of trust, delegations and security constraints drives to the revised version of actor diagrams. This is an iterative process, which can require several revisions.

Design stage. Modeling at this starts with the overall system architecture style selection, using as criteria security requirements of the system. After that system is decomposed in sub components, delegating goals, and responsibilities with respect to the selected architecture and dependency model from the previous stage. Next each sub component is analyzed with respect to its goals and plans.

Knowledge Acquisition in automated Specification extension to Security. This is another goal-modeling language, which was adopted for security (Lamsweerde, 2004). KAOS goal-oriented framework (Dardene *et al*., 1993) addresses security concerns by treating attacks as anti-goals. The KeS extends KAOS framework in such ways:

- extends specification language;
- provides patterns for elicitation of security requirements;
- introduces principle for modeling of threats, system goals, requirements, system vulnerabilities and attacker capabilities.

The KeS is aimed for use at the requirements engineering phase. The main objective of KeS is defined by *goal*. The *goal* can be decomposed into several sub goals. These sub goals are again reviewed till requirements are defined. The detail procedure to meet the goal is defined by an operational model. Risks to the system are represented by anti-goal model. And the countermeasure of the anti-goal model is defined by a new security requirement. The action of the model is defined by using two constructs *achieve* and *avoid*. *Achieve* means the system must have to achieve this goal and *avoid* means system must have to resist it from this goal.

BPMN (White, 2004). Basic flow of BPMN consists of starting event, business decision, gateways, work flow outputs and ending event. BPMN doesn't explicitly consider mechanisms to support security concerns. But it was aligned with security notations (Altuhova *et al*, 2012). It allows modeling system security from the early stages of development. The application of BPMN is divided into three model usage levels: analytical modeling describes activity flow, executable modeling is targeted to system development and descriptive modeling concentrates on business processes by documenting major business flows.

UMLsec (Jürjens, 2002). Developers are used for UML notation, that is why usual UML notation was decided to extend in order to be able to address security concerns during design phase. UMLsec considers security during system design, it allows adding security-related information to UML diagrams, specify requirements on confidentiality and integrity in analysis models, but it doesn't focus on business assets and high-level security requirements. UMLsec main elements are *stereotypes*, *base class*, *tag* and *constraints*. *Stereotypes* define new types of modeling elements by extending the semantics of existing types or classes in the UML meta-model. They can be identified by double angle brackets << >> with name of stereotype in it, which is attached to the extended model element. This model element is then interpreted according to the meaning of the stereotype. One way of explicitly defining a property of the model element is by attaching a *tagged value* to it. Another way of adding information to a model element is by attaching constraints to refine its semantics.

SecureUML (Lodderstedt *et al.*, 2002). SecureUML, the same as UMLsec, is extended from UML diagrams and intended to consider security on the system design stage. It defines a vocabulary for annotating UML based models with information relevant to access control. SecureUML defines a vocabulary for expressing different aspects of access control, like role, permission and user-role assignment. Authorization constraint defines the precondition for granting access to any operation. Such constraints are expressed using Object Constraint Language (OCL) Authorization constraint gives SecureUML the flexibility to define and verify the access decision on dynamically changing data.

Extension of Problem Frames. Problem Frames were developed by Jackson in (2001). Haterbur and Heisel (2005) developed extensions to Problem Frames, called Security Frames, which allowed developers to address security concerns during phase of requirements elicitation. These security frames are related to abuse frames on one hand and to security patterns on the other hand. Secure Frames are of four kinds: two of the security frames concern authentication, third deals with secure transmission of data, and fourth is suitable for generating and storing security information. These frames are described by frame diagrams, which consist of rectangles and links between them. Main elements are *rectangles with double vertical line*, which represent system to be developed, *dashed oval*, denote requirements, *connecting lines*, denote interface that consist of shared *phenomena*, *dashed line* states for requirements reference, and the *arrow* shows that it is a constraining reference. Security frames, are patterns for software development, occurring when security-critical software has to be developed. Also they help to decompose complex security problems to simpler ones. This language helps to cover large parts the development of the security-critical systems, using pattern-based approach.

Abuse Frames (Lin *et al.*, 2004). Abuse frames designed to consider security concerns on the early phase of requirements elicitation; it focuses on finding new threats and vulnerabilities that could be exploited by malicious actors. An attack is defined as a realization of threat. And threat is a potential for use of domains in the system to cause harm. There are so-called anti-requirements, which reproduce requirements of malicious intent. In abuse frames anti-requirements represent security threat. Abuse frames also describe classes of security violation and include: interception, modification and denial of access. Each of them represents a threat that can violate a particular security goal. Interception arises whenever there is some information asset in the physical world that an attacker wishes to obtain, thus violating confidentiality; modification, when attacker wishes to change something- violates integrity; and denial of access to make some information asset unavailable or unusable, and violates availability. The abuse problem is to find a malicious machine that allows the attacker to achieve this. Abuse frames provide means to early structure and bound scope of the security problems, when requirements of future system are on the stage of elicitation. Abuse frames is not a substitute for other security engineering techniques, but it can discover some security vulnerabilities and requirements, so it's useful when defining requirements.

Misuse cases (Sindre & Opdahl, 2005). Misuse cases are a security risk-oriented extension of UML use case diagrams, which support elicitation of functional requirements. Misuse cases have two variants of representation: graphical and textual. Textual description is needed for explicit specification of misuse cases. Firstly, misuse cases were suggested by Sindre and Opdahl (2005). Usual use cases needed some functionality for proper security requirements elicitation was missing. Usual use cases were extended with negative use cases-*misuse cases*- which mean behavior not wanted in the system. Moreover usual use case notation was extended with *misuser, security use cases* constructs, which include *threatens*

and *mitigates* relationships, which allowed addressing security during early requirements stage.

Misuser is an actor, who has intention to harm the system. *Misuse case* is the goal of the *misuser* and has communication association to use case and actor *misuser*, which causes *harm* to the system if actions are performed successfully. *Misuser* executes *misuse case* either by combining efforts of several misuse cases or independently. *Threatens* and *mitigates* are relationships between *use cases* and *misuse cases*. *Threatens* relationship can be used when a misuse case is potentially a *threat* to the use cases and can harm them. *Mitigates* relationship show that a *use case* is a countermeasure against *misuse case* and mitigates it. *Security use case* performs countermeasure against identified *threat*. Thus, misuse cases are integrated in the use case diagrams and show the system unwanted behavior initiated in use case diagrams by *misuser*, who *threatens* the legitimate behavior of the software system.

Mal-activity diagrams (Sindre, 2007). Mal-activity diagrams extend concepts of the UML activity diagrams; they deal with behavioral aspects of security problems. The main idea is to build an activity diagram and then add mal-activity, unwanted behavior, so concept is similar with misuse case diagrams. Mal-activity diagrams add some extra concepts such as *mal-activity, mal-swimlane* and *mal-decision*. These concepts are just opposite of the normal activity diagrams constructs. It also defines *mitigation activity* and *mitigation link* to show the mitigation process.

Summary. We provided a short description of existing languages for security modeling, showing that each of the approaches is intended for particular phase of the development from the waterfall model (Royce, 1987). In the Table 1 we present summarized comparison for visual understandability. As was said before we want to find method, which will allow addressing security concerns at the requirements engineering phase. Languages, which are intended to be used at the requirements elicitation phase are: KeS, Extension to Problem Frames, misuse cases and mal-activity diagrams. Only 3 of them were previously aligned to the ISSRM, which are: KeS, misuse cases, mal-activity diagrams. But KeS, as can be seen from the last column, was not extended regarding ISSRM domain model. Thus we have 2 languages, which intended for requirements elicitation phase, were aligned to the ISSRM domain model and were partially extended according to ISSRM. Misuse cases and mal-activity diagrams model security from different points of view, mal-activity diagrams are dealing with activity flow of the system and misuse cases is showing interaction between different components of the system. For example, misuse cases can't show at what point misuser is entering system and mal-activity diagrams can't represent full set of security notations on the diagrams. However since currently they are intended for different aspects, developers obliged to use both of them to get understanding of system threats and to model secure IS. Thus these languages can complement each other on the requirements elicitation phase, but there is no defined way or set of rules how developers can transfer one language to another, how they can “merge” knowledge gained from misuse cases to mal-activity diagrams and vice versa. So to fulfill this gap and allow developers efficiently use combination of these languages together we should elicit transformation rules for misuse cases and mal-activity diagrams.

To summarize, misuse cases and mal-activity diagrams are chosen in this thesis, because:

- they both are intended for requirements stage of IS development,
- previously misuse cases and mal-activity diagrams already have been aligned to ISSRM domain model (Matulevičius *et al.*, 2008; Chowdhury, 2012),
- extension for usage of these languages at the full extent with ISSRM model is par-

tially done (Soomro & Ahmed, 2012),

- both languages first model usual process and then actions of misuser are added.

Table 1: Comparison of security- oriented modeling languages

Modeling language	Phase of development	Alignment to the ISRRM domain model	Extensions according to ISSRM
Secure Tropos	All	+	+
KeS	Requirements	+	-
BPMN	Requirements, design	+	+
UMLsec	Design	-	-
SecureUML	Design	-	-
Extension of Problem Frames	Requirements	-	-
Abuse Frames	Requirements	-	-
Misuse cases	Requirements	+	+
Mal-activity diagrams	Requirements	+	+/-

Both languages should be aligned to the ISSRM model at the full extent, which was previously done by Matulevičius *et al.* (2008) for misuse-cases diagrams and by Chowdhury *et al.* (2012) for mal-activity diagrams. However it should be revised in order to see if any elements from ISSRM domain model don't have accordance yet in the graphical representation of these two modeling languages. Then gaps, which were found, if any, should be supplemented before transformation, meaning that abstract and concrete syntax should be extended with missing elements.

3.2 Security Risk-Oriented Misuse Cases

Misuse cases are extensions of UML diagrams, which allow developers to address security concerns during early requirements stage. Misuse cases can be presented in graphical form- diagrams and in the table form- textual templates.

Misuse case diagrams provide only general overview of the system and can not capture required functionality at the level needed for developers, so that they can understand threats in the system and implement ways to deal with them. Providing textual templates "encourage developers to write clear and simple action sequences" (Sindre & Opdahl, 2005) and allows understanding of possible system risks in details. According to Sindre and Opdahl (2005), there are two templates, which could be used for textual misuse cases description. First one is *lightweight*, and second is *extensive*. Lightweight textual template is based on the use case template, as introduced by Kulak and Guiney (2000) and Cockburn (2001), extended only with field *Threats*. Extensive textual template proposed by Sindre and Opdahl in (2001, 2005), based on use-case template (Kulak & Guiney, 2000; Cockburn, 2001) as well, although it is extended with more constructs for detailed analysis of the system risks. It is supplemented with such constructs, as *Misuser profile, Stakeholders and risks, Mitigation guarantee* and *Technology and data variations*. In the next chapters we are going to analyze only extensive textual template, since lightweight doesn't have fields for the explicit security analysis.

Misuse case diagrams were aligned to the ISSRM domain model (Matulevičius *et al.*, 2008) and extended in (Soomro & Ahmed, 2012). The main reason for misuse case dia-

grams extension, which resulted in the Security Risk-oriented Misuse Cases (SROMUC), was to enable support of the security risk management for developing IS. The extensions were done on all three components of the language: concrete syntax, meta-model and semantics.

Table 2: Misuse cases alignment to the ISSRM domain model (C-construct, R-relationship)

ISSRM domain model concepts		Type	Misuse-case diagram concepts
Asset-related	Asset	C	Actor and use case
	Business asset	C	Use case
	IS asset	C	Use case, software system
	Security criterion	C	Construct to represent security criterion
	Supports	R	Extends, includes
	Constraint of	R	Constraint of
Risk-related	Risk	C	Combination of constructs for event and impact
	Impact	C	Construct for presenting impact
	Event	C	Combination of constructs for threat and vulnerability
	Threat	C	Misuser and misuse case
	Vulnerability	C	Construct for presenting vulnerability
	Threat agent	C	Misuser
	Attack method	C	Misuse case
	Exploits	R	Exploits
	Negates	R	Negates
	Harms	R	Harms
	Leads to	R	Leads to
	Characteristic of	R	Includes, extends
	Targets	R	Threatens
	Uses	R	-
Risk treatment-related	Risk treatment	C	-
	Security requirement	C	Security use case
	Control	C	-
	Refines	R	-
	Mitigates	R	Mitigates
	Implements	R	-

In correspondence to the ISSRM domain model, alignment was done dividing elements to three concept groups:

- asset-related concept;
- risk-related concept;
- risk treatment-related concept.

Alignment of SROMUC with ISSRM domain model is presented in the Table 2, we

will use it in further chapters. We explain the SROMUC notation on the running example of Registration to the course process. Figure 3 illustrates asset model; Figure 4 shows risk model and Figure 5 presents risk treatment model.

Asset-related concepts (Figure 3). *Users*, which are interacting with the system, are *Student* and *Lecturer*, they are presented outside system boundaries. *System* is *OIS* and is presented as rectangle. Actions which perform each user are connected to him and presented as ovals- *use cases*. Use cases, which are not connected to any user, are performed by the system. Use cases *Send request to the Lecturer* and *Accept student* are connected by relationship *<<include>>*, which shows order of process steps. *Security criterion- Integrity of admission* is presented as hexagon and related to the *See admission* use case.

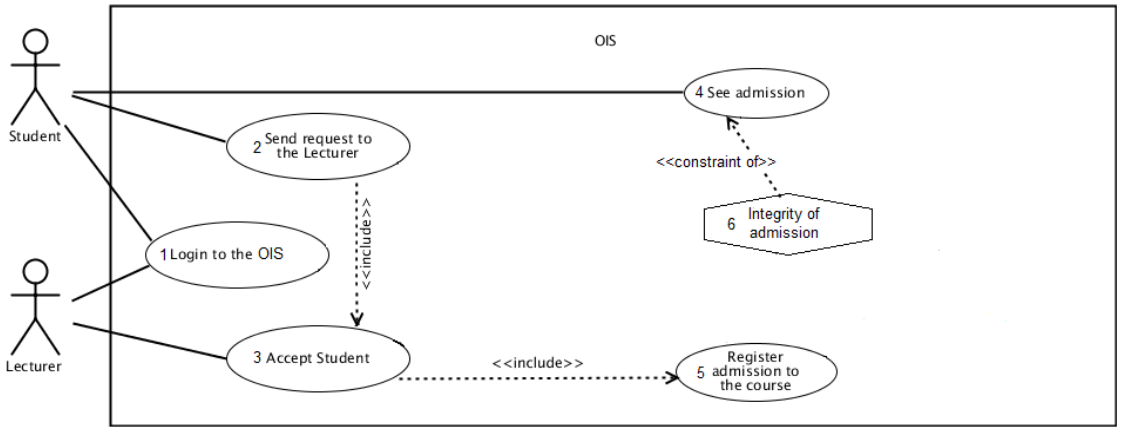


Figure 3: Asset-related concepts. SROMUC. Registration to course process

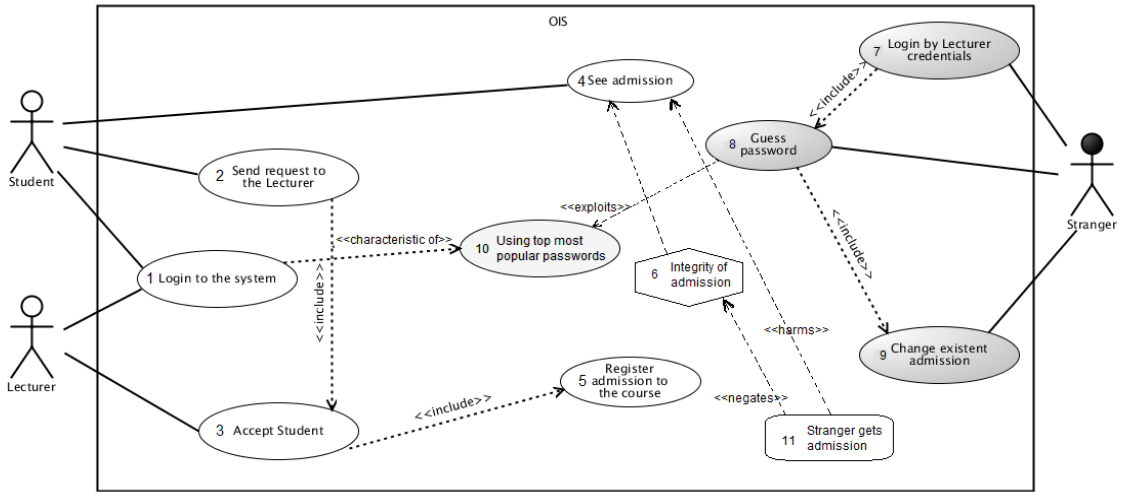


Figure 4: Risk-related constructs. SROMUC. Registration to course process

Risk-related concepts (Figure 4). *Stranger* outside university, *misuser*, who wants to cause harm to the system, is presented outside the system the same as users. Actions, which perform misuser, are presented by *misuse cases* and connected to him. In the Figure 4 they are *Login by Lecturer credentials*, *Guess password*, and *Change existent admission*. *Change existent admission* misuse case *leads to impact- Stranger gets admission*. Password

can be guessed because *Guess password* misuse case *exploits vulnerability- Using top most famous passwords*. Impact *negates* security criterion- *Integrity of admission* and *harms* asset- *See admission*.

Risk treatment-related concepts (Figure 5). To avoid possible threats shown in the Figure 5 there are *security use cases*, which are represented as *use case with a lock inside*. In our example in order to mitigate possible risks can be applied security use case*: Check entered password*, which is part of *Login to the system* and mitigates *Guess password*.

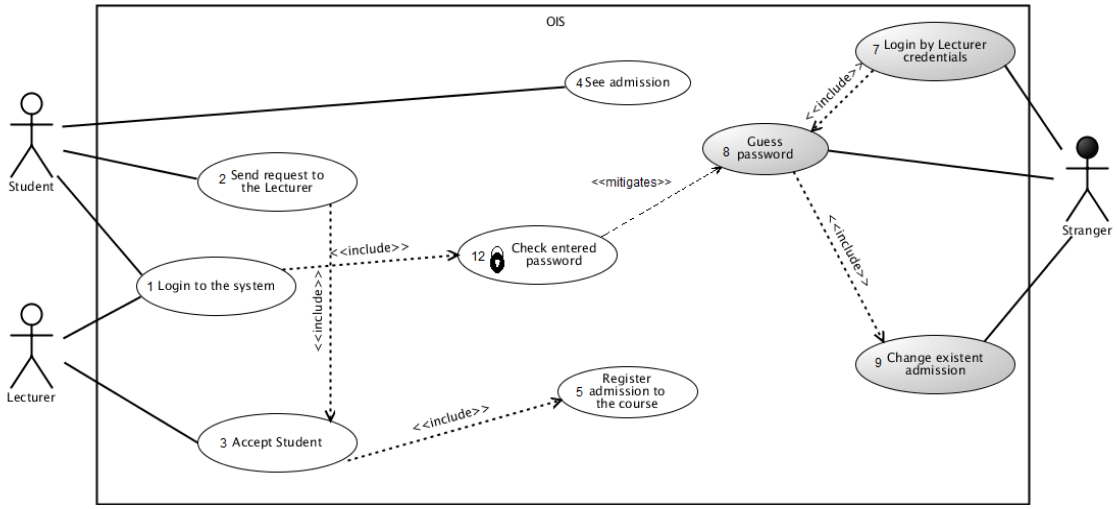


Figure 5: Risk treatment-related concepts. SROMUC. Registration to course process

3.3 Mal-activity diagrams

Mal-activity diagrams (Sindre, 2007) are extensions of the UML activity diagrams for modeling security concerns of the system. They deal with behavior of security problems and shows process flow. Mal-activity diagrams were previously aligned to the ISSRM domain model in (Chowdhury *et al.*, 2012). Table 3 presents the alignment, which we will use in further chapters for the mal-activity diagrams extensions and analysis.

Asset-related concepts. Assets are presented by *activities* with *decisions*, which are situated in the *swimlanes*. Each swimlane presents user or part of the system. In the mal-activity diagrams there is no presentation for security criterion.

Risk-related concepts. Attacker is presented by *mal-swimlane*, and this mal-swimlane contains activities of misuser- *mal-activities* and *mal-decisions*. Impact is presented by one or several *mal-activities*.

Risk treatment-related concepts. Risk treatment decision is taken to reduce risk. Means by which it will be done are presented in the separate swimlane- *security module*. Security requirements are shown as *mitigation activities* and connected to the mal-activity, which it mitigates by *mitigation link*.

Table 3: Mal-activity diagrams alignment with ISSRM domain model

ISSRM domain model		*Mal-activity diagrams*
Asset	Asset	-
	Business asset	Activity, Decision, Control flow
	IS asset	Swimlane, Activity, Decision
	Security criterion	-
Risk	Risk	-
	Impact	Mal-activities
	Event	-
	Vulnerability	-
	Threat	Combination of constructs, representing Threat agent and attack method
	Threat agent	Mal-swimlane
	Attack method	Mal-activities, Mal-decision, Control flow
Risk treatment	Risk treatment decision	-
	Security requirement	Mitigation activity, Mitigation link
	Control	Swimlane

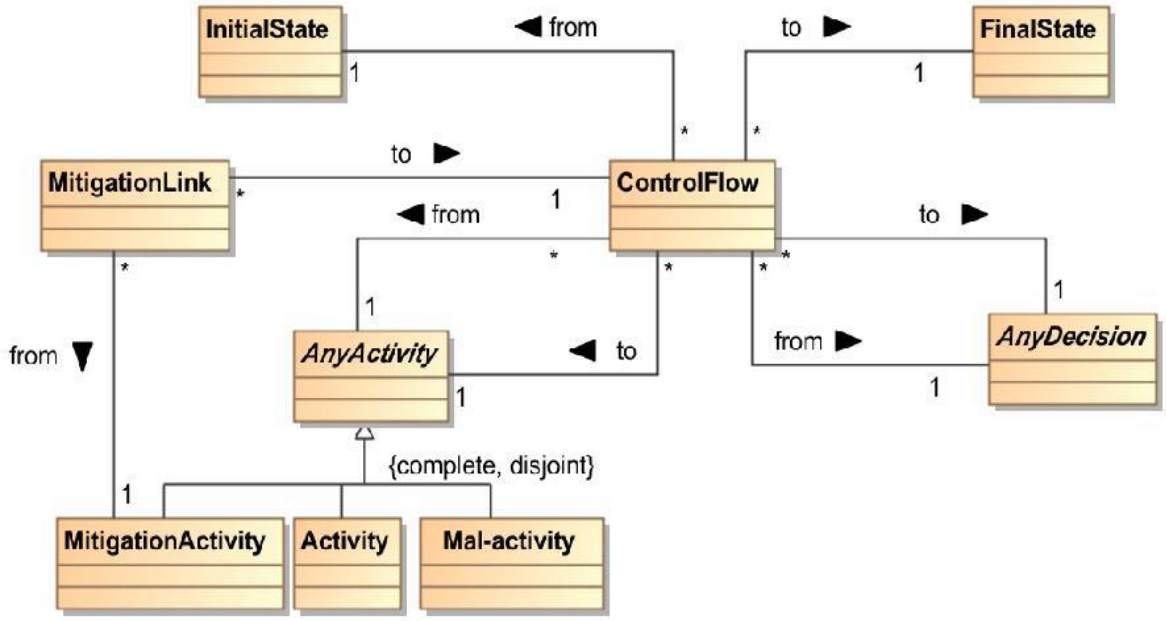


Figure 6: Meta-model of the mal-activity diagrams control flow (Chowdhury et al., 2011)

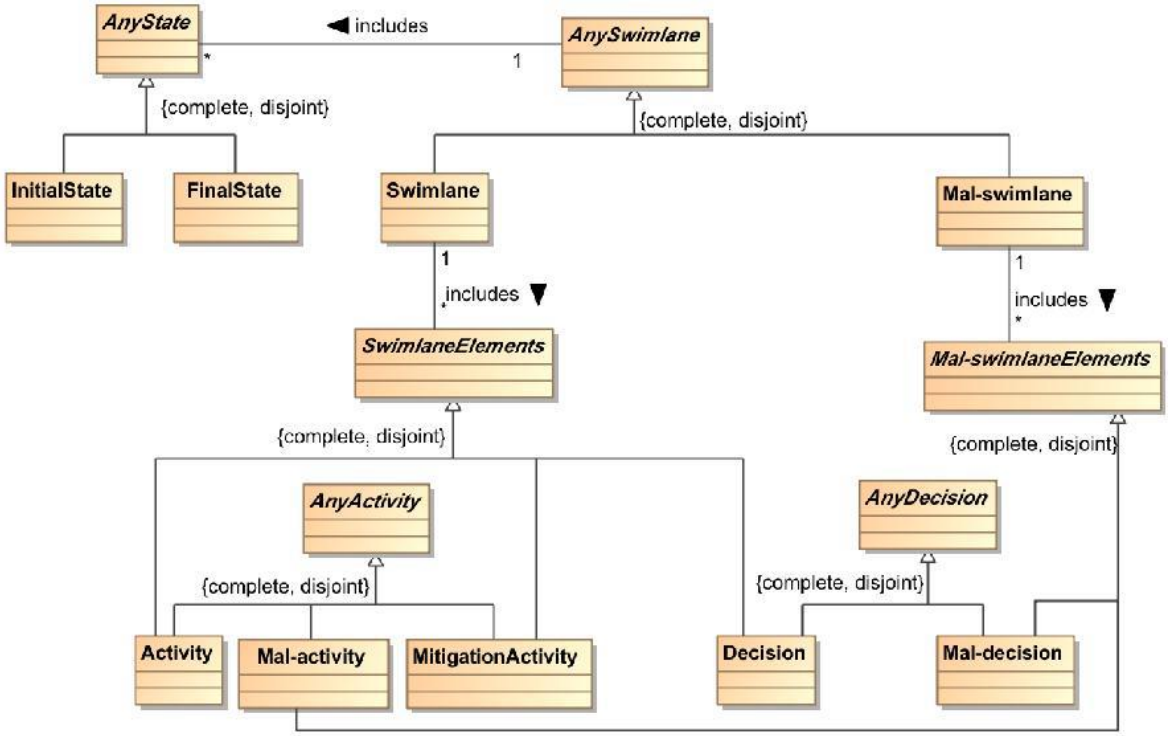


Figure 7: Meta-model of mal-activity diagrams (Chowdhury et al., 2011)

In the Figure 7 presented meta-model of the control flow of mal-activity diagrams. *ControlFlow* goes from *InitialState* to *FinalState*. Control flow has only one initial and one final state. *ControlFlow* goes from *AnyActivity* and to *AnyActivity*, from *AnyDecision* and to *AnyDecision*, from *AnyActivity* to *AnyDecision* and vice versa. *MitigationActivity* is connected to *ControlFlow* through *MitigationLink.*

3.4 Summary

In this chapter we presented short overview of the security risk-oriented modeling languages and their comparison. For each language we provided description of the main elements and explained main modeling principles for particular languages. Next we compared languages, which were introduced, and provided reasons why misuse cases and mal-activity diagrams were chosen in this thesis. Then current state of the research for misuse case and mal-activity diagrams languages is presented. For both languages we presented alignment to ISSRM domain model. Moreover we present meta-model of abstract mal-activity diagrams syntax and meta-model of mal-activity diagrams control flow.

4 Misuse Case Template Extension

In this chapter we are going to present textual template (Sindre & Opdahl, 2005) for misuse cases detailed description. The goal of this chapter is to investigate completeness of the misuse case extensive textual template for modeling security scenarios with regard to the ISSRM domain model. Then we are going to suggest possible extensions, so that misuse case template can cover ISSRM domain model, which was presented in the chapter 2. Extension of the misuse case template is needed, so that potentially they can be used for transforming misuse case to mal-activity diagrams.

4.1 Misuse case extensive textual template and ISSRM

Misuse case extensive textual template alignment to the ISSRM domain model (Matulevičius *et al.,* 2008) is presented in the Table 4.

Table 4: Misuse case extensive textual template alignment with ISSRM domain model (Matulevičius et al., 2008)

ISSRM domain model		*Misuse case extensive template*
Asset	Asset	-
	IS asset	-
	Business asset	Related business rules
	Security criterion	-
Risk	Risk	Stakeholders and risks
	Impact	Worst case threat
	Event	-
	Vulnerability	Assumption, precondition, trigger
	Threat	-
	Threat agent	Misuser profile
	Attack method	Basic path, alternative path, extension points
Risk treatment	Risk treatment	-
	Security requirements	Mitigation points
	Control	-

Asset-related constructs. According to the alignment Table 4, there is representation only for business assets from asset-related group of constructs. They are presented by field *Related business rules*. According to Sindre and Opdahl (2005) *Related business rules* field contains business rules, which will be violated by the misuse, possible together with links to the rules that enable the threat. Thus, we agree that related business ruled field contains business assets of the system. Next element from ISSRM domain model, which should be represented in the misuse case textual template, is *Security criterion*. Currently template doesn't have any field, which holds such information, thus we suggest extending template with field *Security criterion.* We assume it should be added after field *Related business rules*. In this way it will correspond ISSRM domain model, while *Related business rules* field define business assets and security criterion is connected to the business assets.

Table 5: Extensions presented on the Change existent admission misuse case template

№	**9**
Name	**Change existent admission**
Summary	Stranger logged in by Lecturer credentials changes existent admission and enters data, so that he instead of one of the Student is registered to the course
Basic path	Bp1. Choose admission. Bp2. Enter new data to admission (Strangers data). Bp3. OIS accepts and stores new changed admission instead of old (Extp1).
Alternative path	-
Risk treatment decision	Risk reduction
Mitigation points	Mp1. Check entered password
Extension points	Extp1. Impact 'Stranger gets admission'
Trigger	Always true
Assumption	As1. Stranger knows password of the Lecturer
Preconditions	Pc1. Using one of the top most famous passwords for login to the account
Worst case threat	Stranger gets admission
Mitigation guarantee	-
Related business rules	Br1. Student sends request to the Lecturer. Br2. Lecturer accepts student. Br3. OIS registers admission. Br4. Student can view an admission.
Security criterion	Integrity of admission
Misuser profile	Stranger
Scope	OIS
Iteration	-
Level	-
Stakeholders and risks	Student: Is not registered for course Lecturer: Got his data stolen and should be responsible for actions not performed by him.
Technology and data variations	-

Risk-related constructs. In the alignment table (Table 4) there is no representation for *Threat* and *Event* in the textual template. According to the ISSRM domain model *Threat* is combination of *Threat agent* and *Attack method.* We suggest not adding any extra fields; rather one can say that *Threat* is represented by combination of fields *Misuser profile* (Threat agent) and *Basic path, Alternative path* and *Extension points* (Attack method). The same applies for *Event*, which doesn't have representation in the misuse case template. It can be understood as combination of fields *Misuser profile* (Threat agent), *Basic path, Alternative path, Extension points* (Attack method) and *Assumption, Preconditions, Trigger* (Vulnerability). According to the Table 4 other risk-related constructs have representation in the template.

Risk treatment-related constructs. Next set of ISSRM constructs is missing *Risk treatment* decision in the misuse case textual template. For understanding mitigation mech-

anism it is important to understand which risk treatment decision is taken. So we suggest adding field *Risk treatment* to the template, then way of dealing with possible risks will be strictly stated in the template. *Security requirement,* according to the Table 4, is represented by *Mitigation points* field in the template. While risk treatment decision is connected to the security requirements we add field *Risk treatment* before presenting security requirements, so before *Mitigation points* field. We present one of the misuse case extensive textual templates for the running example of the Registration to the courses process. Example explicitly is presented in the Appendices A-E. Textual template is chosen randomly, it is *Change existent admission*, presented in the Table 5. New fields, which we added to the template, are *Security criterion* and *Risk treatment.* They represent constructs with the same name from the ISSRM domain model. For all other constructs from ISSRM domain model we found representation in the existing textual template fields.

In the Table 6 we present extended alignment of the template with ISSRM domain model, underlined elements are the new elements, which we added.

Table 6: Extended misuse case template alignment with ISSRM domain model

ISSRM domain model		*Misuse case extensive template*
Asset	Asset	*IS asset + Business asset*
	IS asset	-
	Business asset	Related business rules
	Security criterion	Security criterion
Risk	Risk	Stakeholders and risks
	Impact	Worst case threat
	Event	*Threat + Vulnerability*
	Vulnerability	Assumption, precondition, trigger
	Threat	*Threat agent + Attack method*
	Threat agent	Misuser profile
	Attack method	Basic path, alternative path, extension points
Risk treatment	Risk treatment	Risk treatment
	Security requirements	Mitigation points
	Control	-

4.2 Templates for security criterion, vulnerability and impact

The ISSRM domain model constructs: security criterion, vulnerability, impact and security use case have representation in the misuse case diagrams notation, but there is no approach, how to express them in the textual way. Thus we suggest 4 new textual templates for security criterion, vulnerability, impact and security use case, which will allow making explicit analysis of the possible risks using textual templates.

In the Table 7 we present example of suggested template for *security criterion*, it is taken from the running example. Row *ID* contains identification number of this element. In the field *Name* should be explicit short name of this security criterion. Field *Summary* shortly describes what this security criterion means. *Constraint of which asset* field provides enumeration of the business assets (use cases), to which applies this security criterion.

In the Table 8 presented template example for vulnerability. *ID* is identification number of the element, field *Name* gives vulnerability name. *Summary* explains shortly why this place at the system is vulnerable. *Characteristic of which asset* states to which IS assets (use case) corresponds vulnerability. *Exploits threat* field contains misuse case name, which exploits this vulnerability.

Table 7: Security criterion template example of the Registration to courses process

ID	**6**
Name	**Integrity of admission**
Summary	Admission should be not possible to modify
Constraint of which asset	See admission, Register admission to the course

Table 8: Vulnerability template example of the Registration to courses process

ID	**10**
Name	**Using top most popular passwords**
Summary	Using one of the top most famous passwords simplify account crack, as password can be guessed by enumeration of the top most famous passwords
Characteristic of which asset	Login to the OIS (Bp3. Actor enters password)
Exploits threat	Guess password

Table 9: Impact template example of the Registration to courses process

ID	**11**
Name	**Stranger gets admission**
Summary	As a result of changing existent admission stranger gets and admission and it is not connected to the Student anymore.
Which asset harms	See admission
Which event leads to impact	Change existent admission

In the Table 9 presented template for impact. *ID, Name* contain element number and name of the impact correspondingly. *Summary* provides short explanation what this impact means. *Which asset harms* field provides names of the assets, which are harmed by this impact. And field *Which event leads to this impact* states misuse case, which leads to the presented impact.

Table 10 shows example of the security use case. *ID, Name* and *Summary* contain the same information as in previous templates. *Basic path* field is intended to show, how this security requirement mitigates defined risk. *Extension points* states use or misuse cases to which connected security use case. *Risk mitigation* field explains which risk is mitigated by this security use case.

Table 10: Security use case template presented on the Registration to the courses process

ID	**12**
Name	**Check entered password**
Summary	Use stronger passwords in order to login into the OIS. One should include capital letters, numbers and special symbols. Moreover password should be not less than 6 characters.
Basic path	1. During registration check if entered password is strong (if it is not less than 6 characters, has at least one capital letter, one special symbol and one number) 2. If password is strong- register in the OIS 3. If password is not strong- request to enter new password.
Extension points	1. Login to the OIS Mitigates 8. Guess password
Risk mitigation	Mitigates risk that stranger using one of the top most famous passwords can change existent admission.

4.3 Summary

In this chapter we analyzed misuse case textual template and 2 fields for extensions were suggested, they are *Security criterion* and *Risk treatment decision.* Representation of other constructs from ISSRM domain model were found in existing fields of template. After proposed extensions extensive textual template fields cover the concepts of the ISSRM domain model. Also we proposed 4 textual templates for security criterion, impact, vulnerability and security use case representations. Thus, it makes possible for developers to analyze security problems using template, they do not need to complement knowledge gained from the diagrams with analyzing textual templates.

5 Mal-Activity Diagrams Extensions

We are going to extend existing mal-activity diagrams, based on the previous alignment of them with ISSRM domain model (Chowdhury *et al.*, 2012). As a result of these extensions developers will be able to use mal-activity diagrams at the full extent with ISSRM domain model for modeling security concerns. These extensions will be used in the next chapters, in order to elicit transformation rules from misuse cases to mal-activity diagrams.

5.1 Mal-activity diagrams and ISSRM domain model

Mal-activity diagrams alignment to the ISSRM domain model (Chowdhury *et al.*, 2012) is presented in the Table 3 (see chapter 3.3). From this alignment it can be seen that graphical representation of the mal-activity diagrams doesn't cover ISSRM domain model (e.g. risk, security criterion etc.). In the asset-related group of constructs there is no correspondence to the asset and security criterion elements. Risk-related concepts don't have elements for risk, event and vulnerability graphical representation. And from risk treatment-related group of elements there is no correspondence to the risk treatment decision. We are going to analyze missing constructs with regard to the ISSRM domain model and suggest possible extensions. Also in the previous work (Chowdhury *et al.*, 2012) there is no explicit explanation and analysis of the mal-activity diagrams relationships, thus we will present meta-model of mal-activity diagrams relationships.

5.2 Security Risk-oriented Mal-activity diagrams. Concrete syntax

Mal-activity elements are presented in the table format with the following discussion. Then extensions are shown on the running example of the Registration to the course process.

Asset-related concepts. In this group of concepts, correspondence for *security criterion* is missing. According to the ISSRM domain model *security criterion* is property/constraint to business asset. Alignment Table 3 shows that *business assets* in the mal-activity diagrams are presented by process flow (activity, decision, and control flow). So we suggest to present *security criterion* as element, connected to particular business asset, to which security criterion corresponds, or to flow, depending on internal logic of process. Security criterion should belong to particular swimlane. We choose oval for presenting security criterion, this shape doesn't have any special meaning and is chosen randomly. According to the ISSRM domain model security criterion is connected to the business asset by <*constraint of*> relationship. We present this relationship as dotted line, in order to distinguish presentation from control flow. Regarding <*supports*> relationships between IS asset and business asset, it doesn't need separate visual representation. IS assets in mal-activity diagrams are presented by swimlane or process flow, business assets are presented by process flow as well, thus relationship <supports> between them is control flow. Graphical representation for asset-related constructs and relationships can be seen in the Table 11.

Risk-related concepts. In this set of constructs there is no correspondence to the risk, event, and vulnerability. *Event*, according to the ISSRM domain model, is combination of threat and vulnerability, so in order to present event, we should have graphical presentation for *threat* and *vulnerability*. *Threat*, according to the Table 3, is

presented by mal-swimlane together with mal-process flow. *Vulnerability* from ISSRM domain model is characteristic of IS asset. *IS asset* in mal-activity diagrams according to the alignment (Table 3) is presented by control flow or mal-activity. Thus we suggest to show *vulnerability* as element, connected to particular mal-activity or to control flow, depending on logic of process. *Vulnerability* the same as security criterion should be situated in the particular swimlane. We use gray-filled rectangle for vulnerability presentation, this shape doesn't have any special meaning and chosen randomly. When we already have graphical representation for the vulnerability, *risk* can be presented by combination of constructs for *event* and *impact*. Graphical presentation for the each risk-related construct is shown in the Table 12.

Table 11: Alignment of mal-activity diagrams with asset-related concepts of ISSRM domain model

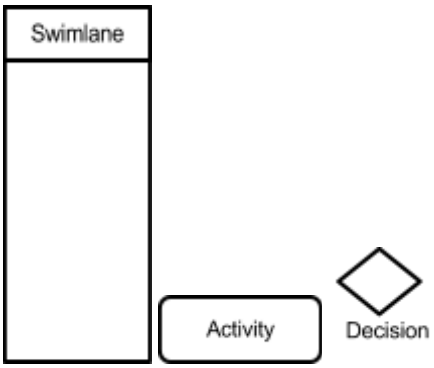
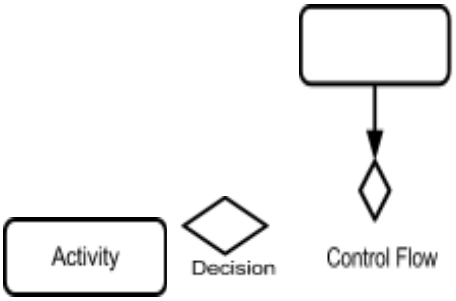
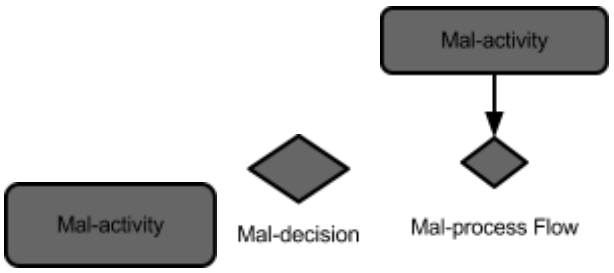
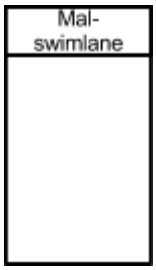
ISSRM	Type	Mal-activity diagrams
Assets	C	IS asset+ Business asset
IS assets	C	
Business assets	C	
Security criterion	C	
supports	R	Control flow
constraint of	R	

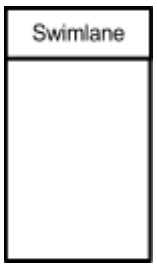
Table 12: Alignment of mal-activity diagrams with risk-related concepts of ISSRM domain model

ISSRM	Type	Mal-activity diagrams
Risk	C	Event + Impact
Impact	C	
Event	C	Threat + Vulnerability
Attack method	C	
Vulnerability	C	
Threat agent	C	
Threat	C	Threat agent + Attack method
exploits	R	-
negates	R	-
harms	R	-
leads to	R	-
characteristics of	R	
targets	R	Control flow
uses	R	-

Relationships in the risk-related set of constructs are: <leads to>, <harms>, <negates>, <exploits>, <characteristic of>, <uses>, <targets>. Threat agent <*uses*> attack method. *Threat agent* in mal-activity diagrams is presented by mal-swimlane and *attack method* is expressed by mal-process flow. Thus <*uses*> relationship between them doesn't need graphical representation, it is assumed by placing mal-process flow in the mal-

swimlane. The same logic applies to the <*leads to*> relationship, which connects *event* and *impact*. *Event* is presented by mal-swimlane and *impact*- by mal-activity, thus separate graphical presentation for <*leads to*> is not needed. According to the ISSRM domain model *impact* is connected by <*negates*> relationship to the *security criterion* and by <*harms*> relationship to the *asset*. *Security criterion* is presented as oval, which is connected to the particular activity or to the process flow. Connecting *security criterion* to mal-activity, which represents *impact* will not allow clearly understand flow of the process, thus we suggest don't present <*negates*> graphically. The same applies to the <*harms*> relationship, which connects *impact* to the *asset* (presented by swimlane or process flow). *Vulnerability*, which is presented by gray-filled rectangle, is connected by relationship <*characteristic of*> to the *IS asset*, presented by process flow or activity. Thus we suggest presenting <*characteristic of*> relationship as dotted line, to distinguish presentation from control flow. Threat <*targets*> IS asset, and threat <*exploits*> vulnerability. Thus presenting <*exploits*> relationship will disturb understanding mal-activity diagram and process flow. Regarding relationship <*targets*>, it is represented in mal-activity diagrams by control flow and don't need any special separate graphical representation. Graphical representation for relationships can be seen in the Table 12.

Table 13: Alignment of mal-activity diagrams with risk treatment-related concepts of ISSRM domain model

ISSRM	Type	Mal-activity diagrams
Risk treatment	C	-
Security requirement	C	
Control	C	
leads to	R	-
mitigates	R	
implements	R	-

Risk treatment-related concepts. *Risk treatment* decision in this set of constructs is missing. But it should not be presented graphically, since it is decision, which developer or software engineer takes, in order to model *control* and *security requirements*. Relationships corresponding to risk treatment-related constructs are: <decision to treat>, <leads to>, <implements>, <mitigates>. Relationships <*decision to treat*> and <*leads to*> don't need visual presentation, because these relationships are connected to the risk treatment decision, which is not presented graphically. Control <*implements*> security requirements. *Control* is

presented by swimlane and *security requirements* are presented by mitigation activities and mitigation-process flow, according to the alignment Table 3. Thus relationship <*implements*> doesn't have separate visual representation, it is reflected by place of mitigation-process flow in the control swimlane. <*Mitigates*> is relationship between security requirement and risk, it is presented in mal-activity diagrams by mitigation link. *Mitigation link* graphically is shown by dotted line, connecting mitigation activities and mal-activities, which it mitigates. Graphical representations for risk treatment-related constructs are presented in the Table 13.

5.3 Example

We present 3 separate diagrams for each set of constructs from ISSRM domain model. Diagrams present running example of Registration to the courses process and were built, based on the information from the Appendices A and B.

Asset-related constructs. In the Figure 8 mal-activity diagram for assets is presented. Security criterion is *Integrity of admission*, which is presented by the oval, and connected to the *Request admission* business asset. Security criterion is connected to the activities by dotted line.

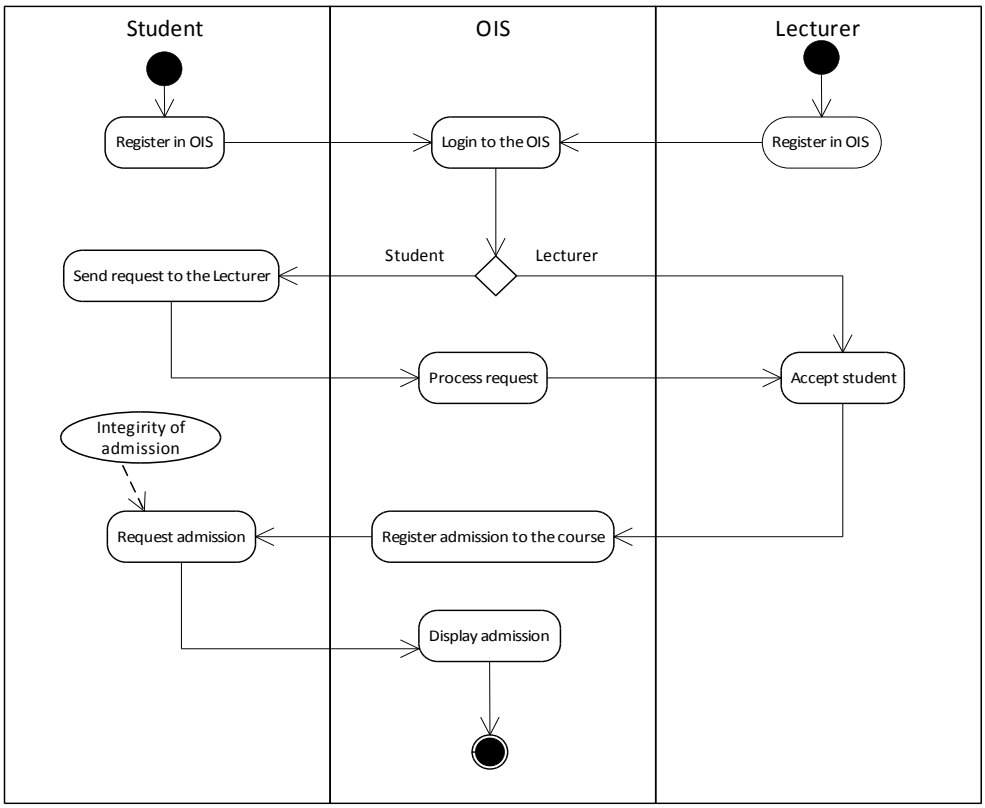


Figure 8: Mal-activity diagram for registration to the courses process (asset-related concepts)

Risk-related constructs. In the Figure 9 mal-activity diagram for risk-related constructs is shown. Vulnerability, *Using top most popular passwords*, is presented by gray-filled rectangle situated in the *OIS* swimlane. Vulnerability is connected by dotted line to the IS asset - *Login to the OIS*. Impact in this case is *Get admission*.
Usually impact in mal-activity diagrams is placed to the separate swimlane, which presents

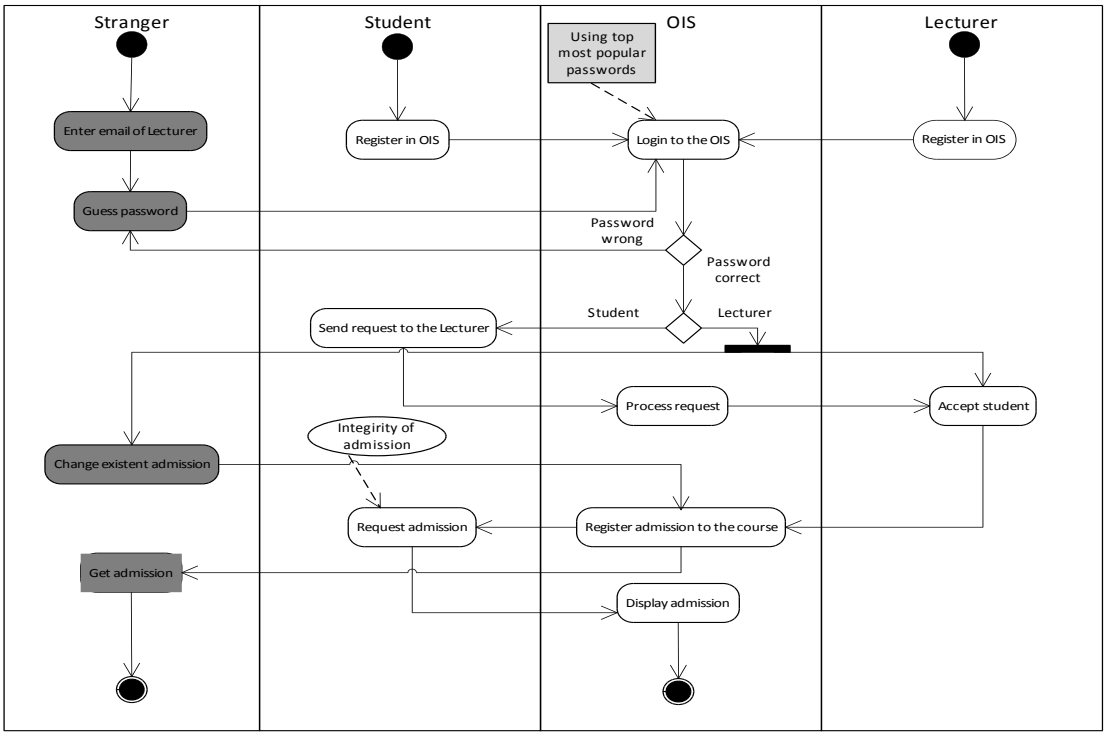


Figure 10: Mal-activity diagram of the registration to the courses process (risk-related constructs)

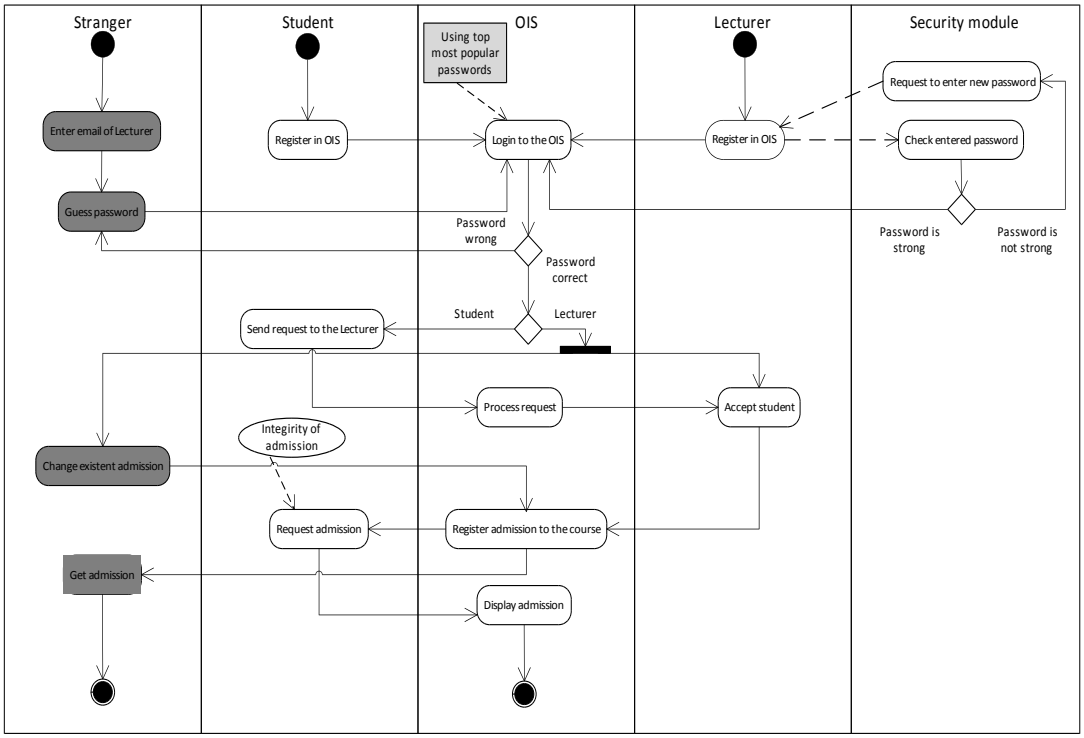


Figure 9: Mal-activity diagram of the registration to the courses process (risk treatment-related conctructs)

means using which misuser performs attack. But in our case *Stranger* attacks directly *OIS*, thus he uses only OIS to perform attack, and separate swimlane for impact is not needed.

Risk treatment-related constructs. Diagram for risk mitigation is presented in the Figure 10. In this set of constructs we didn't propose any extensions. *Security module* swimlane presents control; *Check entered password* and *Request to enter new password* present security requirements.

5.4 Security Risk-oriented Mal-activity diagrams. Abstract syntax

In the chapter 3.2 we presented meta-model proposed by Chowdhury (2012) (Figure 7). But due to extension proposed to the concrete syntax, meta-model should be updated as well. Two elements were added to the mal-activity diagrams syntax, which are *vulnerability* and *security criterion*. Suggested graphical representation for these elements can be found in the Table 7, 8 and in the running example in the Figure 8 and 9. These extensions should be reflected in the mal-activity diagrams meta-model as well.

Security criterion and *vulnerability* are one of the *SwimlaneElements*, since they are always placed in the swimlane. In the previous meta-model abstract class *SwimlaneElements* included *MitigationActivity*. We suggest separating mitigation elements to the distinct class - *MitigationSwimlaneElements,* which includes *MitigationActivity* and *MitigationDecision*. After separation of mitigation elements we have *SwimlaneElements* class, which include only activity and decision, thus vulnerability and security criterion now can be included to the *SwimlaneElements* class. Updated meta-model is presented in the Figure 11.

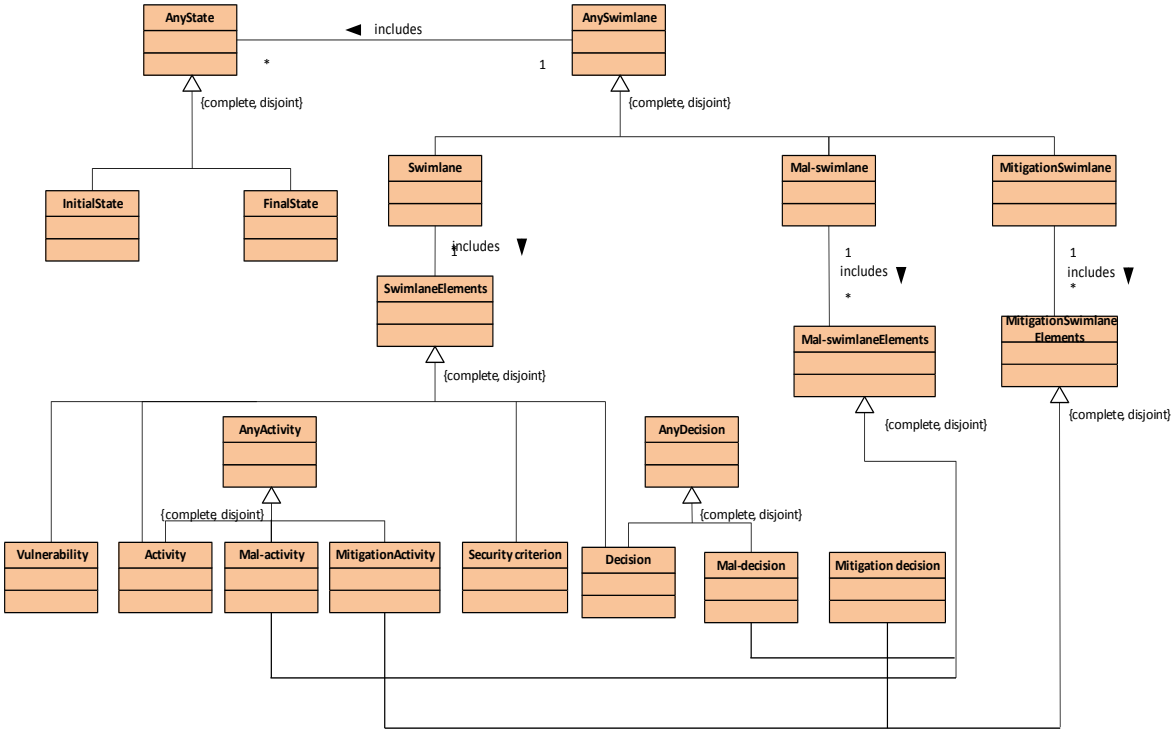


Figure 11: Extended meta-model of the mal-activity diagrams

In the previous works (Chowdhury *et al.*, 2012), there was presented mal-activity diagram meta-model for c*ontrol flow* (Figure 6). New constructs, which were added to the concrete syntax are *security criterion* and *vulnerability*, thus we should show their connection to other mal-activity diagram elements. *Security criterion* and *vulnerability* always are

connected to the activity or control flow. *Security criterion* is connected through <*characteristic of*> relation and *vulnerability* is connected through <*constraint of*>. Thus we add 4 abstract classes: *Security criterion*, *Characteristic of*, *Vulnerability* and *Constrain of* to the previous meta-model (Figure 6). *Security criterion* through *Characteristic of* is connected to the *Control flow* or *Activity*. The same applies for *Vulnerability*, but through *Constraint of* abstract class. After adding extension elements, final meta-model of relationships is presented in the Figure 12.

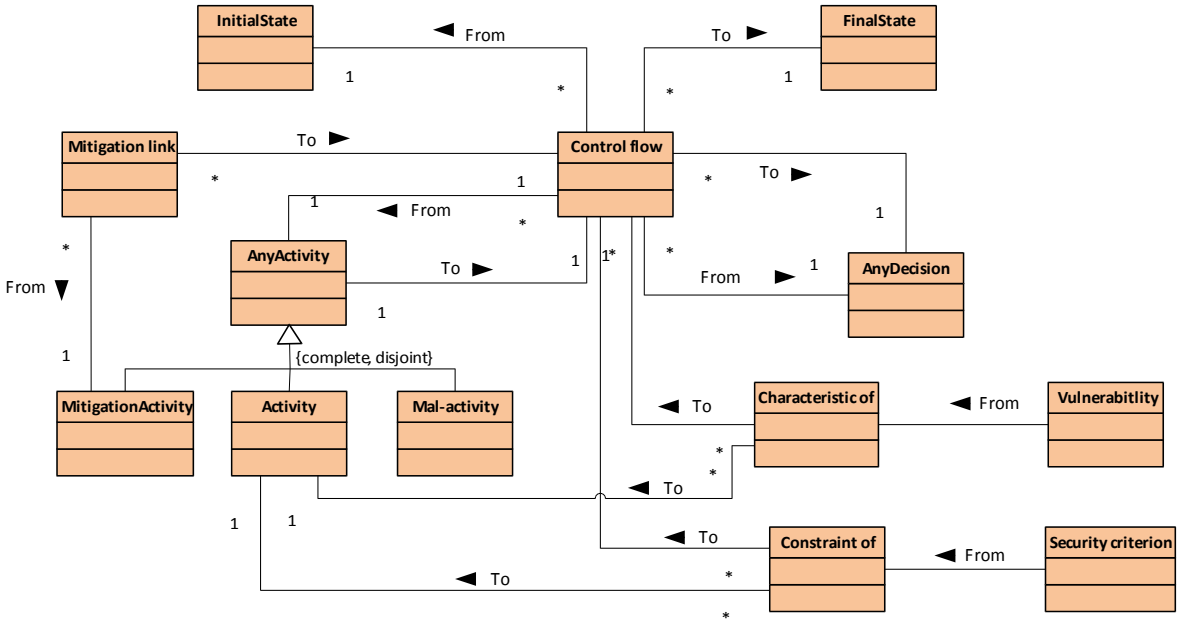


Figure 12: Extended relationships meta-model of the mal-activity diagrams

5.5 Summary

In this chapter we proposed extensions to the mal-activity diagrams. Firstly we extended concrete syntax; we demonstrated extensions on the running example. Extensions proposed to the mal-activity diagrams are: vulnerability and security criterion. Next, we supplemented mal-activity diagrams meta-model with missing abstract classes regarding concrete syntax. Moreover we analyzed how relationships from ISSRM domain model can be expressed in the concrete syntax of mal-activity diagrams. And we showed in control flow meta-model, how security criterion and vulnerability are connected to the other elements of mal-activity diagrams.

6 Transformation Rules: Misuse Case Diagrams -Mal-Activity Diagrams

In this chapter we are going to present transformation rules for model transformation from misuse case diagrams to the mal-activity diagrams. Under transformation rules meant consequent set of operations, which one should perform in order to translate existing misuse case diagram with textual templates and receive corresponding mal-activity diagram. In order to elicit transformation guidelines, first we extended textual templates and mal-activity diagrams with regard to the ISSRM domain model in the chapters 4 and 5 correspondingly. Transformation rules will be elicited, based on the semantic alignment (Matulevičius *et al.,* 2008; Chowdhury, 2011) of both languages to the domain for information security risk management (ISSRM) (Dubois *et al.,* 2010). Graphical transformation is a systematic approach, which will allow developers to elicit and analyze security requirements from two viewpoints at the same time. This will result in the development of protected information systems with built-in security functionality and will reduce costs of taking into account security threats at the later phases of the information system development (e.g. implementation, testing).

6.1 Related work

We have found one paper (El-Attar, 2012), which addresses the same problem of transformation from misuse cases to mal-activities. In the paper (El-Attar, 2012), author proposes to use such called Formal Misuse Case Description (FMCD), which are special textual description of the use cases. FMCD is a transitional step, from misuse case diagrams to mal-activities. One should first describe misuse cases in the form of FMCD and then following rules, which are presented in the paper, misuse cases can be easily translated to the mal-activity diagrams. Approach, which is presented in the paper is similar to one we are going to present in the thesis in the way, that both are based on the domain models (ISSRM in our case, FMCD in the paper). In both approaches misuse cases are analyzed with regard to the domain model and then transformed to the mal-activities following set of rules. The main difference between mentioned paper and approach, which we are going to present in the thesis, is that we base our research on the ISSRM domain model. Using ISSRM domain model one can thoroughly analyze system, present possible risks and suggest mitigation, it is well-grained approach. In contrast, mentioned paper doesn't focus on explicit analysis of risks and doesn't allow step-by-step elicitation of all constructs, important for information system design.

6.2 Misuse case and mal-activity diagrams in regard to the ISSRM domain model

To elicit transformation rules from one security modeling language to another, we need to have some base for transformation, or measure regarding to which transformation will be performed. As this base the ISSRM domain model is taken. In the Table 14 we present alignment of both languages to the ISSRM domain model.

Alignment is presented only for misuse case textual template, thus contains no information about use case template and templates for impact, security criterion, vulnerability and security use case. Nevertheless we will use information for transformation as well.

Table 14: Alignment of the misuse cases and mal-activity diagrams with ISSRM domain model

Line	ISSRM domain model concepts		Type	Misuse case diagram concepts	Misuse case textual template	Mal-activity diagram concepts
A	Asset	Asset	C	Actor and use case	-	-
B		Business asset	C	Use case, actor	Related business rules	Activity, Decision, Control-Flow.
C		IS asset	C	Use case, software system	-	Swimlane, Activity, Decision
D		Security criterion	C	Construct for security criterion	Security criterion	Construct to represent security criterion
E		Supports	R	Extends, includes	-	Control flow
F		Constraint of	R	Constraint of	-	Constraint of
G	Risk	Risk	C	-	Stakeholders and risks	-
H		Impact	C	Construct for impact	Worst case threat	Mal-activity
I		Event	C	-	-	-
J		Threat	C	Misuser and misuse case	Combination of constructs, representing threat agent and attack method	*Threat agent + Attack method*
K		Vulnerability	C	Construct for vulnerability	Assumption, precondition, trigger	Construct for presenting vulnerability
L		Threat agent	C	Misuser	Misuser profile	Mal-swimlane
M		Attack method	C	Misuse case	Basic path, alternative path, extension points	Mal-activities, Mal-decision, ControlFlow
N		Exploits	R	Exploits	-	Control flow
O		Negates	R	Negates	-	-
P		Harms	R	Harms	-	-
Q		Leads to	R	Leads to	-	-
R		Characteristics of	R	Includes, extends	-	Characteristic of
S		Targets	R	Threatens	-	Control flow
T		Uses	R	-	-	-
U	Risk treatment	Risk treatment	C	-	Risk treatment	-
V		Security requirement	C	Security use case	Mitigation points	Mitigation activity, Mitigation link
W		Control	C	-	-	Swimlane for treatment decisions
X		Leads to	R	-	-	-
Y		Mitigates	R	Mitigates	-	Mitigation link
Z		Implements	R	-	-	-

6.3 ***Transformation rules***

We will elicit transformation rules based on the running example of registration to the courses process. Translation rules will be presented, dividing them by 3 groups of concepts, as in ISSRM domain model: assets, risks, and risk treatment. For each group of concepts we present misuse case diagrams and applying set of transformation steps draw corresponding mal-activity diagram.

Asset-related concepts are translated using following transformation rules**:**

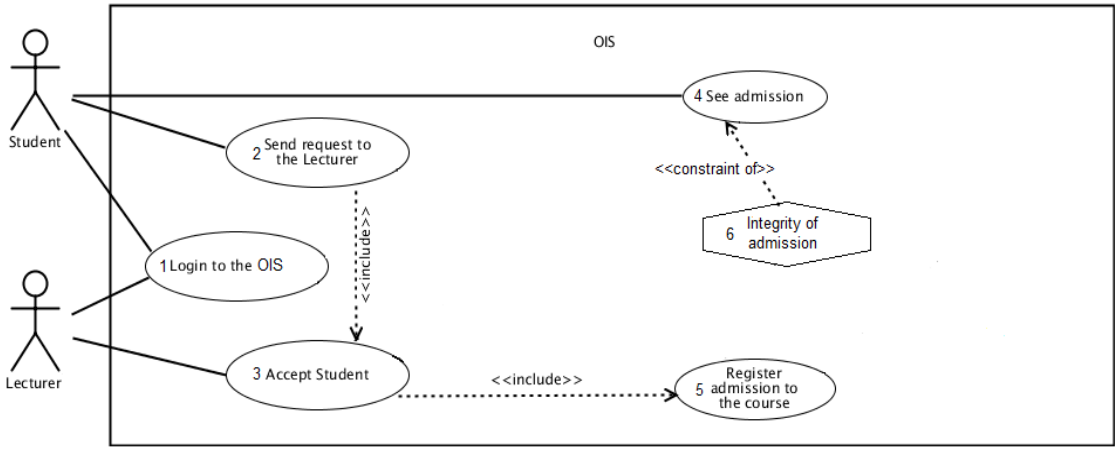


Figure 13: Misuse case diagram for registration to courses process (asset-related concepts)

TR1. Misuse cases *system boundary* is translated to the mal-activity diagrams *swimlane*.

System boundary is an IS asset in terms of ISSRM domain model, thus according to the line C of the alignment Table 14, it can be translated into swimlane. In Figure 13 system boundary is *OIS*, so we have the same swimlane in the mal-activity diagrams (Figure 14).

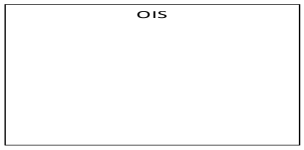


Figure 14: TR1 (MUC-MAD)

TR2. Misuse cases *actor* is represented by *swimlane* in the mal-activity diagrams.

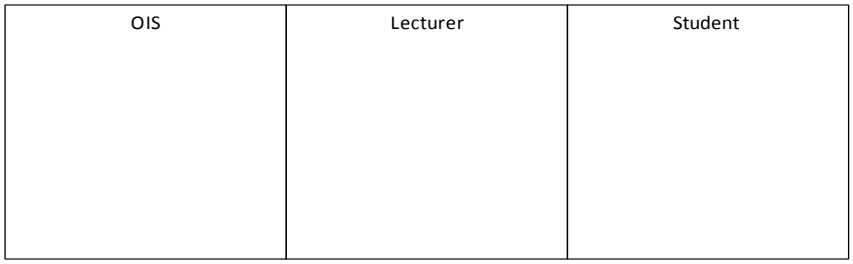


Figure 15: TR2 (MUC- MAD)

In order to understand to which element misuse cases actor should be translated in

the mal-activity diagrams, one should refer to the UML specification (UMLspec, 2011). According to UML specification Actor in use case diagrams is person, who interacts with the system and performs some group of activities. Swimlane for activity diagrams, according to UML specification, is group of activities, which have some characteristic in common. We assume that mal-activity diagram swimlane is set of activities, which has in common actor, who performs them. Thus we translate misuse cases actor to mal-activity diagrams swimlane. In the misuse case diagram actors *Student* and *Lecturer* (Figure 13) are translated into 2 swimlanes with the same names in the mal-activity diagrams (Figure 15).

TR3: Misuse cases *use case* is translated to the mal-activity diagrams *activity.*

Business and IS assets (lines B, C in Table 14), which are represented by misuse cases *use cases*, correspond to mal-activity diagrams *activities*.

G1. Swimlane, where to place activity should be taken from the field *Actor* of the corresponding use case template. If use case is performed by one actor- activity is placed in the corresponding swimlane. If use case is performed by more than one actor- see NOTE1.

This guideline is based on the definition of the field 'Actor' in the use case template according to the (Kulak & Guiney, 2000). Thus in the example, we place *Accept Student* activity to the *Lecturer* swimlane, *Send request to the Lecturer* and *See admission* to the *Student* swimlane and *Register admission to the course* is placed in the *OIS* swimlane. (Figure 16).

NOTE1: *If a use case is business asset and it is performed by actor then corresponding activity should be placed to the swimlane, corresponding to this actor. If use case is information asset it should be placed to the system swimlane.*

This note is based on the definitions of the IS and business assets with regard to the ISSRM domain model (Dubois *et al*, 2010). *Login to the OIS* activity wasn't placed anywhere because it is performed by 2 actors, but since it was defined as IS asset (see Appendix B), we place *Login to the OIS* into *OIS* swimlane (Figure 16).

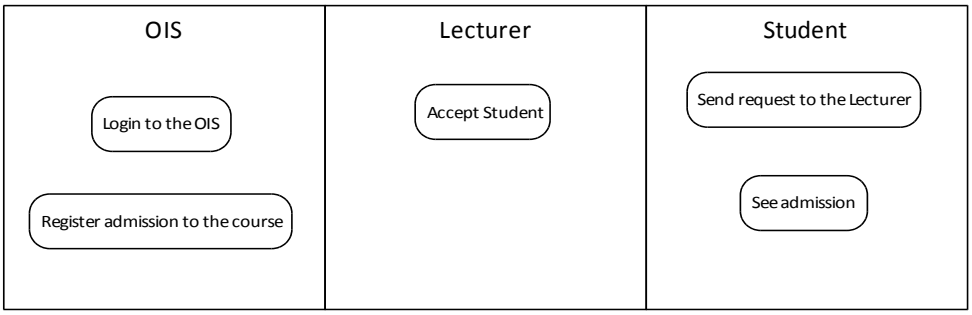


Figure 16: TR3, G1, NOTE1 (MUC-MAD)

G2. Activities should be connected by control flow. Order of activities can be captured from field *Extension points* in the each use case. If from 2 consecutive activities none is situated in the system swimlane- see NOTE2.

This guideline is based on the definition of the ‘Extension points’ field according to the (Kulak & Guiney, 2000). 'Extension points' field shows which use cases follow current (Figure 17).

NOTE2. *One activity situated in the system swimlane should be added in between. This activity can be taken from the field 'Basic path' in the one of the use case templates. Which activity to put, should be decided by developer.*

Mal-activity diagrams represent interaction of the system with the user. If two consecutive activities are not situated in the system swimlane, then it contradicts with idea of mal-activity diagrams, and we should add one system activity between them. In our case we add to the *OIS* swimlane *Proceed request* activity between *Send request to the Lecturer* and *Accept Student* (Figure 17). 'Proceed request' is taken from *Basic path* of the *Send request to the Lecturer* template (see Appendix C).

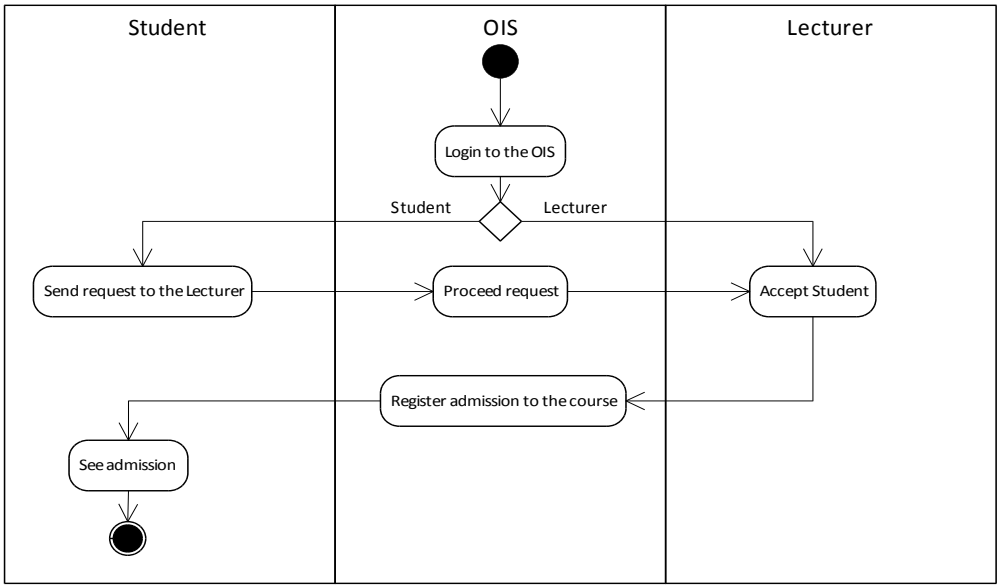


Figure 17: G2, NOTE2, G3 (MUC-MAD)

G3. To the mal-activity diagrams should be added initial and final activities. If use case doesn’t have extension points according to the textual template, then corresponding activity is last and one should add final activity after it.

Place of initial activity can't be decided automatically, since it should be defined logically from which user starts the process flow and who performs first activity. In our example first activity should be *Login to the OIS*, thus we place initial activity in the *OIS* swimlane. *See admission* template doesn't have extension points (see Appendix C), thus we place final activity after it (Figure 18).

TR4. Construct for representing *security criterion* (hexagon) in the misuse case diagrams corresponds to the construct for security criterion (call-out) in the mal-activity diagrams.

This rule based on the line D of the alignment Table 14. In both languages to the security criterion corresponds only one element and it is used for representation only of security criterion. Since admission is stored in the OIS, we place security criterion- *Integrity*

of admission, to the OIS swimlane (Figure 18).

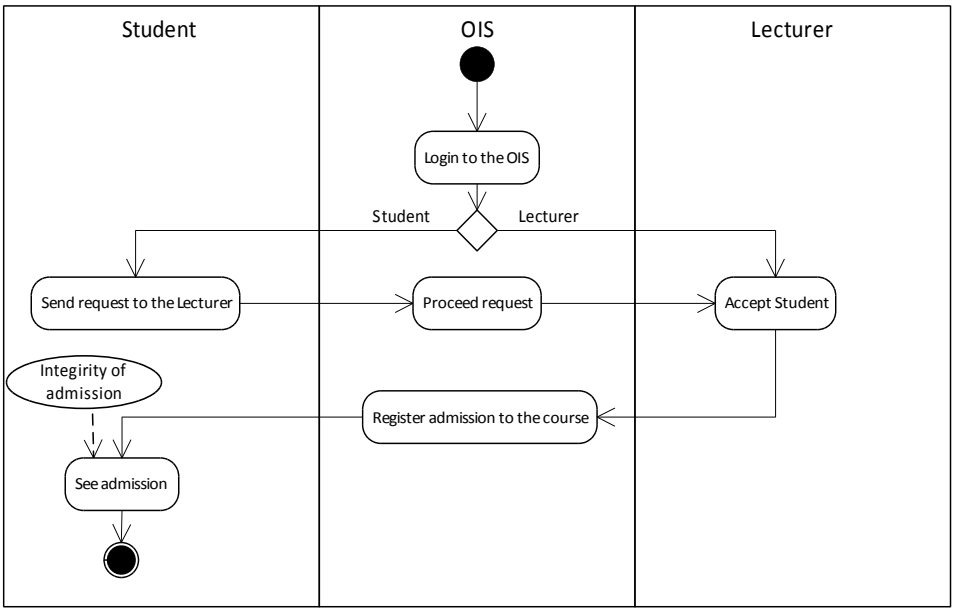


Figure 18: TR4, G4 (MUC-MAD)

G4. *Constraint of which asset* field in the security criterion template shows to which activities should be connected security criterion.

Explicit information regarding security criterion is provided in the security criterion template. Field *Constraint of which asset* is intended to show relation of security criterion to the business assets. Business assets in the mal-activity diagrams are presented by activities. According to the template (see Appendix C), we connect construct for security criterion in the mal-activity diagram to the *See admission* activity (Figure 18).
Transformation rules for ***risk-related concepts***:

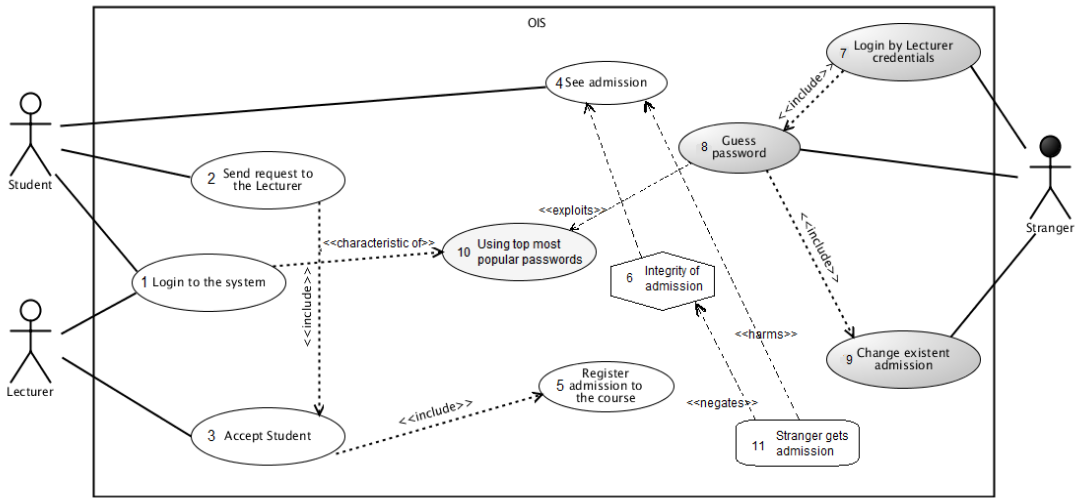


Figure 19: Misuse case diagram for registration to the courses process (risk-related constructs)

TR5. *Misuser* is translated in the *mal-swimlane*

According to the Table 14, in misuse cases to the threat agent corresponds *misuser* and in mal-activity diagrams- *mal-swimlane*. In the example, in the misuse cases there is one misuser- *Stranger* (Figure 19), so we add mal-swimlane in the mal-activity diagrams, with the same name (Figure 20).

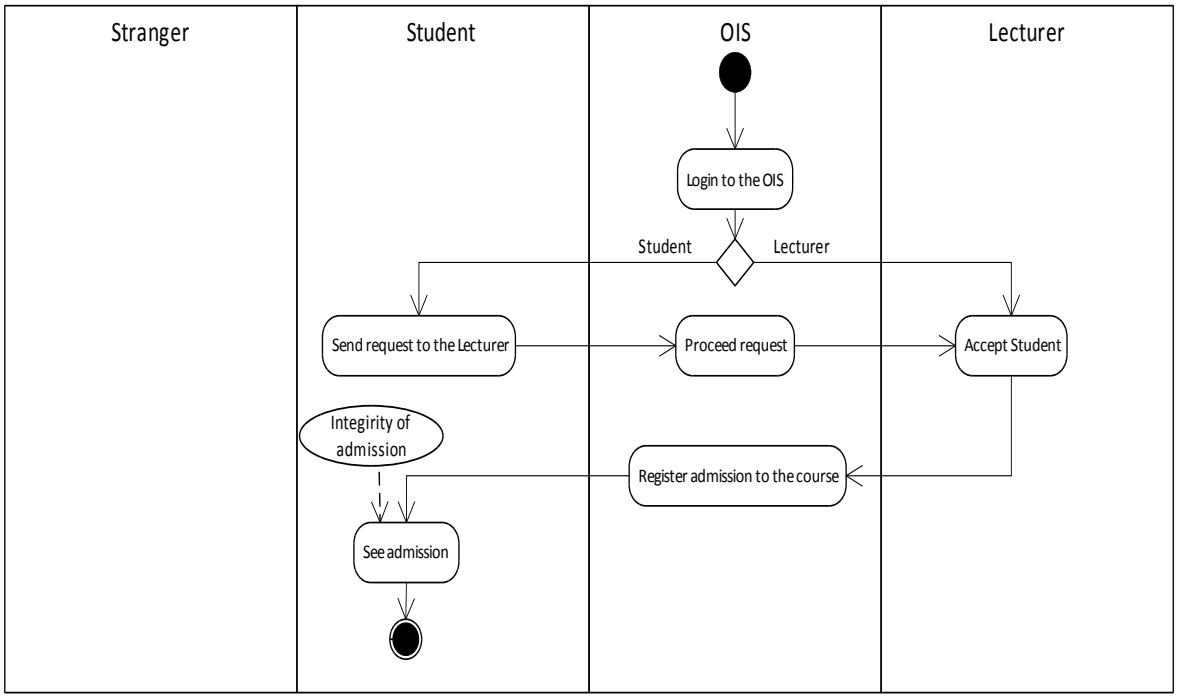


Figure 20: TR5 (MUC-MAD)

TR6. *Misuse cases* transformed into *mal-activities*.

According to the row - Attack method, line M (Table 14), to the *misuse case* corresponds *mal-activity*. Thus we place actions, which are performed by threat agent in the corresponding mal-swimlane. In the running example, misuse cases *Login by Lecturer credentials*, *Guess password*, *Change existent admission* are translated to the corresponding mal-activities, which are placed in the *Stranger* swimlane (Figure 21).

G5. Mal-activities should be connected using control flow. Order of mal-activities and their connection can be captured from the *Extension points* field of the misuse case template.

This guideline is based on the definition of the 'Extension points' field according to the (Kulak & Guiney, 2000). Thus after *Login by Lecturer credentials* placed *Guess password* and after- *Change existent admission* (Figure 21).

G6. If misuse case template has no extension points, it means that it is last mal-activity in the diagram and *final activity* should be added after it. Also *initial activity* should be added in the mal-swimlane before first mal-activity.

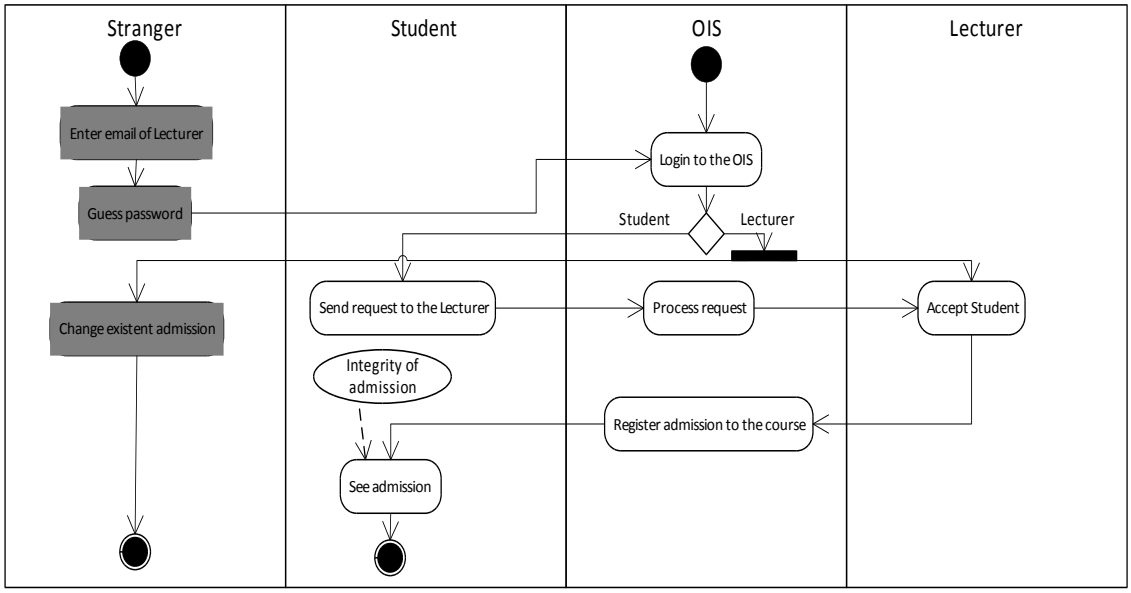


Figure 21: TR6, G5, G6 (MUC-MAD)

TR7. Constructs, which present *vulnerabilities* in misuse cases translated in constructs for vulnerabilities presentation in mal-activity diagrams.

This rule elicited, according to the alignment Table 14, line K. For vulnerability there are special constructs in misuse cases and mal-activity diagrams and they can represent only vulnerabilities (Figure 22).

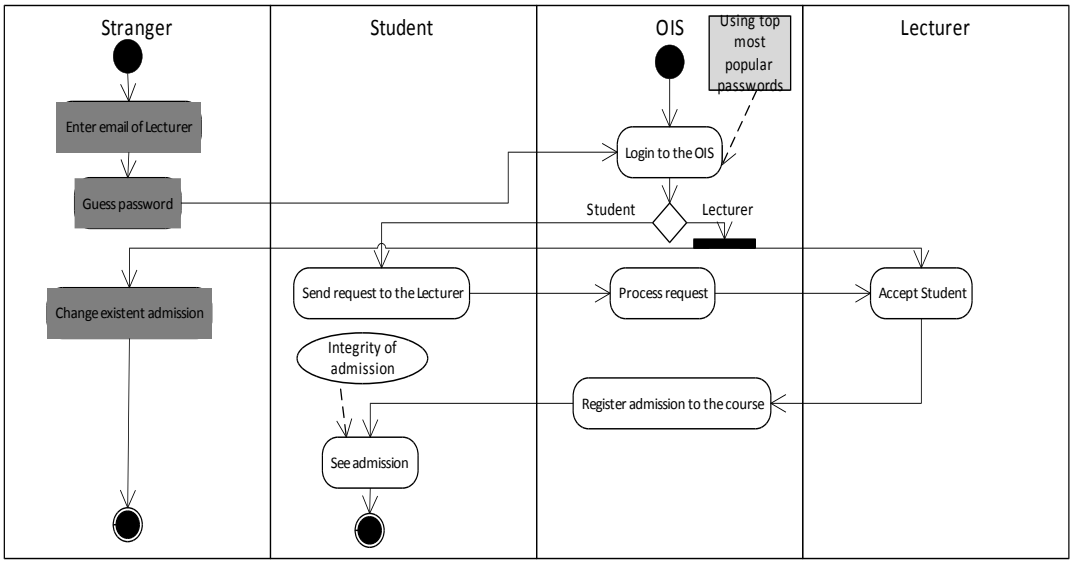


Figure 22: TR7, G7 (MUC-MAD)

G7. In vulnerability template *Characteristic of which asset* field reflects activity to which vulnerability should be connected.

Explicit information about vulnerability is presented in the vulnerability template. According to the definition of '*Characteristic of which asset*' field (see Chapter 4.3), it presents to which assets corresponds vulnerability. *Using top most popular passwords* corre-

sponds to the *Login to the OIS* asset. Thus we connect *Using top most popular passwords* to the *Login to the OIS* (Figure 22).

TR8. Construct for presenting *impact* in the misuse cases is translated into *mal-activity*.

According to the alignment Table 14, line H, element for representing impact is translated in the mal-activity (Figure 23).

G8. Mal-activity corresponding to the *impact* is placed after mal-activity mentioned in the field *Which event leads to impact* of the impact template.

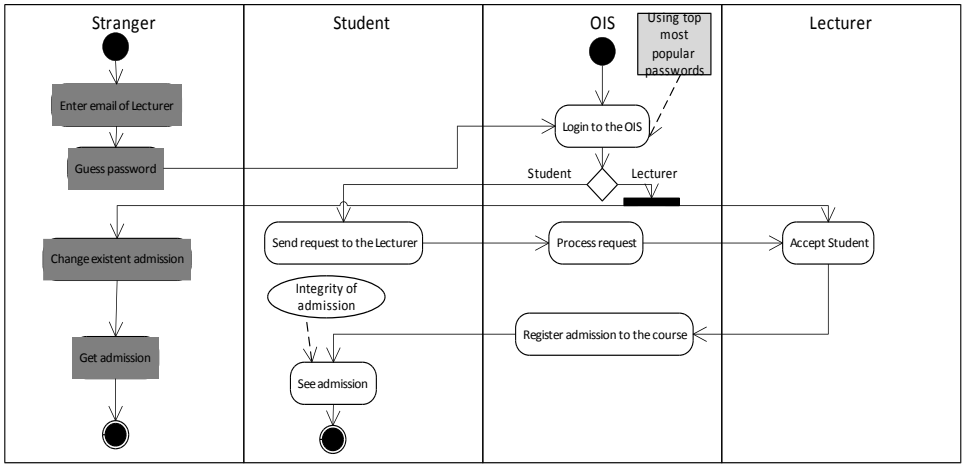


Figure 23: TR8, G8 (MUC-MAD)

Impact template presents detailed information about the impact and field '*Which event leads to impact*', according to the definition (see Chapter 4.3), presents event which causes impact. Event is presented by mal-activity, thus we add impact after *Change existent admission* mal-activity (Figure 23).

Rules for ***risk treatment-related concepts***:

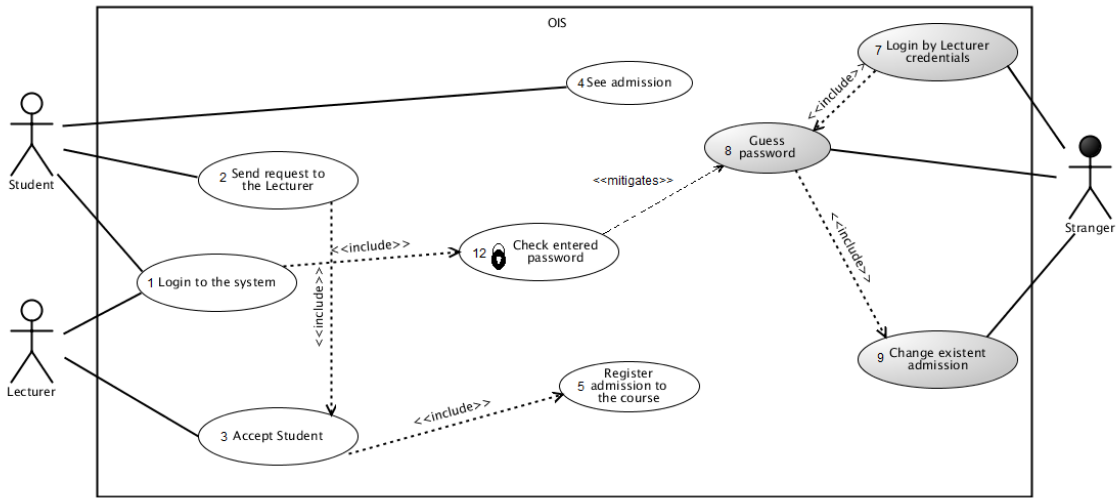


Figure 24: Misuse case diagram for registration to the course process (risk treatment-related concerns)

TR9. *Security use cases* are translated into *mitigation activities*, which are placed in the new security swimlane.

According to the alignment Table 14 (see line V), for *security requirement* in misuse cases corresponds *security use case* and in mal-activity diagrams- *mitigation activity*. In order to separate mitigation activities should be added new swimlane- *Security module*, where all mitigation activities will be placed. For our example, we add *Security module* swimlane and place *Check entered password* mitigation activity there (Figure 25).

G9. Mitigation activity is connected to the mal-activity which is stated in the *Extension points* field of the security use case template.

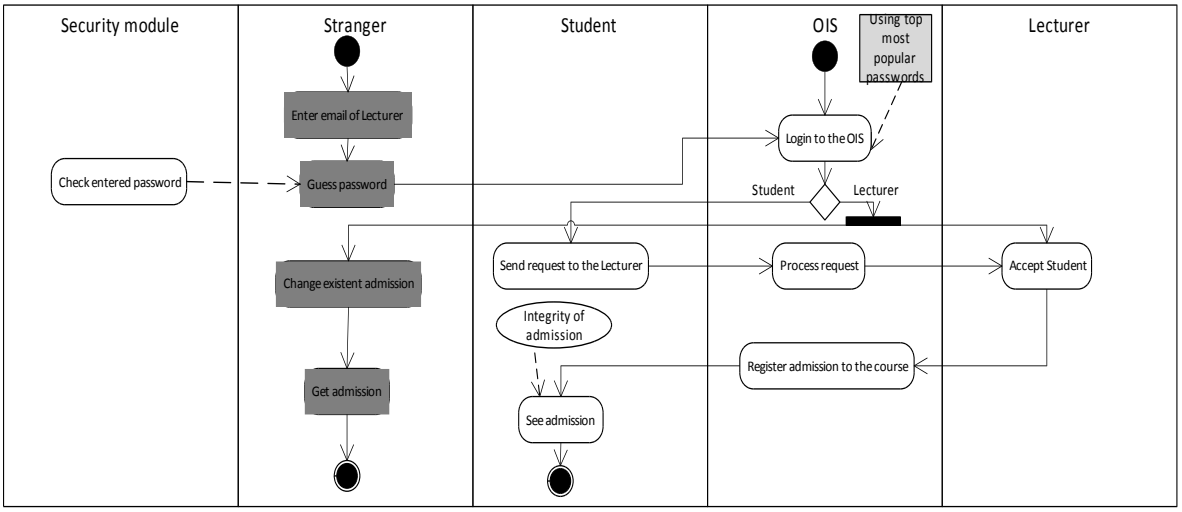


Figure 25: TR9, G9 (MUC-MAD)

Security use case presents detailed information about security requirement, field *Extension points* according to the definition (see Chapter 4.3) presents misuse case, which is mitigated by security requirement. Since misuse case is presented by corresponding mal-activity, we connect *Check entered password* security requirement with mitigation link to the *Guess password* mal-activity (Figure 25).

6.4 Summary

In this chapter we presented set of translation rules from misuse case diagrams to the mal-activity diagrams. There are 9 rules, 9 guidelines and 2 notes to transform asset-related, risk-related and risk treatment related concepts. *Rules* are for translating one exact element from misuse case diagram to another element in the mal-activity diagrams, and *guidelines* are used when information from templates is taken. Using textual templates for transformation allows capturing such information as place or order of activities, which can't be obtained only from misuse case diagrams. In some cases diagrams and templates still can't provide explicit information, needed for building mal-activity diagram. In this case developer should make decisions: in which order to place activities, which user starts the process and so on, based on the internal logic of the process.

7 Validation

In this chapter we will do validation of the results received during research presented in the thesis. Validation is needed in order to prove correctness, completeness or perceptiveness of the proposal. We are going to validate extensions to the misuse case template, which were presented in the chapter 4; extensions to the mal-activity diagrams notation, presented in the chapter 5; and set of transformation rules from misuse cases to the mal-activity diagrams, which are introduced in the chapter 6.

7.1 ***Misuse case template***

Design

Objective of validation is to prove that template is understandable with respect to the ISSRM domain model, meaning that ISSRM domain constructs can be captured from the template fields. Thus r*esearch question*, which will help to prove this, is: How much is understandability percent of misuse case template fields with respect to the ISSRM domain model? In order to answer research question we need to collect and analyze data. We use *direct method of data collection* (Wohlin *et al.*, 2012), which means direct interaction with participants in real time. In order to receive data, which will be analyzed we take one of the textual misuse case templates from the running example of the registration to the courses process (Appendix D). We took *Change existent admission template* (Table 15), it is chosen randomly. Template describes action of misuser, when he is changing existing admission in the system, by entering his data and saving new admission. Next we should choose participants for the experiment, it is crucial step, because it is influencing quality and validity of results. Participants should have background in IT and be acknowledged with domain. Thus we decided to take Master students, who are studying IT. Participants receive chosen template with explanation of the task. The task for participants is to define by which field particular ISSRM domain model construct is presented and to fill this information in the column 'Defined by which field'. Also they should fill information captured from the template to the column 'Information from template'. Answers which we expect to receive from participants are presented in the Table 16, values in the column 'Defined by which field' is taken from the final alignment table of the misuse case template after extensions- Table 6 and column 'Information from template' is taken from tabular representation of the example- Appendix B. After conducting of the experiment, received data should be analyzed. We are defining average percent of the correct answers for each construct with regard to the ISSRM domain model and then make conclusions based on these results.

Execution

In the experiment take part 5 people. Participants are provided with all needed literature before conducting experiment regarding the ISSRM domain model (Dubois *et al.*, 2010), misuse case template (Matulevičius *et al.*, 2008) and analysis presented in the chapter 4. Participants are students of the last year of master programs in Applied Mathematics and Software Security from the National University of Ukraine “Kyiv Polytechnic Institute” (Kyiv, Ukraine). Time for providing answers is unlimited, participants are allowed to use all possible resources and they are not observed by experimenter.

Table 15: Misuse case template extensions validation. Asset-related constructs

№		1	2	3	4	5	%
Business asset	s	100	75	75	0	100	**70**
	v	100	100	75	0	75	**70**
IS asset	s	75	50	25	50	25	**45**
	v	0	50	50	25	0	**25**
Security criterion	s	100	75	100	100	0	**75**
	v	100	75	100	100	75	**90**

Table 16: Misuse case template extensions validation. Risk-related constructs

№		1	2	3	4	5	%
Risk	s	50	50	100	25	75	**60**
	v	50	100	75	25	100	**70**
Impact	s	100	100	75	75	25	**75**
	v	75	100	50	25	25	**55**
Event	s	75	100	50	75	25	**65**
	v	50	100	75	75	100	**80**
Vulnerability	s	100	75	75	100	50	**80**
	v	100	25	75	50	50	**70**
Threat	s	100	75	0	25	50	**50**
	v	100	75	25	0	25	**45**
Threat agent	s	100	100	100	75	50	**85**
	v	75	100	50	100	75	**80**
Attack method	s	75	75	75	0	50	**45**
	v	50	25	75	0	100	**50**

Table 17: Misuse case template extensions validation. Risk treatment-related constructs

№		1	2	3	4	5	%
Risk treatment	s	75	100	100	50	100	**85**
	v	100	100	100	50	100	**90**
Security requirement	s	100	100	75	50	75	**90**
	v	75	50	25	100	0	**50**
Control	s	-	-	-	-	-	**-**
	v	-	-	-	-	-	**-**

Analysis

Results received from the participants are presented in the Appendix H. We are going to analyze percentage of the correctly provided answers. Results of the evaluation are displayed in the Tables 15, 16, 17, they show correctness of each field, filled by particular participant for asset, risk and risk treatment set of constructs. Correct answer with minor inaccuracies receive 100 percent. 25, 50, 75 states for percentage of correctly provided in-

formation. 25 is given when there is correct information provided, but less than half. 50 states for half correct information provided, other half can be wrong information or incomplete answer. 75 is given, when more than half of correct information is provided and just few things are missing. If answer is totally wrong it is- 0. After evaluation we received following results: for asset-related visual- 61%, semantics- 63%; for risk-related visual- 65%, semantics- 66%; for risk treatment -related visual- 70%, semantics- 88%. In the chapter 4, we added fields for security criterion and risk treatment to the misuse case textual template. For both we got big understandability result, both for which field represent particular construct and name of the construct.

Threats to validity

Validity of results may have been influenced by the fact that experiment was conducted on the people with the same background. Also experiment was conducted with small number of participants, only 5 people took part, it is caused by lack of time. And in future it would be good idea to perform the same experiment with bigger number of people with different background.

Conclusions

Validation have shown that experiment was successful and participants were able to capture ISSRM domain model constructs from the template. This means that extensions are beneficial and templates can be used for modeling security threats and their mitigation.

7.2 ***Mal-activity diagrams extensions***

Design

In order to prove that extensions to the mal-activity diagrams, proposed in the chapter 5, are understandable and can be used in the real-life projects, we need to perform a validation. Thus the *objective* of our experiment is to evaluate perception of the mal-activity diagram extensions. In order to fulfill defined objective we formulate such *research questions*:

1. How much is understandability percent for semantics of mal-activity diagrams with respect to the ISSRM domain model?
2. How much is understandability percent of mal-activity diagrams visual constructs with respect to the ISSRM domain model?

In order to answer research questions, relevant data should be collected and analyzed. In our case we have chosen *direct method of data collection* (Wohlin *et al.*, 2012), which means that researchers are in the direct contact with participants and data collection is performed in the real time. To obtain needed data we develop three mal-activity diagrams for registration process, one diagram for each set of ISSRM domain model constructs: asset, risk and risk treatment. Diagrams, which were given to the participants are presented in the Appendix F. Also we have 3 tables, which participants should fill. In the tables in the Appendix F correct answers are presented, which we expect to receive from the participants. It is important to choose appropriate participants, who have background in IT and basic knowledge in security and modeling languages. Choosing appropriate participants is crucial for experiment, because if not correct group of participants is chosen, it will negatively influence results of the experiment. We decided to involve students, who studies IT and are potential security experts. Students are provided with diagrams and asked to fill 3 tables (Appendix F). When participants performed task we need to analyze received data, in order to answer research questions. We calculate average percent of correct answers from all participants for particular semantic or visual ISSRM domain model construct rep-

resentation. After that we make conclusions regarding particular ISSRM domain model construct or set of constructs.

Execution

Case study is performed by 5 IT-related people, last year students of Master programs in Applied Mathematics and Software Security, potential users of the mal-activity diagrams and potential security experts. These Master programs are taught in the National University of Ukraine "Kyiv Polytechnic Institute" (Kyiv, Ukraine). Before experiment participants were provided with required literature on the ISSRM domain model- (Dubois *et al.*, 2010), on the mal-activity diagrams – (Chowdhury *et al.*, 2012) and extensions proposed in the thesis, so chapter 5. Then they receive an explanation of task and were provided with experiment data sheets. The task is to find accordance to definitions from the ISSRM domain model in the diagrams and specify by which language constructs these elements are represented. Students had 1,5 hour to provide solutions and could use any available material. Participants were not observed by experimentator.

Analysis

Results received from the participants are presented in the Appendix I. We evaluate these results and present correctness of each field, filled by particular participant, in the Tables 18, 19, 20. Correct answer with minor inaccuracies receive 100 percent. 25, 50, 75 states for percentage of correctly provided information. 25 is given when there is correct information provided, but less than half. 50 states for half correct information provided, other half can be wrong information or incomplete answer. 75 is given, when more than half of correct information is provided and just few things are missing. If answer is totally wrong it is- 0. For each set of constructs we received both answers regarding semantics and visual representation (in the tables s-semantics, v-visual). Last columns in Tables 18, 19 and 20 present understandability for each element, which is characterized by average percent of correct answers of participants. In total for asset model understandability percent of semantics is 67%, and for visual- 52%. For risk model 60% and 57% for semantics and visual correspondingly. Risk treatment model shows 47% understandability for semantics and 57% for visual.

Table 18: Mal-ativity diagrams extensions validations. Asset-related results

№		**1**	**2**	**3**	**4**	**5**	**%**
Business asset	s	100	75	50	100	25	**75**
	v	25	75	50	25	50	**45**
IS asset	s	100	75	50	75	25	**65**
	v	100	75	50	0	25	**50**
Security criterion	s	100	100	50	50	0	**60**
	v	100	100	50	50	0	**60**

Threats to validity

One of the important factors, which influenced results of experiment, was *small number of participants*. Another factor which may have influenced an experiment can be, that experiment was conducted on the students. They are just in the process of studying domain and are not security experts. Thus, if student didn't understand some construct or didn't have time to read given literature at all, it highly influence the results. Another threat to validity can be such called *construct validity* (Wohlin *et al*., 2012). It means misunder-

standing between researches and participants. We have mentioned that participants had difficulties with defining visual representation for the elements, from the provided answers we assume that it is caused by not clear formulation of question. So participants were not able to understand what kind of information they should provide in the visual representation column.

Table 19: Mal-activity diagrams validation. Risk-related results

№		1	2	3	4	5	%
Risk	s	50	50	25	75	0	**40**
	v	75	75	50	50	0	**50**
Impact	s	50	0	75	100	25	**50**
	v	50	0	25	100	0	**35**
Event	s	100	100	50	75	0	**65**
	v	75	75	50	75	0	**55**
Vulnerability	s	100	75	100	50	75	**80**
	v	100	75	75	50	100	**80**
Threat	s	100	50	50	75	100	**65**
	v	50	75	50	75	100	**75**
Threat agent	s	100	100	0	0	100	**60**
	v	100	75	25	25	100	**65**
Attack method	s	50	0	50	75	100	**55**
	v	0	0	50	50	100	**40**

Table 20: Mal-activity diagrams validation. Risk treatment-related results

№		1	2	3	4	5	%
Risk treatment	s	100	50	100	75	100	**85**
	v	-	-	-	-	-	**-**
Security requirement	s	100	75	100	100	0	**75**
	v	100	75	100	75	0	**70**
Control	s	0	75	75	100	0	**50**
	v	0	75	50	100	0	**45**

Conclusions

We extended notation of mal-activity diagrams in the thesis with constructs for security requirements and vulnerability. Security requirement understandability level is high, especially for semantics (semantics- 75%, visual- 70%). Also for vulnerability understandability percent is high (semantics- 80%, visual- 80%). For some elements understandability percent is low and we mentioned some factors, which may have influenced this. For impact it caused difficulties to define visual representation, understandability is only 35%. We assume that it is because the same visual representation is used for the impact and attack method. Also we observed that participants had problems defining visual representation for ISSRM domain constructs, which can be presented by several elements

in the mal-activity diagrams. For instance, business asset, which according to the Chapter 5.2 can be presented by activities, decisions and control flow, is presented only by activities in our case and it caused confusion of participants, so percent of correct answers is quite low- 45%. In general experiment was successful and participants could capture elements with regard to the ISSRM domain model. Nevertheless, we made a conclusion, that in future, such experiment should be improved, by involving bigger number of participants and with people who are more acknowledged with the domain.

7.3 ***Transformation rules: misuse case diagrams - mal-activity diagrams***

In this chapter we are going to validate transformation rules, which are presented in the chapter 6. We will evaluate quality of the mal-activity model received during transformation (Figure 26), in comparison with mal-activity model (Figure 10) designed from example description (Appendix A,B).

Evaluation method is based on the approach presented in the (Matulevičius *et al.*, 2011), 4-step process is shown in Figure 26 First step is defining evaluation goals. Then we define quality evaluation criteria. After evaluation criteria is defined we use diagrams, designed in the thesis to perform evaluation. Based on the evaluation results we perform analysis and conclusions.

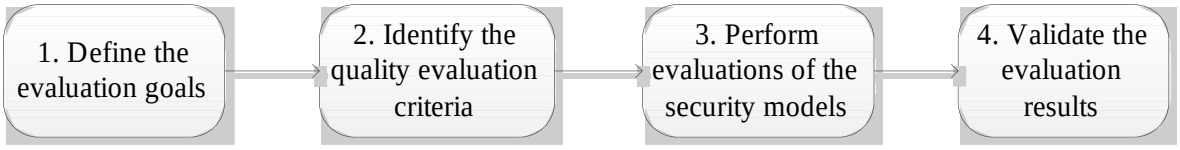


Figure 26: Evaluation of models quality (Matulevičius et al., 2011)

Evaluation goals

According to the (Matulevičius *et al.*, 2011), assessment goals can be understanding of the security problems, defining scope and quality of the security models, and comparison of models with respect to evaluation criteria. We want to learn quality of model developed during transformation from misuse cases to mal-activity diagrams, comparing it to the "ideal" model developed according to the text description of the example. Thus we focus evaluation on the comparison of models regarding quality criteria, which will be defined on the next step. This will help us to make a conclusions regarding value and correctness of the model developed using transformation rules.

Quality evaluation criteria

In this research we are focusing on the two quality types, which are syntax and semantics. Hence we chose 4 assessment criteria, which are *semantic completeness* and *correctness*, *syntactic completeness* and *validity. Semantic quality* defines correspondence between diagram and its semantic domain, which is ISSRM domain model in our case. *Semantic completeness* means how many concepts model includes from the domain, so coverage percent of ISSRM domain model should be calculated. It is calculated as number of ISSRM domain model concepts presented in the diagram divided by total number of ISSRM domain model concepts. *Semantic correctness* means security-related concerns presented in the diagram in comparison to the data-related concerns presented, thus percentage of the security-related statements should be defined. It will be defined as number of security-related concepts presented in the diagram divided by number security-related concepts plus data-related concepts. *Syntactic quality* defines correspondence between diagram and modeling language, which is mal-activity diagrams in our case. *Syntactic validity* is defined

by number of syntactically invalid statements and means that grammatical expressions used in the model should be a part of the modeling language. If number of syntactically invalid statements is high, it means syntactical validity is low. *Syntactic completeness* is defined by syntactically incomplete statements and means that all grammatical expressions are complete and present in the model. We calculate number of syntactically incomplete statements and if it is high, then syntactical completeness is low.

Table 21: Semantic completeness results in the transformation rules validation process

ISSRM domain model	**Mal-activity model**	**Result of transformation**
Asset-related constructs	100%	100%
Risk-related constructs	65%	58%
Risk treatment-related constructs	50%	50%

Table 22: Evaluation result of the quality criteria in the transformation rules validation process

Quality type	Quality criteria	"Ideal" mal-activity model	Mal-activity model received from transformation
Semantic	Semantic completeness	72%	69%
	Semantic correctness	62%	58%
Syntax	Syntactic validity	0	0
	Syntactic completeness	0	1

Evaluation

First, we evaluate diagrams regarding *semantic completeness*. In the Table 21 percent of semantic quality for each set of concepts regarding ISSRM domain model is presented. And in the Table 22 average percents of semantic completeness for each diagram are presented. We can see that asset-related constructs have 100% coverage in both diagrams. Also we can observe that both diagrams have the same coverage percent for risk treatment set of constructs, and this percent of coverage influenced by modeling language, since it the maximum coverage for risk treatment set of constructs in the mal-activity diagrams modeling language. Semantic completeness differs only for risk-related set of constructs. It is caused by the fact that *targets* relationship is not presented in the transformed diagram, in particular, *Change existent admission* attack method is not influencing any IS asset, according to transformed diagram.

Second, we are analyzing *semantic correctness* of the diagrams. We should define security-related and data-related statements. In the initial mal-activity diagram we defined 9 data-related and 15 security-related entries, thus semantic correctness for this diagram is 62%. In the diagram, received as a result of transformation there are 8 data-related entries and 11- security-related, which gives 58% semantic correctness.

Next, we should define *syntactic validity* and *syntactic completeness.* Received after transformation model has one incomplete construct, which is risk mitigation. It is not understandable how this mitigation should be implemented, it should be explicitly presented on the diagram how mitigation is integrated in the system. But both models are syntactically valid, meaning that they contain only expressions, which are part of the mal-activity diagrams syntax.

Threats to validity

In this case external validity could have been influenced, because we considered only one scenario in the validation, so applying transformation rules to other cases can give different results. Nevertheless, scenario, which is presented in the thesis is close to the real life and possible in boundaries of university, we assume that results will not crucially change for different scenarios.

Also validation was conducted by the same people, who developed transformation rules and built “ideal” mal-activity diagram, it possibly influenced internal validity of the validation. But we were objective as possible and tried to minimize such risk.

Conclusions

In the Table 22 results of the validation are presented. Syntactic quality of the transformed model is high, since we didn't find any incomplete or invalid statements. But semantic completeness and correctness are a lower than in initial diagram. In general we assume that quality of the transformed model is high and transformation rules can be used for transformation from misuse cases to the mal-activity diagrams. Potentially transformation rules can be improved by revising relationships presentation in the diagram and reducing number of data-related constructs, concentrating on the security-related entries.

7.4 Summary

In this chapter we provided validation of thesis contribution. Research showed that transformation rules provide mal-activity diagrams with high quality, potentially they can be improved by paying more attention to the semantics of language. Nevertheless proposed extensions are understandable, transformation rules provide qualitative diagrams, which can be used in real-life projects.

8 Conclusions

In this chapter we will summarize what was done in the thesis, list limitations of the work. Also we will provide conclusions and give ideas for future work.

8.1 Limitations

This work as any other research has some limitations. First of them is that validation of the mal-activity diagrams and misuse case template extensions were performed only by 5 participants. Also in both experiments people were with the same background, this may influenced results of validation. When all participants of the experiment have the same level of domain knowledge it increases chances that we will get similar answers. So participants should be with different backgrounds and from different work fields.

Another limitation is that in the thesis research was performed only by one person and reviewed by another. If mal-activity diagrams and misuse cases would be analyzed and reviewed by other people it can show problem from another perspective and lead to another results. The same limitation has running example, we decided to choose risk reduction for reducing possible risks, but other people can choose risk avoidance or risk transfer and process will have different security requirements.

One more limitation is that transformation rules are illustrated and analyzed regarding one process. If transformation rules would be illustrated on the another process it could show some other aspects and minor changes in the proposed transformation rules will be needed.

8.2 Conclusions

In the thesis we have chosen ISSRM domain model for comparison and analysis of the security modeling languages. Main reason why it was chosen, it is because previously research was done using this domain model, in particular such security modeling languages as mal-activity diagrams (Chowdhury *et al.*, 2012), Secure Tropos (Matulevičius *et al.*, 2008A), misuse cases (Matulevičius *et al.*, 2008) were already aligned to the ISSRM domain model.

During comparison of the security modeling languages we have chosen 2 of them for further research in the thesis- mal-activity diagrams and misuse cases. They were chosen because both languages are specially intended for presentation of security concerns, both languages were extended from standard UML diagrams (activity and use case diagrams). They are intended for representing security problems from different viewpoints, so it is beneficial to combine these languages in the approach, which allows presenting security problem from different viewpoints. Both languages were aligned to the ISSRM domain model, but mal-activity diagrams and misuse case textual templates need extensions in order to cover ISSRM domain model. Thus, in the thesis we extended misuse case textual template with such fields as security criterion and risk treatment, which reflects corresponding constructs in the ISSRM domain model. Also we added templates for security criterion, impact, vulnerability and security use cases. These templates allow understanding complete picture of security risks from the textual templates. Next, we proposed extensions to the mal-activity diagrams, in particular we defined elements for vulnerability and security criterion representation. Proposed extensions to mal-activity diagrams and misuse case textual templates allow covering ISSRM domain model, thus we can analyze both lan-

guages with regard to the ISSRM domain model.

We elicited transformation rules, which are based on the misuse case diagrams and complemented them with transformation guidelines, based on the misuse case textual template. Using textual template helped to define order and places of activities in the mal-activity diagrams, what is not possible to capture from the misuse case diagrams. In result we received 9 transformation rules and 9 guidelines, which allows to transform misuse case diagram to mal-activity diagram. Part of transformation can be applied automatically, but minor contribution of developer is still needed, because not all information, which is needed to built mal-activity diagram can be captured from misuse case diagrams and templates.

In order to prove that extensions to the mal-activity diagrams and to misuse case textual templates are understandable and transformation rules provide qualitative mal-activity diagrams, we made a validation. Extensions validation was made in the form of experiment, in both cases we developed case study, gave task to participants and evaluated received results. Validation showed that extensions are understandable and participants of experiments could capture information with regard to the ISSRM domain model. Transformation rules we validated by evaluating quality of the received after transformations mal-activity diagrams in comparison with "ideal" diagram, created manually by running example description. We researched syntactic and semantic quality of both diagrams. Diagrams created during transformation has less percentage in the semantic completeness and correctness, than initial models. Minor difference in the evaluation of initial and received model are not influencing information value, meaning that transformed model can represent the same information as initial one. We assume that transformation is successful and transformation rules can be used by developers in order to present security threats from different viewpoints.

8.3 Future work

This thesis is part of a big project of analyzing security-oriented modeling languages with regard to the ISSRM domain model. In this work we have shown how transformation from misuse case diagrams to the mal-activity diagrams can be performed. We elicited transformation rules, using which misuse case diagrams with help of misuse case templates can be translated to the mal-activity diagrams. However we didn't analyze reverse transformation, from mal-activities to the misuse cases. Also transformation for other combination of languages, such as Secure Tropos (Matulevičius *et al.*, 2008A), EPC (Haterbur & Heisel, 2005), BPMN (Altuhova *et al*, 2012), KAOS extensions to security (Lamsweerde, 2004) are left out of scope. Thus next steps would be analyzing different combination of the security-oriented modeling languages in order to elicit transformation rules from one language to another. Especially, transformation from mal-activity to misuse case diagrams is needed, thus reverse transformation to one, developed in the thesis.

Big contribution would be developing automation tool, which will include currently existing extensions to the security-oriented modeling languages. Also into this tool can be included functionality for partial transformation from one modeling language to another.

Rezümee

Turvalisuse riskihaldus kasutades väärkasutamise juhtumeid ja pahatahtliku tegevuse diagramme

Turvaliste infosüsteemide turvalisuse muresid saab arendamise ajal adresseerida erinevatel arendamise etappidel (näiteks nõuete koostamisel, süsteemi ja tarkvara disainimisel ning implementeerimisel). Turvalisuse analüüsi on võimalik teha kasutades erinevaid selleks mõeldud mudelleerimiskeeli (nagu Secure Tropos, väärkasutamise juhtumid, pahatahtliku tegevuse diagrammid), mis lubavad arendajatel väljendada olulisi probleeme erinevates perspektiivides. Kuigi kõigil keeltel on omad tugevused, siis erinevate perspektiivide üheks sidusaks ja kokkusobivaks mudeliks kombineerimine on jätkuvalt väljakutseid esitav tegevus.

Teesides keskendume me kahe mudelleerimiskeelele, mida kutsutakse väärkasutusjuhtumid (nii diagrammid kui ka tekstilised mallid) ja pahatahtliku tegevuse diagrammid. Need erinevad arenduse perspektiivi poolest, kuid potentsiaalselt saab mõlemaid kasutada süsteemi ja tarkvara disainimise etappides, analüüsis ja turvanõuete dokumenteerimises. Eelmises uuringus uuriti neid mõlemaid keeli seoses infosüsteemide turvariskide halduse domeenimudeliga (ISSRM), mis määrab oluliste ja väärtuslike varade süstemaatilise identifitseerimise protsessi, nende turvariski ja tutvustab turvanõudeid riskide vähendamiseks. Kuid eelmises töös ainult väärkasutuse diagrammid olid täielikult laiendatud seoses ISSRM domeenimudeliga.

Käesoleva magistritöö eesmärk on defineerida põhjalikult meetod, mis hõlbustaks väärkasutamise juhtude tranformatsiooni pahatahtlikuteks tegevusteks. Tuginedes eelpool mainitud uuringutele me esiteks laiendame antud magistritöös väärkasutamise juhtumeid ja pahatahtliku tegevuse diagramme nii, et nad kataksid ISSRM domeeni mudeli kontseptsiooni. Järgmiseks tuginedes nendele laiendustele me tutvustame transformatsiooni reeglite kogumikku, mis juhendab koos väärkasutus juhtude mallidest pahatahtliku tegevuse diagrammideks abiga väärkasutamise juhtude diagrammide tõlkimist.

Me valideerime oma ettepanekut mudeli kvaliteedi analüüsi juhtumi uuringuga. Me loodame, et käesolev magistritöö aitab süsteemi ja tarkvara arendajatel integreerida kaks mudelleerimise vaatenurka, et tabada turvanõuded ja süstemaatiliselt arendada ning rakendada neid süsteemi disainis, aidates luua turvalist infosüsteemi.

References

1. Altuhova O., Matulevičius R., Ahmed N., 2012, Towards Definition of Secure Business Processes, CAiSE Workshops, vol.112 of Lecture Notes in Business Information Processing, p.1-15, Springer (2012).
2. Braber F., Braendeland G., Dahl H., 2006, The CORAS model-based method for Security Risk Analysis, SINTEF, Oslo.
3. Chowdhury M., 2011, Modeling security risks at the system design stage (Alignment of the Mal-Activity Diagrams and SecureUML to the ISSRM domain model), Master thesis, Tartu, 2011.
4. Chowdhury M., Matulevičius R., Sindre G., Karpati P., 2012, Aligning Mal-activity Diagrams and Security Risk Management for Security Requirements Definitions, B. Regnell and D. Damian (Eds.): REFSQ 2012, LNCS 7195, pp. 132–139, Springer-Verlag Berlin Heidelberg (2012).
5. CNN, 2012, http://edition.cnn.com/2012/10/25/tech/web/worst-passwords-2012, access date= 09.04.2013.
6. Cockburn A, 2001, Writing effective use cases. Addison-Wesley, Boston.
7. Dardenne A., Lamsweerde A., Fickas S., "GoalDirected Requirements Acquisition", Science of Computer Programming, Vol. 20, 1993, 3-50
8. DCSSI, 2010, EBIOS-Expression of Needs and Identification of Security Objectives, http://www.ssi.gouv.fr/en/the-anssi/publications-109/methods-to-achieve-iss/ebios-2010-expression-of-needs-and-identification-of-security-objectives.html , France.
9. Dubois, E., Heymans, P., Mayer, N. & Matulevičius, R., 2010. A Systematic Approach to Define the Domain of Information System Security Risk Management. In: S. Nurcan, C. Salinesi, C. Souveyet & J. Ralyte, eds. Intentional Perspectives on Information Systems Engineering. s.l.: Springer-Verlag, pp. 289-306.
10. El-Attar M., 2012, From misuse cases to mal-activity diagrams: bridging the gap between functional security analysis and design, Software and Systems Modeling, March 2012, Springer-Verlag.
11. Giorgini P., Mouratidis H., Zannone N., 2007, Modeling Security and Trust with Secure Tropos, Information Science Publishing, p.161-189**.**
12. Hatebur D.,Heisel M., 2005, Problem Frames and Architectures for Security Problems, International Conference on Computer Safety, Reliability and Security-SAFECOMP, pp. 390-404, Springer Berlin Heidelberg (2005).
13. Jackson M., Problem Frames, Analyzing and structuring software development problems, Addison-Wesley (2001).
14. Jürjens, J., 2002, UMLsec: Extending UML for Secure Systems Development. In: Jezequel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp.412–425. Springer, Heidelberg (2002).
15. Kulak D, Guiney E, 2000, Use cases: requirements in context. SACM Press, New York.
16. Lamsweerde A., 2004, Elaborating Security Requirements by Construction of Intentional Anti-Models, In: Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), pp. 148-157.
17. Lin, L., Nuseibeh, B., Ince, D., Jackson, M., 2004, 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, Los Alamitos.
18. Lodderstedt, T., Basin, D.A., Doser, J., 2002, SecureUML: A UML-based Modeling Language for Model-driven Security. In: J´ez´equel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg.
19. Mayer N, 2009, Model-based Management of Information Security Risk Management, Doctoral thesis for the degree of Doctor of Science.
20. Matulevičius, R., Mayer, N., Heymans, P., 2008, Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Development, In: CAiSE '08 Proceedings of the 20th international conference on Advanced Information Systems Engineering, pp.541-555, Springer-Verlag.
21. Matulevičius, R., Mayer, N., Heymans, P., 2008, 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.(A)
22. Matulevičius R., Lakk H., Lepmets M., 2011, "An approach to assess and compare quality of security models". ComSIS, Volume 8 No 2, Special Issue, (2011), pp.447-476.
23. Mead N.R., Hough E., Stehney T., 2005, "Security Quality Requirements Engineering (SQUARE) Methodology", (CMU/SEI-2005-TR-009), Software Engineering Institute, Carnegie Mellon University.
24. Mouratidis H., Giorgini P., 2007, Secure TROPOS: a Security-Oriented Extension of the TROPOS Metodology, International Journal of Software Engineering and Knowledge Engineering (IJSEKE), vol. 17, no. 2, pp. 285-309.
25. Royce W., Managing the Development of Large Software Systems, ICSE '87 Proceedings of the 9th international conference on Software Engineering, pp.328-338, IEEE Computer Society Press Los Alamitos, 1987.
26. Sindre, G., Opdahl, A., 2001, Templates for Misuse case description, Proceedings of the 7th International Workshop on Requirements Engineering, Foundation for Software Quality (REFSQ'2001).
27. Sindre, G., Opdahl, A., 2005, L. Eliciting Security Requirements with Misuse Cases. Requirements, Eng. 10 (1), Springer-Verlag, 2005.
28. Soomro I., Ahmed N., 2012, Towards Security Risk-Oriented Misuse Cases.
29. Sindre G., "Mal-Activity Diagrams for Capturing Attacks on Business Processes". In proceedings of the Working Conference on Requirements Engineering: Foundation for Software Quality, 2007.
30. UML spec., 2011, http://www.omg.org/spec/UML/2.4.1/ , access date= 17.04.2013
31. White S., Introduction to BPMN, IBM Corporation, 2004.
32. Wohlin C., Runeson P., Hoest M., Ohlsson M.C., Regnell B., Wessln A., Experimentation in Software Engineering, Springer-Verlag Berlin Heidelberg, 2012.

Appendices

A. Textual description

Here we apply 6-step ISSRM process for explicit textual presentation of the running example:

1. Assets definition

In order to register for the course, student registers in the OIS and logins using email and password. Next he sends request to the lecturer. Lecturer registers and logins to the OIS, using email and password. Lecturer accepts student to the course. OIS (=system) registers admission to the course and student can view the admission. From described process, IS assets are: OIS, email, password, Login to the OIS, admission. Business assets are: Send request to Lecturer, Accept student. And to both IS and business assets belong: See admission, Register admission to the course.

2. Security objectives determination

Based on defined assets we assume that admission should not be possible to modify. So security objective, which should be achieved, is integrity of admission.

3. Risk analysis

One of the possible risks, is that attacker, some stranger outside the system, wants to register himself to the course. Stranger knows that in order to add himself to the course he needs to sign into OIS using email and password of the lecturer. We assume that stranger knows email lecturer, because it's not confidential and it is easy to obtain from internet or deans office. After that he is trying enumeration of the top most popular passwords in order to sign into the OIS. In the Table 23 as an example, listed top 25 most famous passwords in 2012, from CNN (2012), but also it can be found bigger list 50 or 100 top most popular passwords. So when password is guessed, stranger logins into OIS by lecturer credentials. After that he changes existent admission of one of the students, as a result stranger gets the admission.

4. Risk treatment decision

In this case a solution will be risk reduction.

5. Security requirements

Security requirements, which is needed to mitigate defined risk is: check entered password during registration. Password should contain capital letters, special characters and numbers. Moreover password should be not less than 6 characters. So during registration password should be verified for satisfying these rules and if it is not strong, it should be requested to enter new password.

6. Control selection and implementation

Usually security requirements are implemented on the last stage of IS development, but if security risk analysis is done after the business process modeling it can lead to the changing whole system from the beginning which result in losing a lot of time and money.

Table 23: Top 25 most popular passwords in the world in 2012 (CNN, 2012)

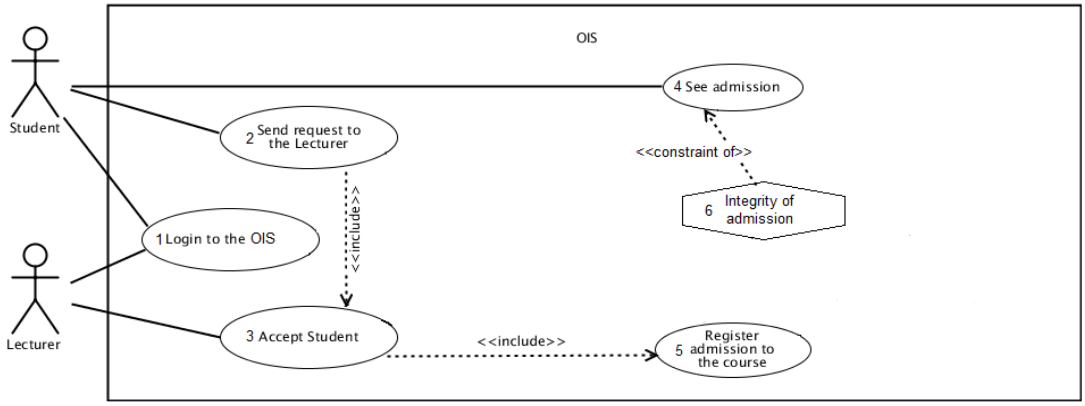
1	password	14	sunshine
2	123456	15	master
3	12345678	16	123123
4	abc123	17	welcome
5	qwerty	18	shadow
6	monkey	19	ashley
7	letmein	20	football
8	dragon	21	jesus
9	111111	22	michael
10	baseball	23	ninja
11	iloveyou	24	mustang
12	trustno1	25	password1
13	1234567		

B. Tabular presentation

Table 24: Tabular presentation of the running example

ISSRM domain model		*Misuse cases*
Asset	Asset	See admission, Register admission to the course
	IS asset	OIS, email, password, Login to the OIS, admission
	Business asset	Send request to the lecturer, Accept student
	Security criterion	Integrity of admission
Risk	Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials, changes existent admission, what leads to Stranger getting admission.
	Impact	Stranger gets admission
	Event	Stranger guesses password, since password is one of the top most famous passwords, logins to the OIS by lecturer credentials, changes existent admission.
	Vulnerability	Using top most popular passwords.
	Threat	Stranger guesses password, logins to the OIS, changes existent admission
	Threat agent	Stranger
	Attack method	Guess password, Login by Lecturer credentials, Change existent admission
Risk treatment	Risk treatment	Risk reduction
	Security requirements	Check entered password
	Control	-

C. Asset-related constructs. Diagram and textual templates.



Use cases:

№	**1**
Name	**Login to the OIS**
Actor	Student, Lecturer
Summary	Actor logins to the OIS using email and password.
Basic path	Bp1. Actor registers in the OIS. Bp2. Actor enters email. Bp3. Actor enters password. Bp4. OIS shows private page of actor (if actor is Student- Extp1; if actor is Lecturer- Extp2)
Alternative path	-
Extension points	Extp1. 2. Send request to the Lecturer Extp2. 3. Accept Student
Preconditions	OIS page is opened, Actor has private email.
Postconditions	Actor is registered, logged into the OIS and has access to his account.

№	**2**
Name	**Send request to the Lecturer**
Actor	Student
Summary	Student sends request to the lecturer, that he wants to be registered for the course
Basic path	Bp1. Student opens list of courses. Bp2. Student chooses course, which he wants be registered to and submits request. Bp3. OIS processes request (Extp1)
Alternative path	-
Extension points	Extp1. 5. Accept student
Preconditions	Student already logged into the OIS.
Postconditions	Request to the lecturer is successfully sent.

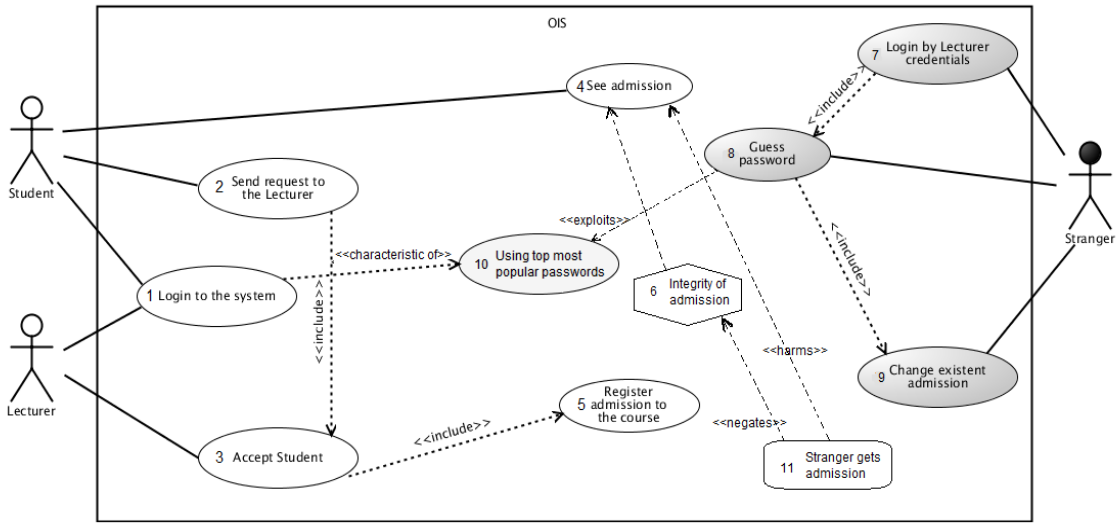
№	**3**
Name	**Accept Student**
Actor	Lecturer
Summary	Lecturer accepts Student for the requested course
Basic path	Bp1. Lecturer views request of the Student. Bp2. Lecturer accepts Student. Bp3. OIS proceeds acceptance (Extp1).
Alternative path	-
Extension points	Extp1. 7.Register admission to the course
Preconditions	Lecturer already logged into OIS, Student have sent a request.
Postconditions	Student is accepted to the course.

№	**4**
Name	**See admission**
Actor	Student
Summary	Student can view the admission after it is registered in the OIS
Basic path	Bp1. Student requests admission. Bp2. OIS retrieves admission from database. Bp3. OIS displays admission.
Alternative path	-
Extension points	-
Preconditions	Student is already logged into OIS.
Postconditions	Student can see his admission.

№	**5**
Name	**Register admission to the course**
Actor	OIS
Summary	OIS registers admission of the Student to the course.
Basic path	Bp1. OIS proceeds Lecturer acceptance. Bp2. OIS prepares document-admission of the Student. Bp3. OIS stores admission.
Alternative path	-
Extension points	4. See admission
Preconditions	Lecturer accepted Student.
Postconditions	Admission for Student is registered and stored into the OIS.

Security criterion:

ID	**6**
Name	**Integrity of admission**
Summary	Admission should be accessible for Student anytime.
Constraint of which asset	See admission



D. Risk-related constructs. Diagram and textual templates.

Misuse cases:

№	**7**
Name	**Enter email of Lecturer**
Summary	Stranger wants to login to the OIS, in order to register himself for the course. In order to do this, he enters email of the Lecturer.
Basic path	Bp1. Stranger opens OIS. Bp2. Stranger enters email of the lecturer (Extp1).
Alternative path	-
Risk treatment decision	Risk reduction
Mitigation points	Mp1. Check entered password (mitigates Bp3).
Extension points	Extp1. 8 'Guess password'.
Trigger	Always true
Assumption	As1. Stranger knows password of the Lecturer
Preconditions	Pc1. Using one of the top most famous passwords for login to the account.
Worst case threat	Stranger gets admission
Mitigation guarantee	-
Related business rules	Br1. Student sends request to the Lecturer. Br2. Lecturer accepts student. Br3. OIS registers admission. Br4. Student can view an admission.
Security criterion	Integrity of admission
Misuser profile	Stranger
Scope	OIS
Iteration	-

Level	-
Stakeholders and risks	Student: Is not registered for the course Lecturer: Got his data stolen and should be responsible for actions not performed by him.
Technology and data variations	-

№	**8**
Name	**Guess password**
Summary	Stranger trying to guess Lecturers password by enumeration of the top most popular passwords (Table 11).
Basic path	Bp1. Enters one of the top most popular passwords (see Table 11) (Extp1). Bp2. OIS logins Stranger as Lecturer (Extp3).
Alternative path	Ap1. Password is not correct, OIS shows message that password is wrong (changes Bp2). Ap2. Stranger enters another password from the list of the top most popular passwords (see Table 11). Ap3.OIS logins Stranger as Lecturer (Extp3).
Risk treatment decision	Risk reduction
Mitigation points	Mp1. Check entered (mitigates Bp2, Ap3) (Extp2).
Extension points	Extp1. 9 'Change existent admission'. Extp2. Include security use case ‘Check entered password’. Extp3. 1.'Login to the OIS'.
Trigger	Always true
Assumption	As1. Stranger knows password of the Lecturer
Preconditions	Pc1. Using one of the top most famous passwords for login to the account
Worst case threat	Stranger gets admission
Mitigation guarantee	-
Related business rules	Br1. Student sends request to the Lecturer. Br2. Lecturer accepts student. Br3. OIS registers admission. Br4. Student can view an admission.
Security criterion	Integrity of admission
Misuser profile	Stranger
Scope	OIS
Iteration	-
Level	-
Stakeholders and risks	Student: Is not registered for course Lecturer: Got his data stolen and should be responsible for actions not performed by him.
Technology and data variations	-

№	**9**
Name	**Change existent admission**

Summary	Stranger logged in by Lecturer credentials changes existent admission and enters data, so that he instead of one of the Student is registered to the course
Basic path	Bp1. Choose admission. Bp2. Enter new data to admission (Strangers data). Bp3. OIS accepts and stores new changed admission instead of old (Extp1).
Alternative path	-
Risk treatment decision	Risk reduction
Mitigation points	Mp1. Check entered password
Extension points	Extp1. Impact 'Stranger gets admission'
Trigger	Always true
Assumption	As1. Stranger knows password of the Lecturer
Preconditions	Pc1. Using one of the top most famous passwords for login to the account
Worst case threat	Stranger gets admission
Mitigation guarantee	-
Related business rules	Br1. Student sends request to the Lecturer. Br2. Lecturer accepts student. Br3. OIS registers admission. Br4. Student can view an admission.
Security criterion	Integrity of admission
Misuser profile	Stranger
Scope	OIS
Iteration	-
Level	-
Stakeholders and risks	Student: Is not registered for course Lecturer: Got his data stolen and should be responsible for actions not performed by him.
Technology and data variations	-

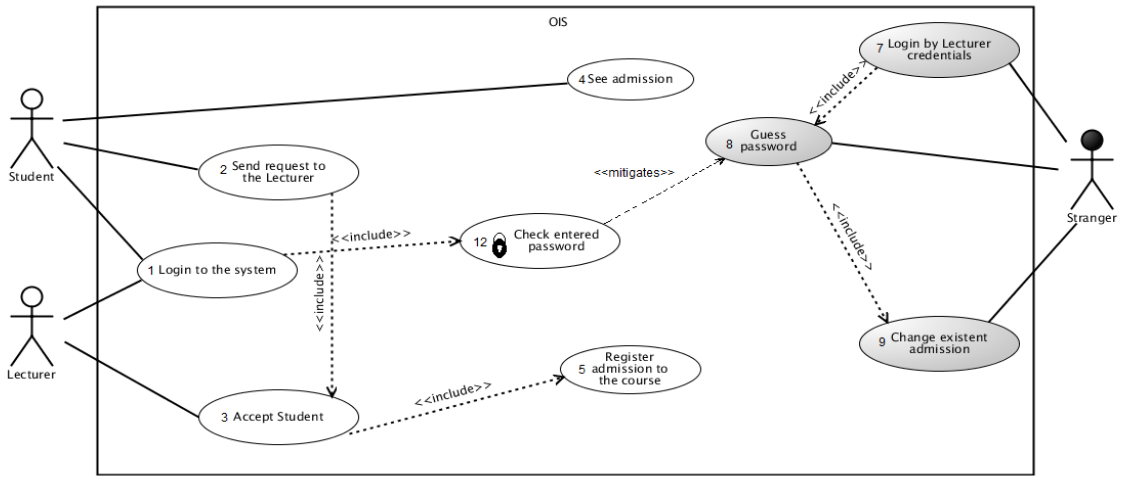
Vulnerabilities:

ID	**10**
Name	**Using top most popular passwords**
Summary	Using one of the top most famous passwords simplify account crack, as password can be guessed by enumeration of the top most famous passwords
Characteristic of which asset	Login to the OIS (Bp3. Actor enters password)
Exploits threat	Guess password

Impact:

ID	**11**
Name	**Stranger gets admission**
Summary	As a result of changing existent admission stranger gets and admission and it is not connected to the Student anymore.
Which asset harms	See admission
Which event leads to impact	Change existent admission

E. Risk treatment-related constructs. Diagram and textual templates.



Security use case:

ID	**12**
Name	**Check entered password**
Summary	Check entered password during registration to the OIS. Password should include capital letters, numbers and special symbols. Moreover password should be not less than 6 characters.
Basic path	1. During registration check if entered password is strong (if it is not less than 6 characters, has at least one capital letter, one special symbol and one number) 2. If password is strong- register in the OIS 3. If password is not strong- request to enter new password.
Extension points	1. Login to the OIS Mitigates 7. Guess password
Risk mitigation	Mitigates risk that Stranger by enumeration of the the top most famous passwords can login to the system by lecturer credentials and change existent admission.

F. Misuse case textual template extension validation. Expected answers

Table 25: Misuse case template validation. Expected results

ISSRM domain model construct	**Information from template**	**Defined by which field**
Business asset	Send request to the lecturer, Accept student	Related business rules
IS asset	OIS, email, password, Login to the OIS, admission	-
Security criterion	Integrity of admission	Security criterion
Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials, changes existent admission, what leads to Stranger getting admission.	Stakeholders and risks
Impact	Stranger gets admission	Worst case threat
Event	Stranger guesses password, since password is one of the top most famous passwords, logins to the OIS by lecturer credentials, changes existent admission.	Threat + Vulnerability
Vulnerability	Using top most popular passwords.	Assumption: precondition
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Threat agent + Attack method
Threat agent	Stranger	Misuser profile
Attack method	Guess password, Login by Lecturer credentials, Change existent admission	Basic path, alternative path, extension points
Risk treatment	Risk reduction	Risk treatment
Security requirement	Check entered password	Mitigation points
Control	-	-

G. Mal-activity diagrams extensions validation. Initial data and expected answers.

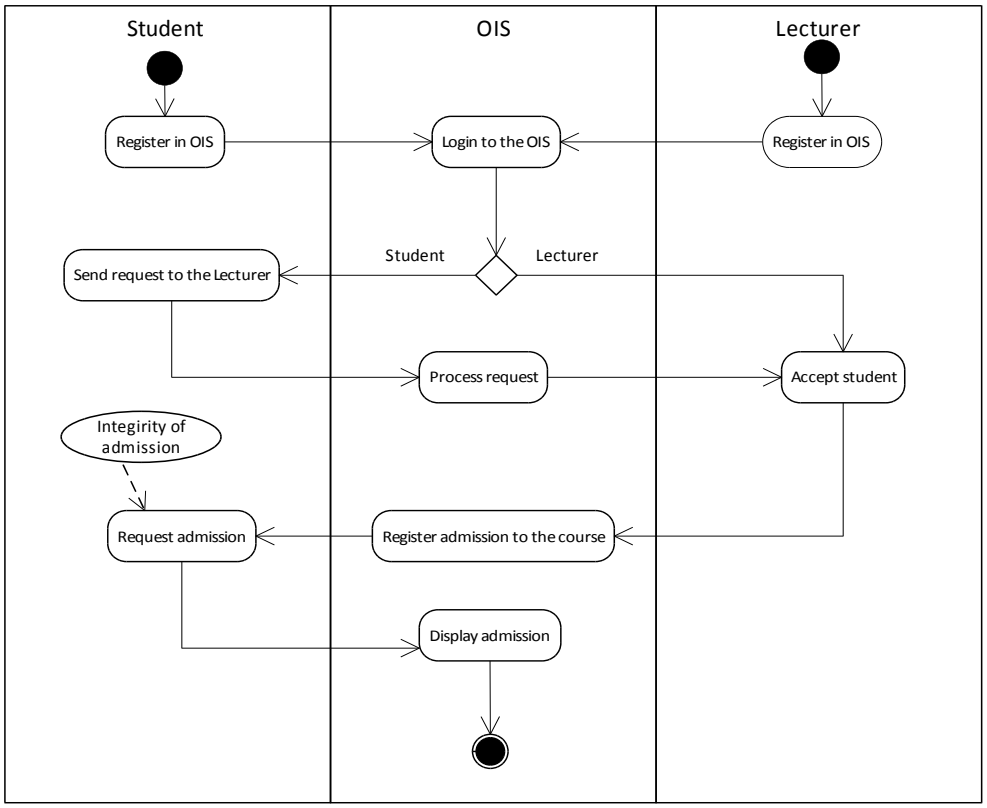


Table 26: Mal-activity diagrams extensions validation. Expected answers (asset-related concepts)

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Business asset	Send request to the lecturer, Accept student	Activity
IS asset	OIS, email, password, Login to the OIS, admission	Activities, control flow, swimlane, sub-swimlanes
Security criterion	Integrity of admission	Oval visual construct for the security criterion

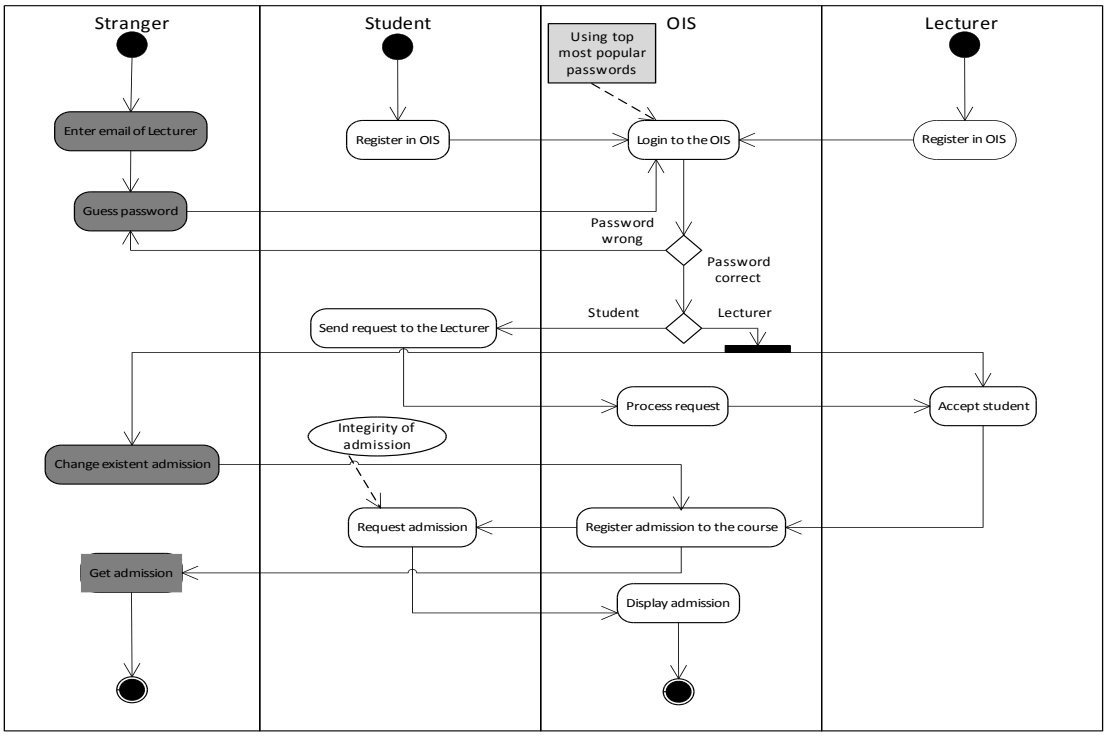


Table 27: Mal-activity diagrams extensions validation. Expected answers (risk-related concepts)

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials, changes existent admission, what leads to Stranger getting admission.	Combination of constructs for threat agent, attack method, vulnerability and impact
Impact	Stranger gets admission	Mal-activities
Event	Stranger guesses password, since password is one of the top most famous passwords, logins to the OIS by lecturer credentials, changes existent admission.	Combination of constructs for threat agent, attack method and vulnerability
Vulnerability	Using top most popular passwords.	Grey rectangle visual construct to represent vulnerability
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Combination of constructs for threat agent and attack method
Threat agent	Stranger	Mal-swimlane
Attack method	Guess password, Login by Lecturer credentials, Change existent admission	Mal-activities, mal-swimlane

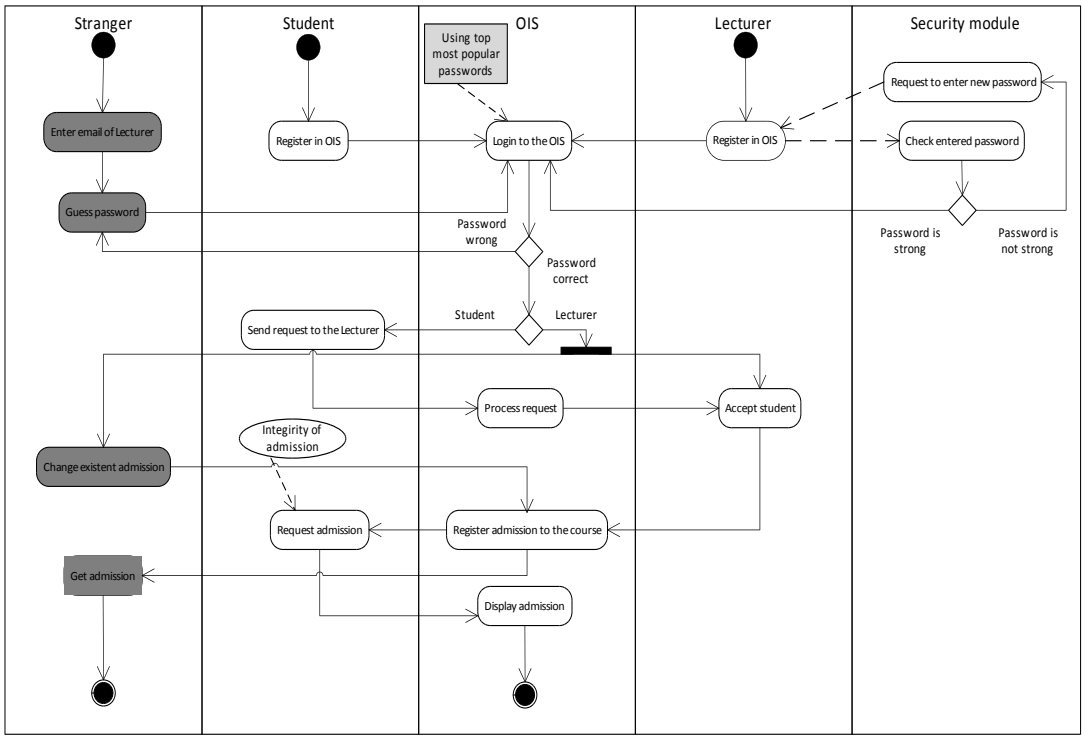


Table 28: Mal-activity diagrams extensions validation. Expected results (risk treatment-related concepts)

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Risk treatment	Risk reduction	-
Security requirement	Check entered password	Mitigation activities, decision
Control	Security module	Swimlane

H. Misuse case textual template extensions validation. Experiment results

Participant 1

Table 29: Misuse case template extensions validation. Solution of participant 1

ISSRM domain model construct	Information from template	Defined by which field
Business asset	Student sends request to the Lecturer, Lecturer accepts student, OIS registers admission, Student view admission	Related business rules
IS asset	-	-
Security criterion	Integrity of admission	Security criterion
Risk	Student is not registered to the course, Lecturer got his data stolen and responsible for action not performed by him.	Stakeholders and risks
Impact	Stranger gets admission	Worst case threat
Event	Stranger enters new data to existent admission	Threat + Vulnerability
Vulnerability	Stranger knows password of the Lecturer, Using top most popular passwords.	Assumption, precondition
Threat	Stranger enters new data to existent admission	Threat agent + Attack method
Threat agent	Stranger	Misuser profile
Attack method	Enter new data to existent admission	Basic path, alternative path, extension points
Risk treatment	Risk reduction	Risk treatment
Security requirement	Check entered password	Mitigation points
Control	-	-

Participant 2

Table 30: Misuse case template extensions validation. Solution of participant 2

ISSRM domain model construct	Information from template	Defined by which field
Business asset	Request to the Lecturer, accept Student	Related business rules
IS asset	OIS	Scope
Security criterion	Integrity	Security criterion
Risk	Change existent admission	Stakeholders and risks
Impact	Stranger gets admission	Worst case threat
Event	Stranger guesses password	-
Vulnerability	Using top most popular passwords.	Assumption, precondition
Threat	Stranger guesses password, logins to the OIS, changes existent admission	-
Threat agent	Stranger	Misuser profile
Attack method	Guess password, Login by Lecturer credentials, Change existent admission	Basic path, alternative path, extension points
Risk treatment	Reduction	Risk treatment
Security requirement	Check password	Mitigation points
Control	-	-

Participant 3

Table 31: Misuse case template extensions validation. Solution of participant 3

ISSRM domain model construct	**Information from template**	**Defined by which field**
Business asset	Send request to the lecturer, Accept student	Related business rules
IS asset	OIS, email, password, Login to the OIS, admission	-
Security criterion	Integrity of admission	Security criterion
Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials, changes existent admission, what leads to Stranger getting admission.	Stakeholders and risks
Impact	Stranger gets admission	Worst case threat
Event	Stranger guesses password, since password is one of the top most famous passwords, logins to the OIS by lecturer credentials, changes existent admission.	Threat + Vulnerability
Vulnerability	Using top most popular passwords.	Assumption: precondition
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Threat agent + Attack method
Threat agent	Stranger	Misuser profile
Attack method	Guess password, Login by Lecturer credentials, Change existent admission	Basic path, alternative path, extension points
Risk treatment	Risk reduction	Risk treatment
Security requirement	Check entered password	Mitigation points
Control	-	-

Participant 4

Table 32: Misuse case template extensions validation. Solution of participant 4

ISSRM domain model construct	**Information from template**	**Defined by which field**
Business asset	Send request to the lecturer, Accept student	Related business rules
IS asset	OIS, email, password, Login to the OIS, admission	-
Security criterion	Integrity of admission	Security criterion
Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials, changes existent admission, what leads to Stranger getting admission.	Stakeholders and risks
Impact	Stranger gets admission	Worst case threat
Event	Stranger guesses password, since password is one of the top most famous passwords, logins to the OIS by lecturer credentials, changes existent admission.	Threat + Vulnerability
Vulnerability	Using top most popular passwords.	Assumption: precondition
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Threat agent + Attack method
Threat agent	Stranger	Misuser profile
Attack method	Guess password, Login by Lecturer credentials, Change existent admission	Basic path, alternative path, extension points
Risk treatment	Risk reduction	Risk treatment

Security requirement	Check entered password	Mitigation points
Control	-	-

Participant 5

Table 33: Misuse case template extensions validation. Solution of participant 5

ISSRM domain model construct	**Information from template**	**Defined by which field**
Business asset	Send request to the lecturer, Accept student	Related business rules
IS asset	OIS, email, password, Login to the OIS, admission	-
Security criterion	Integrity of admission	Security criterion
Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials, changes existent admission, what leads to Stranger getting admission.	Stakeholders and risks
Impact	Stranger gets admission	Worst case threat
Event	Stranger guesses password, since password is one of the top most famous passwords, logins to the OIS by lecturer credentials, changes existent admission.	Threat + Vulnerability
Vulnerability	Using top most popular passwords.	Assumption: precondition
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Threat agent + Attack method
Threat agent	Stranger	Misuser profile
Attack method	Guess password, Login by Lecturer credentials, Change existent admission	Basic path, alternative path, extension points
Risk treatment	Risk reduction	Risk treatment
Security requirement	Check entered password	Mitigation points
Control	-	-

I. Mal-activity diagrams extensions validation. Experiment results

Participant 1

Table 34: Mal-activity diagrams extensions validation. Solution of participant 1

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Business asset	Send request to the lecturer, accept student	Send request activiy
IS asset	OIS, email, password, login to the OIS, admission	Activity
Security criterion	Integrity	Oval
Risk	Stranger guesses password, stranger logins to the OIS, Stranger changes existent admission, Stranger can guess lecturer password, because it is one of the top most famous passwords.	Attack method, vulnerability and impact
Impact	Stranger gets admission and can view it	Activities, decisions
Event	Stranger guesses password, stranger logins to the OIS, Stranger changes existent admission, Stranger can guess lecturer password, because it is one of the top most famous passwords.	Threat agent+ attack method
Vulnerability	Top most popular passwords.	Grey rectangle
Threat	Stranger guesses password, logins to the OIS, changes existent admission	threat agent and attack method
Threat agent	Stranger	Swimlane
Attack method	Changes existent admission, guesses password, logins by lecturer credentials	-
Risk treatment	Reduction	-
Security requirement	Check entered password	Activity, decision
Control	-	-

Participant 2

Table 35: Mal-activity diagrams extensions validation. Solution of participant 2

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Business asset	Send request to the lecturer, accept student, process request	Activity, mal-activity
IS asset	OIS, email, password, admission	Activity -Login to the OIS
Security criterion	Integrity	Circle with integrity of admission insisde
Risk	Stranger uses enumeration of top most famous passwords in order to login to the OIS by lecturer credentials.	Threat agent, attack method, vulnerability.
Impact	Display admission	Activity

Event	Stranger guesses password (password-top most famous password), logins to the OIS, changes existent admission.	Threat agent, attack method, vulnerability
Vulnerability	Top most popular passwords and email of lecturer	Rectangle
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Combination of constructs for threat agent and attack method
Threat agent	Stranger	Swimlane
Attack method	Using top most popular passwords	Rectangle
Risk treatment	Risk reduction and mitigation	-
Security requirement	Check password if password is strong	Mitigation activities
Control	Security module swimlane	Swimlane with mitigation activities

Participant 3

Table 36: Mal-activity diagrams extensions validation. Solution of participant 3

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Business asset	Accept student	Activity, swimalane
IS asset	OIS, email, password.	Credentials, system
Security criterion	Integrity, confidentiality of admission	Oval with activity
Risk	Stranger guesses password because it is top most famous password	Activity, swimlane
Impact	Stranger gets admission	Mal
Event	Guess password, login to OIS, change existent admission	Swimlane, activity, rectangle
Vulnerability	Using top most popular passwords.	Rectangle
Threat	Stranger guesses password, logins to the OIS	Mal-swimlane
Threat agent	-	Mal
Attack method	Guess password	Mal-activitiy
Risk treatment	Risk reduction	-
Security requirement	Check entered password	Mitigation activity, decision
Control	Security module, mitigation swimlane	Activity, mitigation swimlane

Participant 4

Table 37: Mal-activity diagrams extensions validation. Solution of participant 4

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Business asset	Send request to the lecturer, accept student	Activity, decision, swimlane
IS asset	OIS, email, password, admission	-

Security criterion	Integrity of admission, confidentiality of password	Oval, activity
Risk	Stranger logins to the OIS by Lecturer credentials, because it is top most popular passwords and he know email of the lecture.	Mal-activity, mal-swimlane
Impact	Get admission	Mal-activities
Event	Stranger guesses password, because it is easy to guess, it is one of the top most famous passwords.	Mal-activity, mal-swimlane
Vulnerability	Top most popular passwords, email of lecturer	Activity and rectangle
Threat	Stranger guesses password and logins to the OIS.	Mal-activity, mal-swimlane
Threat agent	Swimlane	Mal
Attack method	Login to the system and change existent admission	Activity
Risk treatment	Risk reduction mitigation	-
Security requirement	Check password	Mitigation activities, decision, swimlane
Control	Security module	Mitigation swimlane

Participant 5

Table 38: Mal-activity diagrams extensions validation. Solution of participant 5

ISSRM domain model construct	**Diagram element**	**Language construct/combination of constructs**
Business asset	Request to accept for the course	Activity, decision
IS asset	OIS	Swimlane
Security criterion	Circle	Integrity of admission
Risk	Swimlane, activity	Guess password, login to the OIS, change admission
Impact	Activity of Stanger	Get admission
Event	Threat agent, vulnerability	Guess password, login to the OIS, change admission
Vulnerability	Stranger using top most popular passwords	Quadrat
Threat	Stranger guesses password, logins to the OIS, changes existent admission	Mal-swimlane, mal-activity
Threat agent	Stranger	Mal-swimlane
Attack method	Guess password, login by Lecturer credentials, change existent admission	Mal-activity, mal-swimlane
Risk treatment	Reduction	-
Security requirement	-	-
Control	-	-

Non-exclusive license to reproduce thesis and make thesis public

I, Anastasiia Onchukova, (date of birth: 05.11.1990),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

Security Risk Management using Misuse cases and Mal-activities,

supervised by Raimundas Matulevičius,

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu/Tallinn, August 2013