

UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
Institute of Computer Science

Yenal Turan

**Extension and Application of Event-
driven Process Chain for Information
System Security Risk Management**

Master Thesis (30 EAP)

Supervisor: Raimundas Matulevičius, PhD

Author: “.....“ May 2012

Supervisor: “.....“ May 2012

Professor: “.....“ May 2012

TARTU, 2012

ABSTRACT

Security engineering is one of the important concerns during the system development and it should be addressed throughout the whole system development process. Besides, there are several languages for security modeling that help dealing with security risk management at the requirements stage. In this thesis, first of all, we are focusing on Event-driven Process Chain (EPC), which is used during the business process modeling. More specifically, we investigate how this language supports information system security risk management (ISSRM). The purpose of this investigation is the problem of security requirements need of EPC. As a result, we obtain an alignment table between EPC constructs and ISSRM domain model concepts. Next, we extend the EPC language and its constructs with respect to the alignment table between EPC and ISSRM. As a consequence, we call the extended language as “Security-Oriented EPC”. The extended language contains new set of constructs which refer to ISSRM concepts. Lastly, after clarifying the importance of security requirements at the early system development, we present transformation guidelines to perform forward model translations from Security-Oriented EPC to Mal-Activity Diagrams (MAD). During the transformation, our proposal is based on the systematic and grounded extensions of EPC language and its interdependency to the domain model of ISSRM. Alignment results may help business analysts understand how to model security risks at the system requirement and design stages. Also, transformation results pave the way for interoperability between the modeling languages that are analysed using the same conceptual framework.

ABBREVIATIONS

Acronym	Definition
EPC	Event-driven Process Chain
MAD	Mal-Activity Diagram
BPML	Business Process Modeling Language
YAWL	Yet Another Workflow Language
BPMN	Business Process Modeling Notation
SRM	Security Risk Management
ISSRM	Information System Security Risk Management
KeS	KAOS Extention to Security
SML	Security Modeling Language
UML	Unified Modeling Language
ROI	Return on Investment
ROSI	Return on Security Investment
BPM	Business Process Modeling
IS	Information System
RT	Risk Treatment
IDS	Intrusion Detection System
TR	Transformation Rule
IT	Information Technology

TABLE OF CONTENTS

<i>ABSTRACT</i>	3
<i>ABBREVIATIONS</i>	4
<i>Chapter 1. INTRODUCTION</i>	11
1.1 <i>Scope</i>	11
1.2 <i>Motivation</i>	11
1.3 <i>Problem / Research Question</i>	12
1.4 <i>Introduction of Solution</i>	12
1.5 <i>Structure</i>	13
<i>Chapter 2. BUSINESS PROCESS MODELING LANGUAGES</i>	15
2.1 <i>Activity Diagrams</i>	15
2.2 <i>Petri Nets</i>	16
2.3 <i>Yet Another Workflow Language (YAWL)</i>	17
2.4 <i>Business Process Modeling Notation (BPMN)</i>	19
2.5 <i>Event-Driven Process Chain (EPC)</i>	21
2.5.1 <i>Introduction to Event-Driven Process Chain</i>	22
2.5.2 <i>Meta-Model of EPC and Construct Definitions</i>	23
2.5.3 <i>Integrity Constraints in Meta-Model of EPC</i>	25
2.6 <i>Comparison and Summary</i>	25
<i>Chapter 3. SECURITY MODELING LANGUAGES</i>	27
3.1 <i>KAOS Extention to Security</i>	27
3.2 <i>Misuse Cases</i>	29
3.3 <i>Mal-Activity Diagrams</i>	31
3.4 <i>Comparison and Summary</i>	33
<i>Chapter 4. SECURITY RISK MANAGEMENT</i>	35
4.1 <i>Model-Based Security Analysis with CORAS Method</i>	35
4.2 <i>Goal-Risk Framework</i>	36
4.3 <i>Information System Security Risk Management (ISSRM)</i>	37
4.3.1 <i>The Domain Model of ISSRM</i>	38
4.3.2 <i>Risk Management Process</i>	39
4.3.3 <i>Analysing Security Modeling Languages with ISSRM</i>	40
4.4 <i>Summary</i>	42
<i>Chapter 5. ALIGNMENT OF EPC AND ISSRM</i>	45
5.1 <i>Security Risk Modeling with EPC</i>	45
5.1.1 <i>Context and Asset Identification</i>	45
5.1.2 <i>Determination of Security Objectives</i>	47
5.1.3 <i>Risk Analysis and Assessment</i>	47
5.1.4 <i>Risk Treatment</i>	49
5.1.5 <i>Security Requirements Definition</i>	49
5.1.6 <i>Control Selection and Implementation</i>	50

5.2 ISSRM and EPC Alignment	51
5.3 Summary.....	51
Chapter 6. SECURITY-ORIENTED EPC	55
6.1 Higher-Level of Security Problem Definition	55
6.1.1 Concrete Syntax in High Level.....	55
6.1.2 Abstract Syntax in High Level.....	61
6.2 Lower-Level of Security Problem Definition	61
6.2.1 Concrete Syntax in Low Level.....	61
6.2.2 Abstract Syntax in Low Level.....	65
6.3 Extended Meta-Model of Security-Oriented EPC.....	68
6.4 Summary.....	68
Chapter 7. MEASURING SECURITY RISKS USING SECURITY-ORIENTED EPC.....	71
7.1 Metrics Definitions for Security-Oriented EPC.....	71
7.2 Return on Security Investment (ROSI) of Security-Oriented EPC	74
7.3 Summary.....	80
Chapter 8. TRANSFORMATION GUIDELINES FROM SECURITY-ORIENTED EPC TO MAL- ACTIVITY DIAGRAMS.....	81
8.1 Asset-related Transformation	81
8.2 Risk-related Transformation.....	87
8.3 Risk Treatment-related Transformation	91
8.4 Summary.....	94
Chapter 9. VALIDATION.....	95
9.1 Introduction	95
9.2 Problem Statement	95
9.3 Experiment Planning.....	95
9.4 Experiment Operation	96
9.5 Data Analysis.....	98
9.6 Interpretation of Results	105
9.7 Summary.....	106
Chapter 10. CONCLUSION.....	109
10.1 Summary.....	109
10.2 Limitations	109
10.3 Conclusions.....	110
10.4 Future Work.....	111
RESÜMEE.....	112
BIBLIOGRAPY.....	113

LIST OF FIGURES

FIGURE 1.1 – CoNCEPT ALIGNMENT BETWEEN ISSRM DOMAIN MODEL AND EPC [13].....	13
FIGURE 2.1 – ACTIVITY DIAGRAM OF WITHDRAW MONEY FROM ATM.....	15
FIGURE 2.2 – PLAIN PETRI NET MODEL OF WITHDRAW MONEY FROM ATM.....	17
FIGURE 2.3 – YAWL MODEL OF WITHDRAW MONEY FROM ATM.....	19
FIGURE 2.4 – BPMN MODEL OF WITHDRAW MONEY FROM ATM.....	19
FIGURE 2.5 – EPC DIAGRAM OF WITHDRAW MONEY FROM ATM.....	21
FIGURE 2.6 – EPC META-MODEL.....	23
FIGURE 3.1 – ONLINE BANKING PROBLEM - EXTENDED OPERATIONAL MODEL (CUSTOMER SIDE).....	28
FIGURE 3.2 – ONLINE BANKING PROBLEM - RISK ANALYSIS AND ASSESSMENT MODEL (ATTACKER SIDE)....	28
FIGURE 3.3 – ONLINE BANKING, USE CASES (ON THE LEFT) AND MISUSE CASES (ON THE RIGHT) BASED ON SECURITY REQUIREMENTS.....	30
FIGURE 3.4 – ONLINE BANKING, MAL-ACTIVITY DIAGRAMS OF CLIENT AND ATTACKER IN BANKING SYSTEM.	32
FIGURE 4.1 – THE DOMAIN MODEL OF ISSRM [11].....	38
FIGURE 4.2 – SECURITY RISK MANAGEMENT PROCESS [11].....	40
FIGURE 5.1 – EPC DIAGRAM OF ONLINE REGISTRATION AND LOGIN PROCESSES OF THE INTERNET STORE.....	46
FIGURE 5.2 – EPC DIAGRAM OF ONLINE REGISTRATION AND LOGIN PROCESSES OF THE INTERNET STORE(WITH PROCESSPATHS).....	47
FIGURE 5.3 – MESSAGE HANDLING PROCESS INCLUDING SECURITY RISK ATTACK.....	48
FIGURE 5.4 – MESSAGE HANDLING PROCESS INCLUDING SECURITY REQUIREMENTS.....	50
FIGURE 6.1 – ASSET-RELATED (C)ONCEPTS AND (R)ELATIONSHIPS.....	56
FIGURE 6.2 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE SHOWN IN BUSINESS PROCESS PATH AND IS PROCESS PATH.....	56
FIGURE 6.3 – RISK- RELATED (C)ONCEPTS AND (R)ELATIONSHIPS.....	57
FIGURE 6.4 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE SHOWN IN BUSINESS ASSET PROCESS PATH, IS ASSET PROCESS PATH AND RISK PROCESS PATH.....	58
FIGURE 6.5 – RISK TREATMENT-RELATED (C)ONCEPTS AND (R)ELATIONSHIPS.....	58
FIGURE 6.6 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE IN BUSINESS ASSET PROCESSPATH,IS ASSET PROCESSPATH,RISK PROCESSPATH &RISK TREATMENT PROCESSPATH.....	59
FIGURE 6.7 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE IN BUSINESS ASSET PROCESSPATH,IS ASSET PROCESSPATH,RISK PROCESSPATH &RISK TREATMENT PROCESSPATH.....	59
FIGURE 6.8 – ABSTRACT SYNTAX OF EXTENDED EPC WITH PROCESS PATH AND ISSRM DOMAIN MODEL....	60
FIGURE 6.9 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE SHOWN IN BUSINESS ASSET CONSTRUCTS AND IS ASSET CONSTRUCTS.....	62
FIGURE 6.10 – ASSET-RELATED (C)ONCEPTS AND (R)ELATIONSHIPS.....	62
FIGURE 6.11 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE SHOWN IN BUSINESS ASSET CONSTRUCTS AND IS ASSET CONSTRUCTS INCLUDING NEW CONSTRUCTS EPC – SECURITY CRITERION AND EPC – CONSTRAINT OF.....	63
FIGURE 6.12 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE SHOWN IN BUSINESS ASSET CONSTRUCTS, IS ASSET CONSTRUCTS AND RISK CONSTRUCTS.....	63
FIGURE 6.13 – RISK-RELATED (C)ONCEPTS AND (R)ELATIONSHIPS.....	64
FIGURE 6.14 – PART OF REGISTRATION AND LOGIN TO INTERNET STORE EXAMPLE SHOWN IN BUSINESS ASSET CONSTRUCTS, IS ASSET CONSTRUCTS, RISK CONSTRUCTS AND RISK TREATMENT CONSTRUCTS.....	65
FIGURE 6.15 – RISK TREATMENT-RELATED (C)ONCEPTS AND (R)ELATIONSHIPS.....	66
FIGURE 6.16 – ABSTRACT SYNTAX OF EXTENDED EPC WITH CONSTRUCTS AND ISSRM DOMAIN MODEL.....	67
FIGURE 6.17 – ABSTRACT SYNTAX OF EXTENDED EPC AKA SECURITY-ORIENTED EPC.....	68
FIGURE 6.18 – THREE STEPS OF THE GUIDELINES TO USE SECURITY-ORIENTED EPC.....	69
FIGURE 7.1 – HIGH LEVEL SECURITY-ORIENTED EPC ABSTRACT SYNTAX ENRICHED WITH METRICS.....	72
FIGURE 7.2 – LOW LEVEL SECURITY-ORIENTED EPC ABSTRACT SYNTAX ENRICHED WITH METRICS.....	73

FIGURE 7.3 – INQUIRY BY USER (BUSINESS ASSET), SCANNING OF INQUIRY MESSAGE (RISK-TREATMENT), ATTACK OF VIOLATOR WITH A SPY PROGRAM ATTACHED TO THE INQUIRY MESSAGE (RISK), INQUIRY ACCEPTED AND READ BY ADMIN (IS ASSET).....	75
FIGURE 8.1 – SECURITY-ORIENTED EPC DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE.	83
FIGURE 8.2 – MAL-ACTIVITY DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE.	83
FIGURE 8.3 – SECURITY-ORIENTED EPC DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE INCLUDING SECURITY RISK(S).	87
FIGURE 8.4 – MAL-ACTIVITY DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE INCLUDING SECURITY RISK(S).	88
FIGURE 8.5 – SECURITY-ORIENTED EPC DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE INCLUDING RISK TREATMENT.	92
FIGURE 8.6 – MAL-ACTIVITY DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE INCLUDING RISK TREATMENT.....	92
FIGURE 9.1 – SECURITY-ORIENTED EPC DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE.	97
FIGURE 9.2 – MAL-ACTIVITY DIAGRAM OF ONLINE REGISTRATION (MESSAGE HANDLING) OF THE INTERNET STORE, INCLUDING IDENTIFIED STARS FOR TRANSFORMED CONSTRUCTS.	97
FIGURE 9.3 – SOLUTION OF PARTICIPANT 1.	98
FIGURE 9.4 – SOLUTION OF PARTICIPANT 2.	99
FIGURE 9.5 – SOLUTION OF PARTICIPANT 3.	100
FIGURE 9.6 – SOLUTION OF PARTICIPANT 4.	101
FIGURE 9.7 – SOLUTION OF PARTICIPANT 5.	101
FIGURE 9.8 – SOLUTION OF PARTICIPANT 6.	102
FIGURE 9.9 – SOLUTION OF PARTICIPANT 7.	103
FIGURE 9.10 – SOLUTION OF PARTICIPANT 8.....	104
FIGURE 9.11 – SOLUTION OF PARTICIPANT 9.....	104
FIGURE 9.12 – SOLUTION OF PARTICIPANT 10.....	105

LIST OF TABLES

TABLE 2.1 – LEGEND FOR ACTIVITY DIAGRAMS.	16
TABLE 2.2 – LEGEND FOR PETRI NETS.....	17
TABLE 2.3 – LEGEND FOR YAWL MODELING.	18
TABLE 2.4 – LEGEND FOR BPMN.....	20
TABLE 2.5 – LEGEND FOR EPC.	22
TABLE 2.6 – COMPARISON OF BUSINESS PROCESS MODELING LANGUAGES.	26
TABLE 3.1 – LEGEND FOR KAOS EXTENTION TO SECURITY.	29
TABLE 3.2 – LEGEND FOR MISUSE CASES.	30
TABLE 3.3 – LEGEND FOR MAL-ACTIVITY DIAGRAMS.	32
TABLE 3.4 – COMPARISON OF SECURITY MODELING LANGUAGES.	33
TABLE 5.1 – ALIGNMENT OF THE EPC CONSTRUCTS TO THE ISSRM CONECPTS.	52
TABLE 7.1 – METRIC ANALYSIS TABLE FOR ISO/IEC 27005 [26].	71
TABLE 7.2 – BUSINESS AND IS ASSETS OF INTERNET STORE SYSTEM INQUIRY SENDING PROCESS.	75
TABLE 7.3 – QUALITATIVE SCALE OF VALUE FOR THE VALUE OF BUSINESS ASSETS.	76
TABLE 7.4 – SECURITY CRITERION OF THE INTERNET STORE SYSTEM INQUIRY SENDING PROCESS.	76
TABLE 7.5 – QUALITATIVE SCALE OF VALUE FOR THE SECURITY NEED METRIC.	76
TABLE 7.6 – RISK RELATED CONSTRUCTS IN INTERNET STORE SYSTEM INQUIRY SENDING PROCESS.	77
TABLE 7.7 – QUALITATIVE SCALE OF VALUE FOR THE LIKELIHOOD METRIC.	77
TABLE 7.8 – QUALITATIVE SCALE OF VALUE FOR THE VULNERABILITY LEVEL METRIC.	77
TABLE 7.9 – RISK MATRIX.	78
TABLE 7.10 – RISK LEVEL CALCULATION TABLE.	78
TABLE 7.11 – RISK TREATMENT METHODS AND THEIR DESCRIPTIONS.	79
TABLE 7.12 – SECURITY REQUIREMENTS DEFINITION.	79
TABLE 7.13 – RISK ASSESSMENT AND TREATMENT TABLE.	79
TABLE 8.1 – ALIGNMENT OF THE ISSRM CONCEPTS AND THE SECURITY-ORIENTED EPC AND MAD CONSTRUCTS.	82
TABLE 9.1 – STATISTICS OF THE PARTICIPANTS’ RESULTS BASED ON TRANSFORMATIONS.	106

Chapter 1. INTRODUCTION

There are several established Business Process Modeling Languages (BPMLs) commonly used in industry (e.g. EPC [24] [25], BPMN [9], YAWL [8] and Activity Diagrams [6]). Usually to describe a business process, many forms of information must be integrated into a business process model. BPMLs differ in the extent to which their constructs represent the information that answers what is going to be done, who is going to do it, when and where it will be done, how and why will it be done, and who is dependent on the information. These differences result from the various source domains, and there is a need to secure entities and activities related to the above mentioned questions by implementing secure constructs. Work has not been done to align the business processes with Security Risk Management Model (SRM [11]). SRM can be addressed using different modeling techniques at different enterprise levels; asset level, risk level, and risk treatment level.

1.1 Scope

Information enterprise systems should be secured against potential risks and vulnerable attacks. Event-driven Process Chain (EPC) is a modeling language used to define business processes. Although serving its primary purpose at the high-degree, EPC is not helpful to elicit security concerns when developing information enterprise systems.

Security analysis should start from the early stages, for example from the business process modeling [6] [8] [9] [24] [25]. Business analysts need to invest in security analysis additionally using other approaches and understanding how these approaches could be aligned to the existing business models.

1.2 Motivation

Business processes development includes multiple perspectives and viewpoints [6] [8] [9] [24] [25], thus combined application of these techniques could much improve the understanding of different stakeholders needs with respect to the security risks. It would also contribute to the quality of system security developed through different development stages.

The purpose of this thesis is to develop set of rules and guidelines in order to measure the suitability of Event-driven Process Chain for capturing security concerns. The metric unit during this measurement is the Security Risk Management. The outcome of the analysis of EPC and SRM will also be the answer of the question “*Why security risk management is important?*”. The motive answer is that the use of security risk management helps security professionals align with business objectives rather than focusing entirely on destroying the vulnerability as soon as it raises its head.

1.3 Problem / Research Question

Information system security risk management (ISSRM [11]) is the method used as the path for management of security controls. In this thesis the main expected outcome of the alignment process is the coverage of the ISSRM domain by EPC, since EPC is not helpful to elicit security concerns when developing information enterprise systems.

The result of the alignment will help us to answer our research question “*How EPC could be extended to support security risk management?*” by identifying which constructs or characteristics of EPC are more likely to support ISSRM activities, and how they should be improved for this. The following research question is “*What is the benefit of such extension?*”, where the validation of the extended EPC will be answer of this question.

In addition, we define set of guidelines how to transform extended EPC model to the later stage of system development, i.e. Transformation from extended EPC to the models of Mal-Activity Diagrams (MAD [5]). In such a way we continue secure system definition not only at the business process stage, but also at the requirement analysis and design stages.

1.4 Introduction of Solution

The characteristic of the alignment process is the Security Risk Analysis contribution of ISSRM. By security risk analysis, EPC is analysed with different styles at early stage requirements design. EPC is analysed and grouped into three different concepts which are asset, risk and risk treatment. This grouping clarifies the model and makes the Security Risk Analysis easier to analyse step by step.

To align EPC with the ISSRM domain model, the method shown in Figure 1.1 is applied. As it is shown in Figure 1.1, Meta-Model and glossary of ISSRM domain model are synthesized with the Meta-Model and Documentation of EPC. This synthesis is called as *Concept Alignment*. Consequently, an “ISSRM-oriented” Meta-Model of EPC will be produced after the alignment process. By “ISSRM-oriented”, we mean a Meta-Model aligned on the ISSRM domain model and thus showing only concepts and relationships semantically equivalent to those of the ISSRM domain model. This gives a clear view of the coverage domain of the security-oriented language with regards to ISSRM [13]. In the end we obtain construct extensions and we call the new extended language as Security-Oriented EPC. Later, we define transformation rules from Security-Oriented EPC to MAD and validate these transformations with a case study.

We validate our proposal in a case study where the quality of resulting models would be evaluated and compared. We believe that our proposal will suggest practitioners the means to understand security threats as soon as possible and to address how systematically through the whole system development cycle.

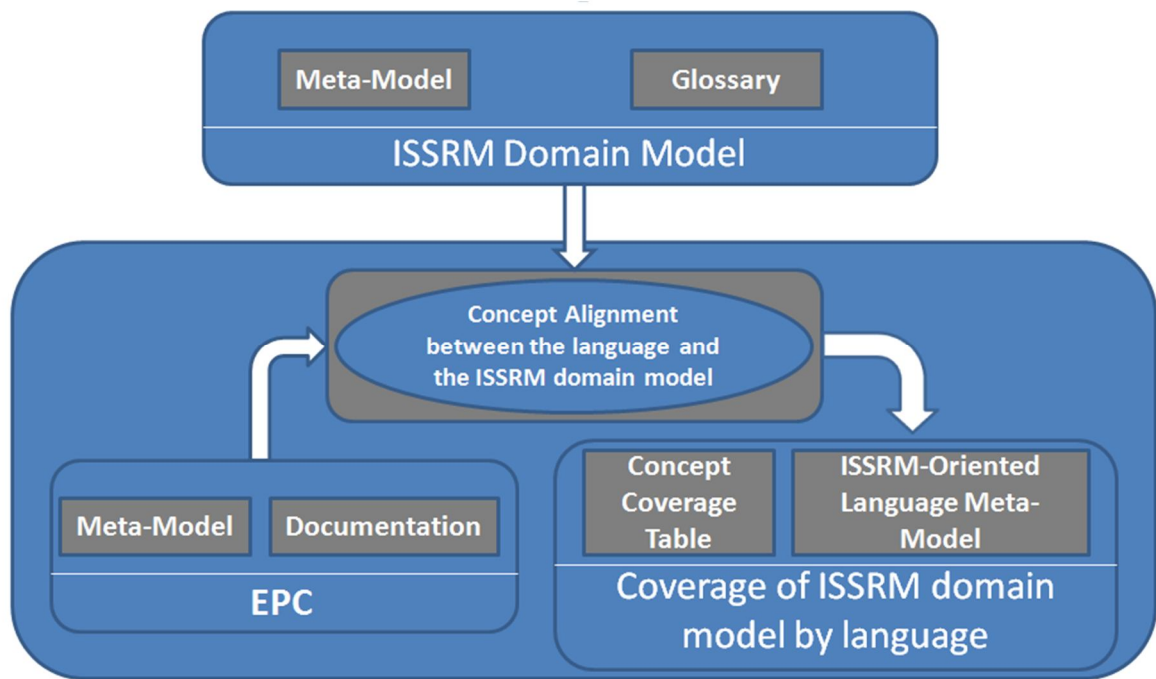


Figure 1.1 – Concept alignment between ISSRM domain model and EPC [13].

1.5 Structure

In this thesis, first of all the existing state of the art is reviewed. BPMLs (Activity Diagrams, Petri Nets, YAWL, BPMN and EPC) and SMLs (KAOS Extension to Security, Misuse Cases and Mal-Activity Diagrams) are analysed in Chapter 2 and Chapter 3 including illustration of models with a running example. In Chapter 4, definition of the domain model of ISSRM, Risk Management Process, and analysis of some security modeling languages with ISSRM is given.

After providing the background information, alignment between EPC and ISSRM is performed in Chapter 5. The alignment process is performed by following six steps of Security Risk Modeling defined in [11]. Alignment process is followed by construct extension of the EPC language. Concrete and abstract syntax EPC extensions are done in Chapter 6 in higher and lower level. The high level extensions contain “process path” construct of EPC whereas low level extensions contain all the constructs of EPC. The extended language is called Security-Oriented EPC.

Chapter 7 illustrates how to capture and measure security risks using Security-Oriented EPC by defining metrics and Return on Security Investment (ROSI [27]).

Chapter 8 consists of transformation rules from Security-Oriented EPC to Mal-Activity Diagrams [5]. Transformation is done into three levels of ISSRM concepts; asset, risk, risk treatment.

Next, validation is done in Chapter 9 with a descriptive case study method [29]. Validation is followed by the conclusion in Chapter 10.

Chapter 2. BUSINESS PROCESS MODELING LANGUAGES

Business Process Modeling Language (BPML) is a language for business process modeling. As the structure of definition process, descriptions including basic graphical elements are given and later with a specific example all these languages are being illustrated. This modeling example is Online Banking and its solutions have features and capabilities in common. The common features will help us to compare the business modeling languages in conclusion and understand why EPC is the language we choose for the alignment and extension process.

2.1 Activity Diagrams

Activity Diagrams [6] define the workflow behavior of a system. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams show activities which are conditional or paralel [6]. Thus, the reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are helpful for *analyzing* a use case by describing what actions need to take place and when they should occur *describing* a complex sequential algorithm and *modeling* applications with parallel processes [6].

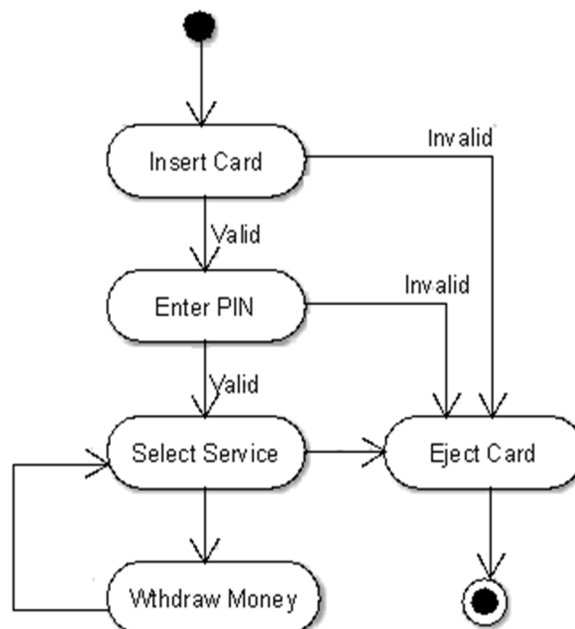





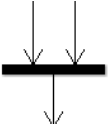
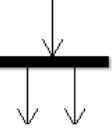

Figure 2.1 – Activity Diagram of withdraw money from ATM.

Activity Diagram displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. If

we illustrate a simple example called “Withdraw Money” based on the Online Banking problem, activity diagram of the model should look like in Figure 2.1. In a single activity diagram, activities belong to one actor. In the example in Figure 2.1, bank user uses ATM to withdraw money and steps of this process shown in order. First, the user inserts his card, if the card is valid user continues his transaction. Later PIN code is controlled, if entered PIN code is valid then user selects the service and withdraws money. Last activity is ejecting card and finishing the transaction.

Besides, activity diagrams fit with security engineering requirements [17]. Activity Diagrams are widely used in security engineering since activities are easy to be controlled and their structure is simple so that in security level they are useful to define security requirements of the corresponding model [17].

Table 2.1 – Legend for Activity Diagrams.

Form	Name	Description
	Activity	The rounded rectangles represent activities that occur. An activity may be physical.
	Initial Node	The filled in circle is the starting point of the diagram. An initial node isn't required although it does make it significantly easier to read the diagram.
	Final Node	The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes.
	Join	A black bar with several flows entering it and one leaving it. All flows going into the join must reach it before processing may continue. This denotes the end of parallel processing.
	Fork	A black bar with one flow going into it and several leaving it. This denotes the beginning of parallel activity.
	Flow	Defines the execution order of activities.

2.2 Petri Nets

Petri Nets is a basic model of parallel and distributed systems which is designed by Carl Adam Petri in 1962 [7] based on his PhD Thesis titled *Kommunikation mit Automaten*. The idea is to describe state changes in a system with transitions. Petri nets contain places and transitions that may be connected by directed arcs. In general, transitions might fire if there are tokens on corresponding places. Firing transitions will remove tokens and place new tokens on new places.

According to the concept *Place/Transition Nets*, we can describe Petri nets and their firing rules as follows; a place might have several tokens which may be interpreted as resources and there might be several input and output arcs between a place and a transition. Besides, the number of these arcs is shown as the weight of a single arc. A transition is enabled if each input place of it contains at least as many tokens as the corresponding input arc

weight indicates. When an enabled transition is fired its input arc weights are subtracted from the input place markings and its output arc weights are added to the output place markings. Furthermore, we illustrate “Withdraw Money” process of Online Banking problem in Figure 2.2.

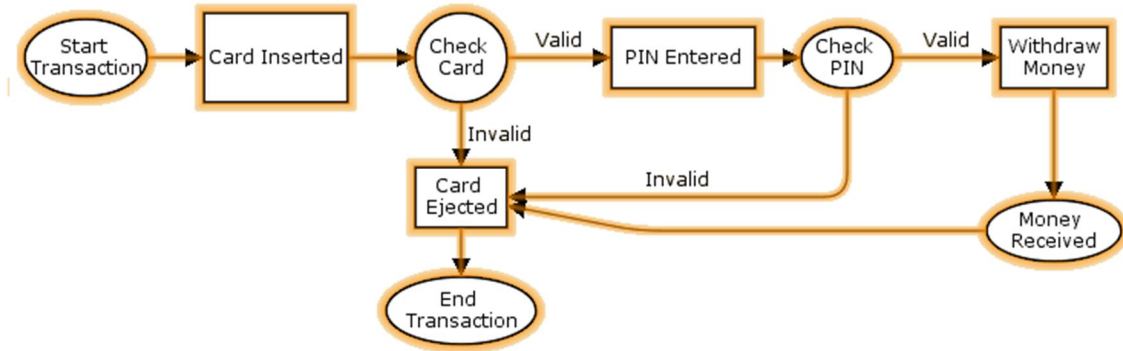


Figure 2.2 – Plain Petri Net Model of withdraw money from ATM.

Table 2.2 – Legend for Petri Nets.

Form	Name	Description
□	Transition	Symbolises actions and a place symbolises states or conditions which need to be met before an action can be achieved.
○	Place	Contains tokens that may move to other places by executing firing actions.
●	Token	Represents the object.
→	Flow	Defines the execution order of transitions and places.

Petri Nets does not have any documentation researches related to security engineering on web, consequently, it makes sense to consider Petri nets as an impractical business process modeling language in security engineering. Although Petri Nets have existed for many decades, they have been recently used to verify cryptographic and security protocols and still needs to be improved [7]. Thus, Petri Nets might be used in the analysis of security protocols and it is recommended to combine different cryptographic algorithms together in the analysis [7].



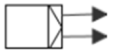
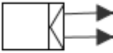




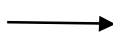


2.3 Yet Another Workflow Language (YAWL)

Yet Another Workflow Language (YAWL) is a fully open-source workflow system or *business process management system* which is based on a workflow definition language. It is capable of capturing all sorts of flow dependencies between tasks.

YAWL has open interfaces based on Web standards which enable developers to plug-in existing applications and to extend and to customise the system in many different ways [8]. It provides a graphical editor with built-in verification functionality which significantly helps developers to takeover workflow models and to detect subtle. Besides, YAWL is, without doubt, the most mature open-source workflow management systems around [8].

Starting from its beginnings as a prototype, YAWL has developed an enterprise-grade workflow engine regards to contributions from organisations and individuals who have used it. As a result, this issue demonstrated commitment from its users and community of developers also ensures the continuity of the system.

Table 2.3 – Legend for YAWL Modeling.

Form	Name	Description
	Starting Point	An input condition which acts as the starting point.
	End Point	An output condition which signals the end.
	XOR-Split	Used to trigger only one outgoing flow. It is best used for automatically choosing between a number of possible exclusive alternatives once a task completes.
	AND-Split	Used to start a number of new pieces of work simultaneously. It can be viewed as a specialisation of the OR-Split where work will be triggered to start on all outgoing flows.
	OR-Split	Used to trigger some, but not necessarily all outgoing flows to other tasks. It is best used when we won't know until run-time exactly what concurrent resultant work can lead from the completion of a task.
	AND-Join	Wait to receive completed work form all of its incoming flows before beginning. It is typically used to synchronise pre-requisite activities that must be completed before some new piece of work may begin.
	XOR-Join	Once any work has completed on an incoming flow, a task with an XOR-Join will be capable of beginning work. It is typically used to allow new work to start so long as one of several different pieces of earlier work have been completed.
	OR-Join	Ensures that a task waits until all incoming flows have either finished, or will never finish. OR-Joins are “smart” [8]; they will only wait for something if it is necessary to wait. However, understanding models with OR-joins can be tricky and therefore OR-joins should be used sparingly.
	Flow	Defines the execution order of tasks, conditions and gates.
	Condition	Decides the flow of the process to the next task according to conditions.
	Task	Main elements of the modeling language, indicates the tasks of users.

Basic terms in YAWL are Business Processes, Workflow Application, Workflow Specification, Workflow System, Workflow Engine, Case (Also known as Workflow Instance), Task (Also known as Activity), Work item (Also known as Task Instance), Worklist and Worklist Handler (Also known as Task Management Service). Figure 2.3 illustrates the example workflow of “Withdraw Money” process in Online Banking problem.

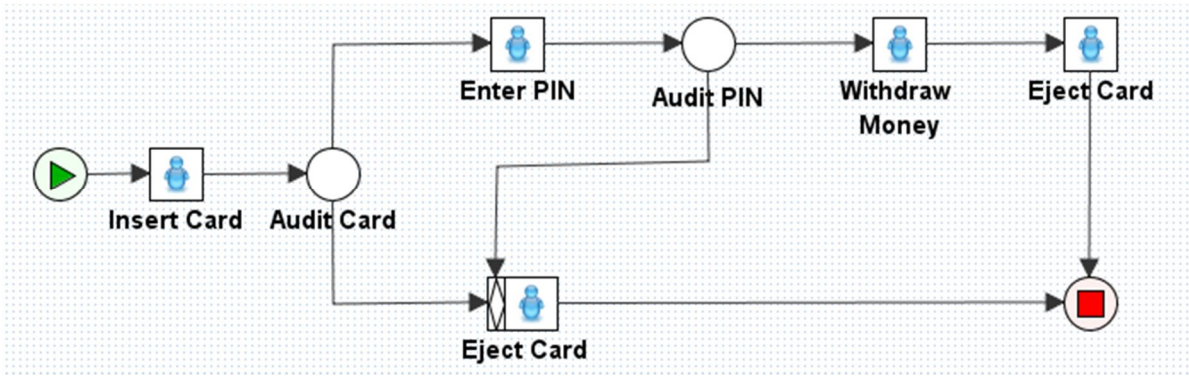


Figure 2.3 – YAWL Model of withdraw money from ATM.

In particular, YAWL is considered as a business process language to be used in security engineering since it is a very prosperous workflow definition language and capable of capturing all sorts of flow dependencies between tasks [18]. Besides, YAWL has developed a confident workflow engine regarding to additions from organisations and individuals who have used it and this workflow engine is also considered as secure [18].

2.4 Business Process Modeling Notation (BPMN)

The Business Process Modeling Notation (BPMN) is a graphical notation which defines the steps in a business process and it is a standard set of diagramming conventions for describing business processes. BPMN is planned in order to visualize a powerful set of process flow semantics within a business process [9].

BPMN is an enabler of Business Process Management (BPM) which is related with the management of business process improvements. The goal of BPMN is to yield a business process modeling notation which is clear by all business users, from business analysts who generate the design of the processes to the technical developers who have the responsibility for implementing the technology that proposed to perform those processes, and lastly to the managers who will manage and monitor those business processes [10].

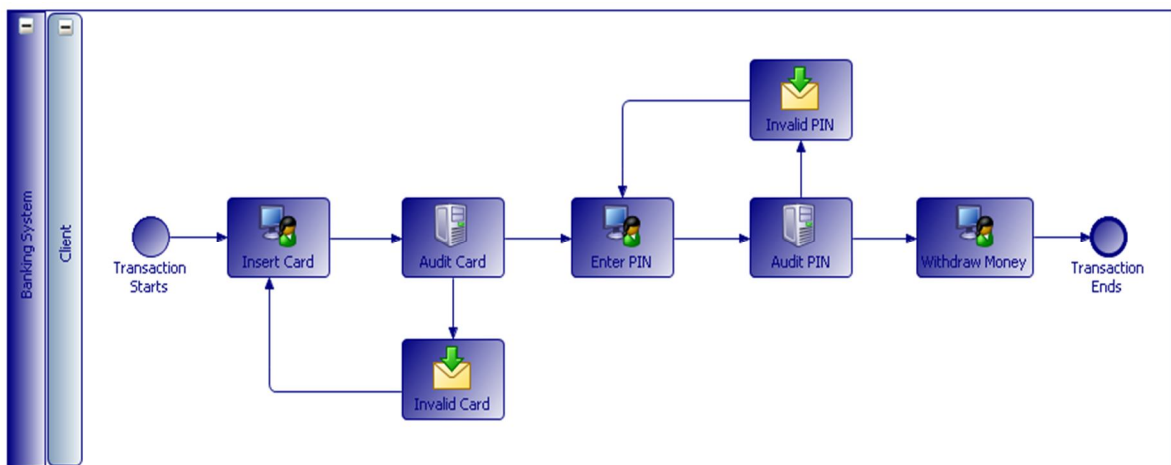













Figure 2.4 – BPMN Model of withdraw money from ATM.

According to the characteristics of BPMN, business process diagrams should be easily read and understood by the business users as well as the process developers should be able to easily read and implement a business process diagram without additional details. BPMN is different from object-oriented modeling techniques since it takes a process-oriented solution to process modeling which is more deputy to the way business analysts model. Therefore, BPMN is determined to provide competence information to let it be the source of an operable process [9].

Table 2.4 – Legend for BPMN.

Form	Name	Description
	Task	A Task is a unit of work, the job to be performed.
	Transaction	A Transaction is a set of activities that logically belong together; it might follow a specified transaction protocol.
	Exclusive Gateway	When splitting, it routes the sequence flow to exactly one of the outgoing branches. When merging, it awaits one incoming branch to complete before triggering the outgoing flow.
	Inclusive Gateway	When splitting, one or more branches are activated. All active incoming branches must complete before merging.
	Parallel Gateway	When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow.
	Start Event	An input condition which acts as the starting point.
	End Event	An output condition which signals the end.
	Sequence Flow	Defines the execution order of activities.
	Event-based Gateway	Is always followed by catching events or receive tasks. Sequence flow is routed to the subsequent event/task which happens first.
	Complex Gateway	Complex merging and branching behavior that is not captured by other gateways.
	Default Flow	The default branch to be chosen if all other conditions evaluate to false.

Structure of BPMN consists of a diagram which is called the Business Process Diagram (BPD). The BPMN Business Process Diagram supplies the ability to model complex business processes but it has been planned in order to be easy to use and to understand. A basic model of a business process workflow consists of the business process starting event, business decisions, workflow branching (gateways) and workflow outputs & results. In

Figure 2.4, the business process workflow indicates a simple “Withdraw Money” process based on Online Banking problem.

BPMN does not explicitly consider mechanisms to represent security requirements [15]. However, among the set of symbols used for the construction of the business process diagram, artifacts can be used to express such requirements. Basically, BPMN opens an opportunity to incorporate security requirements which allows us to improve this aspect of the systems from early stages into software development [15].

2.5 Event-Driven Process Chain (EPC)

Event-Driven Process Chain (EPC) is a method to visualize events and functions by which the logical timing of a business process is shown. As a result, event driven process chain is the description method for business processes. Event driven is equivalent to describing the dynamic part of a business process which means that it is stated in which way and at what time a reaction that causes a change should occur. There are events and functions and the ability to distinguish them is a great importance.

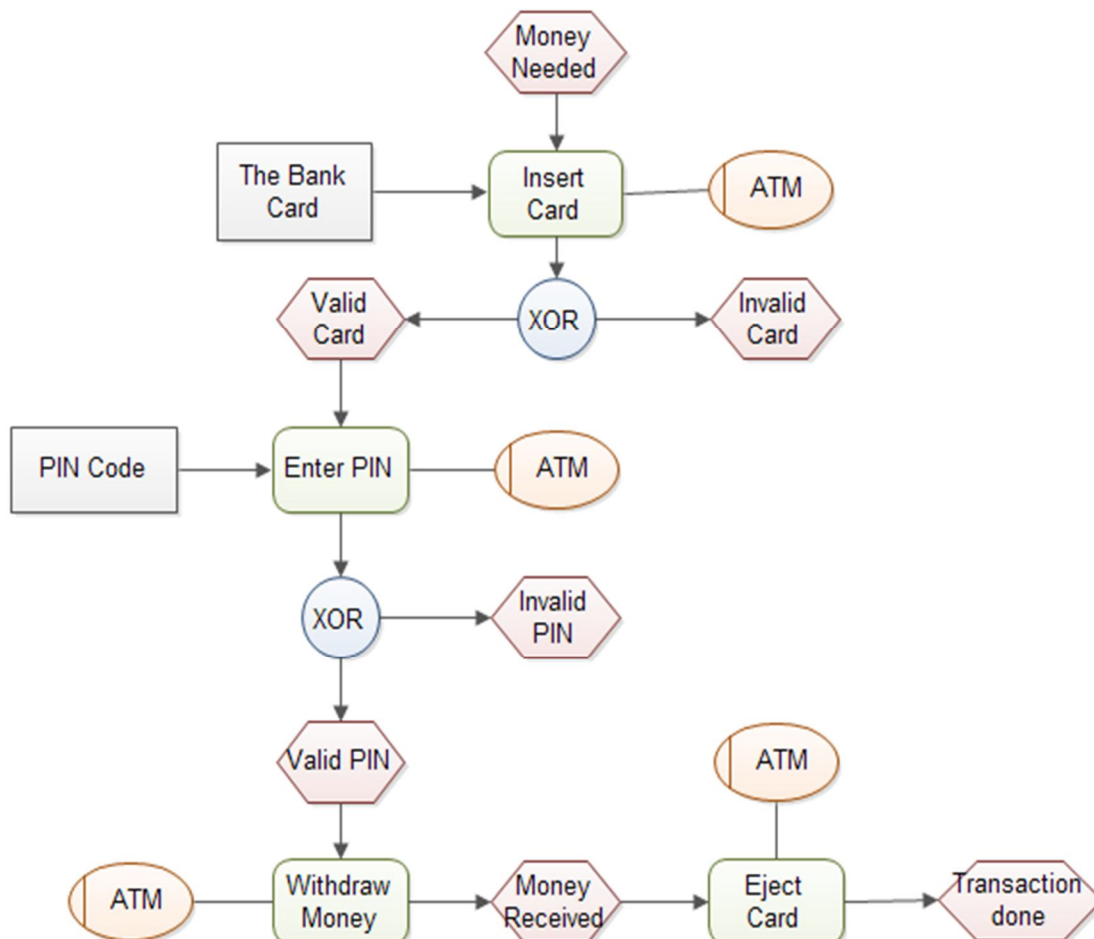
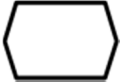
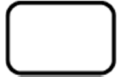





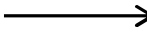





Figure 2.5 – EPC diagram of withdraw money from ATM.

Table 2.5 – Legend for EPC.

Form	Name	Description
	Event	Event is an appearance of an object or change of the expression of an attribute.
	Function	Function is an activity concerning an object to support one or more enterprise objects. Functions refer to an action that uses time and they are registered at the function view.
	AND	Two events / functions must occur.
	OR	One event / function AND / OR another event / function must occur.
	XOR	(= exclusive OR) Either an event / a function OR another event / function must occur.
	Control Flow	Control flow connects the Events, Functions and Decision Gates.
	Organisation Unit	An organisation unit determines which person or organisation within the structure of an enterprise is responsible for a specific function.
	Information Flow	Information flow show the connection between functions and input or output data.
	Assignment	Assignments show the connection between an organisation unit and the function.
	Resource Unit	Resource unit can be input data serving as the basis for a function, or output data produced by a function.
	Process Path	A process path shows the connection from or to other processes.

2.5.1 Introduction to Event-Driven Process Chain

The integration of the data view into event-driven process chain can be defined as functions work on data as input data change to output data and they produce events (data-state changes). According to the Online Banking problem, “Withdraw Money” example is illustrated in Figure 2.5. In EPC form there are events which are shown before/after functions, and related to the functions assigned organisational units are shown as well. Another difference can be seen that some decision gates such as AND, OR and XOR are used here.

Furthermore, based on security engineering, event-driven process chain is not efficient compared to other business process modeling languages since there are not any researches or investigation done about this topic. As a result, EPC notation is mostly used for business modeling purpose without considering the security engineering requirements.

2.5.2 Meta-Model of EPC and Construct Definitions

In previous section, EPC is defined with an illustrated online banking example. In this section, EPC and its Meta-Model will be defined. Moreover, construct definitions will be given.

Metamodeling is the construction of a collection of concepts within a certain domain. A model is an abstraction of phenomena in the real world and a metamodel is yet another abstraction, highlighting properties of the model itself. A model conforms to its metamodel in the way that a computer program conforms to the grammar of the programming language in which it is written.

The Event-Driven Process Chain (EPC) was developed in 1992 at the Institute for Information Systems in Saarbruecken in cooperation with SAP AG. EPC-models are central elements of BPM last but not least due to its use in the SAP R/3 reference model of SAP AG and the ARIS Toolset of IDS Scheer AG. Enterprises model their process data as EPC-models in order to plan, design, simulate and control private enterprise processes. The EPC is a core part of the ARIS-framework and has a big role in combining the different views towards the description of enterprises and information systems in the control view on the conceptual level [24] [25].

One of the main steps of the alignment process of EPC and ISSRM will be the notation of the Meta-Model of EPC in UML Class Diagram model since the domain model of ISSRM is defined as a UML Class Diagram model. ISSRM domain model will be shown and explained in Chapter 4. Figure 2.6 shows the Meta-Model of EPC. This UML Class Diagram based Meta-Model is exactly structured by the constructs of EPC.

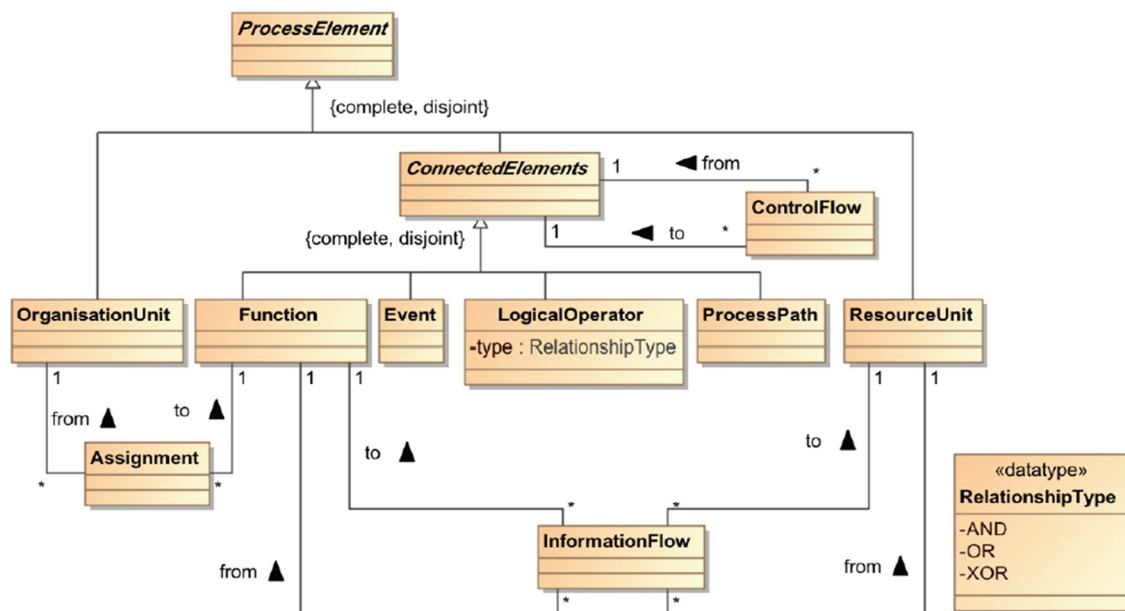


Figure 2.6 – EPC Meta-Model.

In Figure 2.6, if we consider the “Process Element” class, every process element is part of exactly one process and each process consists of one or more process elements. The class process element can be used to create hierarchies of process models. Therefore a function

is detailed by a sub-process. This refining of functions with sub-processes can be done over an unlimited number of levels. One process element can refine no or many functions where a function can either be not refined or be refined by exactly one process element.

“Organisation Unit” class can be assigned to functions. An organisation unit such as applications systems or documents has its resource specific type relation to a function. For example, an organisational unit may have the type of relation “is responsible for”, which is not allowed for a document.

There are nine main constructs exist in EPC and these constructs are shown in previous section in Table 2.5. Below these constructs are defined:

Events are passive elements in EPC. They describe under what conditions a function or a process works or which state a function or a process results in. In particular, an EPC diagram must start with an event and end with an event.

Functions describe transformations from an initial state to a resulting state. In case different resulting states can occur, the selection of the respective resulting state can be modeled explicitly as a decision function using logical connectors. Functions can be refined into another EPC. In this case it is called hierarchical function.

Organisation units decide which person or organisation within the structure of an enterprise is responsible for a specific function.

Resource units can be the information, material, or resource objects portray objects in the real world, for example business objects, entities, etc., which can be input data serving as the basis for a function, or output data produced by a function.

In the EPC the logical relationships between elements in the control flow, that is, events and functions are described by logical connectors. With the help of logical connectors it is possible to split the control flow from one flow to two or more flows and to synchronize the control flow from two or more flows to one flow. There are three kinds of logical relationships defined in EPC:

- **Branch/Merge** : Branch and merge correspond to making decision of which path to choose among several control flows. A branch may have one incoming control flow and two or more outgoing control flows. When the condition is fulfilled, a branch activates exactly only one of the outgoing control flows and deactivates the others. The counterpart of a branch is a merge. A merge may have two or more incoming flows and one outgoing control flow. A merge synchronizes an activated and the deactivated alternatives. The control will then be passed to the next element after the merge. A branch in the EPC is represented by an opening XOR, whereas a merge is represented as a closing XOR connectors.
- **Fork/Join** : Fork and join correspond to activating all paths in the control flow concurrently. A fork may have one incoming control flow and two or more outgoing control flows. When the condition is fulfilled, a fork activates all of the outgoing control flows in parallel. A join may have two or more incoming control flows and one outgoing control flow. A join synchronizes all activated incoming control flows.

- OR : An 'OR' relationship corresponds to activating one or more paths among control flows. An opening 'OR' connector may have one incoming control flow and two or more outgoing control flows.

A control flow connects events with functions, process paths, or logical connectors creating chronological sequence and logical interdependencies between them.

Information flows show the connection between functions and input or output data, upon which the function reads changes or writes.

Organisation unit *assignments* show the connection between an organisation unit and the function it is responsible for.

Process paths serve as navigation aid in the EPC. They show the connection from or to other processes. To employ the process path symbol in an EPC diagram, a symbol is connected to the process path symbol, indicating that the process diagramed incorporates the entirety of a second process which, for diagrammatic simplicity, is represented by a single symbol.

2.5.3 Integrity Constraints in Meta-Model of EPC

Integrity constraints are used to ensure accuracy and consistency of data in a relational database. Data integrity is handled in a relational database through the concept of referential integrity. In EPC Meta-Model, we also identify the integrity constraints due to prevent conflicts about the relationship between the Connected Elements abstract class and the Control Flow class. As it is seen from the model, control flow is the actor of connecting each connected elements to each other. However, there are some connected elements which can not follow or connected to each other during the actual process, in this manner we identify specific elements and their restrictions. In particular, as well as two different events can not be connected to each other, two different functions can not. Events are followed by functions and functions are followed by events.

2.6 Comparison and Summary

In this chapter, five different business process modeling languages (Activity Diagrams [6], EPC [24] [25], Petri Nets [7], YAWL [8] and BPMN [9]) are analysed based on a structure containing the general description of the language, introduction of the principles of the language, application process of the language and the significance of the language according to the security engineering. Table 2.6 indicates the major differences of these five business process modeling languages based on two different criterias; Complexity and Security Coverage. Complexity refers to number of constructs which are described in legend tables of languages and Security Coverage refers to the percentage of usage of these constructs in security criterion. This percentage is calculated through the ratio between the total construct number shown in a legend table of a language and the total construct number which are considered to be related with security criterion. In general, the ratios are, order based on the Table 2.6, (Initial Node, Final Node, Join, Fork, Flow) 5/6, 0/6, 0/4, (Starting Point, End Point, XOR-Split, AND-Split, OR-Split, AND-Join, XOR-Join, OR-Join, Flow, Condition) 10/11, 0/11.

Table 2.6 – Comparison of Business Process Modeling Languages.

<i>Criteria\Language</i>	Activity Diagrams	EPC	Petri Nets	YAWL	BPMN
Complexity	6	6	4	11	11
Security Coverage	83%	0%	0%	91%	0%

In particular, during the answering the question “*How does the overview above contribute to the research question?*”, security coverage is the key point. According to the security coverage percentage of each language, we assume that the alignment of the language which has low percentage will be more complex than the one which has high percentage. Because during the concept alignment, security risk analysis will be performed on language constructs based and as long as these constructs are covered by security criteria the process will be clear. Consequently, according to the research question, alignment of EPC, Petri Nets and BPMN with ISSRM domain model will be more challenging than the alignment of Activity Diagrams and YAWL. Besides, this comparison also showed us that EPC is not helpful to elicit security concerns.

Chapter 3. SECURITY MODELING LANGUAGES

Security modeling languages are the modeling languages which support security requirements. In this chapter, KAOS extension to Security [2], Misuse Cases [3] and Mal-Activity Diagrams [5] are defined. Descriptions include basic graphical elements and later with an example all these languages are being illustrated. This modeling example is Online Banking and its solutions have features and capabilities in common. In the end, based on common and unique capabilities of the languages, a comparison is made in order to choose one of the languages to use during the Security-Oriented EPC transformation.

3.1 KAOS Extention to Security

To define KAOS Extention to security, we present a requirements engineering method for elaborating security requirements based on the incremental building and specification of two concurrent models: [2] an intentional model of the system-to-be and an intentional anti-model yielding vulnerabilities and capabilities required for achieving the anti-goals of threatening security goals from the original model. After the procedure, the original model is enriched with new security requirements derived as countermeasures to the anti-model. This approach extends the KAOS framework for goal-oriented requirements engineering in several ways which are mentioned below [2]:

- *it extends the specification language,*
- *it provides additional specification patterns for formal elicitation of candidate security requirements to start the analysis,*
- *it introduces a duality principle for richer modeling of threats; system goals, requirements, expectations software services, implementable anti-requirements and software vulnerabilities.*

The elements of KAOS modeling are; Goal, Requirement, Operation, Agent, Object, Domain, Input, Output, Performance, G-Refinement, Alternative G-Refinement, Responsibility and Operationalisation. In particular, illustration of KAOS Extention to Security based on Online Banking problem shown in Figure 3.1 (customer side) and Figure 3.2 (attacker side). To analyse the problem, the risk management process is applied. Later, security objectives are determined. It results in introduction and elaboration of new goals. For example, goal Avoid[Account# and PinKnownByThief] addresses the confidentiality of the business assets like Pin and Account number. Next, risk analysis and assessment is considered by negating the goal.

In security engineering, the application layer has received much less attention to date compared with the crypto, protocol and system/language layers [2]. For security assurance at this layer as a precondition, analysts have to ensure that application-specific security requirements are made clear, accurate, competence and non-conflicting with other

requirements and complete. We presented the requirements engineering method for this security assurance at the very beginning of our definition.

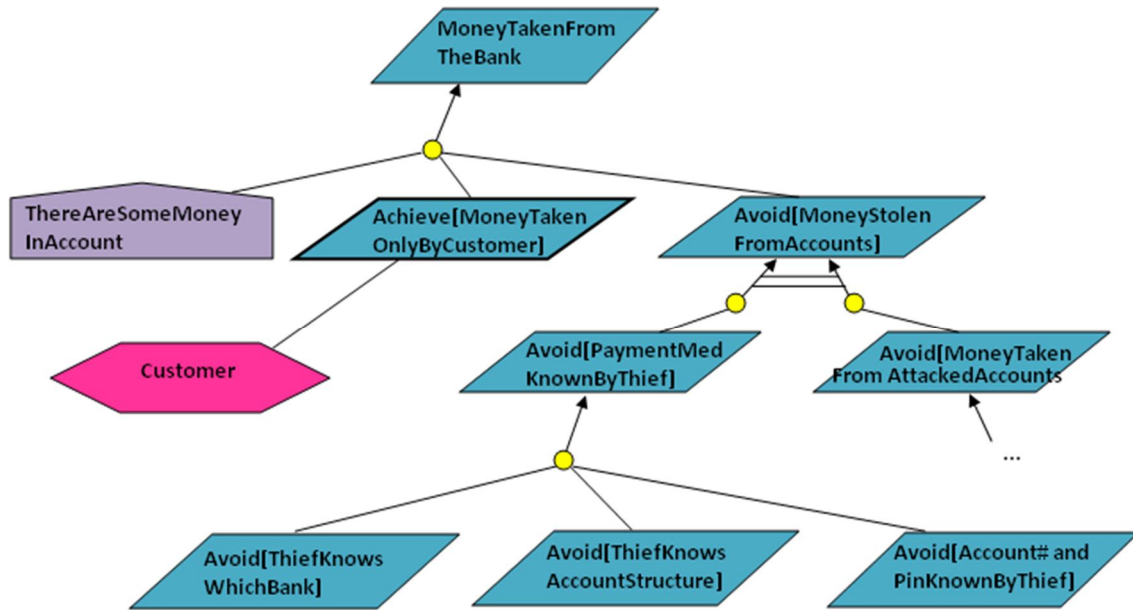


Figure 3.1 – Online Banking Problem - Extended Operational Model (Customer Side).

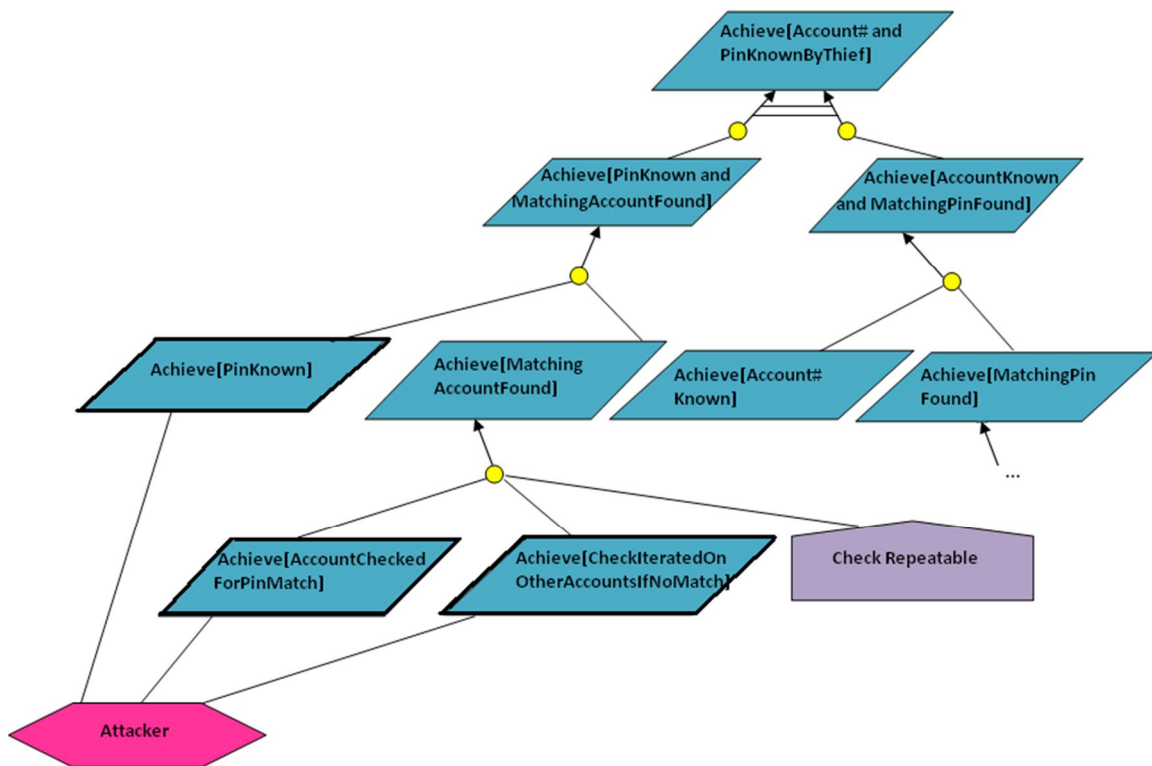




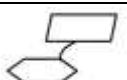
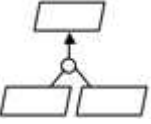
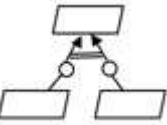




Figure 3.2 – Online Banking Problem - Risk Analysis and Assessment Model (Attacker Side).

Table 3.1 – Legend for Kaos Extension to Security.

Form	Name	Description
	Goal	A prescriptive statement of intent about some system whose satisfaction in general requires the cooperation of some of the agents forming that system.
	Requirement	A terminal goal under responsibility of an agent in the software-to-be.
	Agent	Active components such as humans, devices, legacy software or software-to-be components that play some <i>role</i> towards goal satisfaction.
	Domain Prop.	Descriptive statements about the environment such as physical laws, organisational norms or policies, etc.
	Responsibility	Relationship between agent and goal which refers to responsibility of agent
	G-Refinement	Relate a goal to a set of subgoals (called refinement) possibly conjoined with domain properties.
	Alternative G-Refinement	Relate a goal to a set of alternative refinements.
	Performance	Shows the operations performed by an agent.
	Operation	Actions performed by an agent.

3.2 Misuse Cases

A Misuse Case is Use Case from the point of view of an actor hostile to the system under design which turns out to have many possible applications, [3] and to interact with Use Cases in useful ways. Some misuse cases exist in highly specific situations and some others continually threaten systems. It is possible to develop misuse and use cases recursively, going from system to subsystem levels. Lower-level cases can highlight outlooks not considered at higher levels, which may cause another analysis. The approach offers rich possibilities for exploring, [4] understanding and validating the requirements in any direction.

Drawing the agents and misuse cases explicitly clarifies focus attention on the elements of the scenario. Besides, functional (depends on the design) and non-functional (Reliability, Maintainability, Portability, Testability and so on) requirements exist in Misuse Cases. Basic elements of the model are; (Mis)Users and (Mis)Use Cases and their relationships such as treatment and mitigation.

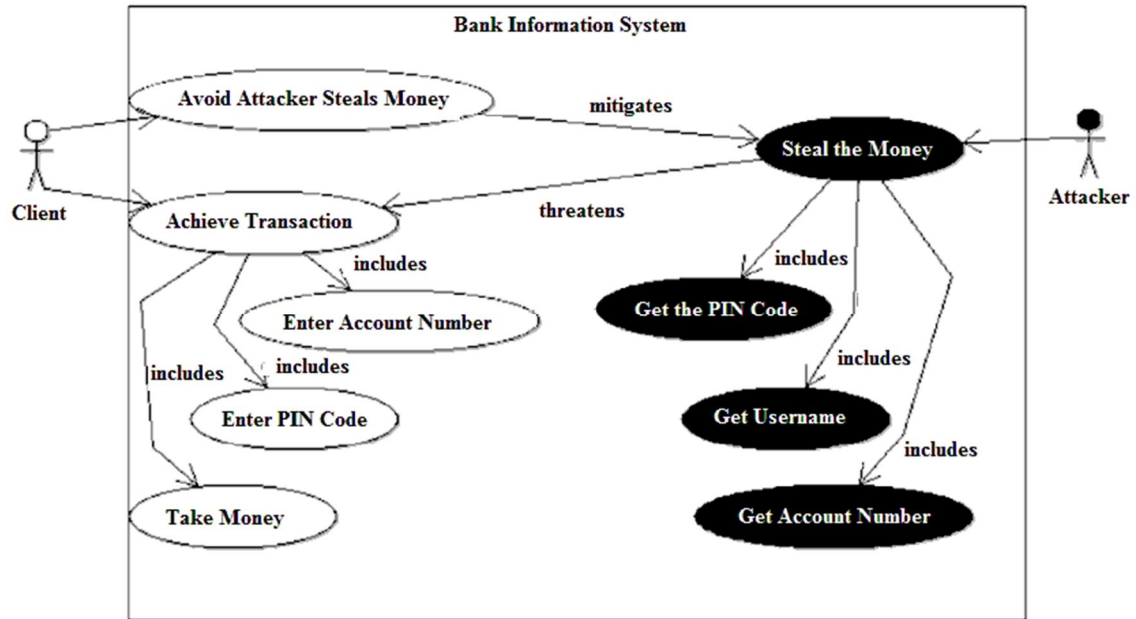


Figure 3.3 – Online Banking, Use Cases (on the left) and Misuse Cases (on the right) based on Security Requirements.

Table 3.2 – Legend for Misuse Cases.

Form	Name	Description
	Use Case	A description of a potential series of interactions between a software module and an external agent, which lead the agent towards something useful.
	Misuse Case	A sequence of actions, including variants, that a system or other entity can perform, interacting with misusers of the entity and causing harm to some stakeholder if the sequence is allowed to complete.
	Actor (User)	An actor that initiates use cases.
	Actor (Misuser)	An actor that initiates misuse cases, either intentionally or inadvertently.
Mitigate 	Use case mitigate misuse case	The use case is a countermeasure against a misuse case, i.e., the use case reduces the misuse case's chance of succeeding.
Threaten 	Misuse case threaten use case	The use case is exploited or hindered by a misuse case.
	Include	Shows the subcases of a case in detailed.

In Figure 3.3, the Use Case (client side) and the Misuse Case (attacker side) diagrams are shown according to Online Banking problem. As it is seen in the figure, the client can

achieve the money transaction including necessary processes to succeed. Accordingly, the client has to avoid attacker steal his money by the mitigation process. Since the attacker is misuser, his acts are shown as misuse cases in black circles which threaten the acts of the client.

To elicit security requirements with misuse case, following five processes are proposed; identifying critical assets in the system, defining security goals for each asset, identifying threats, identifying and analyzing risks and defining security requirements. As long as these five significant processes are successfully run, Misuse Cases might be very effective in security engineering in many different projects such as [3] Knowledge Map Application in an EU-funded Research Project, Open Web Application Security Project, E-shop and Telemedicine Projects and so on.

3.3 Mal-Activity Diagrams

Mal(icious)-Activity Diagrams are same with ordinary UML Activity diagrams with their syntax and semantics, additionally activities in Mal-Activity Diagrams are shown with icons which are the inverse of normal activity icons. Besides, actors are indicated with swim-lanes where the actor name is shown inverse [5]. Also in Mal-Activity Diagrams decision boxes are shown as the inverse of normal decision boxes.

The Mal(icious)-Activity Diagram is not the only notation utilizing inverted icons to indicate security threats [5]. For instance, considering the difference between mal-activity diagrams and misuse case diagrams, the main difference is the same as the difference between normal activity diagrams and use case diagrams, which means that they are both useful for separate purposes. Besides, Mal-Activity diagrams would not indicate sequences of activities like an activity diagram, and also not exactly where a certain malicious activity might fit into a business process or how the process could be changed to deal with it.

If we illustrate the Mal-Activity model of Online Banking problem we obtain a diagram shown in Figure 3.4. There are three actors with their swimlanes. Client performs activities in order to transact by using ATM banking system while attacker performs activities indicating the threat and cases of security failure during the transaction process. Besides, attacker performs activities where he can steal the bank card and PIN information in order to use malicious online transactions.

Based on security engineering, Mal-Activity diagrams are convenient to clarify situations and make assumption by relating it with security requirements [16]. Besides, Mal-Activity diagrams are also used in social engineering since they are considered as one of the best appropriate way to describe social engineering attacks [16].

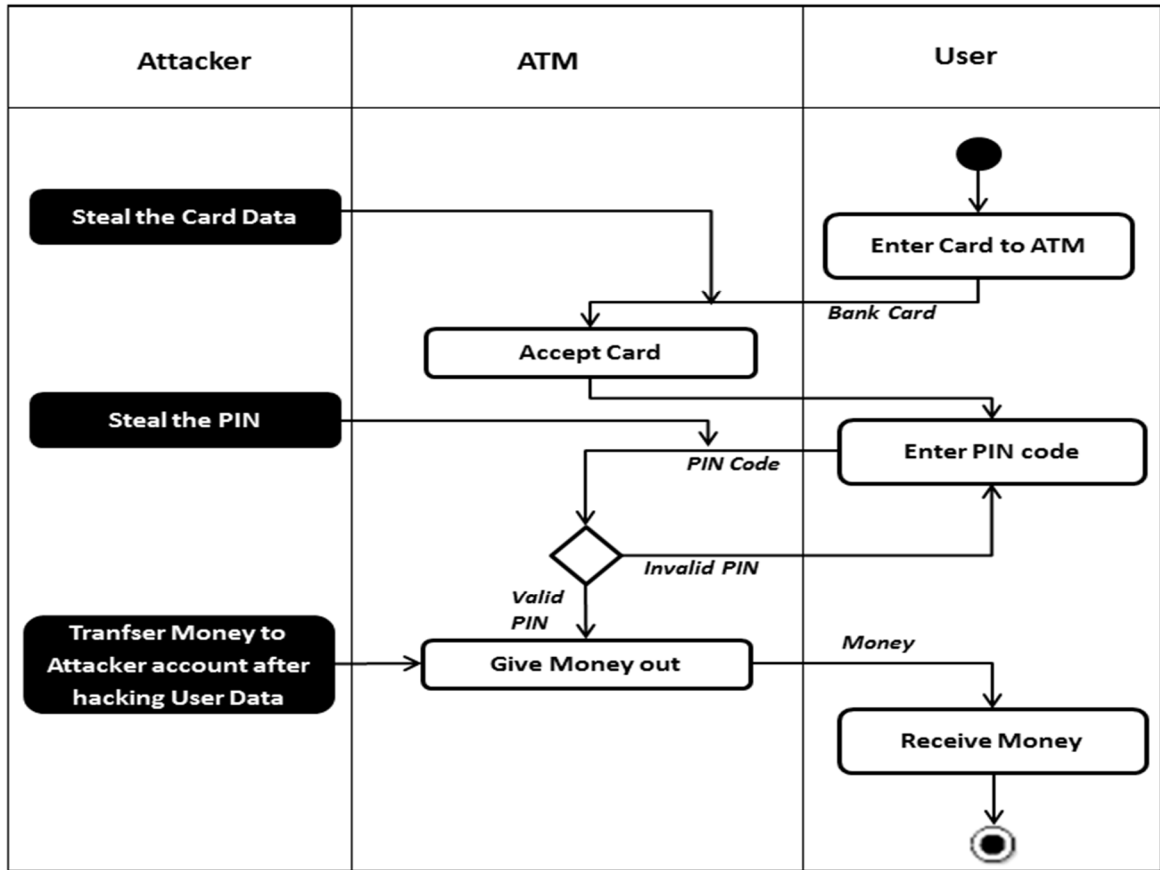


Figure 3.4 – Online Banking, Mal-Activity diagrams of client and attacker in banking system.

Table 3.3 – Legend for Mal-Activity Diagrams.

Form	Name	Description
	(Mal) Swimlane	An actor initiates activities.
	Activity	The rounded rectangles represent activities that occur. An activity may be physical.
	Mal-Activity	The rounded rectangles represent malicious activities that occur. A malicious activity may be physical.
	Initial Node	The filled in circle is the starting point of the diagram. An initial node isn't required although it does make it significantly easier to read the diagram.
	Final Node	The filled circle with a border is the ending point. An activity diagram can have zero or more activity final nodes.
	Decision	Decision gates which are used where different decisions can occur.
	ControlFlow	Defines the execution order of activities.

3.4 Comparison and Summary

Table 3.4 indicates the major differences of three security modeling languages (KAOS Extention to Security [2], Misuse Cases [3] and Mal-Activity Diagrams [5]) based on two different criterias; Complexity and Security Coverage. Complexity refers to number of constructs which are described in legend tables of languages and Security Coverage refers to the percentage of usage of these constructs in security criterion. This percentage is calculated through the ratio between the total construct number shown in a legend table of languages and the total construct number which are considered to be related with security criterion. In general, the ratios are, order based on the Table 3.4, (Goal, Requirement, Agent, Domain Prop., Responsibility, G-Refinement, Alternative G- Refinement for KeS) 7/9, (Misuse Case, Actor/Misuser, Use case mitigate Misuse Case, Misuse case threaten Use Case, Include for Misuse Cases) 5/7, (Mal-Swimlane, Mal-Activity, Initial Node, Final Node, Decision, ControlFlow for MAD) 6/7.

Table 3.4 – Comparison of Security Modeling Languages.

<i>Criteria\Language</i>	KAOS Extention To Security	Misuse Cases	Mal-Activity Diagrams
Complexity	9	7	7
Security Coverage	78%	71%	85%

According to the security coverage percentage of each language, we assume that the alignment of the language which has low percentage will be more complex than the one which has high percentage. Since during the concept alignment, security risk analysis will be performed on language constructs based and as long as these constructs are covered by security criteria the process will be less complex. Consequently, alignment of Mal-Activity Diagrams with ISSRM domain model will be less challenging than the alignment of Misuse Cases and KeS. However, it is necessary to mention that these security coverage percentages are based on the constructs of the language. It definitely does not mean that if a language has less percentage then it will have less efficiency of alignment, because there are other concepts and definitions which affect the alignment process. Also, since MAD has the highest constructual based security coverage, we choose it as a potential language during the transforming constructs from extended EPC (Security-Oriented EPC) to a security modeling language.

Chapter 4. SECURITY RISK MANAGEMENT

Security Risk Management is the method used as the path to reach reasonable and appropriate spending and management of security controls. This chapter is divided into three parts; Model-Based Security Analysis in seven steps with CORAS method [19], Goal-Risk Framework [20] and Information System Security Risk Management (ISSRM [11]).

4.1 Model-Based Security Analysis with CORAS Method

A security risk analysis provides answers to many questions. CORAS is a method for managing security risk analysis and it provides a customised language for threat and risk modeling, and comes with detailed guidelines explaining how the language should be used to capture and model relevant information during the various stages of the security analysis [19]. In addition, it is necessary to mention that CORAS is model-based and Unified Modeling Language (UML) can be used to model the target of the analysis. Seven steps of security analysis in CORAS are summarised below and the structure of each step includes *tasks*, *participants* of these tasks and *the modeling guideline*.

Step 1 – Introductory Meeting

Tasks: The security analysis method is introduced, then client presents goals and the target of the analysis, also the focus and scope of the analysis is set, lastly meetings and workshops are planned.

Participants: Analysis Leader (required), Analysis Secretary (required), Representatives of the Client.

Modeling guideline: System description.

Step 2 – High-Level Analysis

Tasks: The target as understood by the analysts is presented, the assets are identified and a high-level analysis is conducted.

Participants: Security Analysis Leader (required), Security Analysis Secretary (required), Representatives of the Client.

Modeling guideline: Asset diagrams, Target descriptions.

Step 3 – Approval

Tasks: The client approves target descriptions and asset descriptions, the assets should be ranked according to importance, consequence scales must be set for each asset within the scope of the analysis, a likelihood scale must be defined and the client must decide risk evaluation criteria for each asset within the scope of the analysis.

Participants: Security Analysis Leader (required), Security Analysis Secretary (required), Representatives of the Client.

Besides, no modeling guidelines are defined in this step.

Step 4 – Risk Identification

Tasks: The initial threat diagrams should be completed with identified threats, vulnerabilities, threat scenarios and unwanted incidents.

Participants: Security Analysis Leader (required), Security Analysis Secretary (required), Representatives of the Client.

Modeling guideline: Threat diagrams.

Step 5 – Risk Estimation

Tasks: Every threat scenario must be given a likelihood estimate and unwanted incident likelihoods are based on these and every relation between an unwanted incident and an asset must be given a consequence estimate.

Participants: Security Analysis Leader (required), Security Analysis Secretary (required), Representatives of the Client.

Modeling guideline: Risk estimation on threat diagrams.

Step 6 – Risk Evaluation

Tasks: Likelihood and consequence estimates should be confirmed or adjusted, the final adjustments of the acceptable area in the risk matrices should be made and an overview of the risk may be given in a risk diagram.

Participants: Security Analysis Leader (required), Security Analysis Secretary (required), Representatives of the Client.

Modeling guideline: Risk diagrams.

Step 7 – Risk Treatment

Tasks: Adding treatments to threat diagrams, estimating the cost/benefit of each treatment and decide which ones to use and showing treatments in risk overview diagrams.

Participants: Security Analysis Leader (required), Security Analysis Secretary (required), Representatives of the Client.

Modeling guidelines: Treatment diagrams, Treatment overview diagrams.

In conclusion, the focus on security risk of information systems, the heavy use of models to guide and structure the analysis and the specialised language for documenting, and communicating intermediate as well as the final results of the analysis characterise CORAS [19].

4.2 Goal-Risk Framework

Research on Goal-Risk Framework is divided into three major areas [20]: Requirements Engineering, Secure and Dependable Engineering, and Risk Analysis.

In Requirements Engineering, according to KAOS, a goal-oriented requirements engineering methodology aiming at modeling not only what and how aspect of requirements, but also why, who, and when [20]. Later, KAOS introduces also the concept of obstacle and anti-goal to analyze boundary conditions and failure situations for a design. Obstacles can be defined as situations which can lead to goal failure. Anti-goals can be defined as goals associated with malicious stakeholders, such as an attacker. Consequently, it is possible to say that obstacles are unintended risks and anti goals are threats or intended risks. These features make KAOS suitable for analyzing requirements for secure and/or dependable systems. I* modeling framework is extended to analyze risk and security issues

during requirement analysis [20]. The framework models business assets of an organisation and assets of its IT systems. Afterwards, countermeasures are selected to mitigate risks, thereby ensuring that risks will not affect any assets.

In the area of Secure and Dependable Systems, the most popular analysis frameworks are Fault Tree Analysis (FTA), Failure Modes, Effects, and Criticality Analysis (FMECA) [20]. In security engineering, concepts such as attack trees and threat trees are similar to FTA, while other proposals such as UMLSec, SecureUML, Abuse Case, and Misuse Case constitute UML extensions intended to deal specifically with security concerns [20]. The most relevant work to clarify Goal-Risk Framework is the Defect Detection and Prevention (DDP). DDP consists of a three-layer model consisting, respectively of Objectives, Risks, and Mitigations. Each objective has a weight to represent its importance, each risk has a likelihood of occurrence, while every mitigation has a cost for its accomplishment. Severity of a risk can be represented by an impact relationship between an objective and a risk. Besides, a DDP model specifies how to compute the level of objective achievements and the cost of mitigations. This calculation lets one to evaluate the impact of a collection of countermeasures, thereby supporting risk analysis. Also the DDP model can be integrated with other quantitative frameworks (e.g., FMECA, FTA) in order to model and assess risks/failures [20].

In the area of Risk Analysis, uncertain events such as threats and failures are quantified with two attributes: likelihood and severity. Probabilistic Risk Analysis (PRA) is widely used for quantitative risk assessment, while approaches like FMECA quantify risk into qualitative values: frequent, reasonably probable, occasional, remote, and extremely unlikely [20]. Events are prioritized using the notion of “expectancy loss” resulted from them which is defined as the product of its likelihood and severity. Priority here reflects the criticality of an event. When resources are limited, an analyst may decide to adopt countermeasures for mitigating events on the basis of their priority. However, estimation of probabilities is generally imprecise, as they typically strongly depend on expert judgment. Approaches such as Multi-Attribute Risk Assessment can improve the risk assessment process by considering multi-attribute analysis [20]. In this process, many factors that can impact the quality of a system—such as reliability, availability, safety and confidentiality—are analyzed for potential risks [20]. For instance, an Air Traffic Management system is required to always be available and safe. Certain conditions such as radar noise can affect the normal behavior of the system and as a result impact on its safety. In many cases, the best way to deal with radar noise is to restart the system. This, however, impacts on its availability. This inter dependence of quality factors introduces the need for analysis that finds the right tradeoffs [20].

4.3 Information System Security Risk Management (ISSRM)

There are literally hundreds of ISSRM methods and standards exist and generally consist of process guidelines which help identifying defenceless assets, determine security objectives, assess risks and define security requirements to treat the risks. By these methods and standards it is possible to reduce the losses which might result from security problems.

4.3.1 The Domain Model of ISSRM

The domain model of ISSRM, shown in Figure 4.1, consists of concepts related to each other. These domain model concepts are created during the alignment process of basic ISSRM concepts. All these concepts are linked to each other according to their relationships as well as grouped under three principal concepts [11]: *asset*-related concepts, *risk*-related concepts and *risk treatment*-related concepts. Definitions are introduced below.

Asset can be anything which has value to the organisation and is important for achieving its objectives. There are different kinds of assets exist. Business asset is the information, process, skill inherent to the business of the organisation which has value to the organisation in terms of its business model. IS asset is a component of the IS which has value to the organisation and it is significant for achieving its objectives and supporting business assets. An IS asset can be a component of the IT system [11] such as hardware, software or network, but also people or facilities playing a role in the IS and therefore in its security. Security criterion (aka security property) is the property or constraint on business assets which characterises their security needs.

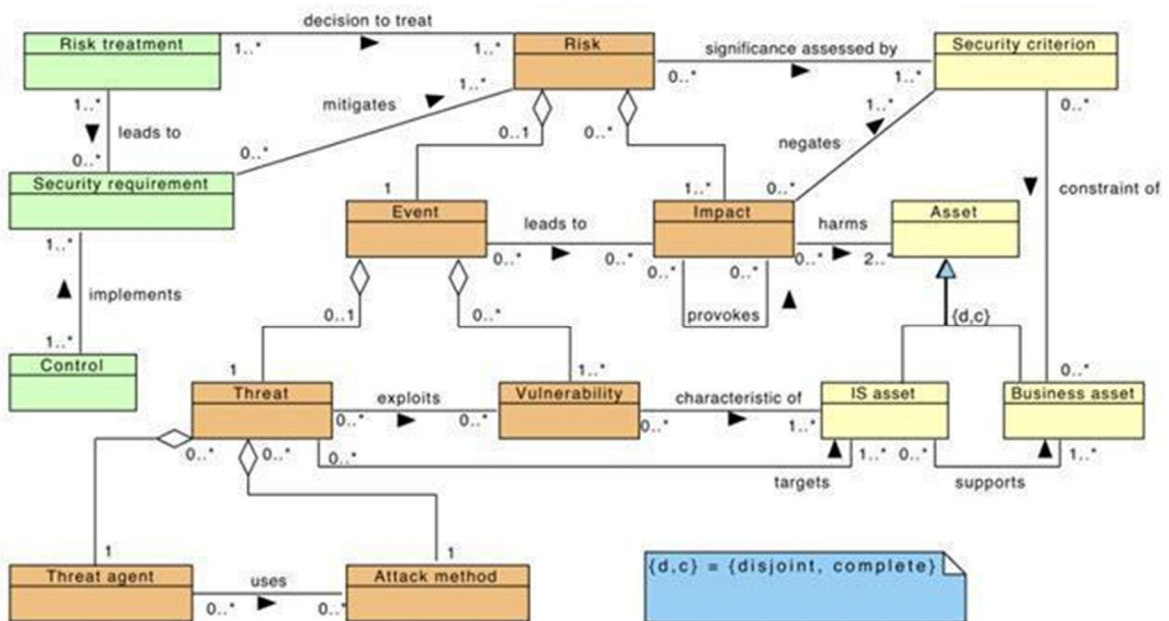


Figure 4.1 – The domain model of ISSRM [11].

Risk is the synthesis of a threat with one or more vulnerabilities causing to a negative impact harming one or more of the assets. Threat and vulnerabilities are part of the risk event and impact is the result of the risk. As it is mentioned during the definition of Risk, Impact is the negative result of a risk which may damage assets of a system or an organisation, when a threat or an event is successful. It is possible to describe the impact at the level of IS assets or at the level of business assets where it negates security criteria [11]. We can define Event as the synthesis of a threat and one or more vulnerabilities. Besides, Vulnerability is the characteristic of an information system asset or group of information system assets that can form a weakness in terms of information system security. Threat is the potential attack which is achieved by an agent, that targets one or more information system assets and that may cause to harm to assets. Also a threat is formed of a threat agent and an attack method. Related to Threat, Threat agent is an agent

which can lead harm to assets of the information system. A threat agent triggers a threat and is also the source of a risk. Attack method is the standard means by which a threat agent carries out a threat.

Risk treatment is defined as the decision of how to treat the specifically identified risks. The possible treatment should satisfy a security need which is expressed in generic and functional terms [11]. Categories of risk treatment decisions include; *Avoiding* the risk (risk avoidance decision), *Reducing* the risk (risk reduction decision), *Transferring* the risk (risk transfer decision), *Retaining* the risk (risk retention decision). There is a security requirement during the treatment process and it can be defined as a condition over the phenomena of the stage where it is purposed to make true by installing the information system in order to reduce risks. Lastly, Control (aka countermeasure or safeguard) is specified by a specific security requirement and it is designed to improve the security and implemented to fit with that requirement. Security controls can be processes, policies, devices, practices or other actions or components of the information system.

4.3.2 Risk Management Process

The application of the Risk Management Process is shown step by step in Figure 4.2. Below all these steps are described separately in order [11]:

Context and asset identification can be defined as the description of organisation and its environment. Sensitive activities related to information security such as design of technical plans can be counted as an example of this step.

Determination of security objectives is the process of determining the security objectives to be reached. During this work Confidentiality, Integrity and Availability are significant concepts. If we illustrate an example in this step it is possible to mention that during the design, technical plans should be kept confidential.

Risk analysis and assessment is the process of identifying risks and estimating them qualitatively or quantitatively. A rival of tries to use common operating system and network protocol weaknesses to penetrate on the personal computer of an employee can be an example of this step.

Risk treatment process consists of risk avoidance, risk reduction, risk transfer and risk retention. Reducing the preceding risk with some security controls implemented in the information system is an effective example at this step.

Security requirements definition is basically the process of defining security solutions to mitigate the risks. If security requirements are unsatisfactory, it is essential to revise the risk treatment step and revise all of the preceding steps.

Control selection and implementation is the process of implementing system countermeasures within organisation. The selection and the implementation of a firewall and an Intrusion Detection System (IDS) is an illustrated example of this step.

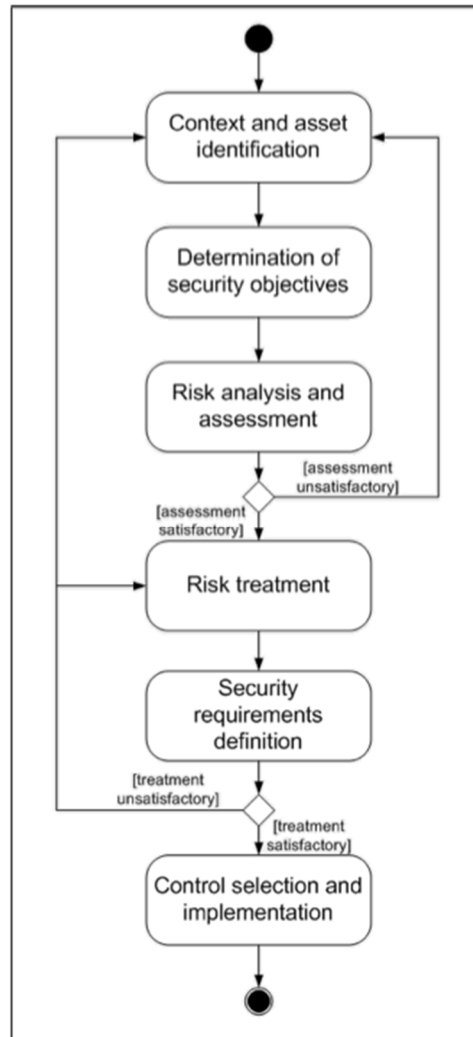


Figure 4.2 – Security Risk Management Process [11].

4.3.3 Analysing Security Modeling Languages with ISSRM

In this section we analyse KeS, Misuse Cases and MAD security modeling languages with ISSRM. During the state of art, in Chapter 3, we have given definitions about these three languages and in comparison and conclusion section we have compared their security coverage based on constructs in high level. Now we will review these languages and their alignment with ISSRM model in lower level.

4.3.3.1 Alignment of KeS and ISSRM Model

The alignment between KAOS Extention to Security (KeS) and ISSRM is analysed in three steps based on the three main concepts of ISSRM:

Asset-related concepts – KeS is basically focused on the security of the system-to-be but it does not make a separation between the information system and business aspects. Besides, the alignment process includes all three ISSRM concepts concerning assets with the KAOS' goal, requirement and expectation. The main idea of the security goals is to guard

system states against unauthorised access. Besides, in terms of KAOS this means that the security goals must describe confidentiality, privacy, integrity and availability goal and object properties which are concerned by potential risk events and threats [13]. Moreover, both goals and object attributes concerned by anti-goal with ISSRM security criteria are aligned.

Risk-related concepts – At higher abstraction levels, an anti-goal might be defined as the event which is a combination of a threat and one or more vulnerabilities. At lower abstraction levels, an anti-goal might be defined as a threat which is a potential attack. In KAOS an anti-agent displays or controls objects and their behaviour. Besides, in KAOS an anti-agent performs operations which satisfy an anti-goal. These operations' role is to change the state of the system-to-be using input/output relationships over the objects and their behaviour which means that by performing these operations the anti-agent breaks the security criteria. Thus, KAOS does not address two concepts from the ISSRM domain model which are; *risk* and *impact*. [13] This situation can be stated by the fact that KAOS was not specifically planned to evaluate the business context of an information system.

Risk treatment-related concepts – ISSRM's risk treatment resembles the countermeasures which are designed detailed after definition of the anti-goals [13]. Countermeasures can be defined as modeling expressions or “patterns” adopted by modelers. In KAOS, the countermeasures generally occur in new security goals which need to be set off in realizable security requirements. The set off operation and operationalisation of the new security goals lead to new system-to-be components recognizing the significant security means. Moreover, according to the ISSRM domain model, these new system-to-be components resemble controls.

4.3.3.2 Alignment of Misuse Cases and ISSRM Model

The alignment between Misuse Cases and ISSRM is analysed in three steps based on the three main concepts of ISSRM:

Asset-related concepts – In misuse case, literature founded confusion seeking a correspondence for the notion of ISSRM security criteria [14]. The most significant assets in an organisation are defined as the knowledge and the skills of the workers but they are only defenceless remote through the misuse of other more concrete assets.

Risk-related concepts – In this level, the risk is defined as the estimated likelihood of occurrence and cost of the damage if the threat occurs. This definition corresponds to definition of risk in ISSRM in terms of concerned concepts [14]. The notion of impact in misuse cases rises as the cost of the damage. Besides, according to the expression “The security threats identified can be described as misuse cases and misusers”, it is possible to mention that this expression corresponds to the ISSRM threat which is composed of a threat agent and an attack method. Moreover, specific correspondences between the misuser who is the actor that initiates misuse case and ISSRM threat agent are identified. Later align the misuse case which is a sequence of actions interacting with misuser and causing harm to stakeholder and the ISSRM attack method is aligned. At the end, threatens relationship which shows how a use case is exploited or hindered by a misuse case can be understood clearly as the target relationship between threat and information system asset.

Risk treatment-related concepts – It is recommended that for each defined threat and taking its risk into account, it is necessary to state requirements to reduce the threat. [14] This recommendation basically means that convenient security requirements should be stated and specified. Also, security requirements identified are specified as independent security use cases and the security use case should have a reduced relationship to a misuse case. It is possible to conclude that security use cases correspond to the ISSRM security requirements. Besides, the misuse case reduces link corresponds to the ISSRM reduces relationship [14]. The misuse cases do not specify anything which would correspond to the ISSRM notions of risk treatment or controls.

4.3.3.3 Alignment of MAD and ISSRM Model

The alignment between Mal-activity Diagrams (MAD) and ISSRM is analysed again in three steps based on the three main concepts of ISSRM:

Asset-related Concepts – As we defined earlier, the ISSRM asset represents something of value for the organisation. The business asset is defined as the information, process or skill that is important to for the business. Activity diagrams are used to show the business workflow by combining constructs together, like: *Activity*, *Decision* and *ControlFlow* [28]. These constructs are mapped to the ISSRM business asset. The *Swimlane* construct holds the constructs (e.g. *Activity* and *Decision*) that are needed to support execution of business workflows. Thus, all these constructs are aligned to the IS assets. Consequently, *Activity*, *Decision*, *WorkFlow* and *Swimlane* are considered as IS asset. Lastly, if we consider the security criterion, it is possible to say that there is not such a construct in MAD to align security criterion.

Risk-related Concepts – In MAD, *Mal-Swimlane* is used to define malicious actor that will harm the system by malicious activities, e.g. the Mal-activity constructs that are combined using *Mal-decision* and *ControlFlow* constructs [28]. *Mal-swimlane* is aligned to the ISSRM threat agent and process defined by combining Mal-activity constructs, to the ISSRM attack method. *Mal-swimlane* construct is later aligned to the ISSRM attack method since in MAD the malicious actor could use some means, which are defined as *Mal-swimlane*. Next, it is possible to say there is not any Mal-activity construct to align to the ISSRM vulnerabilities. In MAD the ISSRM impact can be expressed by using Mal-activity constructs that belong to the *Mal-Swimlane*, characterized as the ISSRM attack method [28].

Risk Treatment-related Concepts – In MAD, the *MitigationActivity* construct is considered as a countermeasure [28]. The *Swimlane* holding the *MitigationActivity* constructs implements the countermeasures. Thus, such a *Swimlane* is aligned to the ISSRM controls.

4.4 Summary

Security Risk Management application is defined with different approaches in this chapter. Basically, with CORAS method [19] we focused on managing security risk analysis and providing a customised language for threat and risk modeling, this operation is done in seven steps. Based on Goal-Risk Framework [20], Security Risk Management is divided into three major areas: Requirements Engineering, Secure and Dependable Engineering,

and Risk Analysis. Afterwards, (ISSRM [11]) Information System Security Risk Management is defined with its domain model and three concepts; Asset, Risk and Risk Treatment, of that domain model. Lastly, Security Risk Management Process referenced by ISSRM is defined step by step. This process is similar with the CORAS approach since it is divided into steps and each step is followed by each other. In conclusion, we will continue with ISSRM approach during the alignment of EPC with security risk management since ISSRM is analysed in information system level and have construct based alignment with Mal-Activity Diagrams as well. The alignment results between MAD and ISSRM will also help us during the transforming the extended EPC to MAD.

Chapter 5. ALIGNMENT OF EPC AND ISSRM

Modern information systems should be secured against potential risks and vulnerable attacks. Event-driven Process Chain (EPC [24] [25]) is a modeling language used to define business processes. Although serving its primary purpose at the high-degree, EPC is not helpful to elicit security concerns when developing information enterprise systems.

Preferably security analysis should start from the early stages, for example from the business process modeling. Business analysts need to invest in security analysis additionally using other approaches and understanding how these approaches could be aligned to the existing business models.

In this chapter we consider how the EPC approach could be applied according to ISSRM [11]. In particular, we will follow the six steps of the ISSRM process to investigate security risks in a running example (online registration process of the Internet Store) modeled using the EPC approach. Later, we will summarize our observation to the alignment of EPC to ISSRM. After all, we will obtain a new language called Security-Oriented EPC.

The main motivation of the alignment process is to cover the existing model with its security requirements. We will try to identify the elements that need to be secured using ISSRM and in the end we will define its security needs and security requirements.

5.1 Security Risk Modeling with EPC

The running example shown in Figure 5.1 is “Online registration process of the Internet Store”. The model will be analysed with defining asset related concepts, risk related concepts and risk treatment related concepts in next sections. When following the ISSRM process, first, we identify the content and valuable assets. Later security objectives are defined by considering the security needs of assets defined. Afterwards, risk analysis is done based on the defined security objectives. Lastly, risk treatment process is done with respect to security requirements definitions.

5.1.1 Context and Asset Identification

Let’s consider the following situation where the user wishes to start using the Internet Store System. In order to get details about the registration, the user sends an inquiry to the system administrator. The system accepts or denies the inquiry. In case of acceptance, administrator reads the inquiry and replies with guidelines (event *positive demand for registration*). This process is called message handling and shown in Figure 5.1.

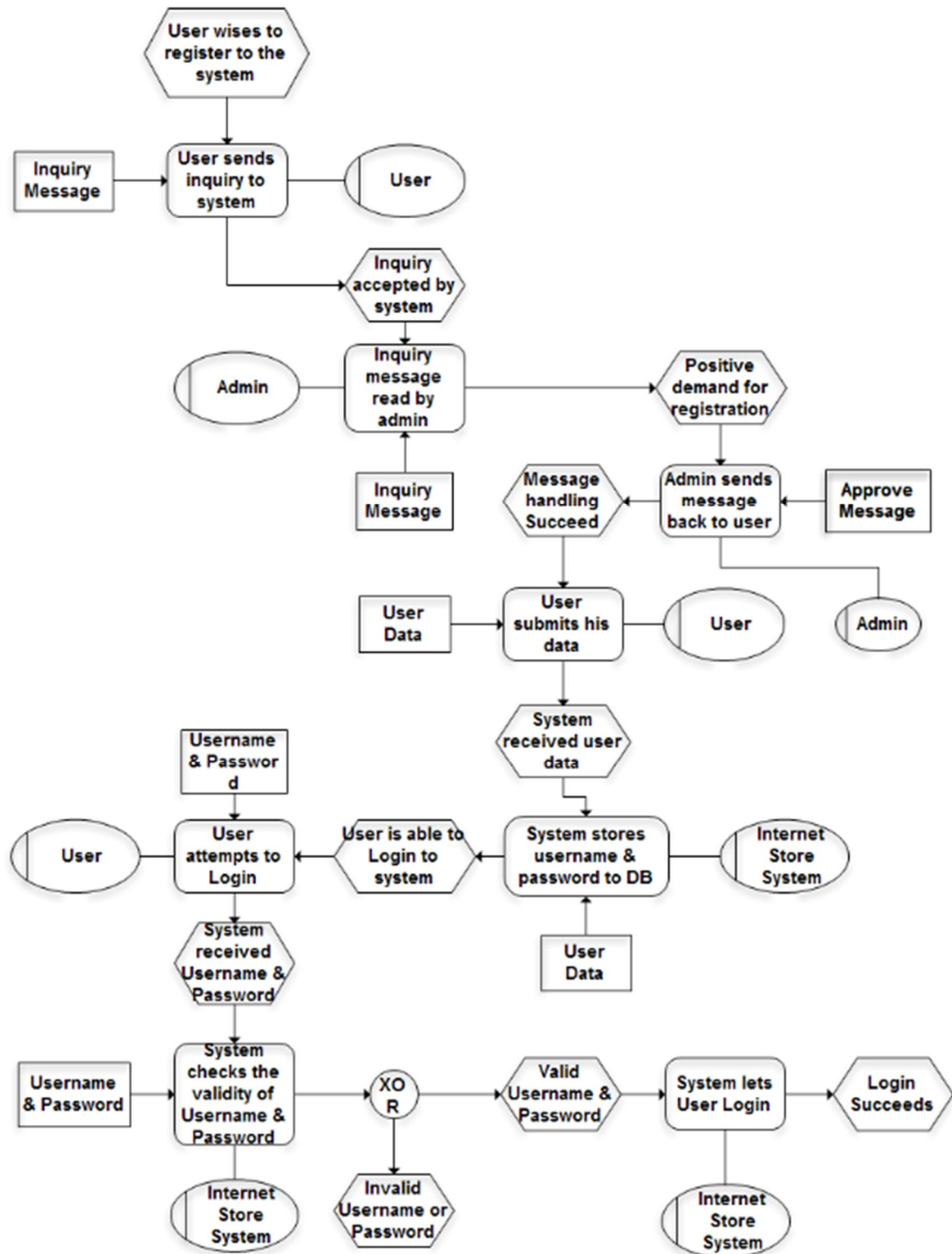


Figure 5.1 – EPC diagram of online registration and login processes of the Internet Store.

After receiving the guidelines, the user registers to the internet store by submitting his data (resource unit *User Data* which includes Username and Password as well). The system then accepts registration information and stores the data into Database of the system. After registering the valid username and password, now the user is able to login to Internet Store

System as illustrated in Figure 5.1. The system checks the validity of username and password and accordingly let user login or fail login process.

With the presented running example, all assets are defined and classified in alignment table (Table 5.1) in section 5.2.

Since the representation of the whole process consists of large amount of constructs, the construct of *Process Path* could be used to show whole model at the higher level of abstraction. Such a representation is shown in Figure 5.2.

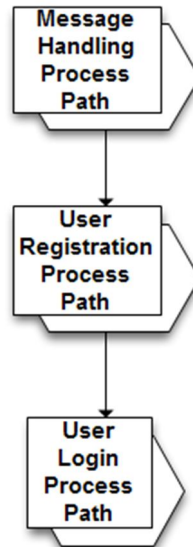


Figure 5.2 – EPC diagram of online registration and login processes of the Internet Store (with Process Paths).

5.1.2 Determination of Security Objectives

We identify several assets which need protection against security risks. Firstly, we need to ensure the *Confidentiality of User Data (Username & Password)*. If confidentiality is revealed, the system violators could use the user’s personal data for the unintended purposes. In addition, *Integrity of all the Business Processes* (the ones in Figure 5.1) has to be ensured. If integrity is negated the system might be used not according to the intended purpose.

5.1.3 Risk Analysis and Assessment

In Figure 5.3, a part of the previous model visualizes a potential security risk. Let’s say that there exist a violator who would like to login to the system without registering his personal user account.

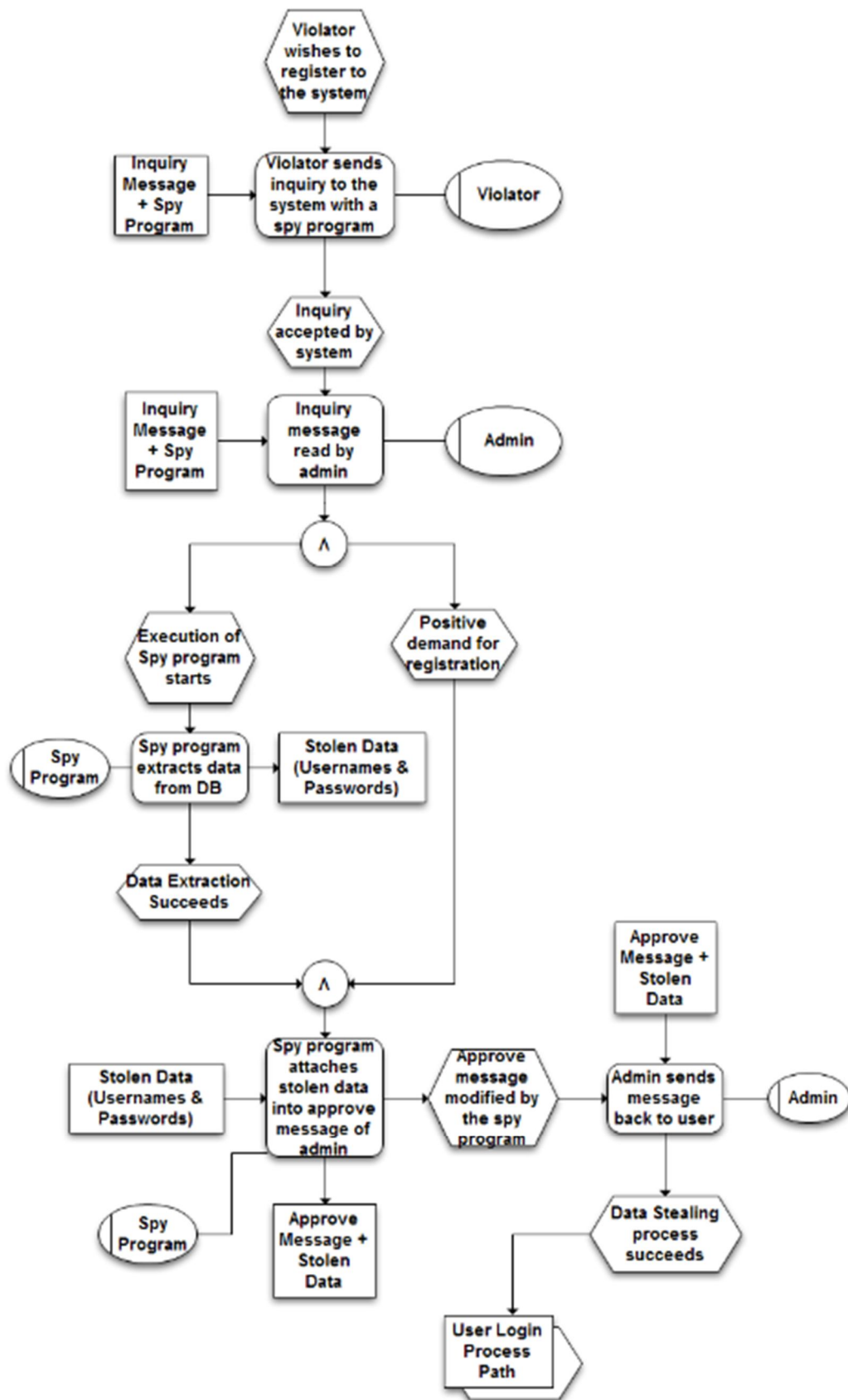


Figure 5.3 – Message Handling process including security risk attack.

In the model in Figure 5.3, violator sends an inquiry message to the system as it is done in Figure 5.1 by user. However, this time the inquiry message includes a spy program which is started after the admin accepts and reads the message. The spy program initializes a new event *Execution of Spy program starts*, and it is followed by the initialization of a new function called *Spy program extracts data from DB* which sends an inquiry to the database and extracts the usernames and passwords of existing users. This function is followed by another event which indicates the purpose of the attack, *Data Extraction Succeeds*. Simultaneously, positive demand for the registration state is approved. By the spy program, the stolen data then attached to the approval reply message of admin which is sent to the violator, this function is called *Spy program attaches stolen data into approve message of admin* and indicates the result of the attack. As a result, violator reaches his target and, therefore, in this model User Registration process path is not shown since violator is now able to continue his process by logging in.

In this analysis we are able to identify the ISSRM threat agent (Violator) and the ISSRM attack method (Message including a spy program and extracting data from DB). Combination of these elements forms a security threat. The direct impact of this threat is the negation of the *Confidentiality of User Data (Username & Password)*. In addition, this ISSRM impact provokes another impact (such as the Violator accesses the system with a stolen username and password and change the business processes according to his needs), which negates the *Integrity of all the Business Processes*.

5.1.4 Risk Treatment

Risk treatment is the part where it is decided how the identified security flows could be mitigated. In our running example, as all the threats are defined in previous sections we can take a *risk reduction decision*, which reduces the probability of the negative consequences, based on mentioned threats. In particular, we will modify our current online registration and login processes of the Internet Store.

5.1.5 Security Requirements Definition

To decrease the possibility of accepting the message which includes a spy program, first we introduce the message scanning in our existing model. Message scanning consists of *Scanning system activated*, *Message is not safe* and *Registration failed* events; *Message is scanned* and *System blocks the user and deletes the message* functions, shown in Figure 5.4. If scanning of the message reports a problem, the message is deleted and the message sender is blocked. Based on second security requirement, the control activity of DB access is modeled, including *DB access attempt found*, *DB access attempt is not found* and *Registration failed* events; *Control activity of DB access* and *Block DB access* functions. If there is an attempt to access the database during the message handling process, it is blocked. In addition, control activity of DB access and reading inquiry message functions are concurrent to each other, this concurrency is presented with join and fork AND gates. The final security requirement leads us modeling the control of outgoing/sent information which consists of *Traffic is safe*, *Traffic is not safe* and *Registration failed* events; *Outcoming traffic control* and *Stop the operation* functions. This operation investigates if the response message is of the same length as it was initially defined by the system admin. If this check reports a problem, the system stops the message sending operation.

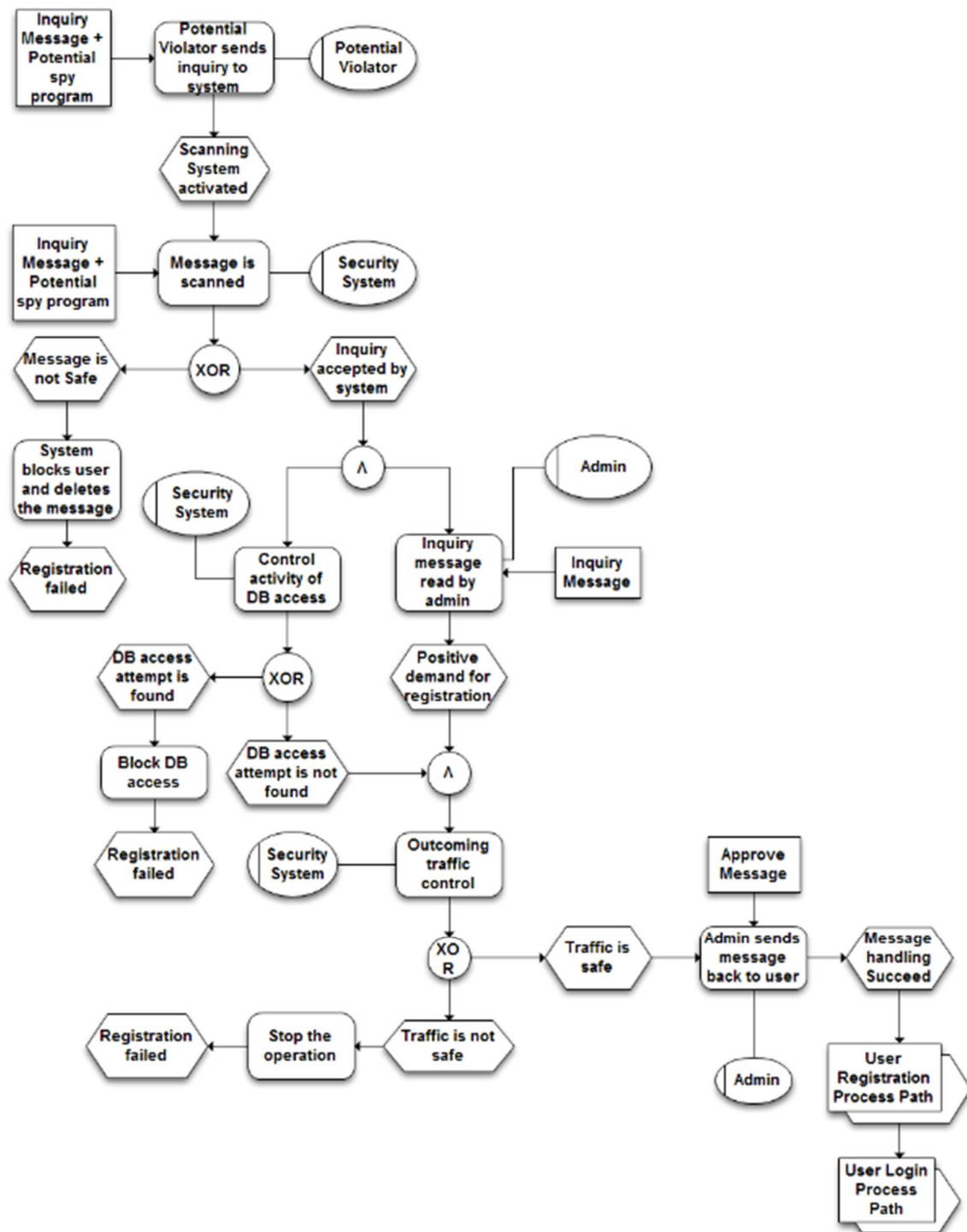


Figure 5.4 – Message Handling process including security requirements.

5.1.6 Control Selection and Implementation

The EPC application is typically performed at the business process management and modeling stages and implementation of the security requirements usually remains postponed for the later system development stages. However, it is important to make the security risk analysis and the modeling at together, otherwise, if the security risk analysis is

done after the business process modeling business analysts might have to change the whole model from the beginning, which causes loss of time and money.

5.2 ISSRM and EPC Alignment

The result of previous analysis is the semantic alignment between the EPC constructs and the concepts of the ISSRM domain model. A summary is given below including alignment which is shown in Table 5.1.

Asset-related Concepts: As it is comprehensively described in section 5.1.1, all assets, business assets and information system assets in Table 5.1 are categorized. Event, Function, Organisation Unit, Resource Unit, Control Flow, Logical Operator, Information Flow, Assignment and Process Path constructs of EPC are considered as both business asset and IS asset of ISSRM domain model since these constructs are continuously used in the running example(s). In addition, the relationship between these constructs and Asset-related concepts of ISSRM domain model is very strong because of the similarity between concept definitions and construct definitions.

Risk-related Concepts: These concepts present how the risk itself can be defined and what major principles should be taken into account when defining the potential risks. Based on the running example and the wide explanations in risk analysis and assessment section, it is possible to illustrate the EPC constructs included risk scenario in the table only in *Threat Agent* and *Attack Method* parts. There are two threat agents in this level; Violator and Spy Program, and they both have different roles in different risks of different security levels. Threat agent concept is represented with the organisation unit construct whereas attack method is represented with the rest of the EPC constructs (Event, Function, Logical Operator, Resource Unit, Information Flow, Assignment, Control Flow, Process Path) since the organisation unit is the actor who uses all these constructs similarly threat agent performs the attack method.

Risk Treatment-related concepts: These concepts describe the decisions that should be taken, and controls to be implemented in order to mitigate the identified risks during the risk analysis and assessment. As the running example and risk treatment methods are described in section 5.1.5, we only show in Table 5.1 which constructs are involved in the three different mitigation processes. All the EPC constructs (Event, Function, Organisation Unit, Logical Operator, Resource Unit, Information Flow, Assignment, Control Flow, Process Path) are clustered under the Security Requirement concept of ISSRM domain model since all the risk treatment process is done with these constructs by defining the security requirements.

5.3 Summary

In this chapter we have focused on the common characteristics of the EPC constructs and the ISSRM domain model concepts. After analyzing the similarities and common characteristics of the EPC constructs and the ISSRM concepts with the running examples, we align EPC and ISSRM in a table which presents alignment of the EPC constructs to the ISSRM concepts, shown in Table 5.1. All EPC constructs are clustered under a specific ISSRM concept. The alignment process leads us to extend the EPC language and make it

more secure. The main challenge for the extension process is that to classify some constructs which are used and clustered under the different ISSRM concepts (e.g. Event, Function, Organisation Unit and so on). Further discussion and analysis are done in next chapter with the extension of the EPC language.

Table 5.1 – Alignment of the EPC constructs to the ISSRM concepts.

The ISSRM Domain Model		EPC Constructs	Example
Asset-Related Concepts	Asset	-	-
	Business Asset	Event, Function, Organisation Unit, Resource Unit, Control Flow, Logical Operator, Information Flow, Assignment, Process Path	<p>Event[<i>User wishes to register to the system</i>] Organisation Unit[<i>User</i>] Function[<i>User sends inquiry to system</i>] Resource Unit[<i>Inquiry Message</i>]</p> <p>Organisation Unit[<i>User</i>] Function[<i>User submits his data</i>] Resource Unit[<i>User Data</i>]</p> <p>Organisation Unit[<i>User</i>] Function[<i>User attempts to login</i>] Resource Unit[<i>Username & Password</i>]</p>
	IS Asset	Event, Function, Organisation Unit, Resource Unit, Control Flow, Logical Operator, Information Flow, Assignment, Process Path	<p>Event[<i>Positive demand for registration</i>] Organisation Unit[<i>Admin</i>] Function[<i>Inquiry message read by admin</i>] Resource Unit[<i>Inquiry Message</i>]</p> <p>Organisation Unit[<i>Admin</i>] Function[<i>Admin sends message back to user</i>] Resource Unit[<i>Approve Message</i>]</p> <p>Organisation Unit[<i>Internet Store System (ISS)</i>] Function[<i>System stores username & password to DB</i>] Resource Unit[<i>User Data</i>]</p> <p>Organisation Unit[<i>ISS</i>] Function[<i>System checks the validity of username & password</i>] Resource Unit[<i>Username & Password</i>]</p> <p>Event[<i>Invalid username or password, Valid username & password</i>] Organisation Unit[<i>ISS</i>] Function[<i>System lets user login</i>]</p>
	Security Criterion	-	<ul style="list-style-type: none"> Confidentiality of Username and Password (<i>Business Resource Unit</i>), Integrity of Business Process Paths.
	Risk	-	-
Risk-Related Concepts	Impact	-	<ul style="list-style-type: none"> Confidentiality of Username and Password is broken, Integrity of functions and events is negated.
	Event	-	-
	Threat	-	A violator sends message containing a spy program. Later, the spy program extracts info from database and finally sends it back to the violator by attaching stolen data into approve message.

	Vulnerability	-	<ul style="list-style-type: none"> • Message is being handled without any scanning, • The outgoing traffic is not monitored, • The access to DB is not controlled.
	Threat Agent	Organisation Unit	Organisation Unit [<i>Violator</i>] Organisation Unit [<i>Spy Program</i>]
	Attack Method	Event, Function, Logical Operator, Resource Unit, Information Flow, Assignment, Control Flow, Process Path	<ul style="list-style-type: none"> • A violator sends a message containing a spy program, Event[<i>Violator wishes to register to the system</i>] Function[<i>Violator sends inquiry to the system with a spy program</i>] Resource Unit[<i>Inquiry Message + Spy program</i>] <ul style="list-style-type: none"> • Spy program extracts info from database, Event[<i>Execution of spy program starts, Data extraction succeeds</i>] Function[<i>Spy Program extracts data from DB</i>] Resource Unit[<i>Stolen Data</i>] <ul style="list-style-type: none"> • Spy program sends it back to the violator by attaching stolen data into approve message. Event[<i>Approve message modified by the spy program</i>] Function[<i>Spy Program attaches stolen data into approve message of admin</i>] Resource Unit[<i>Stolen data, Approve message + Stolen Data</i>]
Risk Treatment-Related Concepts	Risk Treatment	-	Risk Reduction
	Security Requirement	Event, Function, Organisation Unit, Logical Operator, Resource Unit, Information Flow, Assignment, Control Flow, Process Path	<ul style="list-style-type: none"> • Inquiry message scanning, Event[<i>Scanning system activated, Message is not safe, Registration failed</i>] Organisation Unit[<i>Security System</i>] Function[<i>Message is scanned, System blocks user and deletes the message</i>] Resource Unit[<i>Inquiry message + potential spy program</i>] <ul style="list-style-type: none"> • Database access control, Event[<i>DB access attempt found, DB access attempt is not found, Registration failed</i>] Organisation Unit[<i>Security System</i>] Function[<i>Control activity of DB access, Block DB access</i>] <ul style="list-style-type: none"> • Outgoing traffic control. Event[<i>Traffic is safe, Traffic is not safe, Registration failed</i>] Organisation Unit[<i>Security System</i>] Function[<i>Outcoming traffic control, Stop the operation</i>]
	Control	-	-

Chapter 6. SECURITY-ORIENTED EPC

In this chapter, we develop syntactic, semantic and methodological extensions to Event-driven Process Chain [25] that would support modeling security risks and their countermeasures. First, we analyse the extensions to the concrete syntax and then present how concrete syntax extensions are addressed in the abstract syntax. Next, methodological guidelines are defined, lastly, extensions with respect to Information System Security Risk Management domain model are defined. Security-Oriented EPC is obtained after the extensions in two different levels; high level and low level. The high level means less constructs since we only focus on “process path” construct of EPC. Process paths contain all other constructs in it like a cluster. Low level refers to more constructs since we focus on all constructs of EPC by ignoring the case “process path contains other constructs”. The main reason to make two different levels of analysis is to highlight the importance of the complexity of the EPC language.

6.1 Higher-Level of Security Problem Definition

The high level extension process is done in two main parts: Concrete Syntax and Abstract Syntax by analyzing the constructs of the model.

6.1.1 Concrete Syntax in High Level

In section 5.2, during the alignment of EPC and ISSRM, the concrete syntax of EPC is analysed according to the three categories: asset-related concepts, risk-related concepts and risk treatment-related concepts. In Figure 6.1, 6.3 and 6.5, constructs of EPC are categorized according to the ISSRM concepts and showed separately according to their usage in the model. Besides, in these figures the ISSRM relationships are also expressed (if possible) with EPC. In this section we focus on the higher level problem definition by using the process path construct(s).

Asset-related Concepts:

As it is seen in Figure 6.1 both Business Asset (red) and IS Asset (green) have their own *Process Paths*. Based on ISSRM domain model, we extended the EPC language with new relationships and constructs in high level. These relationships are *Supports* (with *Control Flow* in high level, with *Control Flow* and *Event* in low level) and *EPC-Constraint of*. The new construct is *EPC-Security Criterion* and all these new constructs and relationships are illustrated in the example in Figure 6.2.



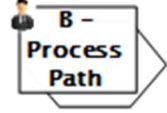




ISSRM	Type	EPC	
		Constructs	Concrete Syntax
Asset	C	-	-
IS Asset	C	Control Flow, Process Path	 
Business Asset	C		 
Supports	R	Relations: Control Flow	
Security Criterion	C	EPC–Security Criterion	
Constraint of	C, R	Relations: EPC–Constraint of	

Figure 6.1 – Asset-related (C)oncepts and (R)elationships.

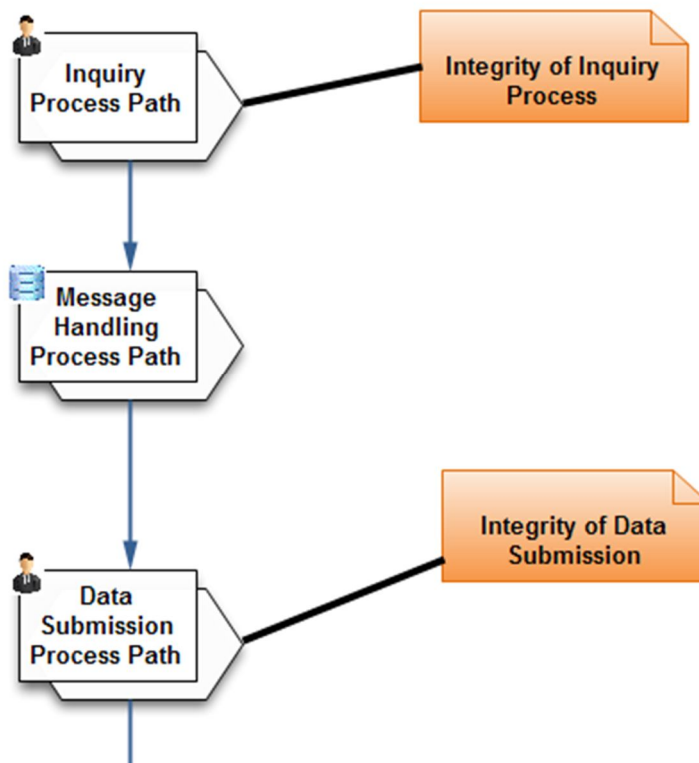


Figure 6.2 – Part of Registration and Login to Internet Store example shown in Business Process Path and IS Process Path including new constructs EPC – Security Criterion and EPC – Constraint of.

Risk-related Concepts:

In Risk-related concepts we also have new relationships and new constructs as shown in Figure 6.3. Again we classified these relationships, constructs and their connections between each other based on ISSRM domain model.



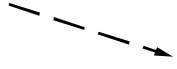
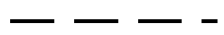

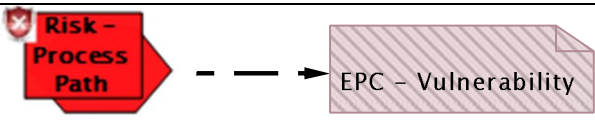



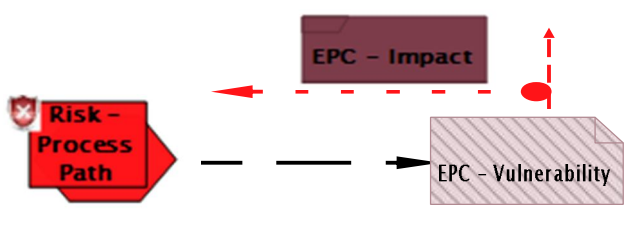
ISSRM	Type	EPC	
		Constructs	Concrete Syntax
Threat Agent	C	-	-
Attack Method	C	-	-
Uses	R	-	-
Threat	C	Process Path	
Vulnerability	C	EPC-Vulnerability	
Exploits	C, R	Relations: EPC-Exploits	
Characteristic of	C, R	Relations: EPC-Characteristic of	
Targets	R	Relations: Control Flow	
Event	C	(Process Path) + (EPC-Exploits) + (EPC-Vulnerability)	
Impact	C	EPC-Impact	
Harms	C, R	Relations: EPC-Harms	
Leads to/Negates	C, R	Relations: EPC-Leads to/Negates	
Risk	C	(Process Path) + (EPC-Exploits) + (EPC-Vulnerability) + (EPC-Impact) + (EPC-Harms) + (EPC-Leads to/Negates)	
Significance Assessed by	R	-	-

Figure 6.3 – Risk- related (C)oncepts and (R)elationships.

Risk-related process is included in the Business Asset level as it can be seen in the Figure 6.4. Threat *Process Path* targets (with *Control Flow*) the IS Asset *Process Path* (which has a *Vulnerability*). *Vulnerability* is connected to the IS Asset *Process Path* with *EPC-Characteristic of* and threat *Process Path* exploits the *Vulnerability* with *EPC-Exploits*.

Besides, the *Vulnerability* leads to an impact and negates the security criterion with the new construct *EPC-Leads to/Negates* and from this construct a branch called *EPC-Harms* goes to the IS Asset *Process Path* which indicates the harm and its impact definition with the *EPC-Impact*.

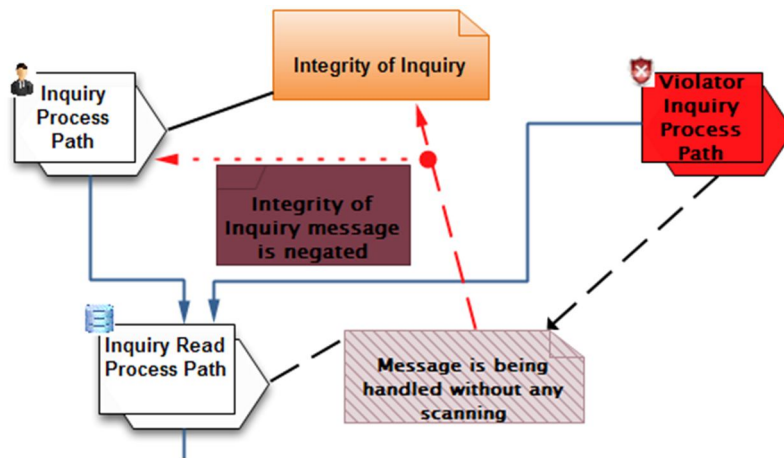


Figure 6.4 – Part of Registration and Login to Internet Store example shown in Business Asset Process Path, IS Asset Process Path and Risk Process Path including new constructs EPC – Security Criterion, EPC – Constraint of, EPC – Exploits, EPC – Vulnerability, EPC – Harms, EPC – Impact, EPC – Characteristic of and EPC – Leads to/Negates.

Risk Treatment-related Concepts:

The last extended and analysed concept is Risk Treatment-related concepts. Here also there is a new *Process Path* shown in Figure 6.5 which characterize the treatment functionality and, in general, mitigates the vulnerability of the IS Asset in the Control level.




ISSRM	Type	EPC	
		Constructs	Concrete Syntax
Risk Treatment	C	-	-
Decision to Treat	R	-	-
Security Requirement	C	-	-
Control	C	Control Flow, Process Path	 
Refines	R	-	-
Mitigates	C, R	Relations: EPC–Mitigates	
Implements	R	-	-

Figure 6.5 – Risk treatment-related (C)oncepts and (R)elationships.

The Risk Treatment-related process is included to the system right after Business Asset or Risk-related process level as it is illustrated in Figure 6.6 and 6.7. All in all, control *Process Path* mitigates the IS Asset *Vulnerability* with the new construct *EPC-Mitigates*.

Once the Vulnerability is mitigated, all other Risk-related concepts are also ignored and the system is safe now.

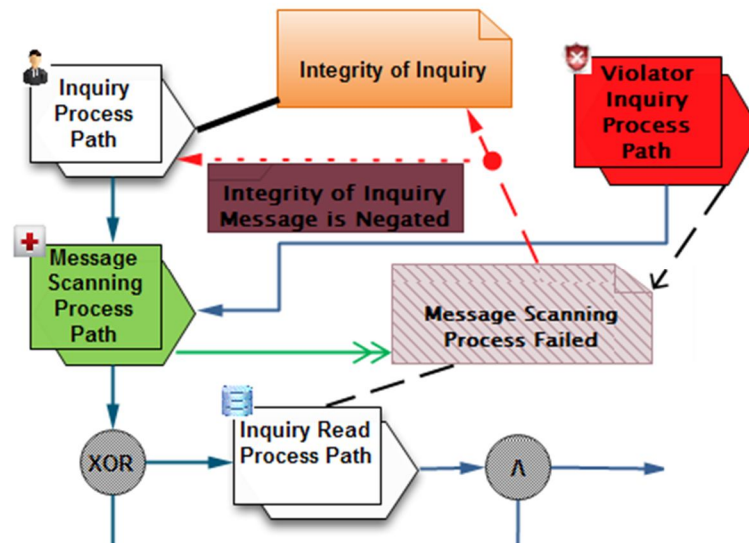


Figure 6.6 – Part of Registration and Login to Internet Store example shown in Business Asset Process Path, IS Asset Process Path, Risk Process Path and Risk Treatment Process Path including new constructs EPC – Security Criterion, EPC – Constraint of, EPC – Exploits, EPC – Vulnerability, EPC – Harms, EPC – Impact, EPC – Characteristic of, EPC – Leads to/Negates and EPC – Mitigates.

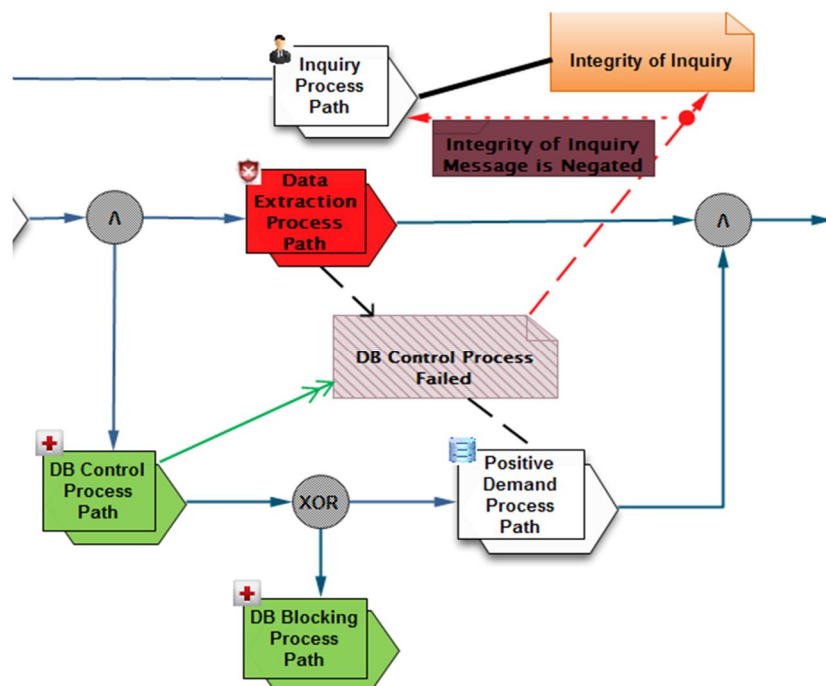


Figure 6.7 – Part of Registration and Login to Internet Store example shown in Business Asset Process Path, IS Asset Process Path, Risk Process Path and Risk Treatment Process Path including new constructs EPC – Security Criterion, EPC – Constraint of, EPC – Exploits, EPC – Vulnerability, EPC – Harms, EPC – Impact, EPC – Characteristic of, EPC – Leads to/Negates and EPC – Mitigates.

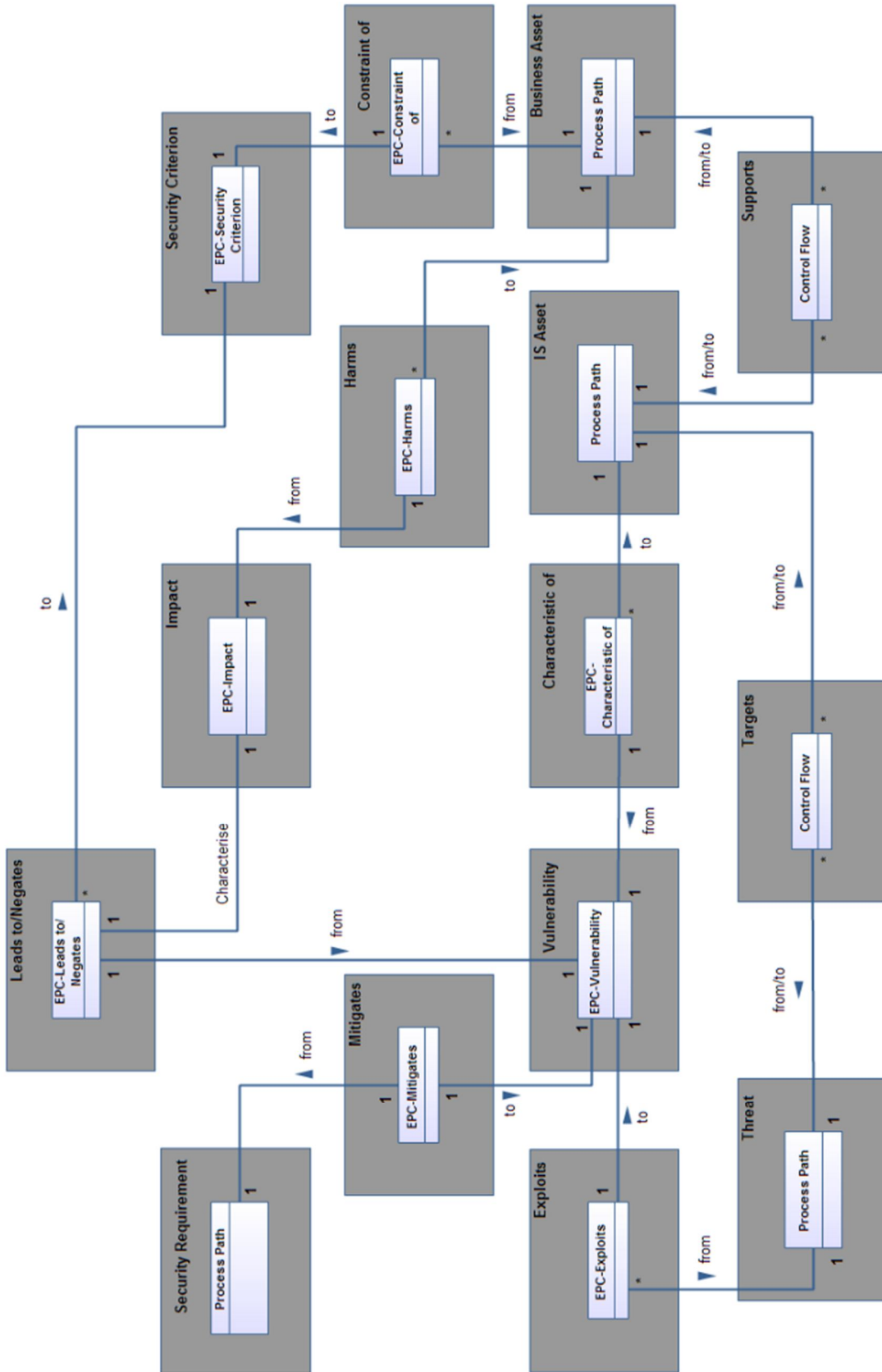


Figure 6.8 – Abstract Syntax of Extended EPC with Process Path and ISSRM Domain Model.

6.1.2 Abstract Syntax in High Level

In previous sections abstract syntax of EPC has not been presented due to the need of the simple introduction of the language itself. However, to illustrate how the proposed syntactic EPC extensions are used, we need to present abstract syntax elements and the rules how they can be combined together.

The abstract syntax of Security-Oriented EPC is illustrated in a model. The model we obtained is called *The Security Enhanced Model with Process Paths* shown in Figure 6.8. The model is based on the ISSRM domain model concepts, these concepts' relationships to each other and on the other side the EPC meta-model constructs (only Process Path) and the associations of these constructs between each other.

6.2 Lower-Level of Security Problem Definition

In this section we focus on the main constructs of the EPC language except process path. Here also we extend the language in two parts; concrete syntax and abstract syntax.

6.2.1 Concrete Syntax in Low Level

In low level also the concrete syntax of EPC is analysed according to the three construct categories: asset-related concepts, risk-related concepts and risk treatment-related concepts. In Figure 6.1, 6.3 and 6.5, high level constructs (*Process Path* level) of EPC are categorized according to the ISSRM concepts and showed separately according to their usage in the model. In this section we analyse low level construct extensions in Figure 6.10, 6.13 and 6.15.

Asset-related Concepts:

In this section we focus on the constructs in low level, which means we will show the extended relationships and constructs with usual EPC constructs, not with *Process Paths*. As all the constructs are classified in Figure 6.10, now we start analyzing the example in Figure 6.9 and Figure 6.11 and see the differences between low level and high level extensions. In Figure 6.9 and 6.11 Security-Oriented EPC asset constructs are illustrated in lower level. Extended construct *Security Criterion* and *EPC-Constraint of* are also shown in these figures.

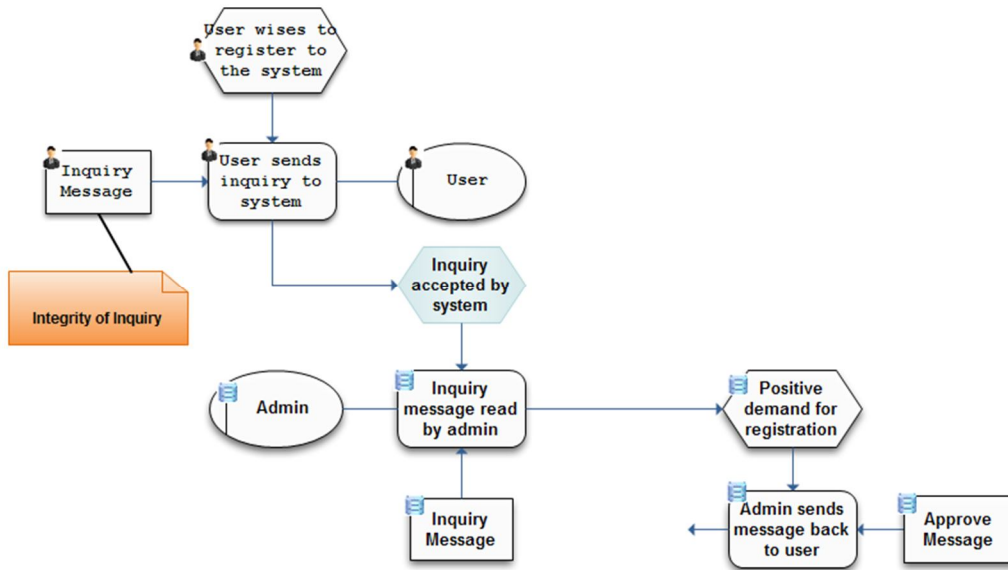


Figure 6.9 – Part of Registration and Login to Internet Store example shown in Business Asset Constructs and IS Asset Constructs including new constructs EPC – Security Criterion and EPC – Constraint of.

ISSRM	Type	EPC	
		Constructs	Concrete Syntax
Asset	C	-	-
IS Asset	C	Event, Function, Organisation Unit, Resource Unit, Information Flow, Control Flow, Logical Operator, Assignment	
Business Asset	C	B - Event, B - Function, B - Log. Op., B - Resource Unit, B - Org. Unit	
Supports	R	Relations: Control Flow, Event	
Security Criterion	C	EPC–Security Criterion	
Constraint of	C, R	Relations: EPC–Constraint of	

Figure 6.10 – Asset-related (C)oncepts and (R)elationships.

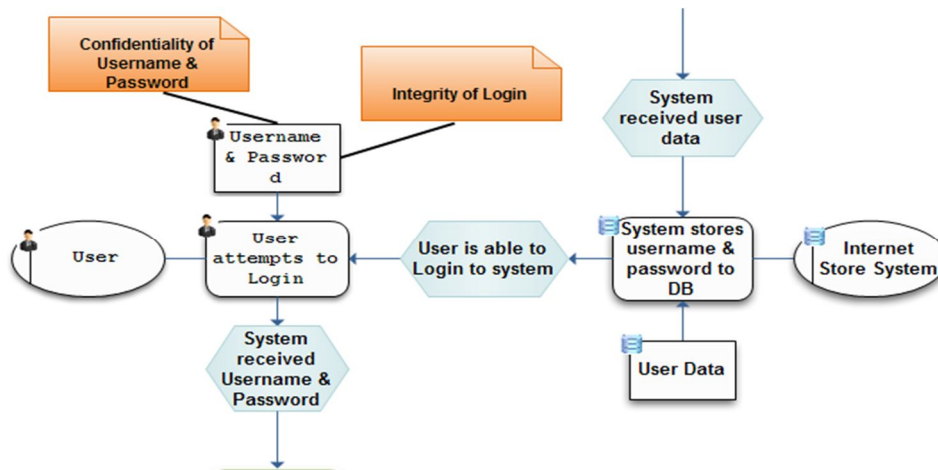


Figure 6.11 – Part of Registration and Login to Internet Store example shown in Business Asset Constructs and IS Asset Constructs including new constructs EPC – Security Criterion and EPC – Constraint of.

Risk-related Concepts:

In Risk-related concepts, we have the similar relationships and constructs that we have extended in higher-level, shown in Figure 6.13. We will just indicate which low level construct will be connected to each other, this is important because we use this information when we define the abstract syntax of the extensions.

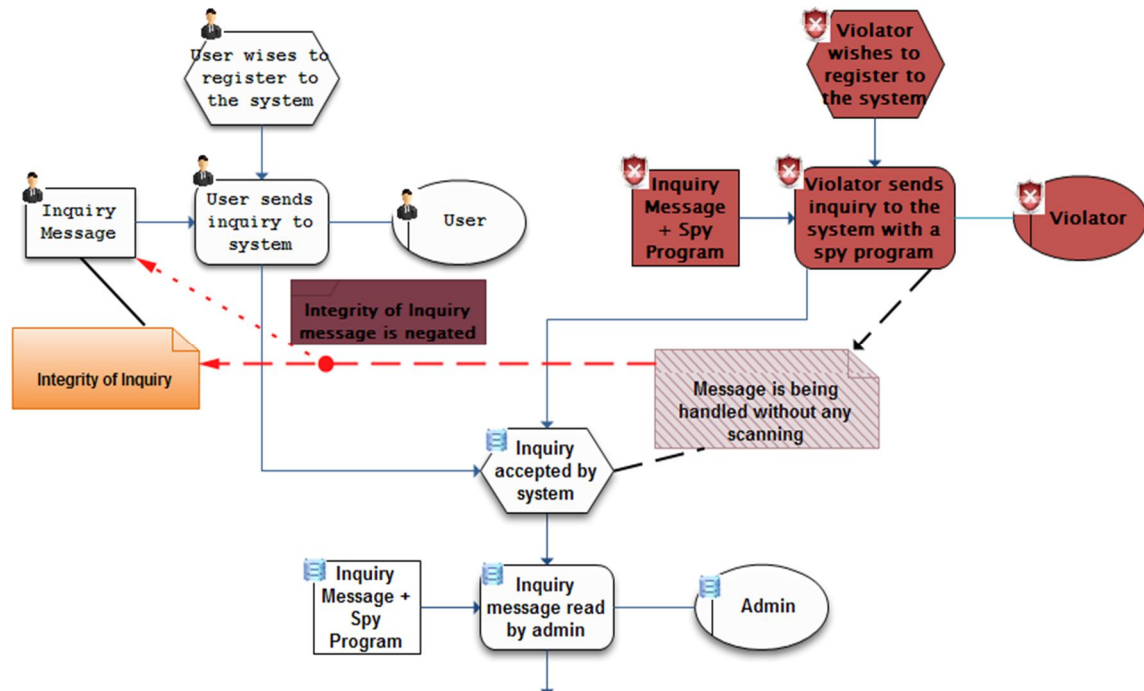


Figure 6.12 – Part of Registration and Login to Internet Store example shown in Business Asset Constructs, IS Asset Constructs and Risk Constructs including new constructs EPC – Security Criterion, EPC – Constraint of, EPC – Exploits, EPC – Vulnerability, EPC – Harms, EPC – Impact, EPC – Characteristic of and EPC – Leads to/Negates.

ISSRM	Type	EPC	
		Constructs	Concrete Syntax
Threat Agent	C	Organisation Unit	
Attack Method	C	Event, Function, Resource Unit, Information Flow, Control Flow, Logical Operator	
Uses	R	Relations: Assignment	
Threat	C	Event, Function, Resource Unit, Information Flow, Control Flow, Logical Operator, Organisation Unit	
Vulnerability	C	EPC-Vulnerability	
Exploits	C, R	Relations: EPC-Exploits	
Characteristic of	C, R	Relations: EPC-Characteristic of	
Targets	R	Relations: Control Flow	
Event	C	(Function) + (EPC-Exploits) + (EPC-Vulnerability) + (Event) + (Resource Unit) + (Org. Unit)	
Impact	C	EPC-Impact	
Harms	C, R	Relations: EPC-Harms	
Leads to/Negates	C, R	Relations: EPC-Leads to/Negates	
Risk	C	(Function) + (EPC-Exploits) + (EPC-Vulnerability) + (Event) + (Resource Unit) + (Org. Unit) + (EPC-Impact) + (EPC-Harms) + (EPC-Leads to/Negates)	
Significance Assessed by	R	-	-

Figure 6.13 – Risk-related (C)oncepts and (R)elationships.

In Figure 6.12, the constructs of the attack method are included in the same level with Business Asset constructs. Briefly, the *Supports* relationship between Business Asset and

IS Asset this time is provided by only *Control Flow* because in risk level we assume that *Event* is part of IS Asset and also attack method *Function* targets the IS Asset *Event*. Next, attack method *Function* exploits the *Vulnerability* with *EPC-Exploits*, and *Vulnerability* is characteristic of IS Asset *Event* and they are connected to each other with *EPC-Characteristic of*. *Vulnerability* leads to/negates the *EPC-Security Criterion* with *EPC-Leads to/Negates* and again a branch connected to the *EPC-Leads to/Negates* called *EPC-Harms* targets the IS Asset *Resource Unit* and this branch has also the *EPC-Impact* construct which gives information about the impact.

Risk Treatment-related Concepts:

In Risk Treatment-related concepts of low level, the important connections between constructs are; *Business Asset Function Supports* and *Threat Function Targets* the *Control Event* with a *Control Flow*, and *Control Function* mitigates the *EPC-Vulnerability* with *EPC-Mitigates* as we can see from Figure 6.14.

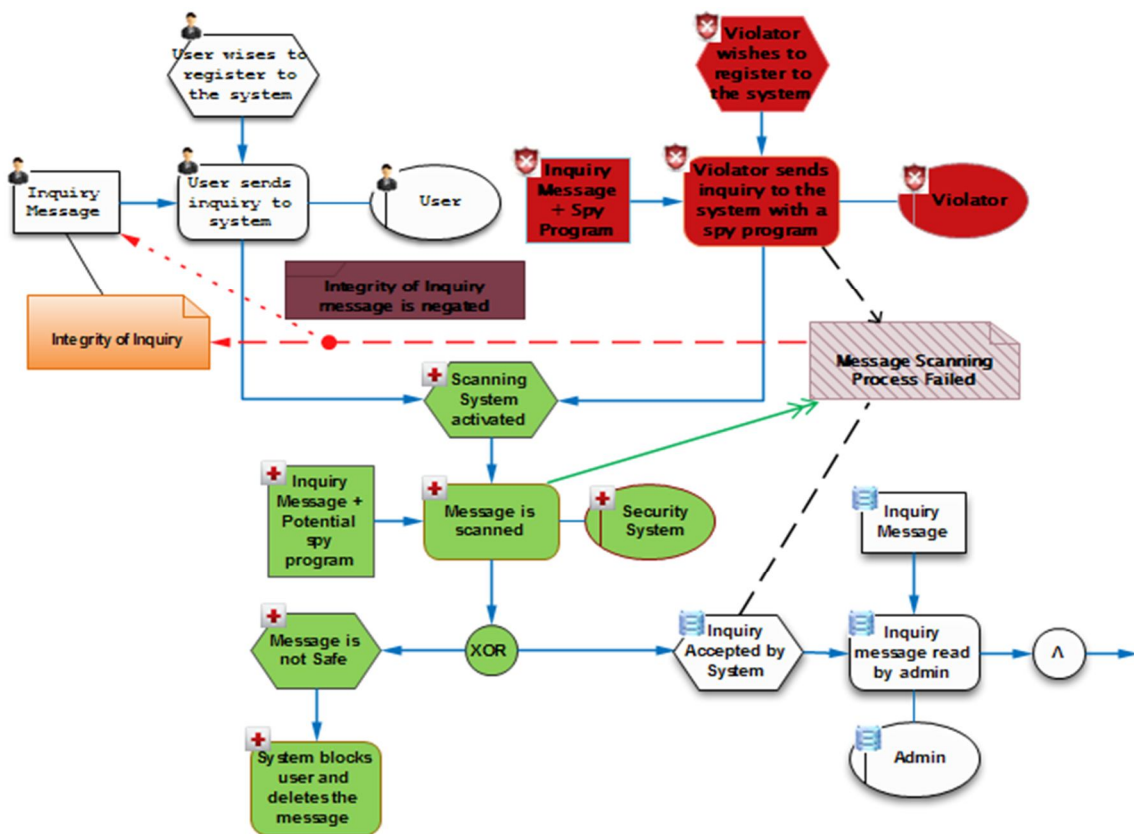


Figure 6.14 – Part of Registration and Login to Internet Store example shown in Business Asset Constructs, IS Asset Constructs, Risk Constructs and Risk Treatment Constructs including new constructs EPC – Security Criterion, EPC – Constraint of, EPC – Exploits, EPC – Vulnerability, EPC – Harms, EPC – Impact, EPC – Characteristic of, EPC – Leads to/Negates and EPC – Mitigates.

6.2.2 Abstract Syntax in Low Level

The first model we obtained during the higher level abstract definition was called *The Security Enhanced Model with Process Paths* shown in Figure 6.8. The second model is

called *The Security Enhanced Model with Subconstructs* is shown in Figure 6.16. The model is based on the ISSRM domain model concepts, these concepts' relationships to each other and on the other side the EPC meta-model constructs (not including Process Path) and the associations of these constructs between each other.

ISSRM	Type	EPC	
		Constructs	Concrete Syntax
Risk Treatment	C	-	-
Decision to Treat	R	-	-
Control	C	-	-
Security Requirement	C	Event, Function, Organisation Unit, Resource Unit, Information Flow, Control Flow, Logical Operator, Assignment	
Refines	R	-	-
Mitigates	C, R	Relations: EPC-Mitigates	
Implements	R	-	-

Figure 6.15 – Risk Treatment-related (C)oncepts and (R)elationships.

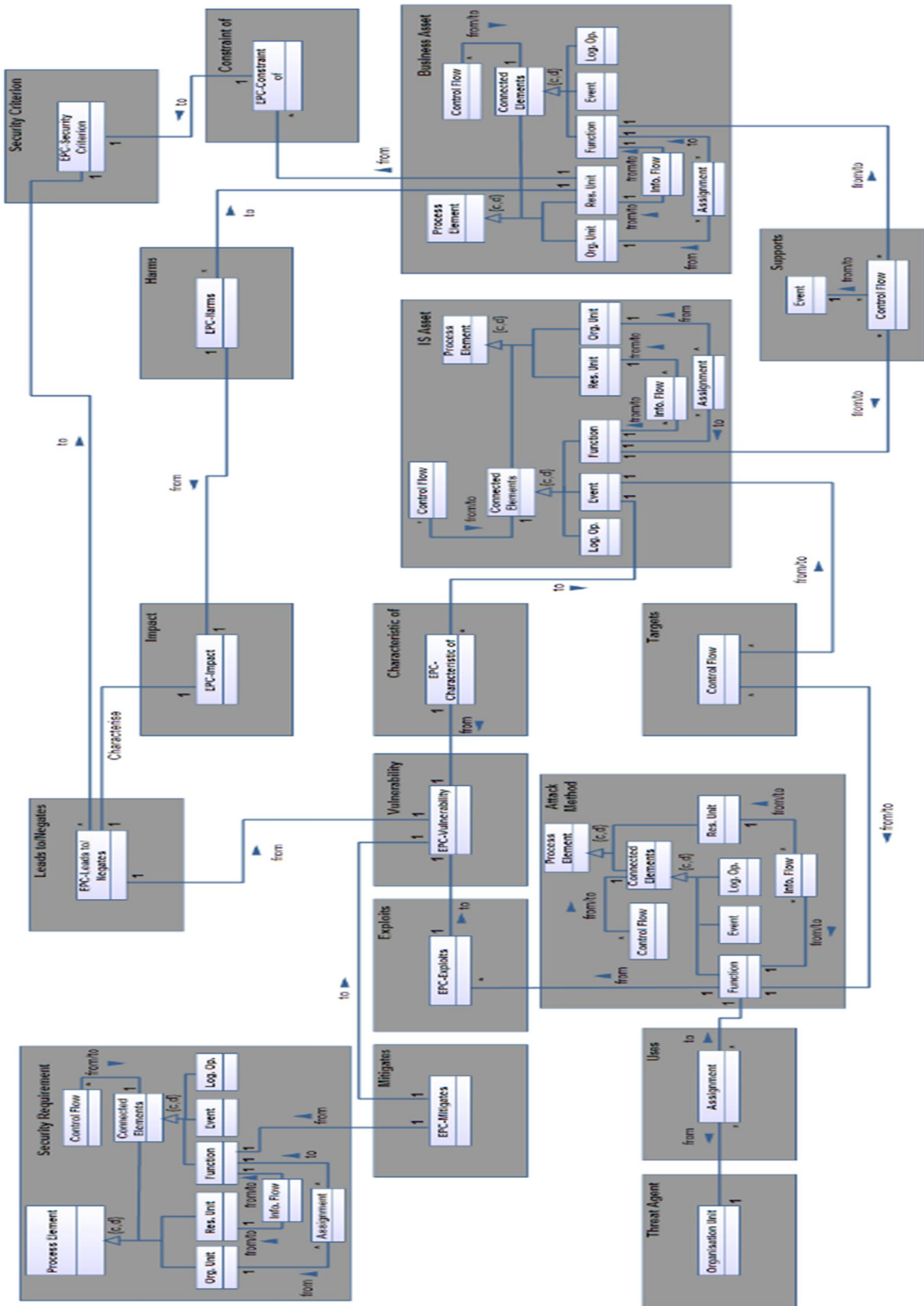


Figure 6.16 – Abstract Syntax of Extended EPC with Constructs and ISSRM Domain Model.

6.3 Extended Meta-Model of Security-Oriented EPC

In previous sections we have extended the EPC language in Risk/Security oriented way. In the end we have obtained two abstrax syntaxes, one based on high level with Process Paths, the other based on low level with other sub constructs. If we combine these two abstrax syntaxes on the EPC meta-model template, we gain the below model in Figure 6.17 which can be called as the Security-Oriented Meta-Model of EPC or Abstrax Syntax of Security-Oriented EPC.

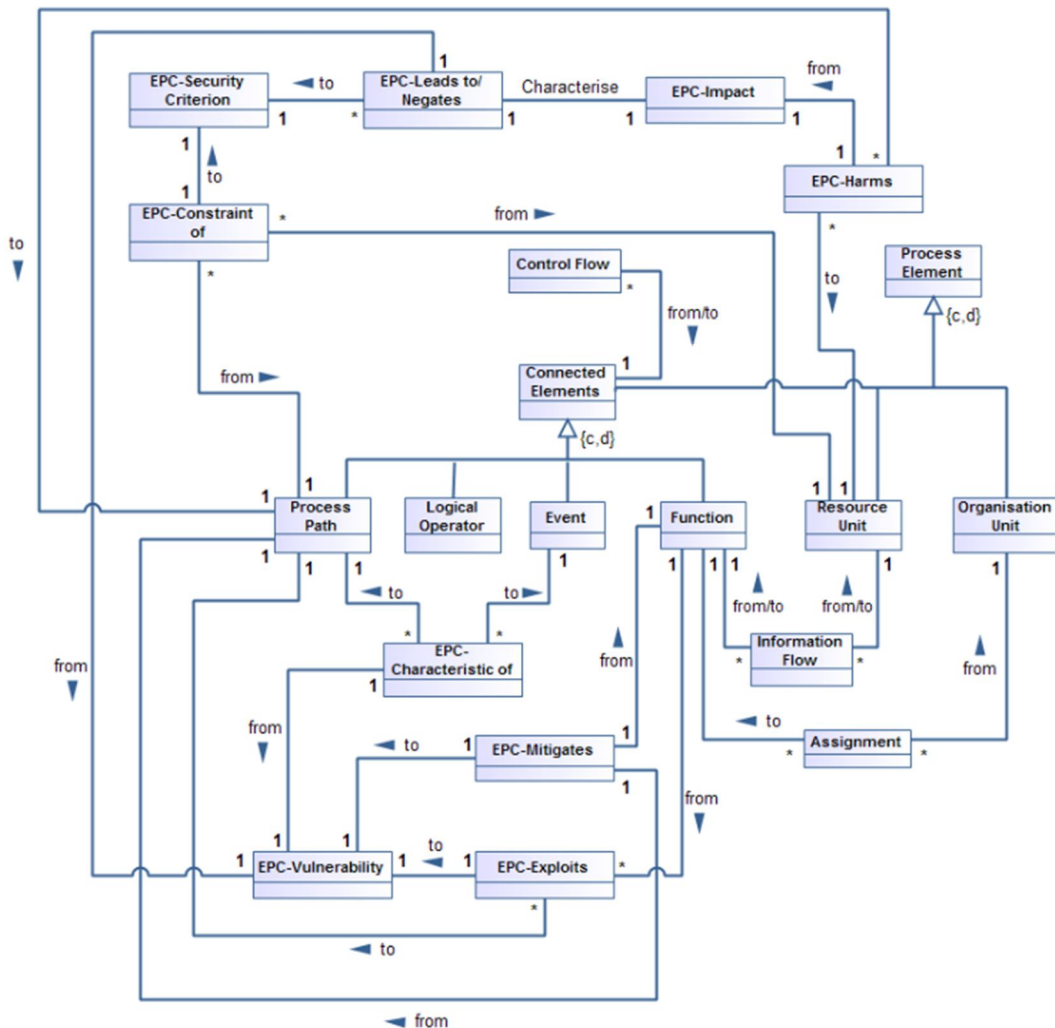


Figure 6.17 – Abstract Syntax of Extended EPC aka Security-Oriented EPC.

6.4 Summary

The question “*Why do we need to use Security-Oriented EPC in Business Modeling?*” rises explicitly after the syntactic and symantic extensions of the language. Many business process modelers might want to ignore the security extensions due to *avoiding the complexity*. At this point, we need to introduce the solution with purpose in order to disprove the main idea “There is no mean to do security analysis in business process modeling”.

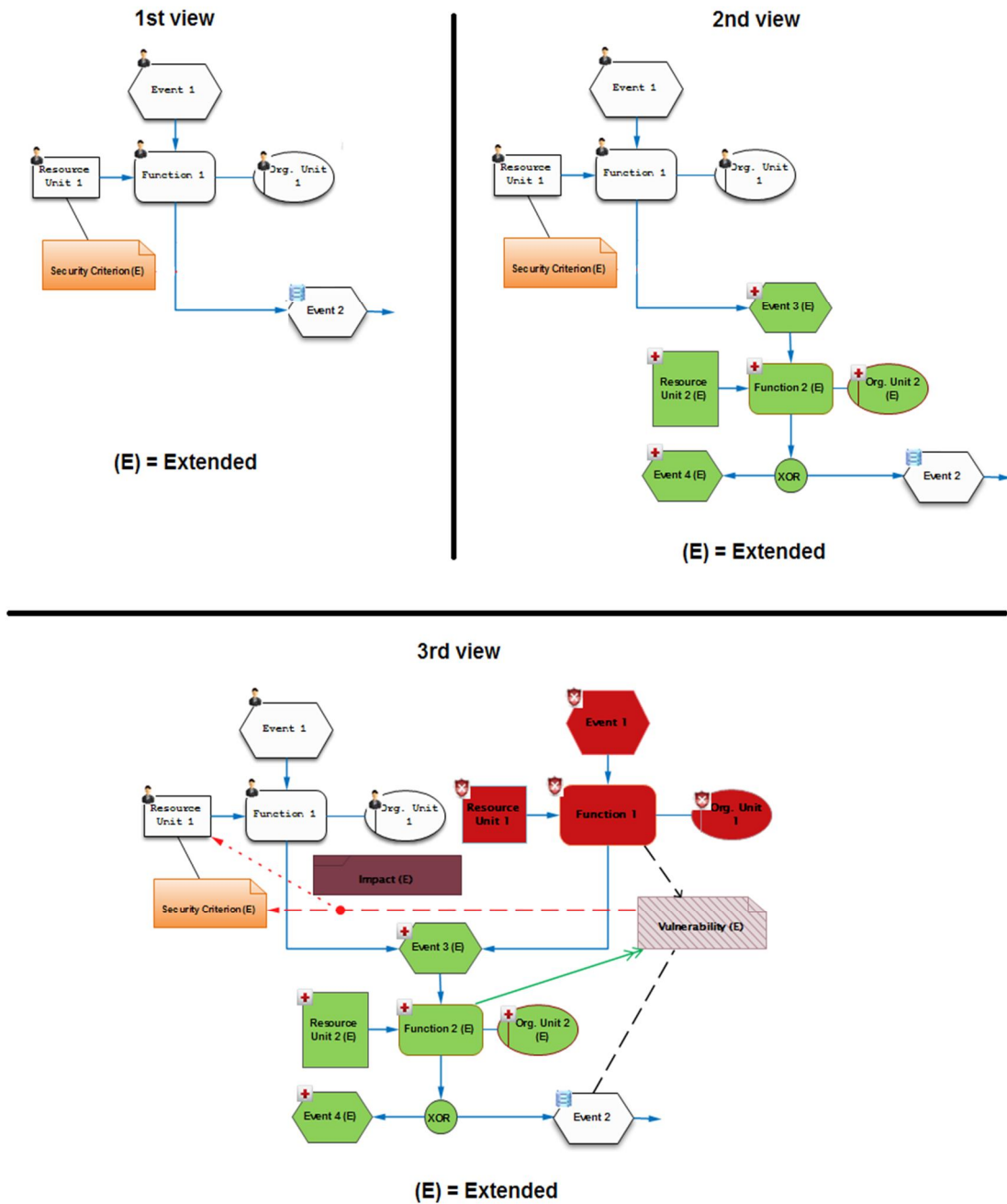


Figure 6.18 – Three steps of the guidelines to use Security-Oriented EPC.

In conclusion, the guidelines to use Security-Oriented EPC could be defined in three steps in order to point out the significance of security needs during the business process modeling. The first step is basically the main business process model and we just analyse, as a security risk analyser, and find out the parts of the model where security requirements might be defined. To do this in first operation we just identify the security criterion(s).

During the second step we start editing the existing model by introducing the extended EPC, defining security requirement(s) and how security criterion(s) is fulfilled by the defined security requirement(s). In the end, the third step is to analyse the metrics of the extended constructs and based on these metrics find the result(s) how the system is treated and what would the cost be. Third step will also lead us to analyse the return on security

investment which will answer the question “*Why do we need to use Security-Oriented EPC in Business Modeling?*”. Figure 6.18 illustrates the architecture of these three steps. We have already analysed all three steps in previous sections. We will focus on step three deeper in following chapter(s).

Chapter 7. MEASURING SECURITY RISKS USING SECURITY-ORIENTED EPC

In this chapter we define metrics of security-oriented EPC. Metrics are important during the calculation of return on investment (ROI [27]). We will use metrics with a scenario in our Internet Store running example and later will measure the security risks and calculate the cost on the treatment and return on security investment (ROSI [27]).

7.1 Metrics Definitions for Security-Oriented EPC

We first present and then use the predefined ISSRM metrics definitions in order to define the metrics values in our extended EPC language. We use this method because in our previous analysis during the all extension process, we chose ISSRM as basis. The metrics analysis table for ISO/IEC 27005 [26] is shown below.

By considering the Table 7.1, we can start analyzing the extended abstract syntaxes of EPC language in low level (based on usual constructs) and high level (based on Process Path), due to the fact that in previous chapter we have combined the EPC meta-model and ISSRM domain model and this approach will help us during the metrics analysis.

Table 7.1 – Metric Analysis Table for ISO/IEC 27005 [26].

ISSRM Concept	Concept	Metric	ISSRM Metric
Asset	Asset	Value	-
Business Asset	Primary Asset	Value	Value
IS Asset	Supporting Asset	-	-
Risk	Risk	Risk Level	Risk Level
Event	Event	Likelihood	Potentiality
Impact	Consequence	Business Impact Value	Impact Level
Threat	Threat	Frequency of Occurrence	Likelihood
Vulnerability	Vulnerability	Easiness of Exploitation	Vulnerability Level
Security Requirement Control	Control	Effectiveness	Risk Reduction

As an initialization, let's start with high level abstract syntax metrics analysis. Compared to the low level model we have less constructs in high level abstract model. However, the metrics analysis of new constructs which are defined by us in the extended EPC won't change in both models, since the main difference between two models is that in high level model we show the constructs in abstract level with Process Paths.

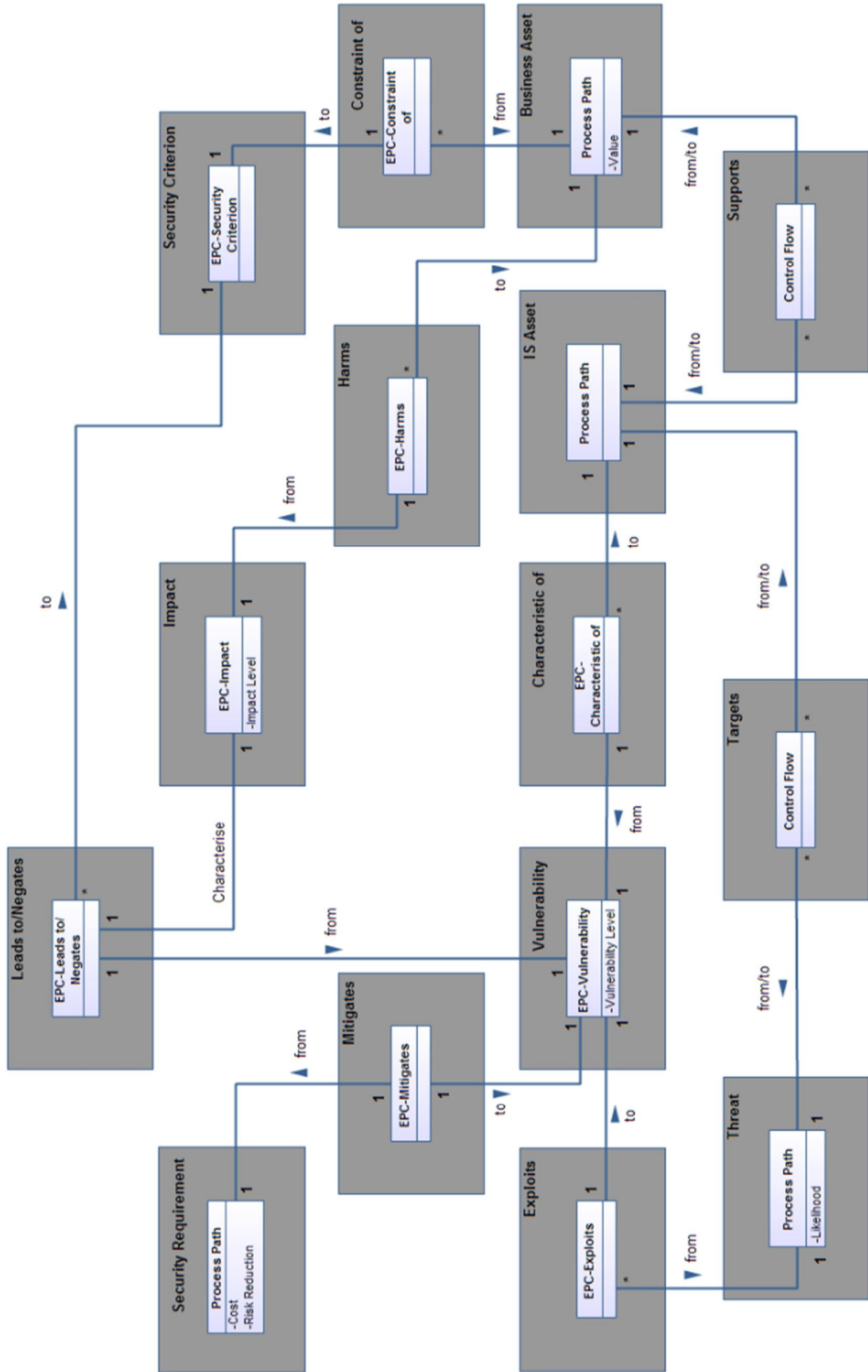


Figure 7.1 – High Level Security-Oriented EPC Abstract Syntax enriched with Metrics.

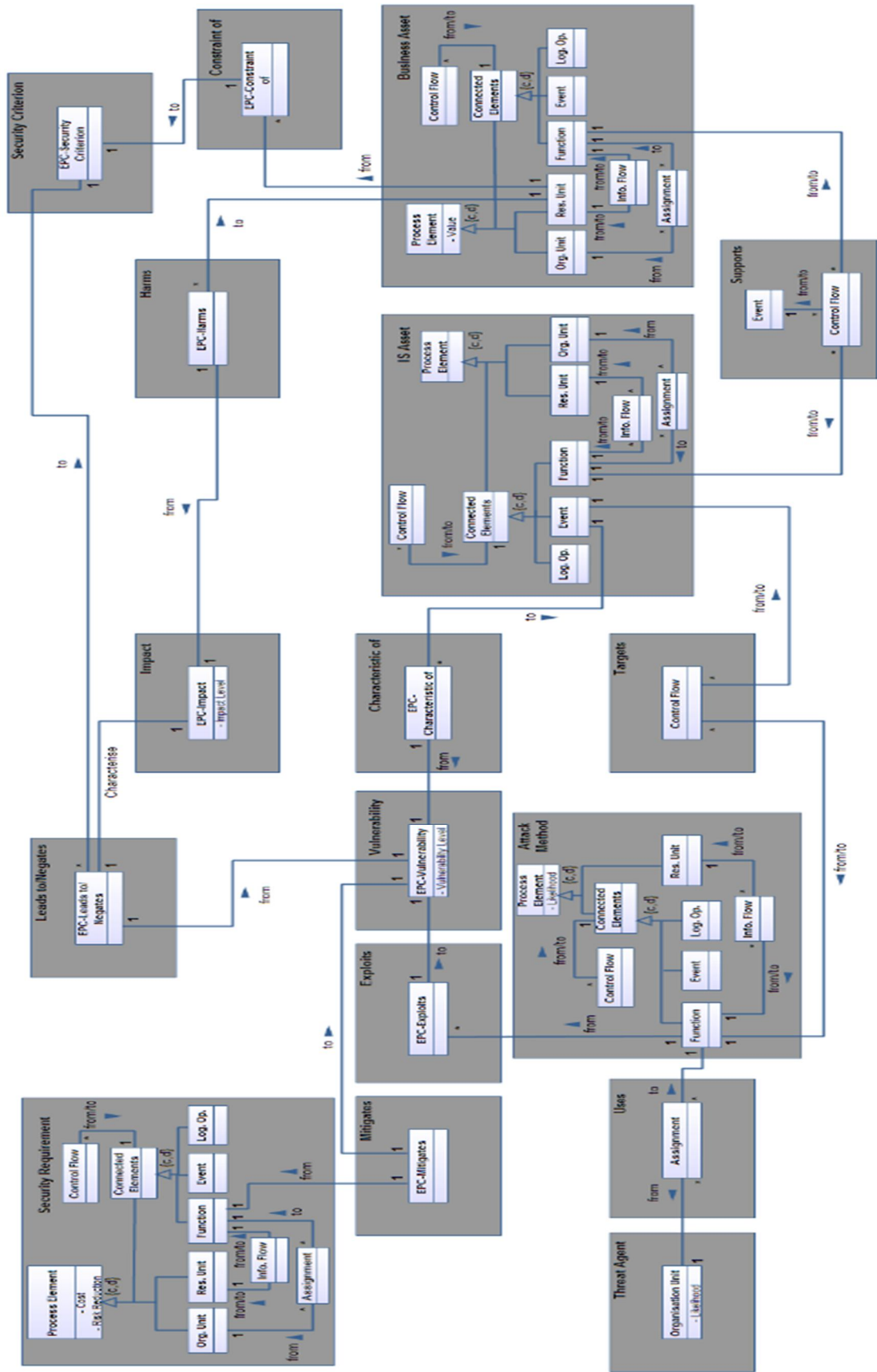


Figure 7.2 – Low Level Security-Oriented EPC Abstract Syntax enriched with Metrics.

In particular, if we start with Business Asset cluster in Figure 6.8 in Chapter 6, it is possible to say that Business Process Path in that cluster is a primary asset and its metric can be defined as “Value”. As another instance, we can define the metric of the EPC-Impact construct as “Impact Level” because it exists in the Impact cluster. Similarly, we define all the metrics of the constructs of high level abstract syntax according to the Table 7.1, and in the end we obtain a new meta-model which is called as “*High Level Security-Oriented EPC Abstract Syntax enriched with Metrics.*” and shown in Figure 7.1. On the other hand, since we do not have a unique construct which refers to Event concept of the ISSRM domain model (see Figure 6.3 in Chapter 6), we can not match the “Potentiality” metric with any of our EPC constructs. Same consequence is applicable for the Risk concept.

Similarly, we can define the metrics in low level abstract syntax as well (see Chapter 6, Figure 6.16). The only difference is that we have to define which construct is related with the metric. In high level construct (Process Path level) we defined the metrics inside the Process Path classes in Figure 7.1. In Figure 7.2 we define the metrics in the class of Process Element since all other constructs are subclasses of Process Element. In next sections we will define how the metrics value, cost or level be separated inside the constructs.

7.2 Return on Security Investment (ROSI) of Security-Oriented EPC

The definition of Return on Security Investment is the following: $ROSI = \text{monetary risk mitigation} - \text{cost of control}$. Therefore, a security investment is judged to be profitable, if the risk mitigation effect is greater than the expected costs [27].

Evaluation of the ISSRM Metrics

This section is dedicated to the experimentation of the ISSRM metrics on our business case in order to analyse all the security requirements and in the end calculate the Return on Security Investment of our case.

Process and Approach

The process followed is based on the ISO/IEC 27005 standard, which provides guidelines for performing security risk management. As seen in previous section, the ISO/IEC 27005 standard promotes the use of metrics. In our case we try to adapt and complete these metrics with regards to the Online Internet Store System. We will consider the case in Figure 7.3, the first attack of the violator, afterwards we will analyse the metrics on different constructs in order to calculate the ROSI.

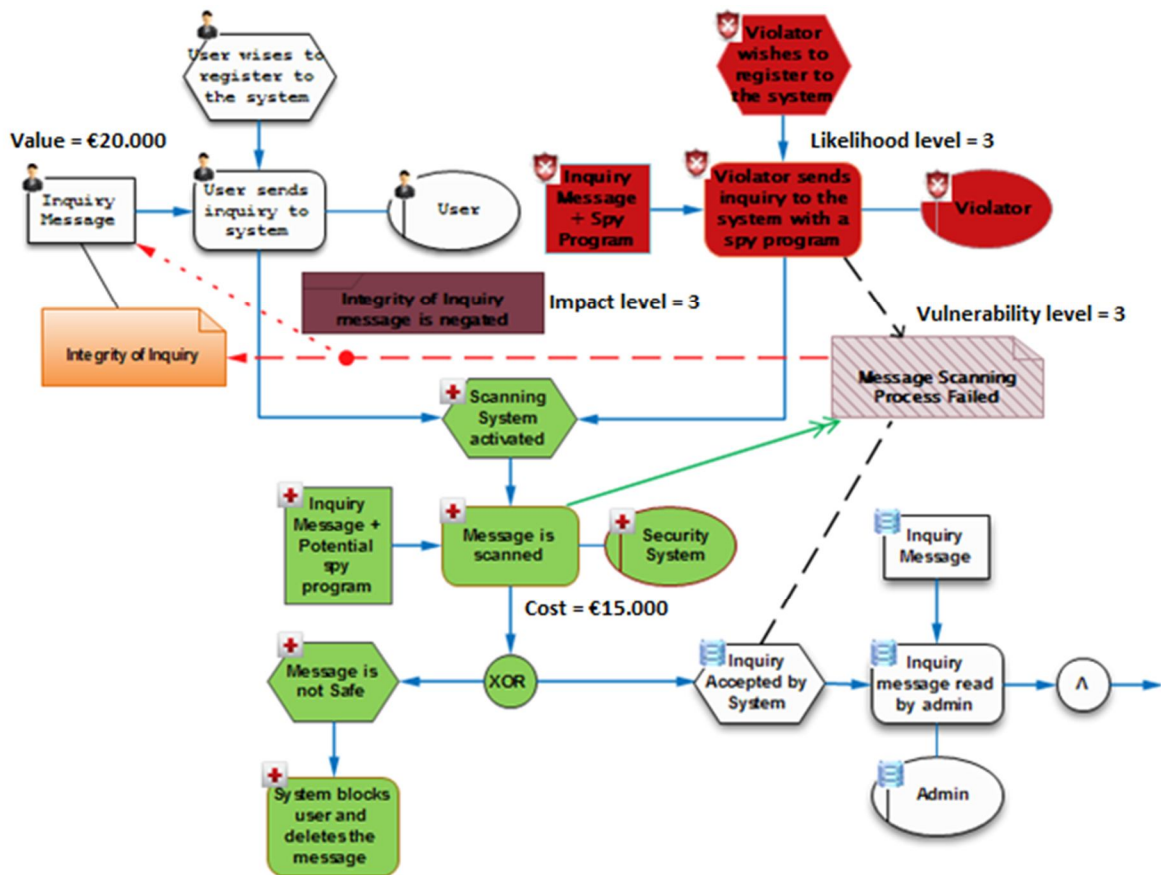


Figure 7.3 – Inquiry by user (Business Asset), Scanning of the inquiry message (Risk-treatment), Attack of the violator with a spy program attached to the inquiry message (Risk), Inquiry accepted and read by admin (IS Asset).

Context and Assets Identification

In this step, the assets of Internet Store System are identified. We first analyse what the business assets of the organization are, later, based on the inventory of the IS assets we map each IS asset to its related business asset(s).

Table 7.2 – Business and IS Assets of Internet Store System Inquiry sending process.

Business Assets	Organisation Unit[User], Function[User sends inquiry], Resource Unit[Inquiry Message], Event[User wishes to register]
IS Assets	Organisation Unit[Admin], Event[Inquiry accepted], Function[Inquiry message read], Resource Unit[Inquiry Message]

According to the business asset value metrics identification, we should focus on the importance and privacy of the inquiry message (resource unit). When a user is registering to the system, he/she might enter a lot of private information with an inquiry message, or vice versa. Based on this situation, our values and costs are defined in Table 7.3.

Table 7.3 – Qualitative scale of value for the value of business assets.

Business Asset Value		
Inquiry Message Information		
Value Estimate	Approx. Value	Description
High	€ 20.000	Inquiry message contains a lot of private information
Normal	€ 10.000	Inquiry message contains few private information
Low	€ 4.000	Inquiry message contains almost no private information

Security Objectives Determination

In this step, we analyse the security criterion for the business asset. As it can be seen from the Figure 7.3, our security criterion and related constructs are defined on the following tables.

Table 7.4 – Security Criterion of the Internet Store System Inquiry Sending process.

Security Criterion	EPC – Constraint of, EPC – Security Criterion[Integrity of Inquiry]
---------------------------	---

As we defined in the previous chapters, there are three important concepts in security criterion part; Confidentiality, Integrity and Availability. In our case, we consider the all possible security need for the related security risk. To do this, we analyse these three factors in different levels. Table 7.5 clearly identifies the levels of the security need and their effects on the business asset. During our analysis, we choose which level of security need(s) has to be assured.

Table 7.5 – Qualitative scale of value for the security need metric.

Security Objective			
Security Need for Inquiry Message Information			
	Need for Confidentiality	Need for Integrity	Need for Availability
0	No need of confidentiality	No need of integrity	No need of availability
1	Disclosure of the information restricted to the Internet Store System	-	Disruption only internal to Internet Store system
2	Disclosure of the information restricted to the Internet Store System and its users	Modifications that does not influence the user registration and login	Disruption with minor effect on users
3	Disclosure of the information restricted to Internet Store System Users	Modifications that influences the user registration and login	Disruption with major effect on users

Risk Analysis and Assessment

Once security needs of assets are defined for each security criterion, we analyse the risks. We first start by identifying the threats relevant to the IS. We analyse all the Security Risk related constructs in Table 7.6. Accordingly, we identify the likelihood of the threat in our system. Likelihood identification is done in Table 7.7. The lower level of likelihood represents a more secure system.

Table 7.6 – Risk related constructs in Internet Store System Inquiry sending process.

Threat Agent	Organisation Unit[Violator]
Attack Method	Function[Violator sends inquiry with spy program], Event[Violator wishes to register system], Resource Unit[Inquiry message + Spy program]
Vulnerability	EPC-Vulnerability
Impact	EPC-Impact
Threat	Attack Method + Threat Agent
Event	Threat (Attack Method + Threat Agent) + Vulnerability
Risk	Event (Threat + Vulnerability) + Impact

Table 7.7 – Qualitative scale of value for the likelihood metric.

Likelihood that the violator will try to register to the Internet Store System	
Level	Description
1	Unlikely according to statistics
2	Can happen at most 4 times in a year
3	Can happen more than 4 times in a year

Threats are associated with some vulnerabilities that are exploited by the threat for the risk to take place effectively. Once again, the vulnerabilities are identified through a brainstorming with some key actors of the organisation and then complemented by an analysis of the available knowledge bases. In Table 7.8, vulnerability levels and descriptions are clarified for the Internet Store System Inquiry process.

Table 7.8 – Qualitative scale of value for the vulnerability level metric.

Level of Vulnerability check Internet Store System accepts Inquiries	
Level	Description
0	Very Low – security measures in place and so far no threat has happened
1	Medium – security measures in place (but once a threat has happened)
2	High – no effective security measures in system (approx. 4 attacks per year succeeded)
3	Very High – extremely inadequate security measures (not applied or expired)

Table 7.9 – Risk matrix.

Potentiality Impact Level	0	1	2	3	4	5
1	0	1	2	3	4	5
2	0	2	4	6	8	10
3	0	3	6	9	12	15

Impact level is directly related with the potentiality. It is necessary to make such calculation assumptions for determining the risk level, and determining the risk level is a must for the certification. Then a risk matrix is created, in Table 7.9. This risk matrix indicated the risk level, based on the potentiality and the maximum impact level of the concerned impacts for the studies business assets.

$$Potentiality = likelihood + vulnerability level - 1$$

Risk Level

In our case, likelihood is high level since we assume that the Violator attempted to register to the system five times in a year, which is more than four. In this case, vulnerability level is also high as we can see from the Table 7.8. Also we only take into account the integrity of the inquiry message since there is not such related security criterion about confidentiality or availability of inquiry message. Consequently:

Table 7.10 – Risk level calculation table.

Business Asset		Inquiry Message Information	
Security Need		C = 0	I = 3
	A = 0		
Threat	Likelihood	Can happen more than 4 times in a year	3
Vulnerability	Vulnerability Level	System is very vulnerable due to the lack of security requirements	3
Risk Level		RL	

$$\text{Since; } Potentiality = likelihood + vulnerability level - 1,$$

$$\text{Considering our business asset, } Potentiality = 3 + 3 - 1 = 5,$$

Where our Impact Level is 3 and RL (Risk Level) is **15** (see Table 7.9).

Risk Treatment Decisions

For each risk, it is now necessary to choose a suited risk treatment and associated security requirements. In our method based on ISO/IEC 27005, each risk having a level inferior to the risk acceptance level is systematically accepted. If the risk level was superior to the risk acceptance level, it is necessary to reduce, transfer or avoid the risk.

Table 7.11 – Risk treatment methods and their descriptions.

Risk Treatment Decisions	Definition
<u>Risk Reduction</u>	Action to lessen the probability, negative consequences, or both, associated with a risk
Risk Avoidance	Decision not to be involved in, or to withdraw from a risk
Risk Transfer	Sharing with another party the burden of loss for a risk
Risk Retain	Accepting the burden of loss from a risk

Table 7.12 – Security Requirements Definition

Risk Treatment	Risk Reduction
Security Requirements	Function[Message scanning], Event[Scanning system activated], Organisation Unit[Security system], Resource Unit[Inquiry message + Potential spy program], Function[Block user and delete inquiry message]
Control	Implementing and adding a message scanning functionality to the system (effectiveness assumption 80%)

Considering our security requirements definitions, received inquiry message is scanned in order to clarify if it includes a spy program or not, if spy program is detected, then security system blocks the user and deletes the inquiry message. However, in some circumstances the spy program is attached to the inquiry message so professionally that scanning system can not detect the threat (the case of 80% effectiveness).

Control Selection and Implementations

Table 7.13 – Risk assessment and treatment table.

Risk Treatment		Risk Reduction by introducing the new additional scanning security system (which has approx. 80% effectiveness) to the system	
Security Requirement	New Vulnerability Level	Scan all messages and block approx. 80% of them which include spy program	1 (80% effectiveness reduced the threat to once in a year)
New Risk Level		9	
Risk Reduction		6	

As we can see from the Table 7.13, after implementing the scanning security system we reduced the risk level to 9. Since we have calculated the risk reduction, in our case we can now calculate the ROSI:

Cost of implementing additional scanning system for the incoming inquiry messages is €15.000.

$$ROSI = \{[(Risk Exposure * Risk Mitigated) - Solution Cost] / Solution Cost\} * 100\%$$

Risk Exposure = 5 * 20.000 = € 100.000

Risk Mitigated = Risk Reduction / Risk Level = 6/15

Solution Cost = € 15.000

ROSI (Scanning Module Implementation) = 167%

7.3 Summary

Return on security investment (ROSI) has been difficult to calculate successfully. In the absence of actual data on the number of incidents, organisations are often forced to make estimates. Also, the impact of an individual incident can be difficult to assess. In this chapter we have analysed the Security-Oriented EPC with ISSRM metrics defined in [26]. The purpose of this analysis is to understand how the return on security investment (ROSI) of extended EPC could be calculated. As a result, we can say that extended EPC helps to measure the risk. Because by the ROSI calculation we analysed the cost with a security requirement definition and with a solution cost of € 15.000 the system gets return on investment the percentage of 167. Which means that with that price risk reduction can be achieved in high level.

Chapter 8. TRANSFORMATION GUIDELINES FROM SECURITY-ORIENTED EPC TO MAL-ACTIVITY DIAGRAMS

In this chapter we will define transformation rules from Security-Oriented EPC to Mal-Activity Diagrams with a running example by using the alignment of the languages with ISSRM domain model. MAD is chosen for the transformation process since conclusion of Chapter 3 and alignment of MAD with ISSM in Chapter 4 showed us that MAD is a highly security-oriented modeling language and it will be suitable during the generation of construct-based transformation rules.

Table 8.1 shows the alignment between ISSRM, Security-Oriented EPC and Mal-activity Diagrams. Security-Oriented EPC column is based on the alignment table we have obtained in chapter five, and Mal-activity Diagram column belongs to [28].

We use our running example (online registration to internet store) in three different levels of ISSRM concepts; Asset-related, Risk-related and Risk Treatment-related, and illustrate the Security-Oriented EPC and MAD models in order to identify the similarities and transformation rules between Security-Oriented EPC and MAD.

8.1 Asset-related Transformation

During the asset-related transformations we focus on the constructs and relationships which we defined as an asset in previous chapters. The running example is the one which we used during the alignment and EPC language extension processes. Our running example can shortly be described as, user wishes to register to the system and sends inquiry to the system, when inquiry message is accepted by system it is handled and read by admin. As you can see in the Figure 8.1, since we use the Security-Oriented EPC, we can notice the extended constructs “Security Criterion” and “EPC-Constraint of”. All other constructs are the main constructs (Event, Function, Organisation Unit, Resource Unit, Control Flow and Information Flow) of EPC.

Table 8.1 – Alignment of the ISSRM Concepts and the Security-Oriented EPC and MAD Constructs.

ISSRM Model		ID	Security-Oriented EPC	Mal-activity Diagrams
Asset	<i>Asset</i>	a	-	-
	<i>Business Asset</i>	b	(B) Event, Function, Organisation Unit, Information Flow, Control Flow, Logical Operator, Resource Unit, Assignment	Activity, Decision, ControlFlow, Swimlane
	<i>IS Asset</i>	c	(IS) Event, Function, Organisation Unit, Information Flow, Control Flow, Logical Operator, Resource Unit, Assignment	Swimlane, Activity, Decision, ControlFlow
	<i>Supports</i>	r7	Control Flow	ControlFlow
	<i>Security Criterion</i>	d	EPC-Security Criterion	MAD-Security Criterion
	<i>Constraint of</i>	r8	EPC-Constraint of	
Risk	<i>Risk</i>	e	Combination of Event and Impact constructs	Combination of Event and Impact constructs
	<i>Impact</i>	f	EPC-Impact	Mal-activities
	<i>Event</i>	g	Combination of Vulnerability and Threat constructs	Combination of Vulnerability and Threat constructs
	<i>Exploits</i>	r4	EPC-Exploits	-
	<i>Characteristic of</i>	r9	EPC-Characteristic of	MAD-
	<i>Vulnerability</i>	h	EPC-Vulnerability	Vulnerability
	<i>Targets</i>	r5	Control Flow	ControlFlow
	<i>Threat</i>	i	Combination of Attack Method and Threat Agent constructs	Combination of Attack Method and Threat Agent constructs
	<i>Significance assessed by</i>	r12	-	-
	<i>Harms</i>	r6	EPC-Harms	-
	<i>Threat Agent</i>	j	Organisation Unit	Mal-swimlane
	<i>Leads to / Negates</i>	r1	EPC-Leads to / Negates	Negates
	<i>Uses</i>	r3	Assignment	Swimlane contains Mal-Activity
	<i>Attack Method</i>	k	(Risk) Event, Function, Resource Unit, Information Flow, Control Flow, Logical Operator	Mal-activities, Mal-decision, ControlFlow, Mal-swimlane
Risk Treatment	<i>Risk Treatment</i>	l	-	-
	<i>Security Requirement</i>	m	(RT) Event, Function, Resource Unit, Organisation Unit, Information Flow, Control Flow, Logical Operator, Assignment	MitigationActivity, Decision, Swimlane, ControlFlow
	<i>Mitigates</i>	r2	EPC-Mitigates	MitigationLink
	<i>Control</i>	n	-	Swimlane
	<i>Refines</i>	r10	-	-
	<i>Implements</i>	r11	-	-

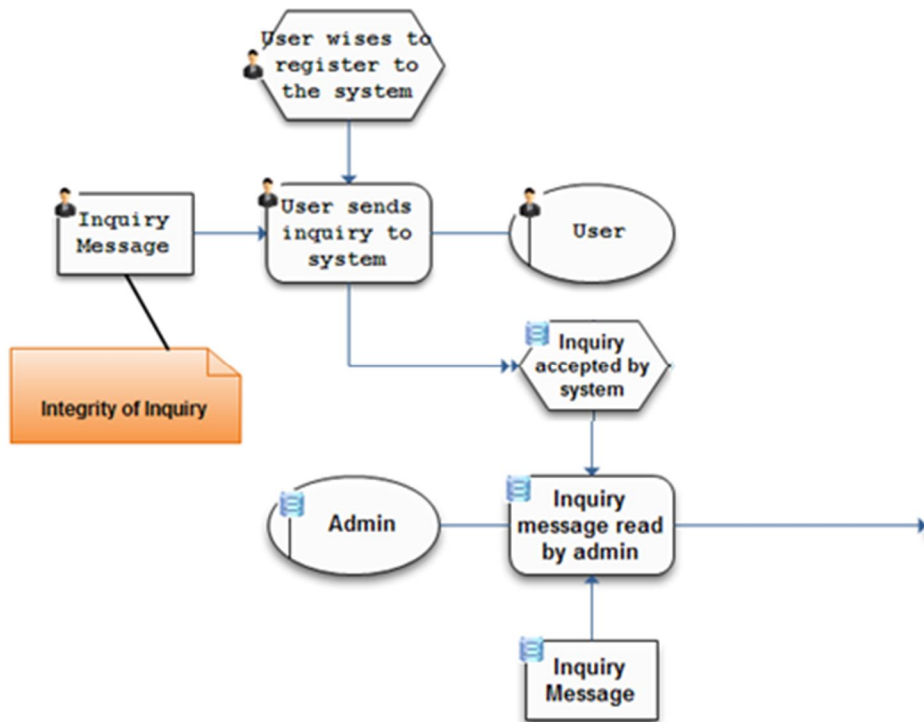


Figure 8.1 – Security-Oriented EPC diagram of online registration (message handling) of the Internet Store.

Similarly, we model the MAD of the running example based on the same scenario. The main difference is the concrete syntax of the languages. In order to reduce this difference, we define transformation rules which will help us to transform a model from Security-Oriented EPC model to MAD.

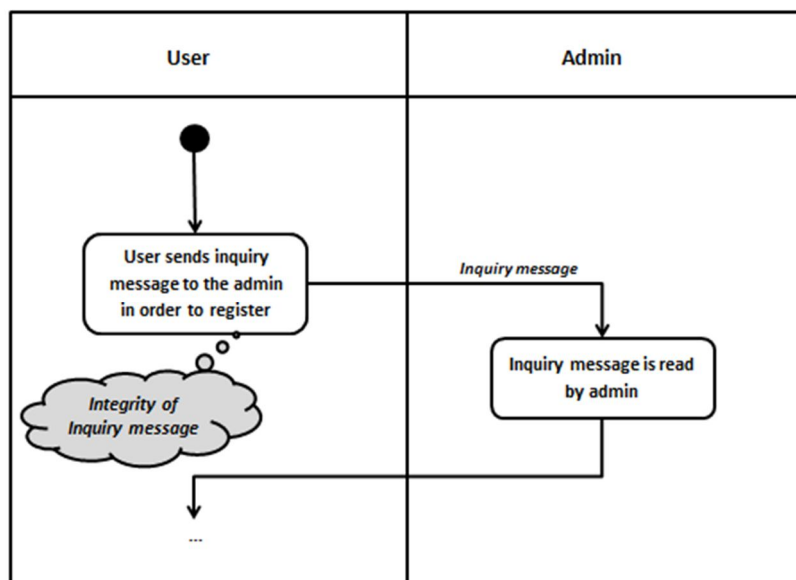
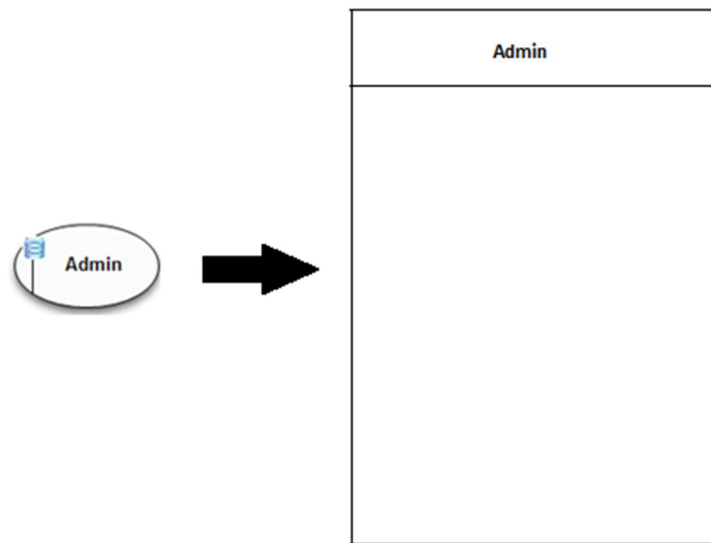


Figure 8.2 – Mal-Activity Diagram of online registration (message handling) of the Internet Store.

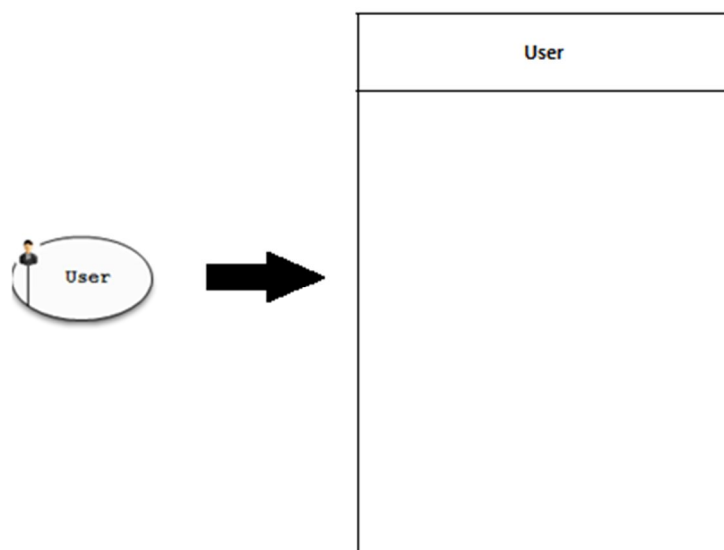
Transformation rules for Asset-related transforming:

TR1 = From *Organisation Unit* to *Swimlane*

- Transformation rule one is based on ID-c in Table 8.1. Refers to IS Asset; e.g. Admin ↔ Admin. The purpose and the characteristics of these two constructs are very similar. The only difference is that in MAD the *Swimlane* construct includes all other constructs in it, acts like a cluster which includes the whole operation in it whereas the *Organisation Unit* in EPC is just connected to a *Function*. Therefore, the example below indicates the IS asset since the „Admin“ in our scenario is an IS asset.



- Transformation rule one is also based on ID-b in Table 8.1. Refers to Business Asset; e.g. User ↔ User. Similar to the previous example, the example below indicates the Business asset since the „User“ in our scenario is a Business asset.



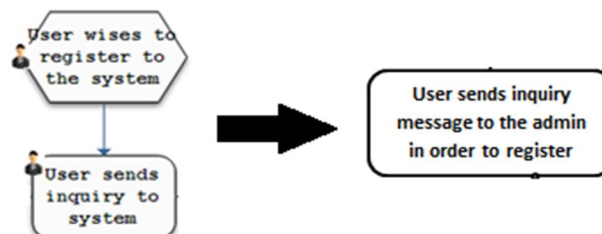
TR2 = From *EPC-Security Criterion* + *EPC-Constraint of* to *MAD-Security Criterion*

- Transformation rule two is based on ID-d and ID-r8. Refers to Security Criterion concept and „Constraint of“ relationship; e.g. Integrity of Inquiry ↔ Integrity of Inquiry Message. The only difference between *EPC-Security Criterion* and *MAD-Security Criterion* is that they are linked to the different constructs. *EPC-Security Criterion* is an extended construct and it is linked to the *Resource Unit* with another extended construct called *EPC-Constraint of* whereas the *MAD-Security Criterion* is linked to an *Activity* as we can see in Figure 8.2.

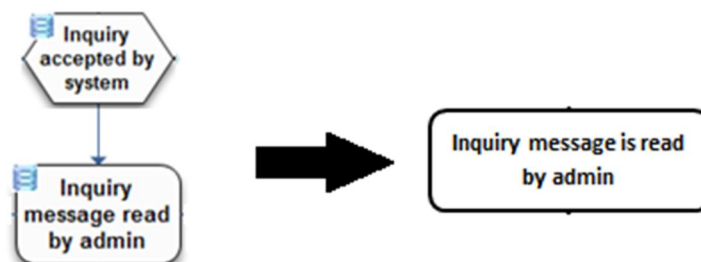


TR3 = From *Event* + *Function* + *Control Flow* to *Activity*

- Transformation rule three is based on ID-b. Refers to Business Asset; e.g. „User wishes to register“ and „User sends inquiry“ and *Control Flow* ↔ User sends inquiry message to admin in order to register. The transformation can not be done one-to-one constructs since in EPC the main process is divided into three parts, *Event*, *Control Flow* and *Function*. In particular, in MAD there aren't any constructs which clarifies the purpose or preliminary phase of the *Activity*. Example below indicates the business asset since the process is performed by user. Also, in MAD, an *Activity* is followed by another *Activity* or *Decision*.

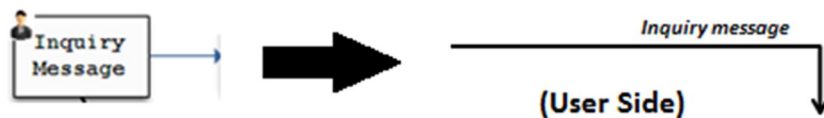


- Transformation rule three is based on ID-c. Refers to IS Asset; e.g. „Inquiry accepted“ and *Control Flow* and „Inquiry message is read by admin“ ↔ Inquiry message is read by admin. Similar to the previous example, here we show the example of IS asset since these processes are performed by admin in system side.

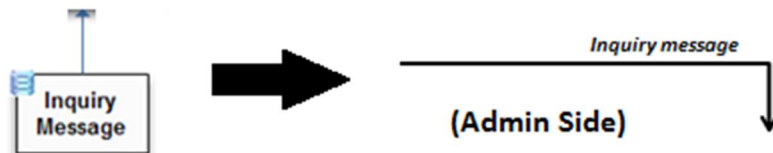


TR4 = From *Resource Unit + Information Flow* to *ControlFlow*

- Transformation rule four is based on ID-b. Refers to Business Asset; e.g. „Inquiry Message“ and *Information Flow* ↔ *ControlFlow* with a message. This transformation is little bit different then others and we have to consider two cases (Business asset and IS asset) since in MAD we do not show the same construct in different *Swimlanes* separately, instead, the construct is owned by both of the *Swimlanes*. On the other hand, in EPC, we can show the same construct in different levels of users. As illustrated in the below example, the *Resource Unit* which is connected to a *Function* with an *Information Flow* is transformed to the *ControlFlow* which appears in „User“ *Swimlane* in our scenario, on the Business asset side. Besides, here *ControlFlow* is connected to *Activity* in MAD.



- Transformation rule four is based on ID-c. Refers to IS Asset; e.g. „Inquiry Message“ and *Information Flow* ↔ *ControlFlow* with a message. Similarly, the example below analyses the same case in IS asset side since the Resource Unit shown below is connected to the *Function* which is performed by admin and the *ControlFlow* (in MAD diagram) appears in admin *Swimlane*. In particular, the *ControlFlow* acts as a single construct lying in two *Swimlanes* according to our scenario.



TR5 = From *Control Flow* to *ControlFlow*

- Transformation rule five is based on ID-r7. Refers to „Supports“ relationship. As we discussed in previous chapters, *Control Flow* in EPC acts like a support construct between IS asset and Business asset. Basically, the flow continues from IS asset to Business asset with a *Control Flow*. Similarly, in MAD, *ControlFlow* construct has the same role. When the process flows from one *Swimlane* to another one, support relationship is provided by *ControlFlow* construct.



8.2 Risk-related Transformation

During the risk-related transformations we focus on the constructs and relationships which we defined as risk in previous chapters. The running example is same but the difference is that the “Violator” as a threat is included and we represent the “Risk” constructs which are shown in red in Figure 8.3. White constructs represents “Asset” concept. The extended constructs (EPC-Vulnerability, EPC-Impact and so on) are also shown in Figure 8.3. Basically, Violator sends inquiry which includes a spy program and the inquiry message is accepted by the system without any scan, which causes an impact on the integrity of inquiry message of user, because the inquiry message contains important data of the user. The same scenario is illustrated in Figure 8.4 with MAD.

Transformation rules for Risk-related transforming:

TR6 = From *EPC-Vulnerability* + *EPC-Characteristic of* to *MAD-Vulnerability*

- Transformation rule six is based on ID-h and ID-r9. Refers to Vulnerability concept and „Characteristic of“ relationship; e.g. Message is being handled without any scanning ↔ Message is handled without scanning. The point we need to consider is that *MAD-Vulnerability* is connected to a *ControlFlow* on IS asset side whereas *EPC-Vulnerability* is connected (with *EPC-Characteristic of*) to an *Event* of the admin which is an IS *Organisation Unit*.

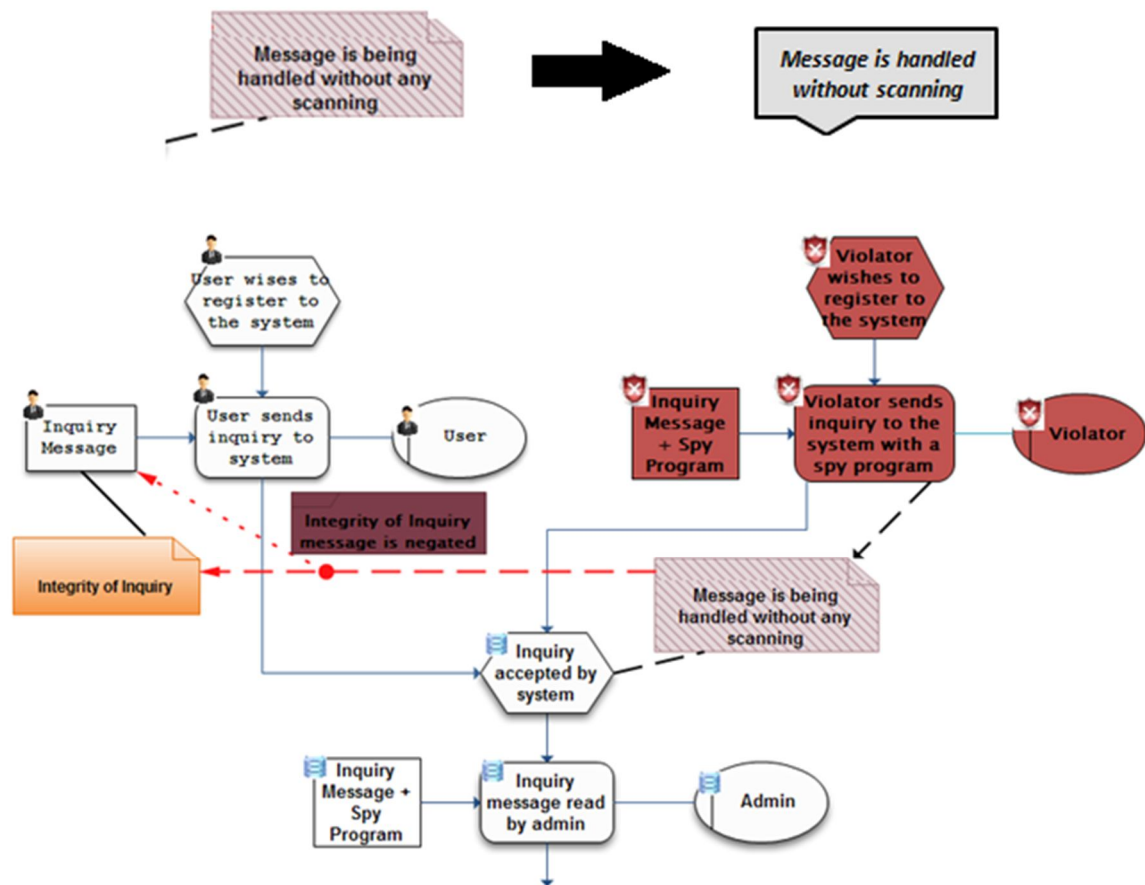


Figure 8.3 – Security-Oriented EPC diagram of online registration (message handling) of the Internet Store including security risk(s).

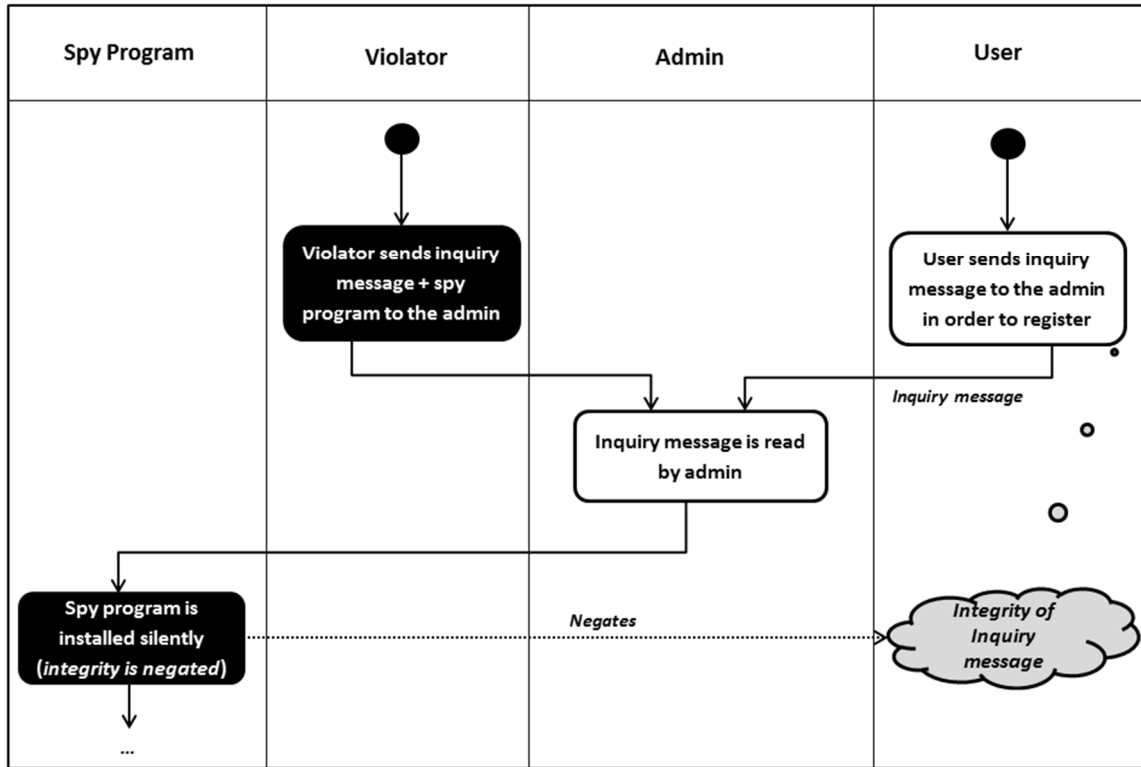


Figure 8.4 – Mal-activity diagram of online registration (message handling) of the Internet Store including security risk(s).

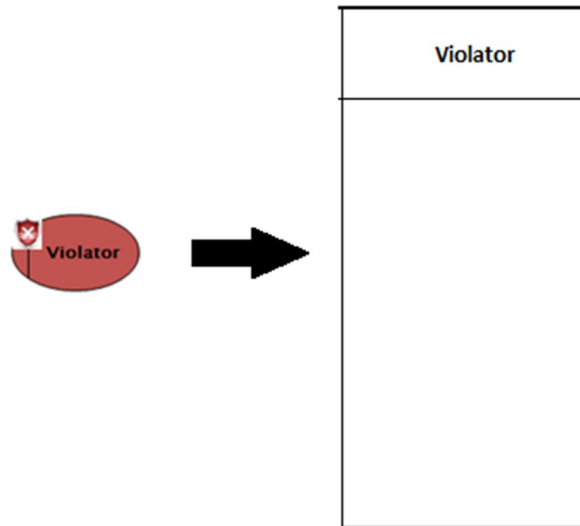
TR7 = From *EPC-Impact* to *Mal-activity*

- Transformation rule seven is based on ID-f. Refers to Impact; e.g. Integrity of inquiry message is negated ↔ Spy program is installed silently (integrity is negated). MAL-Activity lies on *MAL-Swimlane* and represents the impact as we can see on the example below. *EPC-Impact* is connected to *EPC-Harms* which is linked to *Resource Unit* as we can see in Figure 8.3.



TR8 = From (Risk) *Organisation Unit* to *Mal-Swimlane*

- Transformation rule eight is based on ID-j. Refers to Threat Agent; e.g. Violator ↔ Violator (or Spy Program). As we explained during the asset transformations, *Organisation Unit* and *Swimlanes* act similarly, only difference here is that the constructs belongs to Risk and indicates malicious actors (See transformation rule 1).



TR9 = From *EPC-Leads to/Negates* to *Negates*

- Transformation rule nine is based on ID-r1 in Table 8.1. Refers to „Leads to / Negates“ relationship; e.g. *EPC-Leads to/Negates* ↔ *Negates*. In MAD *Negates* construct initializes at *Mal-Activity* and linked to *MAD-Vulnerability* whereas *EPC-Leads to/Negates* initializes at *EPC-Vulnerability* and linked to *EPC-Security Criterion*.



TR10 = From (Risk) *Event + Function + Control Flow* to *Mal-activity*

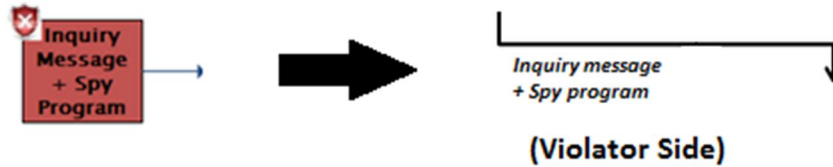
- Transformation rule ten is based on ID-k. Refers to Attack Method; e.g. „Violator wishes to register to the system“ and „Violator sends inquiry with a spy program“ and *Control Flow* ↔ Violator sends inquiry message and spy program to the admin. As we explained earlier, *Event, Function* and *Control Flow* of EPC acts like an *Activity*, here only difference from TR3 is that EPC constructs are Risk constructs shown in red and MAD construct is *Mal-Activity* which refers to malicious activity (See transformation rule 3).



TR11 = From (Risk) *Resource Unit + Information Flow* to *ControlFlow*

- Transformation rule eleven is based on ID-k. Refers to Attack Method; e.g. „Inquiry message plus Spy program“ and *Information Flow* ↔ *ControlFlow* with a

message. As we explained earlier, the *ControlFlow* here also lies in two *Swimlanes* (violator side and admin side) as shown in Figure 8.4. Below example indicates the transformation in Risk side (See transformation rule 4).



- Transformation rule ten is based on ID-c. Refers to IS Asset; e.g. „Inquiry message plus Spy program“ and *Information Flow* ↔ *ControlFlow* with a message. Below example indicates the transformation in IS asset side.



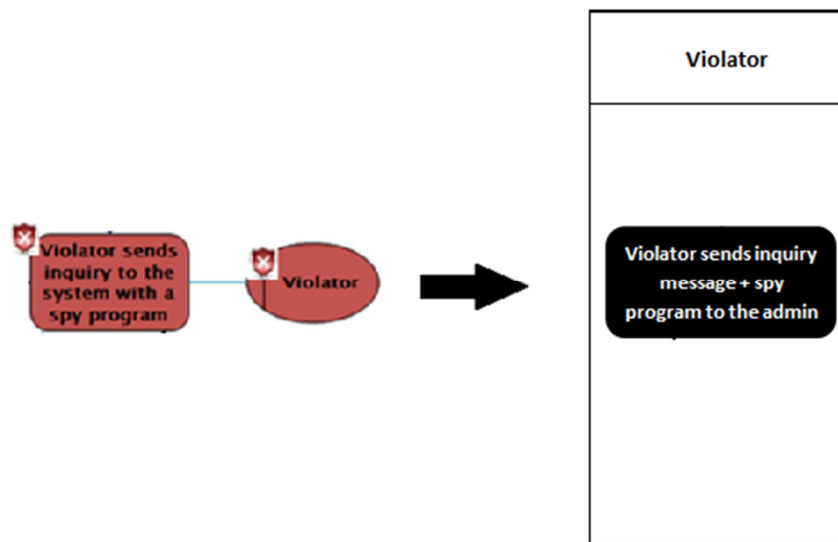
TR12 = From (Risk) *Control Flow* to *ControlFlow*

Transformation rule twelve is based on ID-r5. Refers to „Targets“ relationship. In EPC, *Control Flow* has another role which covers the relationship between threat and IS asset. When the process is flowing from threat to IS asset, *Control Flow* construct is used as target construct. In MAD, *ControlFlow* construct has the same role when the process is flowing from threat *Swimlane* to IS asset *Swimlane*.



TR13 = From (Risk) *Assignment* to *Swimlane contains Mal-Activity*

Transformation rule thirteen is based on ID-r3. Refers to „Uses“ relationship. The relationship between threat agent and attack method is defined by *Assignment* construct during the EPC and ISSRM alignment process since the *Organisation Unit* uses the attack method constructs. Similarly, *Swimlane* containing (uses) *Mal-Activity* has the same role. Following example illustrates the transformation rule, *Assignment* is the link between *Function* and *Organisation Unit*.



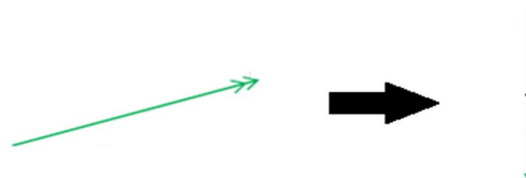
8.3 Risk Treatment-related Transformation

During the risk treatment-related transformations we focus on the constructs and relationships which we defined as risk in previous chapters. Risk treatment method and security requirements are included in our running example shown in Figure 8.5. The difference between previous models is that this time inquiry message is scanned by a security system before admin reads it. By the scanning process, threat which is created by violator is mitigated. The scenario is illustrated in Figure 8.6 with MAD as well.

Transformation rules for Risk Treatment-related transforming:

TR14 = From *EPC-Mitigates* (mitigates *EPC-Vulnerability*) to *MitigationLink* (mitigates *ControlFlow*)

- Transformation rule fourteen is based on ID-r2. Refers to „Mitigates“ relationship; e.g. Mitigation with *EPC-Mitigates* from „Message is scanned“ to „Message scanning process failed“ ↔ Mitigation with *MitigationLink* from „Scanning inquiry message“ to *ControlFlow* with a message. In MAD *MitigationLink* initializes at *MitigationActivity* and linked to *ControlFlow* whereas in EPC *EPC-Mitigates* initializes at risk-treatment *Function* and is linked to *EPC-Vulnerability*.



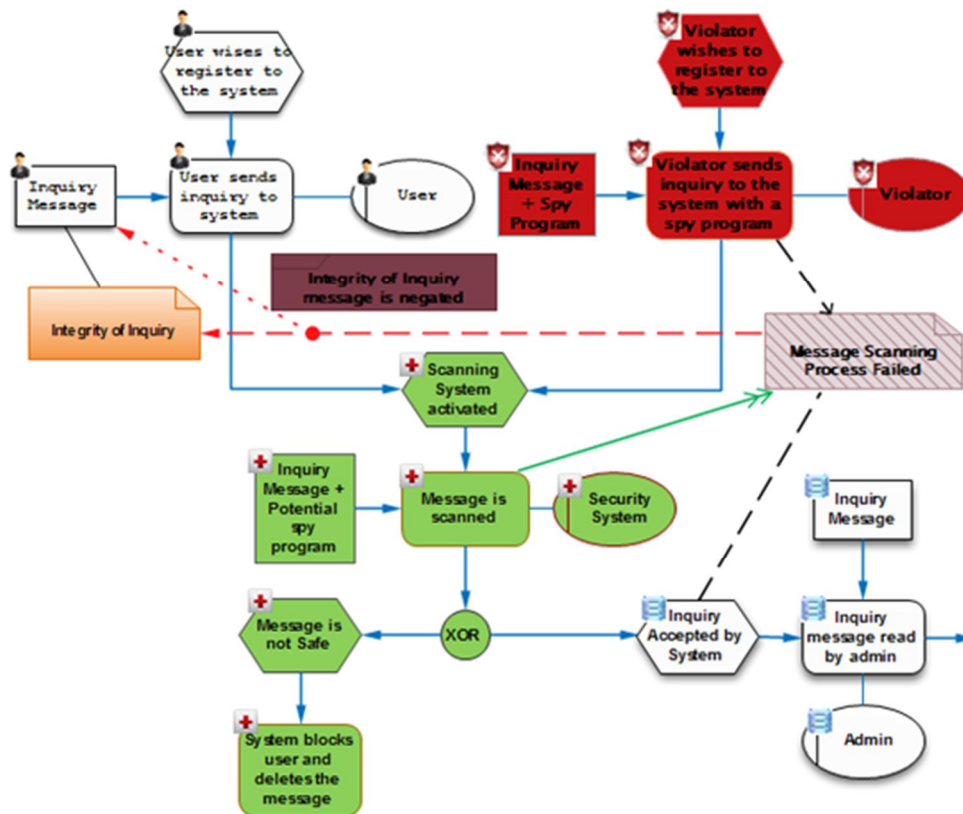


Figure 8.5 – Security-Oriented EPC diagram of online registration (message handling) of the Internet Store including risk treatment.

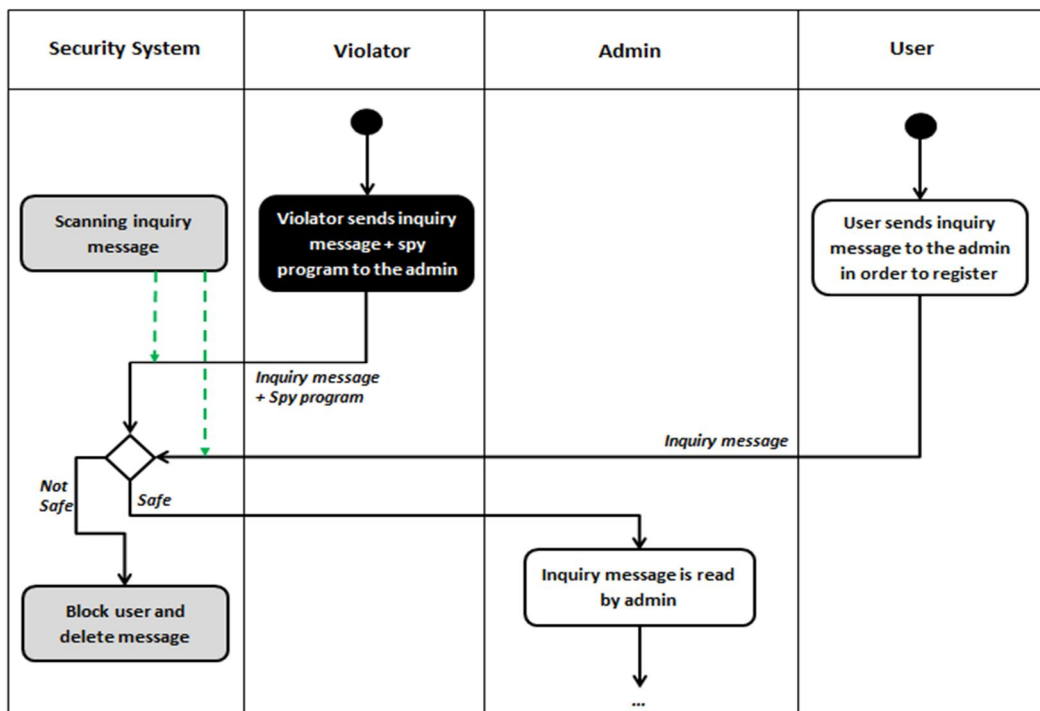
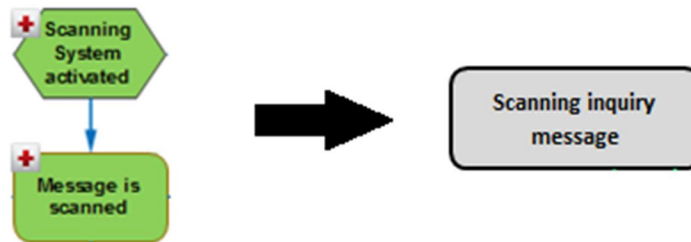


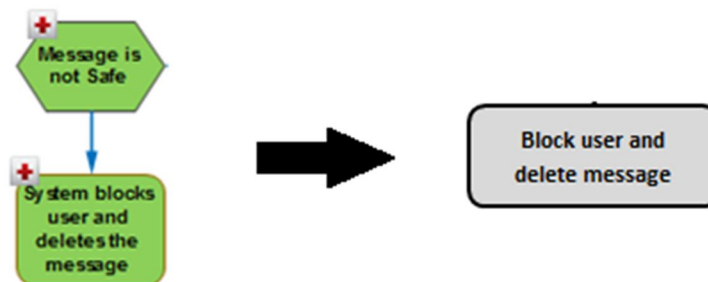
Figure 8.6 – Mal-activity diagram of online registration (message handling) of the Internet Store including risk treatment.

TR15 = From (RT) *Event + Function + Control Flow* to *MitigationActivity*

- Transformation rule fifteen is based on ID-**m**. Refers to Security Requirement; e.g. „Scanning system activated“ and „Message is scanned“ and *Control Flow* ↔ Scanning inquiry message. As we described the similarity between asset *Function*, *Event*, *Control Flow* and *Activity* earlier, here also similar transformation is indicated between risk-treatment *Function*, *Event*, *Control Flow* and *MitigationActivity* (See transformation rule 3 and 10).



- e.g. „Message is not safe“ and „System blocks the user and deletes the message“ and *Control Flow* ↔ Block user and delete message. Similar example is shown below.



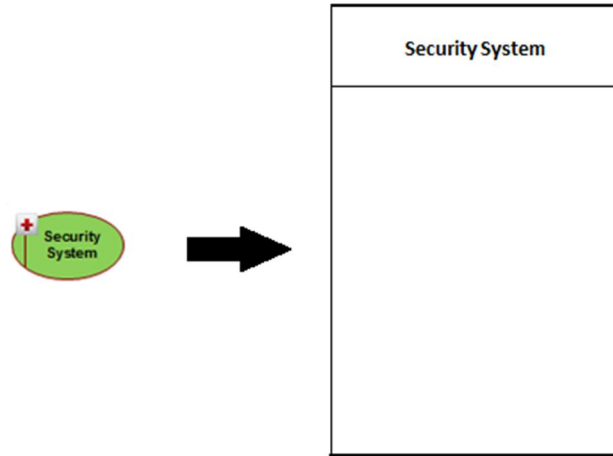
TR16 = From (RT) *Logical Operator* to *Decision*

- Transformation rule sixteen is based on ID-**m**. Refers to Security Requirement; e.g. XOR ↔ decision gate. The *Logical Operator* in EPC and *Decision* in MAD are simple structured constructs which act as a decision maker according to the flow of the process. They both are connected to control flows during the process flow.



TR17 = From (RT) *Organisation Unit* to *Swimlane*

- Transformation rule seventeen is based on ID-**m**. Refers to Security Requirement; e.g. Security System ↔ Security System (See transformation rule 1 and 8).



(Common for Asset, Risk and Risk Treatment concepts)

TR18 = From *Control Flow* to *ControlFlow*

- Transformation rule eighteen is based on ID-**c,k** and **m**. This transformation rule belongs to all three concepts defined below. Control flow might just indicate the flow of the process, it can be part of Asset, Risk or Risk Treatment as a construct.



The purpose of transformation from Security-Oriented EPC to MAD is to relate two completely different models to each other and provide business analysts different way of solution for a given problem in different perspectives. By transformation, we also see in which level of security EPC and MAD differs from each other and in which level of security they act in same way.

8.4 Summary

We have used the online registration to internet store running example in three different levels of ISSRM concepts; Asset-related, Risk-related and Risk Treatment-related. Also we have illustrated the Security-Oriented EPC and MAD models in order to identify the similarities and transformation rules between Security-Oriented EPC and MAD. In the end, we have obtained eighteen transformation rules which refer to the construct transformations from Security-Oriented EPC constructs to MAD constructs. Although transformation rules are identified and illustrated with models one by one, their effectiveness will be controversial until we validate and approve them.

Chapter 9. VALIDATION

Validation chapter consists of an introduction to the research area, description of the problem statement, detailed experiment planning, operation, presentation of the data analysis, interpretation of results and discussions about the findings and the conclusions.

9.1 Introduction

In validation chapter, we validate our transformation rules which have been generated in previous chapter. Validation process is done through a descriptive case study which is a test and performed by ten different IT-related people (Project Managers, Software Engineers, Electricians and Electronics Engineers).

9.2 Problem Statement

The purpose of the descriptive case study during the validation is to see how a participant can understand the transformation rules defined and illustrate the transformation of a given model by analysing the concepts and rules. We will now introduce the case study experiment planning and analyse each of the solutions performed by the participants and in the end discuss which parts of the transformation rules are clear and which are not.

9.3 Experiment Planning

The case study consists of the transformation rules and figures which are shown during the Chapter 8. All transformation rules are given with a clear explanation, including the Security-Oriented EPC models (Asset, Risk and Risk-treatment levels) of the online registration (message handling) of the Internet Store scenario. Only information hidden is the Mal-Activity Diagrams of the scenario, in the end of the test the participant is asked to draw MAD of Figure 8.5 of Chapter 8 by using the defined transformation rules. The answer is already shown in previous chapter with Figure 8.6. To make the test little bit easier the MAD of Figure 8.1 of Chapter 8 is given to the participant, the Asset model transformation. In particular, we identify and count each transformation rules used in the correct solution and we will compare the participant's solution with the correct one and get the percentage of the correctness of the result. Also, if the transformed construct is used in inaccurate flow then the rule will be counted as half point, even if the transformed construct is accurate itself. In the end we will get the average of these ten results and by this method we will validate the transformation rules in statistical test based percentage. Although these results will not give us the exact and 100% consistent numbers, we will have approximate idea and conclusion about how our transformation rules are successful, clear and effective.

9.4 Experiment Operation

Security-Oriented EPC model of online registration (message handling) of the Internet Store (to be transformed in Figure 9.1) and MAD of the same scenario (the result after transformation in Figure 9.2) are illustrated in figures. In the MAD, you can see the stars referring each transformed construct including transformation numbers. In total there are twenty transformations done by using fourteen different transformation rules, four transformation rules are not used in the model and we will discuss the reason during the interpretation and conclusion chapters:

1. Transformation rule number one is used (Business Asset-related transformation)
2. Transformation rule number three is used (Business Asset-related transformation)
3. Transformation rule number four is used (Business Asset-related transformation)
4. Transformation rule number five is used (Business Asset-related transformation)
5. Transformation rule number eight is used (Risk-related transformation)
6. Transformation rule number ten is used (Risk-related transformation)
7. Transformation rule number eleven is used (Risk-related transformation)
8. Transformation rule number twelve is used (Risk-related transformation)
9. Transformation rule number thirteen is used (Risk-related transformation)
10. Transformation rule number seventeen is used (Risk Treatment-related transformation)
11. Transformation rule number fifteen is used (Risk Treatment-related transformation)
12. Transformation rule number sixteen is used (Risk Treatment-related transformation)
13. Transformation rule number fifteen is used (Risk Treatment-related transformation)
14. Transformation rule number eighteen is used (Common transformation rule)
15. Transformation rule number eighteen is used (Common transformation rule)
16. Transformation rule number fourteen is used (Risk Treatment-related transformation)
17. Transformation rule number fourteen is used (Risk Treatment-related transformation)
18. Transformation rule number one is used (Asset-related transformation)
19. Transformation rule number three is used (Asset-related transformation)
20. Transformation rule number eighteen is used (Common transformation rule)

Now we can start analyzing the participant's data by using our result set including twenty transformations. Most of the results are similar to the correct solution, however we did not expect none of participants complete it 100% correctly. Because some people even did not know what MAD is. Also, we have underestimated some mistakes such as unnecessarily used constructs and transformations. Besides, some drawings look different than each other since different tools are used by participants.

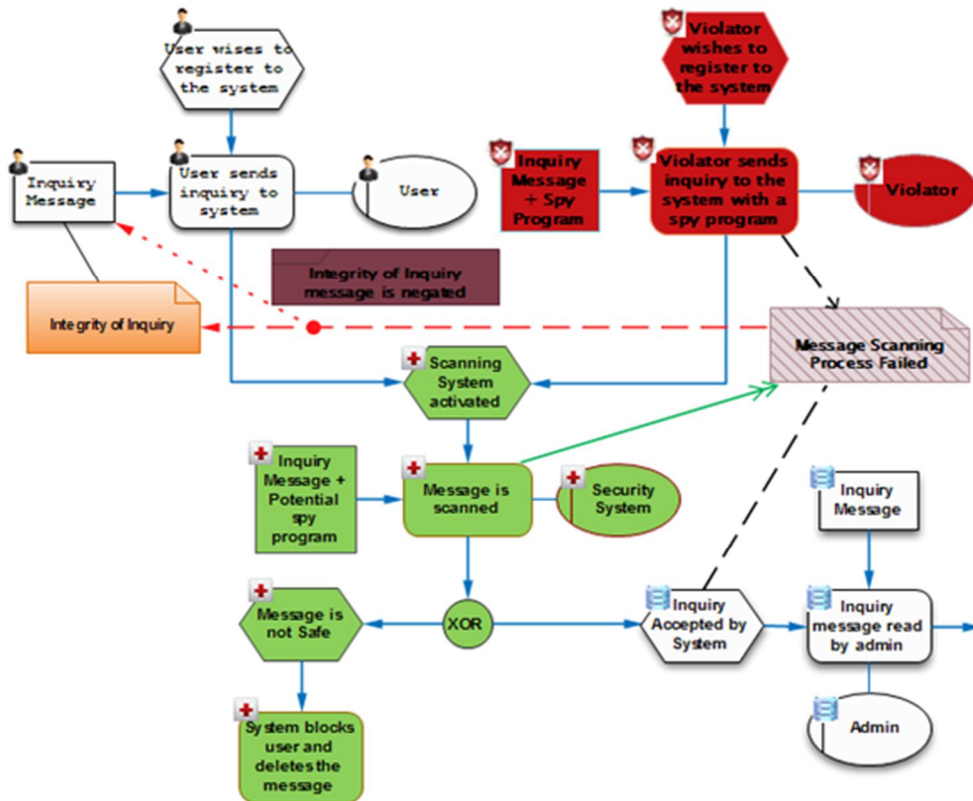


Figure 9.1 – Security-Oriented EPC diagram of online registration (message handling) of the Internet Store.

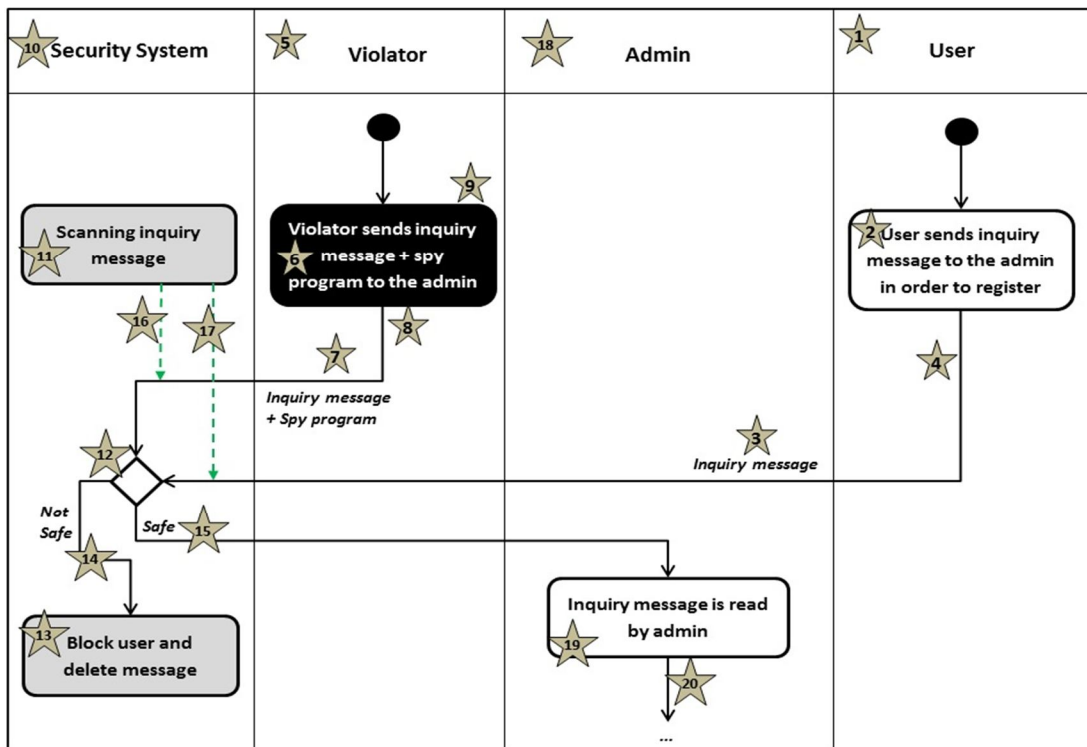


Figure 9.2 – Mal-activity Diagram of online registration (message handling) of the Internet Store, including identified stars for transformed constructs.

9.5 Data Analysis

Participant 1:

In Figure 9.3 the solution of Participant 1 is illustrated. Except 17th and 20th transformations, all transformations are used perfectly. Participant 1 forgot to use the mitigation link to the inquiry message coming from user side. The security system does not know if the message includes spy program or not, in this case scanning function mitigates all potential incoming messages. Also, Participant 1 forgot to use the last control flow after admin reads the message, however it is not very necessary detail. In conclusion, 18 over 20 of transformations are correct and success percentage of Participant 1 is 90%.

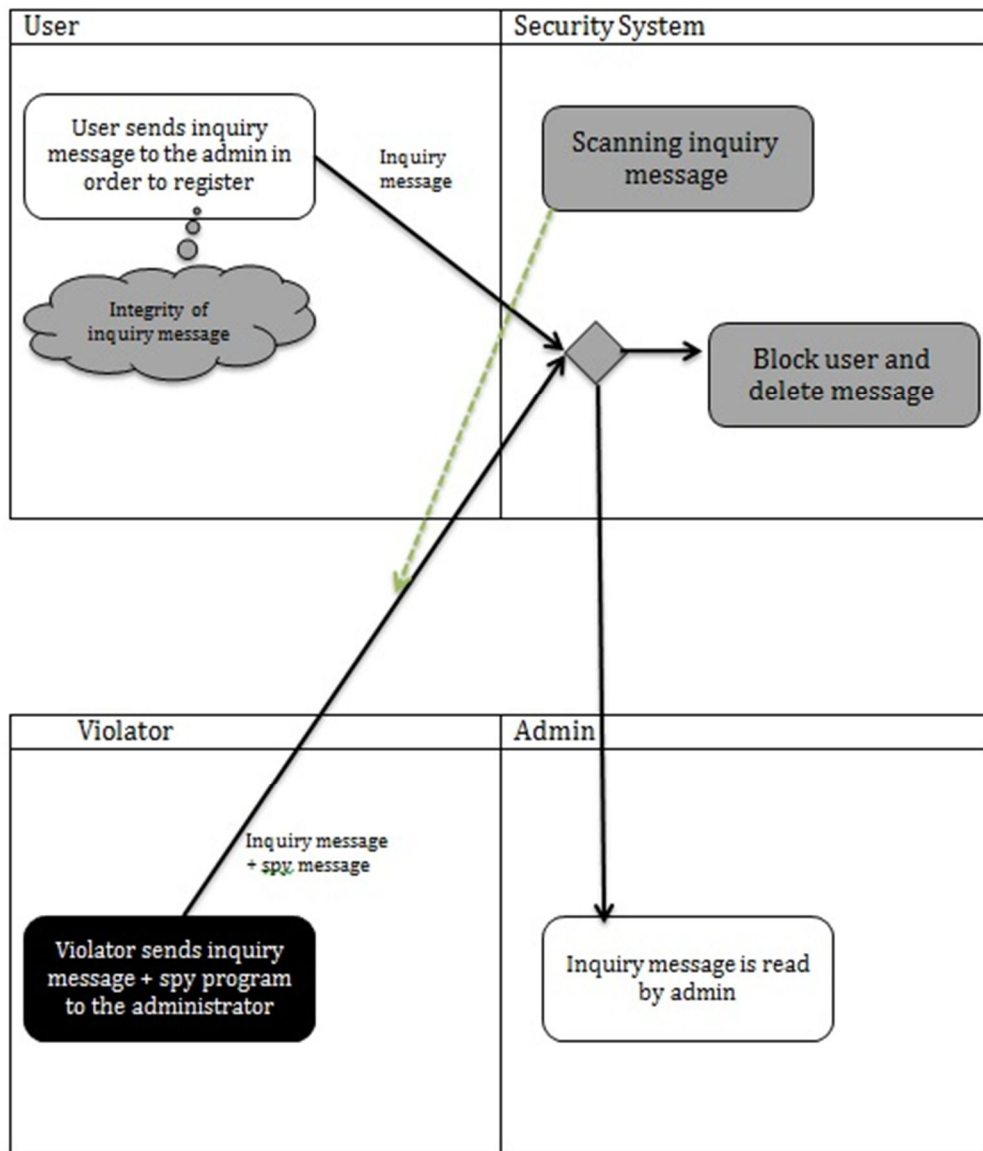


Figure 9.3 – Solution of Participant 1.

Participant 2:

In Figure 9.4 the solution of Participant 2 is illustrated. Participant 2 also managed to transform all constructs except 17th and 20th transformations. Participant 2 forgot to use the mitigation link to the inquiry message coming from user side and also forgot to use the last control flow after admin reads the message, however it is not very necessary absence. In conclusion, 18 over 20 of transformations are correct and success percentage of Participant 2 is 90% as well.

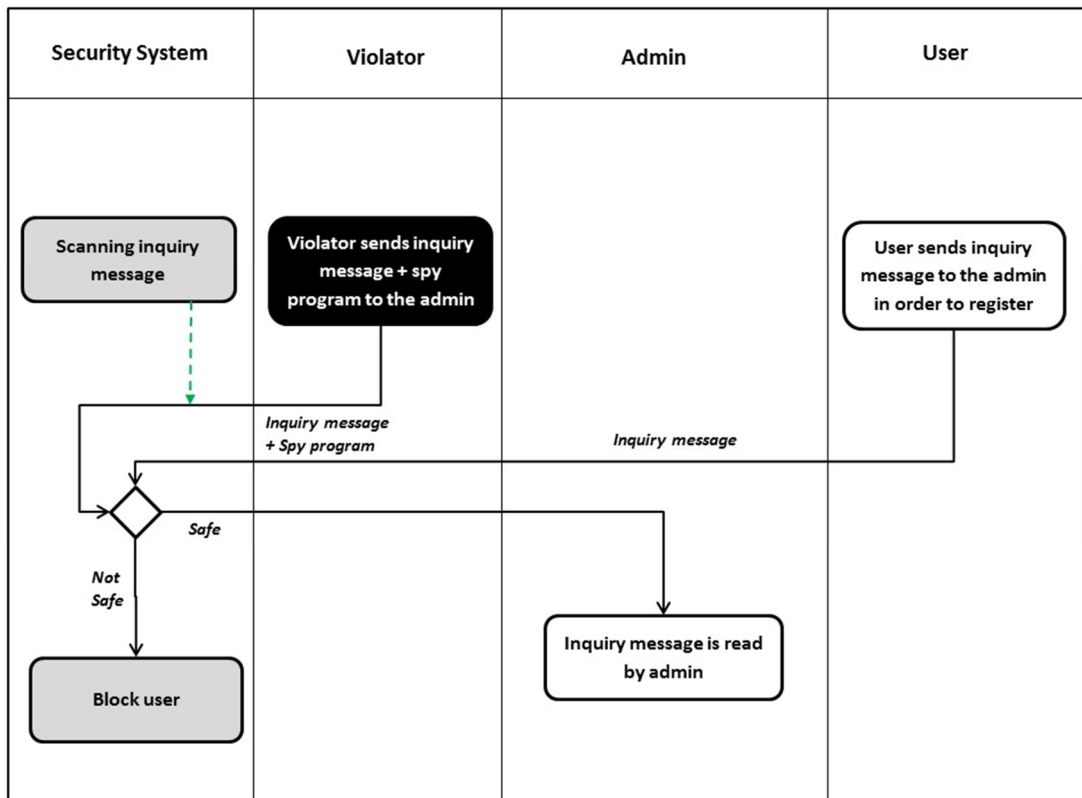


Figure 9.4 – Solution of Participant 2.

Participant 3:

In Figure 9.5 the solution of Participant 3 is illustrated. Participant 3 did not use 16th and 17th transformations which represent mitigation link. It seems he did not understand the mitigation process and how it works. Also, he used wrong color in 6th transformation, he should have used black color when he was representing the mal-activity. During the 7th transformation he should have ended the inquiry message control flow in decision gate, not in mitigation activity. Lastly, 3rd transformation should have ended in gate, not in admin function. In conclusion, correct transformations number of Participant 3 is 16,5 and his success percentage is 83%.

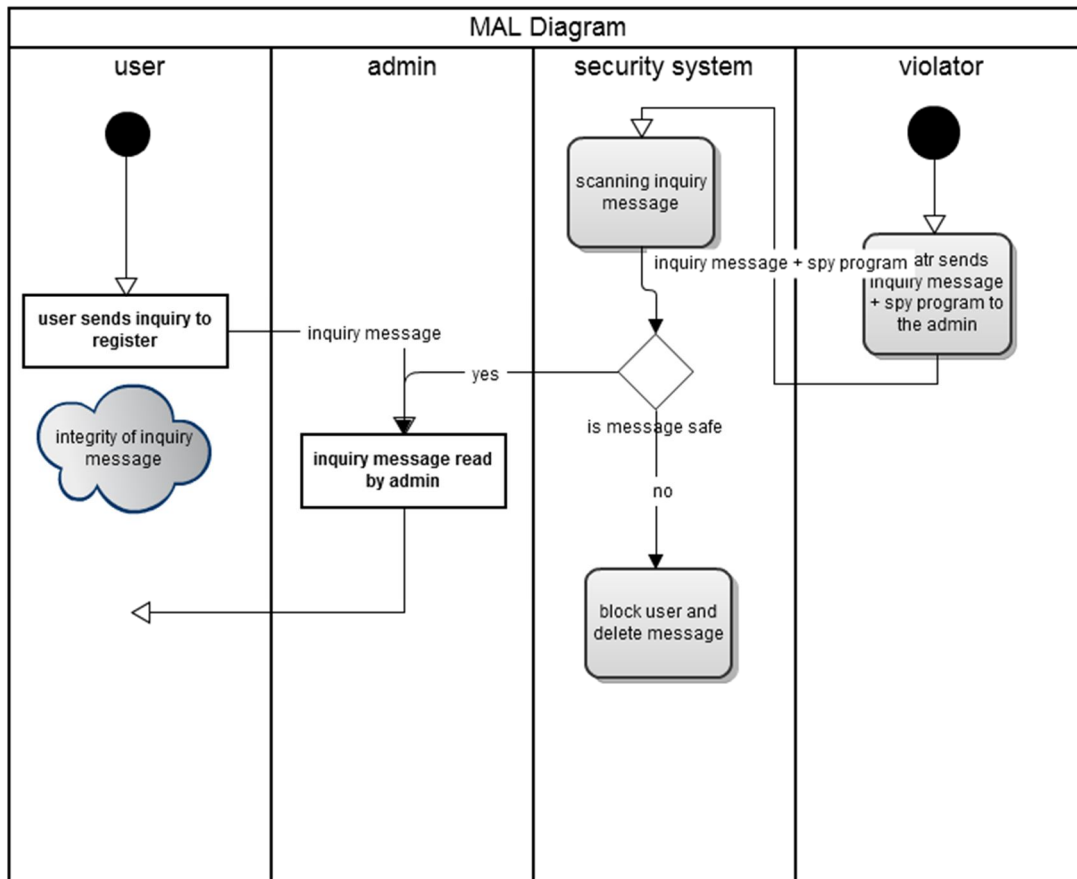


Figure 9.5 – Solution of Participant 3.

Participant 4:

In Figure 9.6 solution of Participant 4 is illustrated. Participant 4 did not use the transformation number 3, 4, 7, 8, 16 and 17. Transformation number 3 and 4 represents inquiry message which initializes at user activity and ends at security system decision gate, Participant 4 just indicated the communications through the swimlanes. The same issue happened in the inquiry message and spy program on 7th and 8th transformations. Also, Participant 4 skipped using mitigation links where he was supposed to transform at 16th and 17th transformations. He used wrong color during the 6th transformation (mal-activity was supposed to be black color) and wrong arrow type during the 15th transformation (control flow should not be a dashed arrow). Correct transformation number of Participant 4 is 13 and his success percentage is 65%.

Participant 5:

In Figure 9.7 solution of Participant 5 is illustrated. 3rd transformation ends in wrong place and it is wrong messaging type as well. The inquiry message coming from user side should not include spy program, and also inquiry message control flow should have ended at security system decision gate, not at violator function. Participant 5 have used wrong color during the 6th transformation, (mal-activity should have been in black color). Also, during the 7th transformation message on the control flow coming from the violator is missing. Participant 5 did not use transformation number 16 and 20 as well. Mitigation link is used only once to the inquiry message coming from user, whereas it should have been used for

the control flow coming from violator as well. Correct transformation number of Participant 5 is 16 and his success percentage is 80%.

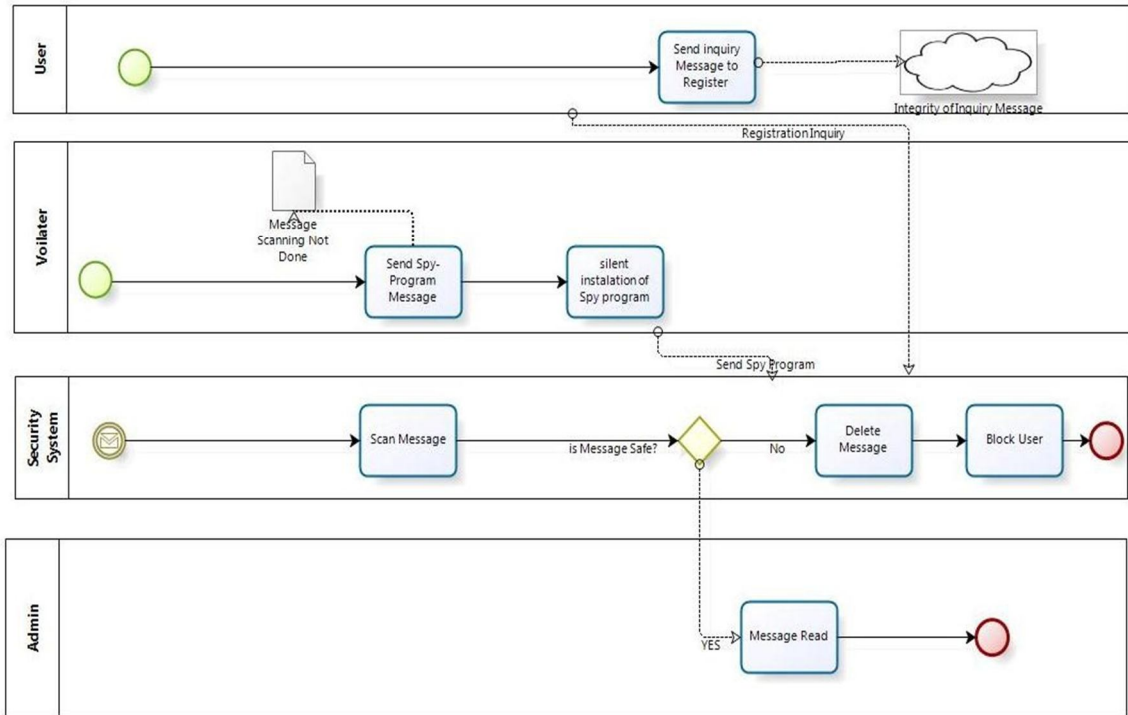


Figure 9.6 – Solution of Participant 4.

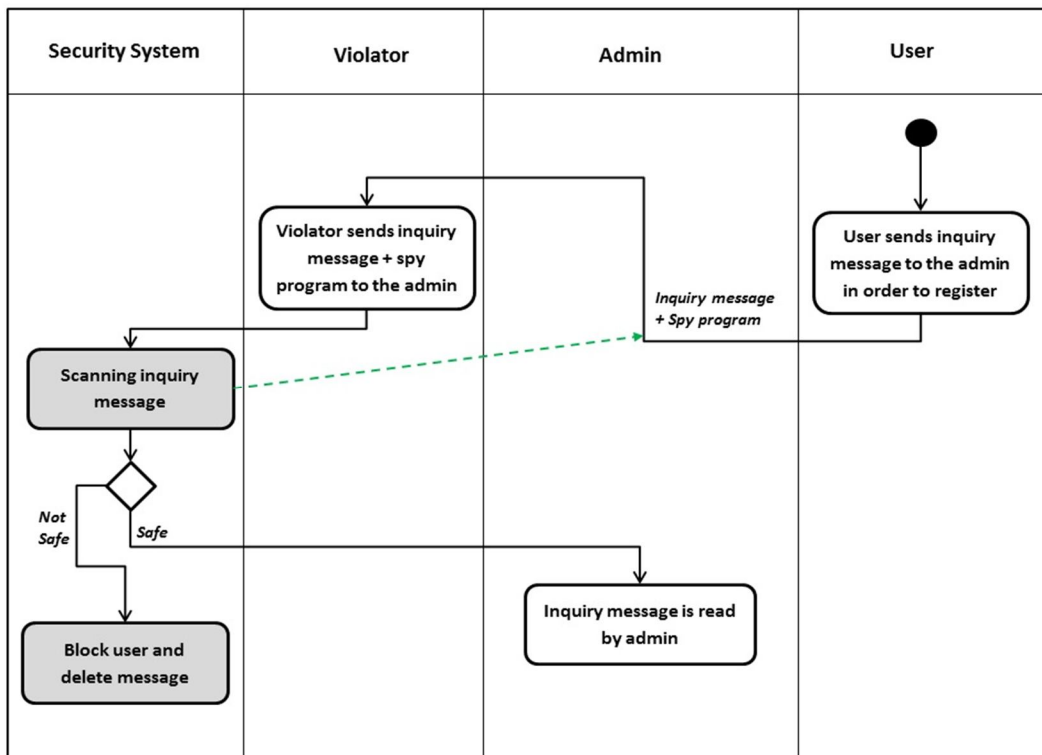


Figure 9.7 – Solution of Participant 5.

Participant 6:

In Figure 9.8 the solution of Participant 6 is illustrated. Participant 6 did not use the transformation number 16, 17 and 20. 16th and 17th transformations represent mitigation link, and 20th transformation represents control flow, however 20th transformation is not very critical and just tests the attention of the participant during the experiment. In conclusion, correct transformation number of Participant 6 is 17 and his success percentage is 85%.

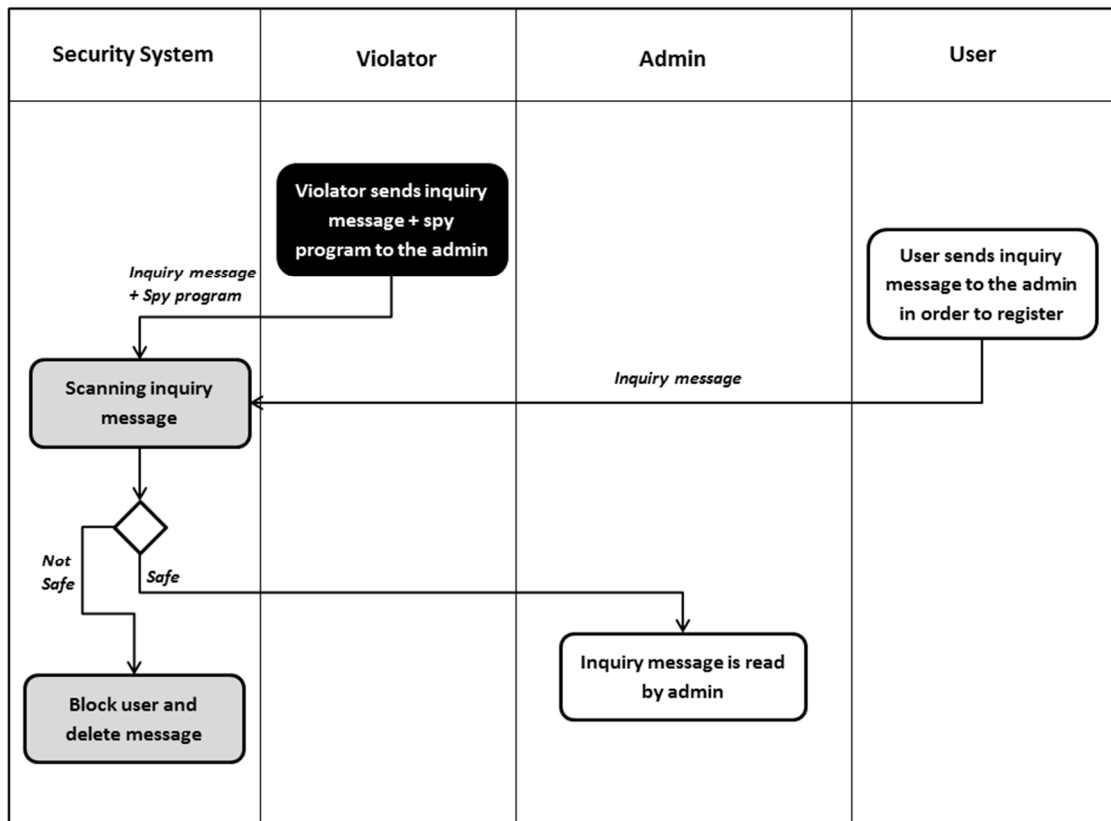


Figure 9.8 – Solution of Participant 6.

Participant 7:

In Figure 9.9 the solution of Participant 7 is illustrated. In his solution, 3rd and 7th transformations end in wrong place, control flows coming from user and violator should have ended at security system decision gate, not at security system mitigation activity. Also, wrong positioning is used during the 16th transformation, mitigation link initializes correctly at mitigation activity but ends incorrectly at vulnerability which is unnecessary construct in this model. 16th transformation should have ended at inquiry message and spy program control flow coming from violator. Also 17th transformation is missing, the other mitigation link which is supposed to link the inquiry message coming from user. Lastly, wrong color is used during the 19th transformation, it should have the same color with transformation number 2, which is white. In conclusion, correct transformation number of Participant 7 is 17 and his success percentage is 85%.

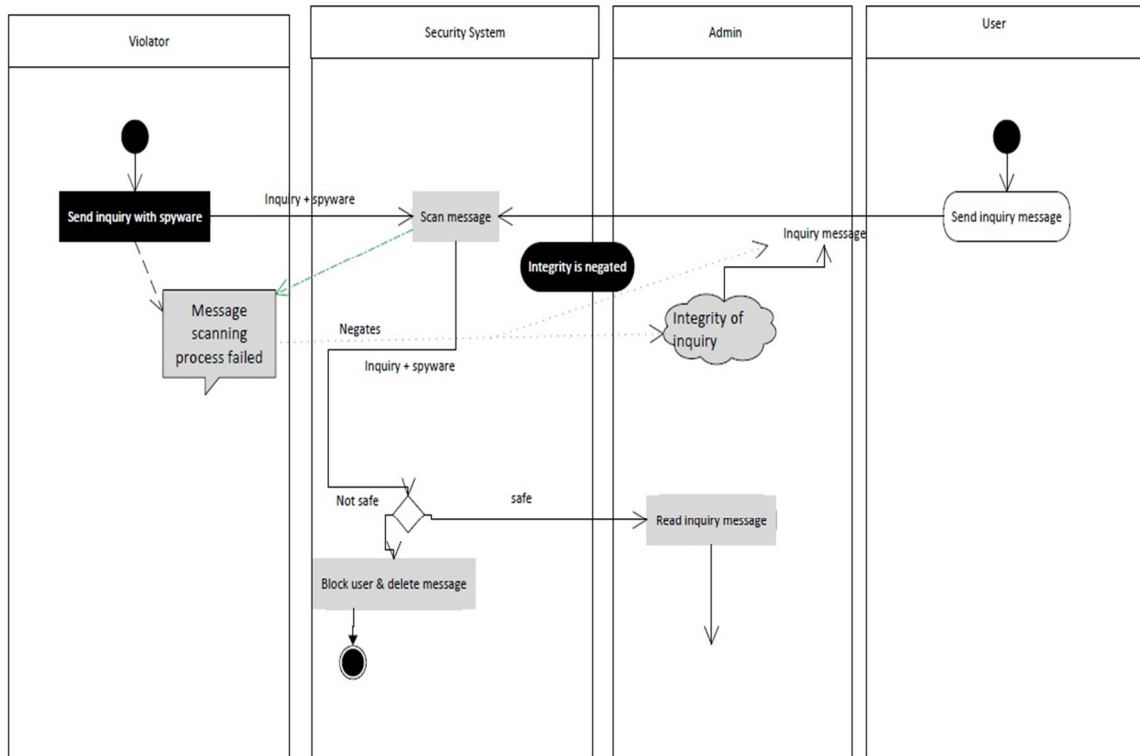


Figure 9.9 – Solution of Participant 7.

Participant 8:

In Figure 9.10 solution of Participant 8 is illustrated. Participant 8 has used wrong type of arrow during the 3rd and 4th transformations, control flow should not be a dashed arrow. Also he used wrong color (white, instead of black) during the 6th transformation when he was indicating mal-activity. During the 7th and 8th transformations, wrong positioning and also wrong arrow type (dashed) are used. Therefore, during the 15th transformation wrong arrow type (dashed) is used. So far Participant 8 is the only one who has used both mitigation links (16th and 17th transformations), but he used wrong color (black instead of green). Correct transformation number of Participant 8 is 16 and his success percentage is 80%.

Participant 9:

In Figure 9.11 the solution of Participant 9 is illustrated. Participant 9 made a mistake during the 3rd and 7th transformations, they both should have ended at security system decision gate, not at security system mitigation activity. Also, 16th transformation has wrong initialization, it should have initialized from the scanning mitigation activity. Participant 9 did not use 17th transformation which is the other mitigation link. Also 20th transformation is missing. In conclusion, correct transformation number of Participant 9 is 16,5 and his success percentage is 83%.

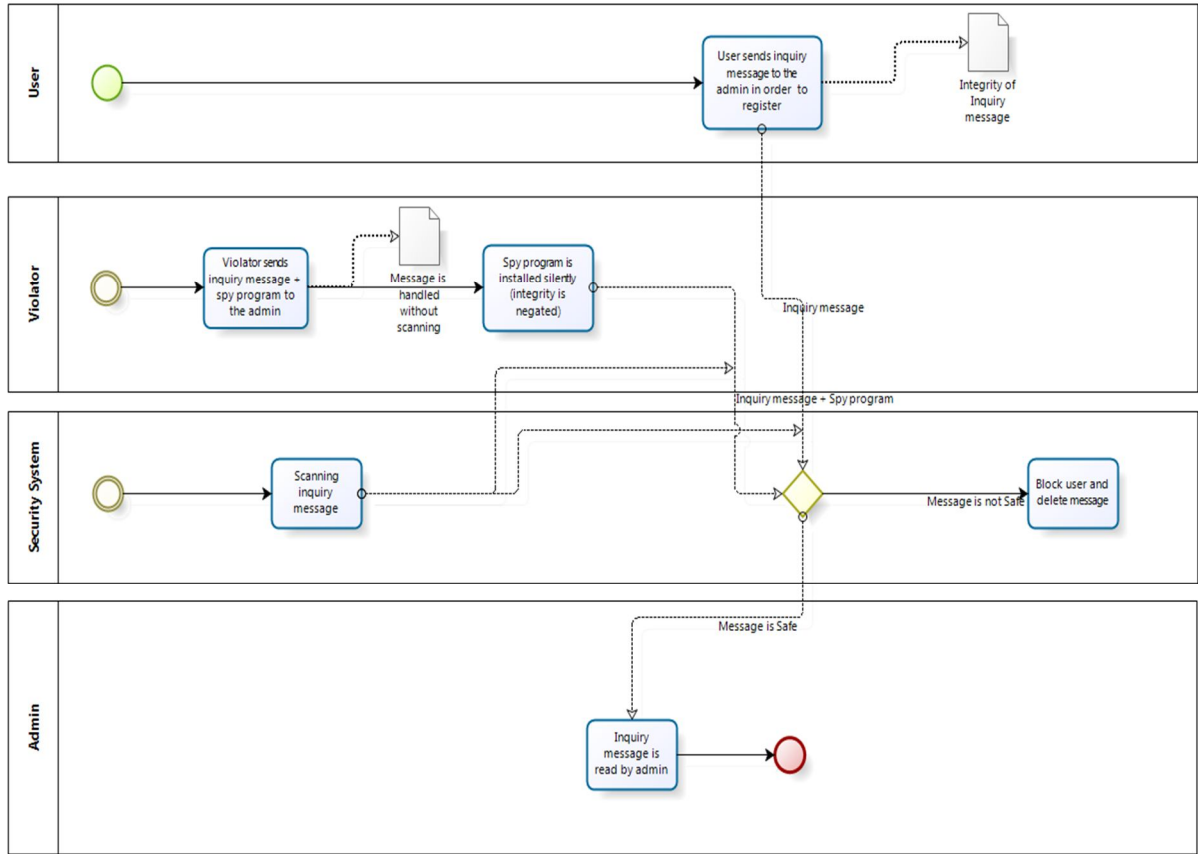


Figure 9.10 – Solution of Participant 8.

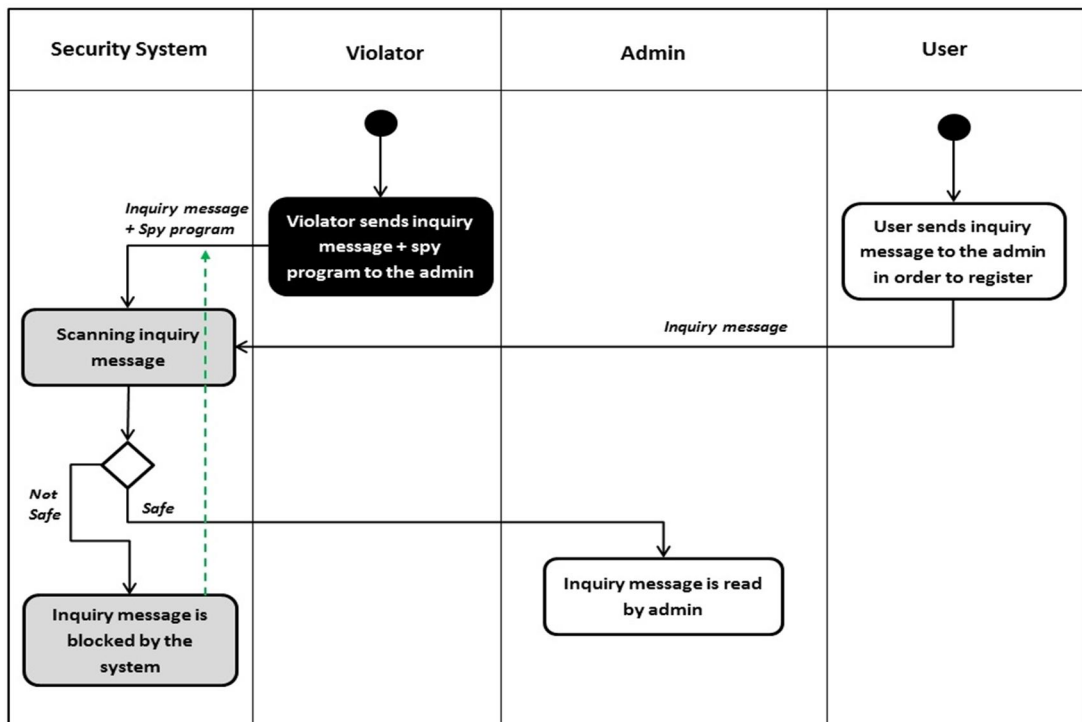


Figure 9.11 – Solution of Participant 9.

Participant 10:

In Figure 9.12 the solution of Participant 10 is illustrated. In Participant 10's solution, 3rd and 7th transformations (control flows coming from user and violator) end at wrong place, as many other participants made the same mistake. 16th transformation (mitigation link to control flow of violator) is missing, transformation number 17 (mitigation link to control flow of user) is used with wrong positioning, it should have ended at control flow of user. Lastly, 20th transformation is missing. Correct transformation number of Participant 10 is 16,5 and his success percentage is 83%.

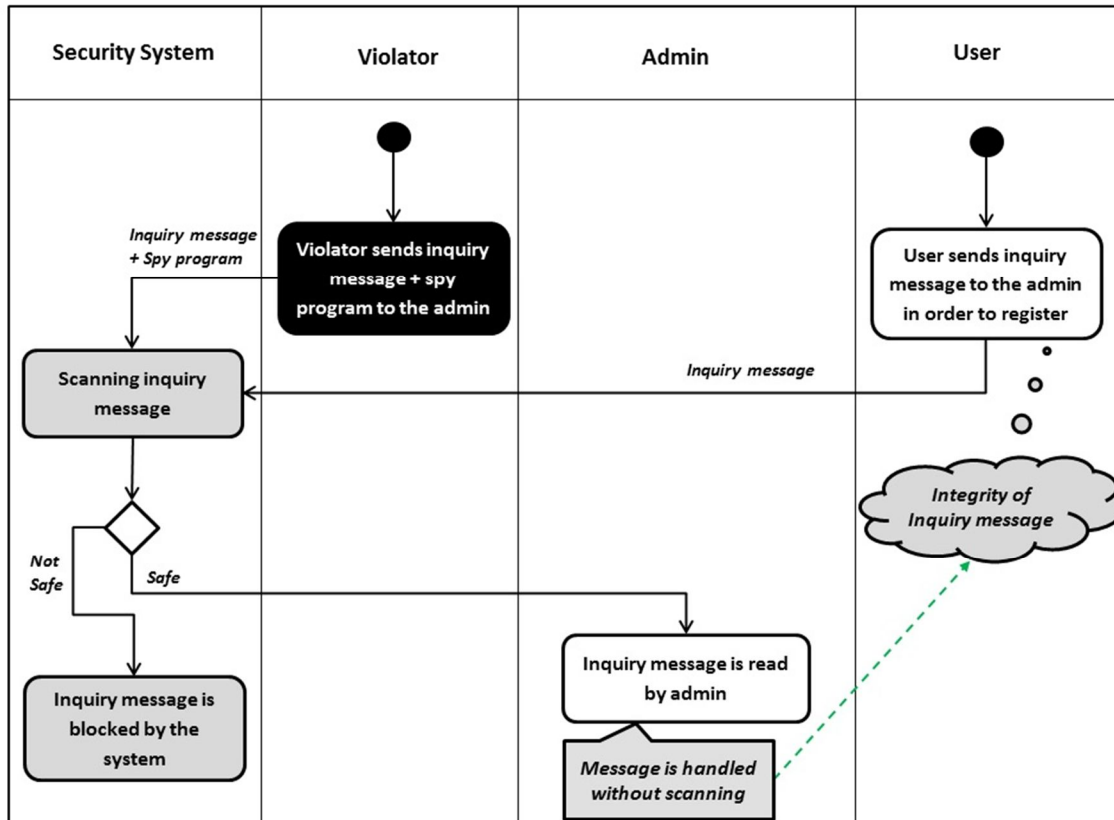


Figure 9.12 – Solution of Participant 10.

9.6 Interpretation of Results

If we calculate the average success rate of the case study participants:

$$\text{Total Success Percentage} = (90\% + 90\% + 83\% + 65\% + 80\% + 85\% + 85\% + 80\% + 83\% + 83\%) / 10$$

$$\text{Total Success Percentage} = 83\%$$

Total success percentage is based on twenty transformations done in the case study model, however in total fourteen of eighteen transformation rules are used in the solution. The transformation rules which are not used in the solution are; TR2 (Security Criterion), TR6 (Vulnerability), TR7 (Impact) and TR9 (Negates). The main reasons why these transformations are not used is the difference between EPC and MAD abstract syntaxes in

the solution phase. In EPC, we can show the process in more sophisticated way, however, in MAD, the process is considered in more simple way. For example, security criterion (TR2) is not used because after the risk treatment process spy program is not installed which means there is no negation (TR9) to the security criterion, and no vulnerability (TR6) any more in the system. As a result there is no impact (TR7) too. In Security-Oriented EPC, in Figure 9.1, we showed these constructs because we have done the alignment with ISSRM and it worthed to illustrate all the constructs together in a single model; Risk-treatment model. However we do not need to show all the constructs in one model during MAD design since we did not align MAD and ISSRM with our running example and that is why we ignore unnecessary constructs during the design of corresponding concept level (e.g. during the risk-treatment analysis we ignore the risk or asset constructs which are not part of the process flow anymore).

Besides, in this manner, it is possible to point “inadequate preoperational explication of constructs” threat since the participants had lack of information about the background of the whole research that is why they just tried to transform each construct. Another threat during the validation was “interaction of history and treatment”, which means that our experiment is conducted on a time which affected the results. Some participants were complaining that they could not spend enough time to create the solution due to their personal occupation.

9.7 Summary

All in all, it is possible to say that the success percentage of our experiment based on transformation rules defined in Chapter 8 is approximately 80% which is a satisfying result.

Table 9.1 – Statistics of the participants’ results based on transformations.

Participant	Missing transformation	Incorrectly used transformation
Participant 1	<i>17, 20</i>	-
Participant 2	<i>17, 20</i>	-
Participant 3	<i>16, 17</i>	<i>3, 6, 7</i>
Participant 4	<i>3, 4, 7, 8, 16, 17</i>	<i>6, 15</i>
Participant 5	<i>7, 16, 20</i>	<i>3, 6</i>
Participant 6	<i>16, 17, 20</i>	-
Participant 7	<i>17</i>	<i>3, 7, 16, 19</i>
Participant 8	-	<i>3, 4, 6, 7, 8, 15, 16, 17</i>
Participant 9	<i>17, 20</i>	<i>3, 7, 16</i>
Participant 10	<i>16, 20</i>	<i>3, 7, 17</i>

Now let’s focus on fourteen transformation rules which are included in the case study model. All the participants have made mistakes, some are similar and some are different. Table 9.1 indicates the statistics of the missing and incorrectly used transformations.

According to Table 9.1, occurrences of the missing transformations can be listed in descending order as; 17 (TR14 – 7 times), 20 (TR18 – 6 times), 16 (TR14 – 5 times), 7 (TR11 – 2 times), 3 (TR4), 4 (TR5) and 8 (TR12 – 1 times). Basically, participants mainly

forgot to use transformation rule number 14 and 18. We can underestimate the absence of 18th transformation rule since it just indicates a usual control flow at the end of our scenario. However, 14th transformation rule is very important because it indicates the mitigation link during the risk treatment process. We assume this absence happened due to the difference of the abstract syntax of the modeling languages in risk treatment level, however, in definition of TR14 in Chapter 9 the differences are defined. In conclusion, ten participants have clearly used TR1, TR3, TR8, TR10, TR13, TR15, TR16 and TR17, which gives us the success ratio of 8/14. However, single transformation rule based analysis is not very efficient since we have ten different participants and if each participant skips only one but different transformation rules than each other, our success ratio would be 4:14, whereas the total success percentage would be 95%. That is why we consider the whole model during the decision of success percentage.

Occurrences of the incorrectly used transformations can be listed in descending order as; 3 (TR4 – 6 times), 7 (TR11 – 5 times), 6 (TR10 – 4 times), 16 (TR14 – 3 times), 15 (TR18) and 17 (TR14 – 2 times), 4 (TR5), 8 (TR12) and 19 (TR3 – 1 time). To make conclusion about incorrectly usage of transformation rules is more complex than to make conclusion about missing transformation rules, because incorrectly does not mean a specific mistake, various types of mistakes can be done by the participants such as using wrong color, using wrong initialize or end and so on. In particular, we can assume that participants made wrong linking during the 4th and 11th transformation rules since it is perfectly normal to use the control flow incorrectly during the process flow (e.g. ending at activity instead of decision gate or not writing the message on the control flow). 10th transformation rule mistake is obviously clear since many participants forgot to use black color when using malicious activity. And lastly, some mitigation links are used incorrectly during the usage of 14th transformation rule.

Chapter 10. CONCLUSION

We divide conclusion chapter into four parts; short summarize of the thesis, limitations of the thesis, conclusions of the thesis and future work. Conclusions section states the contributions between recalled chapter and the research question of the thesis.

10.1 Summary

In this thesis, after required knowledge is provided in state of art chapters, the alignment between EPC and ISSRM is performed by using security risk modeling methods. The background information about business process modeling languages, security modeling languages and ISSRM are given in order to let reader understand and have the ability to compare the languages and concepts between each other in security requirements level. During the alignment process, in Chapter 5, EPC language is analysed in six different steps; Context and Asset Identification, Determination of Security Objectives, Risk Analysis and Assessment, Risk Treatment, Security Requirements Definition and Control Selection and Implementation. Later, extensions to EPC are done and the extended language is called as Security-Oriented EPC. To relate the extensions of EPC with real life, in Chapter 7, a running example with metric values is illustrated by measuring security risks of Security-Oriented EPC. In the end of the measurement analysis, we have obtained approximate ROSI of Security-Oriented EPC. Later, in order to test and emphasize the availability and adaptability of Security-Oriented EPC, we have defined set of transformation rules from Security-Oriented EPC to Mal-Activity Diagrams in Chapter 8. In the end, in Chapter 9, we have validated the transformation rules by a descriptive case study.

10.2 Limitations

Although the research in this thesis has reached its aims, there were some unavoidable limitations. First, because of the time limit, limited (ten participants in total) participants have involved during the validation of thesis. Therefore, to generalize the results for larger groups, validation case study should have involved more participants in at different levels. Also, another limitation was the lack of various resources on Information System Security Risk Management method. Almost all the resources which are used in this thesis as a reference are created by same authors. Another limitation is that we have used a single running example in order to illustrate different modeling language examples parallel to each other.

10.3 Conclusions

The main conclusion of giving background information in this thesis is instructing the reader for the further analysis and understanding the research question and the theory better. In Chapter 2, after analyzing business process modeling languages, we conclude that EPC is not effective to elicit security concerns and we decided to align it with ISSRM. Conclusion of Chapter 3 is that MAD is the most suitable language to use during the transformation and validation of the extended EPC since the alignment between MAD and ISSRM showed us that MAD is an effective security-oriented modeling language. In Chapter 4, we conclude that the risk management process with ISSRM is the way to align and extend EPC.

The main contribution of the alignment process is making EPC more secured against potential risks and vulnerable attacks. Although EPC is serving its primary purpose at the high-degree, it is not helpful to elicit security concerns when developing information enterprise systems. In the end of the alignment process with a running example, we have obtained an alignment table between EPC and ISSRM. By obtaining alignment table, we had a chance to see the security needs of the EPC language in corresponding levels; Asset, Risk, or Risk Treatment. This conclusion helped us during the analysis of the extension of the EPC language. Besides, conclusion of the alignment process is the answer of our research question "*How EPC could be extended to support security risk management?*". Alignment of EPC led us to extend language in high and low construct levels according to security needs and requirements.

In Chapter 6, the extensions on EPC language are done in two different ways; High Level with Process Paths and Low Level with other language constructs which can be contained by Process Path during the modeling. Such method is chosen in order to emphasize the sophistication of EPC and also to reduce the complexity optionally (complexity reduction might be helpful during the future work). As a conclusion of extensions, we have developed syntactic, semantic and methodological extensions to EPC that would support modeling security risks and their countermeasures. Besides, extensions will help business analysts to see the business model in different levels of security, this answers the second research question "*What is the benefit of such extension?*". Another conclusion of the extension process is that we have obtained guidelines to use Security-Oriented EPC. This conclusion is answering both research questions as well.

The conclusion of the ROSI analysis is answering the question of the research questions in such a way that a business analyst can understand the purpose better by estimating the cost of using Security-Oriented EPC. Also, we can conclude that extended EPC helps to measure the risks, because, by the ROSI calculation we have analysed the cost with a security requirement definition and the system got return on investment the percentage of 167.

During the state of art when we analysed the MAD with ISSRM, we have pointed that MAD is a suitable language to measure security needs and requirements of a model. This consequence helped us to define many transformation rules from Security-Oriented EPC to MAD easily. Regarding our research question "*What is the benefit of such extension?*", we can say that one of the benefits is obtaining a lot of transformation rules and a common alignment table between ISSRM, Security-Oriented EPC and MAD.

Conclusion of the case study experiment contributes the previous work and research question by proving the effectiveness, validity and coordination of the Security-Oriented EPC language.

10.4 Future Work

Regarding future work, we can discuss what the next steps could be and if we think that certain paths seem to be more promising than others or they do not.

The next step could be the implementation of the Security-Oriented EPC language in a modeling tool and let an organisation to use it during a business process modeling of a sophisticated problem which contains security needs. By this method we could see the efficiency of our work in real life in a more realistic way. Besides, another step could be analyzing the EPC and ISSRM alignment with a less or more complicated running example.

In this thesis, each analysis is planned and contribution is done several times from different perspectives. As a result, we can't identify certain paths which seem to be more promising than others.

RESÜMEE

Event-driven Process Chaini laiendused ja rakendus Infosüsteemi Turberiskihalduseks

Turvatehnika konstrueerimine on üks suuremaid murekohti süsteemi arenduses ja sellele tuleks tähelepanu pöörata kogu arendusprotsessi jooksul. Turvaliseks modelleerimiseks on mitmeid erinevaid keeli, mis aitavad hallata turvariske juba nõuete staadiumis. Käesolevas töös keskendutakse esmalt Event-driven Process Chain (EPC)-le, mida kasutatakse äriprotsesside modelleerimisel. Täpsemalt öeldes uuritakse, kuidas antud keel toetab infosüsteemi turberiskihaldust (ISSRM). Uurimuse eesmärk on välja selgitada EPC jaoks vajalikud turbenõuded. Nende tulemusena saame vastavustabeli EPC konstruktsioonide ja ISSRM domeeni mudeli kontseptide vahel. Järgnevalt laiendame EPC keelt ja selle konstruktsioone EPC ja ISSRM vastavustabeli seostega. Tekkinud laiendatud keelt kutsume "Security-Oriented EPC". Laiendatud modelleerimiskeel sisaldab uut konstruktsioonide kogumikku, mis viitab ISSRM kontseptidele. Olles selgitanud turvanõuete olulisust varajases arendusstaadiumis, esitleme töötamise suunised, et viia ellu tõlked Security-Oriented EPC ja Mal-Activity Diagrams (MAD) vahel. Meie ettepanek põhineb EPC keele süstemaatiliste ja maandatud laiendustel ja selle vastastikusest sõltuvusest ISSRM domeeni mudelisse. Vastavuses olevad tulemused aitavad ärianalüütikutel mõista, kuidas modelleerida turvariske süsteemi nõuete ja disainimise staadiumites. Lisaks annavad töötamise tulemused võimaluse koostööks erinevate modelleerimiskeelte vahel, mida analüüsitakse kasutades sama kontseptuaalset raamistikku.

BIBLIOGRAPY

1. Giorgini P., Mouratidis H. - *Secure Tropos: A Security-Oriented Extension of the Tropos Methodology*, *International Journal of Software Engineering and Knowledge Engineering*, World Scientific 2007
2. Lamsweerde A. v. - *Elaborating Security Requirements by Construction of Intentional Anti-Models*, Belgium, 2004
3. Opdahl A. L., Sindre G. - *Eliciting security requirements with misuse cases*, Norway, 2004
4. Alexander I. - *Misuse Cases, Use Cases with Hostile Intent*, *IEEE Software*, 2003
5. Sindre G. - *Mal-Activity Diagrams for Capturing Attacks on Business Processes*, *Requirements Engineering: Foundation for Software Quality*, 2007
6. Gooch T. - *Unified Modeling Language (UML) Tutorial*, Kennesaw State University, 2000
7. Muscholl A. - *Petri Nets*, Université Bordeaux, 2006
8. Bradford L., Dumas M. - *Getting Started with YAWL*, Technical Paper, YAWL Foundation, 2007
9. White S. A. - *Introduction to BPMN*, IBM Corporation, 2004
10. Article on PNMSOFT - *Business Process Modeling Notation Tutorial*, http://www.pnmssoft.com/bpm_workflow_tutorial.aspx, 2012
11. Dubois E., Heymans P., Mayer N., Matulevičius R. - *A Systematic Approach to Define the Domain of Information System Security Risk Management*, Nurcan, S., Salinesi C., Souveyet C., Ralyte, J. (eds.) *Intentional Perspectives on Information Systems Engineering* pp. 289-306, 2010
12. Dubois E., Genon N., Heymans P., Mouratidis H., Mayer N., Matulevičius R. - *Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Development*, *Proceedings 20. International Conference on Advanced Information Systems Engineering*, Montpellier, France, *Lecture Notes in Computer Science*, 2008
13. Dubois E., Heymans P., Mayer N., Matulevičius R. - *A Framework for Analysing the Interoperability of Security Modeling Languages with Risk Management Methods*, Section 7, Not-published draft, 2009
14. Heymans P., Mayer N., Matulevičius R. - *Alignment of Misuse Cases with Security Risk Management*, *Proceedings of the SREIS*, 2008
15. Medina E., Piattini M., Rodriguez A. - *A BPMN Extension for the Modeling of Security Requirements in Business Processes*, 2007
16. Sindre G. - *Mal-Activity Diagrams for Capturing Attacks on Business Processes*, *Requirements Engineering: Foundation for Software Quality*, 2007
17. Jürjens J. - *Model-based Security Engineering with UML*, Competence Center for IT Security - Software & Systems Engineering TU Munich, 2004
18. YAWL User Manual V2.2 Version - www.yawlfoundation.org, 2012
19. Den Braber F., Hogganvik I., Lund M. S., Stølen K., Vraalsen F. - *Model-based security analysis in seven steps — a guided tour to the CORAS method*, *BT Technology Journal*, 2007

20. Asnar Y., Giorgini P., Mylopoulos J. - *Goal-driven risk assessment in requirements engineering*, Springer-Verlag London Limited, 2010
21. Firesmith D. - *Common Concepts Underlying Safety, Security, and Survivability Engineering*, Technical Note CMU/SEI-2003-TN-03, 2003
22. Dubois E., Heymans P., Mayer N., Matulevičius R. - *A Systematic Approach to Define the Domain of Information System Security Risk Management*, *Intentional Perspectives on Information Systems Engineering*, 2010
23. Mayer N. - *Model-based Management of Information System Security Risk*, PhD theses, University of Namur, HenriTudor public research centre, 2009
24. Seel C., Vanderhaeghen D. - *Meta-Model based Extensions of the EPC for Inter-Organisational Process Modeling*, *Institute for Information Systems (IWi) at the German Research Center for Artificial Intelligence (DFKI)*, 2005
25. Article at University of Wien, ERP WebTrainer web page – http://www.wu.ac.at/erp/webtrainer/epc_webtrainer, 2012
26. ISO/IEC 27005: Information Technology, Security Techniques – *Information Security Risk Management*, 2008
27. Locher C. - *Methodologies for evaluating information security investments*, 2005
28. Chowdhury M., Karpati P., Matulevičius R., Sindre G. – *Aligning Mal-activity Diagrams and Security Risk Management for Security Requirements Definitions*, 2012
29. Zaidah Z. – *Case study as a research method*, Universiti Teknologi Malaysia, 2007