

UNIVERSITY OF TARTU
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
INSTITUTE OF COMPUTER SCIENCE

Olga Altuhhova

**An Extension of Business Process Model and Notation
for Security Risk Management**

Master's thesis (30 ECTS)

Supervisor: Raimundas Matulevičius

Author: "....." June 2013

Supervisor: "....." June 2013

Allow to defence

Professor: "....." June 2013

TARTU 2013

Abstract

Modern Information System (IS) development supports different techniques for business process modelling. Recently Business Process Model and Notation (BPMN) has become a standard that allows modellers to visualise organizational business processes. However, despite the fact that BPMN is a good approach to introduce and understand business processes, there is no opportunity to address security concerns while analysing the business needs. This is a problem, since both business processes and security concerns should be understood in parallel to support a development of the secure systems. In current thesis we introduce the extensions for BPMN 2.0 regarding security aspects. The following proposal is based on alignment of the modelling notation with IS security risk management (ISSRM). We apply a structured approach to understand major aspects of BPMN and propose extensions for security risk management based on the BPMN alignment to the ISSRM concepts. We demonstrate the use of extensions, illustrating how the extended BPMN could express assets, risks and risk treatment on few running examples related to the Internet store assets' confidentiality, integrity and availability. We believe that our proposal would allow system analysts to understand how to develop security requirements to secure important assets defined through business processes.

We also attempt to observe the following approach in the broader sense and we open a possibility for the business and security model interoperability and the model transformation between BPMN and another modelling approach also aligned to ISSRM, Secure Tropos.

Table of Contents

| | |
|--|-----------|
| Abstract..... | iii |
| Table of Contents..... | v |
| Table of Figures..... | vii |
| CHAPTER 1. Introduction | 1 |
| 1.2 Research questions and contribution | 1 |
| 1.3 Scope | 2 |
| 1.4 Structure..... | 2 |
| PART I. BACKGROUND | 5 |
| CHAPTER 2. Security Engineering..... | 7 |
| 2.1 Security Risk Management..... | 7 |
| 2.1.1 CORAS..... | 7 |
| 2.1.2 Automated Risk and Utility Management..... | 8 |
| 2.1.3 Goal-Risk driven assessment | 9 |
| 2.1.4 Information System Security Risk Management | 10 |
| 2.2 Comparison of Security Risk Management Methodologies | 11 |
| 2.3 Why ISSRM?..... | 12 |
| 2.4 Security risk-oriented modelling languages | 13 |
| 2.4.1 Secure Tropos | 13 |
| CHAPTER 3. Business Process Modelling | 17 |
| 3.1 Approaches to do BPM..... | 17 |
| 3.1.1 UML Activity Diagrams..... | 17 |
| 3.1.2 Yet Another Workflow Language | 18 |
| 3.1.3 Event-driven Process Chain | 20 |
| 3.1.4 Business Process Model and Notation..... | 21 |
| 3.2 Comparison of the Languages | 22 |
| 3.3 Why BPMN? | 24 |
| CHAPTER 4. BPMN and Security..... | 25 |
| 4.1 Previous work..... | 25 |
| 4.2 Related Studies | 26 |
| 4.3 Discussion..... | 28 |
| PART II. CONTRIBUTION | 29 |
| CHAPTER 5. Security Risk-oriented BPMN..... | 31 |
| 5.1 Research Method | 31 |
| 5.2 Extensions on Concrete Syntax | 31 |
| 5.3 BPMN Meta-Model..... | 35 |

| | |
|---|-----------|
| 5.4 Examples | 37 |
| 5.4.1 Confidentiality | 37 |
| 5.4.2 Availability | 39 |
| 5.4.3 Integrity | 42 |
| 5.5 Summary | 43 |
| Chapter 6. Transformation rules from BPMN to Secure Tropos | 45 |
| 6.1 Transformation rules | 45 |
| 6.1.1 Rules for asset definition | 45 |
| 6.1.2 Rules for risk identification | 46 |
| 6.1.3 Rules for risk treatment solution | 47 |
| 6.2 Transformation examples | 47 |
| 6.2.1 Availability | 47 |
| 6.2.2 Confidentiality | 53 |
| 6.2.3 Integrity | 59 |
| 6.2.4. Summary | 66 |
| | |
| PART III. VALIDATION | 67 |
| | |
| Chapter 7. Validation of BPMN extensions..... | 69 |
| 7.1 Description of the experiment | 69 |
| 7.2 Introducing the variables and assessment | 69 |
| 7.3 Data interpretation | 70 |
| 7.3 Threads to validity | 73 |
| 7.4 Data analysis results | 73 |
| | |
| Chapter 8. Transformation quality assessment | 75 |
| 8.1 Research method..... | 75 |
| 8.2 Evaluation goals | 75 |
| 8.3 Quality evaluation criteria | 75 |
| 8.4 Analysis of evaluation results..... | 76 |
| 8.5 Threads to validity | 77 |
| 8.6 Summary..... | 82 |
| | |
| Chapter 9. Conclusions..... | 83 |
| 9.1 Limitations..... | 83 |
| 9.2 Conclusions | 83 |
| 9.3 Future work..... | 84 |
| Rezümee | 85 |
| <i>Olga Altuhhova</i> | 85 |
| | |
| Magistritöö..... | 85 |
| References | 87 |
| | |
| Appendix A | 89 |

| | |
|--|-----|
| A1. Availability example transformation | 89 |
| A.1.1 Availability: Assets | 89 |
| A.1.2 Availability: Risk | 93 |
| A.1.3 Availability: Risk Treatment | 96 |
| A2. Confidentiality example transformation | 99 |
| A 2.1 Confidentiality: Assets..... | 99 |
| A 2.2 Confidentiality: Risk..... | 103 |
| A 2.3 Confidentiality: Risk Treatment..... | 106 |
| A3. Integrity example transformation..... | 109 |
| A 3.1 Integrity: Assets..... | 109 |
| A 3.2 Integrity: Risk | 113 |
| A 3.3 Integrity: Risk Treatment | 117 |
| Non-exclusive licence to reproduce thesis and make thesis public | 119 |

Table of Figures

| | |
|--|----|
| Figure 1. Eight steps of CORAS method guidance; adapted from (Lund et al., 2010) | 8 |
| Figure 2. The ISSRM Domain Model; adapted from (Dubois et al., 2010) | 11 |
| Figure 3. UML semantics | 18 |
| Figure 4. UML Activity Diagram: Order fulfilment process | 18 |
| Figure 5. YAWL: order fulfilment process | 19 |
| Figure 6. YAWL semantics | 19 |
| Figure 7. EPC semantics | 20 |
| Figure 8. EPC example: Order fulfilment | 20 |
| Figure 10. BPMN semantics | 21 |
| Figure 9. BPMN example: Order fulfilment process | 21 |
| Figure 11. BPMN Extended Abstract Syntax: Concept Classification | 36 |
| Figure 12. BPMN Extended Abstract Syntax: Relationships | 36 |
| Figure 13. BPMN Extended Abstract Syntax: Relationships | 37 |
| Figure 14. Message Handling Process: Asset Identification | 38 |
| Figure 15. Message Handling Process Including Security Risk Attack | 39 |
| Figure 16. Message Handling Process Including Security Requirements. | 40 |
| Figure 17. Service providing process; assets and security objectives | 41 |
| Figure 18. Service providing process; security risk | 41 |
| Figure 19. Service providing process; risk treatment | 42 |
| Figure 20. Process of logging in; assets and security objectives | 43 |
| Figure 21. Process of logging in; security risk definition | 43 |
| Figure 22. BPMN: Availability analysis - Message handling process assets | 48 |
| Figure 23. Secure Tropos: Availability analysis – assets | 48 |
| Figure 24. Secure Tropos: Availability analysis - assets (complete) | 49 |
| Figure 25. BPMN: Availability analysis - Denial of service risk | 50 |
| Figure 26. Secure Tropos: Availability analysis – Risk | 50 |
| Figure 27. Secure Tropos: Availability analysis – Risk (complete) | 51 |

| | |
|---|----|
| Figure 28. Secure Tropos: Availability analysis – Risk (Threat construct) | 51 |
| Figure 29. BPMN: Availability analysis - security requirements | 52 |
| Figure 30. Secure Tropos: Availability analysis – security requirements | 52 |
| Figure 31. Secure Tropos: Availability analysis – security requirements (complete) | 53 |
| Figure 32. BPMN: Confidentiality analysis – Handle request message | 53 |
| Figure 33. BPMN: Confidentiality analysis – User registration process | 54 |
| Figure 34. Secure Tropos: Confidentiality analysis – assets | 55 |
| Figure 35. Secure Tropos: Confidentiality analysis – assets (complete) | 55 |
| Figure 36. BPMN: Confidentiality analysis – Security Risk | 56 |
| Figure 37. Secure Tropos: Confidentiality analysis – risk model | 56 |
| Figure 38. Secure Tropos: Confidentiality analysis – risk model (complete) | 57 |
| Figure 39. Secure Tropos: Confidentiality analysis – risk model (Threat construct) | 57 |
| Figure 40. BPMN: Confidentiality analysis – Security Requirements | 58 |
| Figure 41. Secure Tropos: Confidentiality analysis – Security Requirements | 58 |
| Figure 42. Secure Tropos: Confidentiality analysis – Security Requirements (complete) | 59 |
| Figure 43. BPMN: Integrity analysis – asset identification (collapsed process) | 59 |
| Figure 44. BPMN: Integrity analysis – asset identification (expanded process) | 60 |
| Figure 45. Secure Tropos: Integrity analysis – asset identification | 60 |
| Figure 46. Secure Tropos: Integrity analysis – asset identification (complete) | 61 |
| Figure 47. BPMN: Integrity analysis – security risk | 61 |
| Figure 48. Secure Tropos: Integrity analysis – security risk | 62 |
| Figure 49. Secure Tropos: Integrity analysis – security risk (complete) | 62 |
| Figure 50. Secure Tropos: Integrity analysis – security risk (Threat construct) | 63 |
| Figure 51. Secure Tropos: Integrity analysis – risk treatment | 63 |
| Figure 52. Quality assessment process (adapted from Matulevičius et al, 2011) | 75 |
| Figure 53. Secure Tropos: Asset identification – confidentiality (created originally in Secure Tropos) | 79 |
| Figure 54. Secure Tropos: Risk identification – confidentiality (created originally in Secure Tropos) | 79 |
| Figure 55. Secure Tropos: Risk identification (collapsed to Impact concept) – confidentiality (created originally in Secure Tropos) | 80 |
| Figure 56. Secure Tropos: Risk Treatment – confidentiality (created originally in Secure Tropos) | 80 |
| Figure 57. Semantic correctness of models (number of elements used to express ISSRM concepts) | 81 |

CHAPTER 1. Introduction

The concept of *security* stands for ability of a product to protect information and data from the illegal access, performed by any unauthorized person or system in order to change, read or delete it. The research papers and experiences described in security knowledge domain show that the process of security integration into the information systems is not completely and appropriately understood. The analysis of security requirements is usually performed during system maintenance and implementation stages, meaning that the business process defined at early stages theoretically excludes consideration of security (Jürjens, 2005). That leads to the following problem; the security engineers receive little feedback about the needs for system security. And considering the idea that risks are difficult to calculate, this in turn increases the possibility of successful attack being reproduced in some other component of the security-critical systems.

The *business process* understanding and modelling in allows analysing needs and purposes of proposing services that we consider and use in practise in every kind of business. We believe that early consideration of security in business processes allows system and security analysts to define and investigate potential threats, study risks and its impacts, and also design and plan countermeasures, which will benefit secure system architecture and functionality in its future development. So the necessity of analyzing and designing new methodologies or investigating into the extensions of existing tools for modelling security in business processes remains actual and motivating.

1.2 Research questions and contribution

In current thesis we raise the importance of two research questions:

RQ1: What extensions should be implemented into the modelling language in order to express security concerns?

We investigate the opportunity to address security in organizational business processes using the Business Process Model and Notation (BPMN, version 2.0), a multi-vendor standard controlled by Object Management Group (White, 2004). In our previous work (Altuhhova et al., 2012) we performed an alignment between BPMN constructs and Information System Security Risk Management (ISSRM) (Dubois et al., 2010) domain concepts. Our purpose was to define the volume of security related concepts that can be covered by current version of language. This research resulted into the alignment table that shows the following: security concepts can be addressed by existing BPMN semantics and syntax only partially; another part of ISSRM domain remains uncovered. So in this study we continue our research and develop the set of extensions on semantics concrete and abstract syntax. We demonstrate how to apply extensions on running example of the Internet Store, including risk scenarios for three security criteria: *confidentiality*, *integrity* and *availability*. We then validate our study in the form of experiment and analyse the understandability of proposed extensions. We believe that the following proposal will allow system analysts to model and analyse the business processes and security concerns at early stages of IS development with a help of one language.

RQ2: How to transform security concerns between different modelling perspectives?

In frames of this research question we demonstrate our contribution in a broader sense. For instance, in some cases application of the BPMN security extensions would not be applicable because of the language nature to model organization's business processes, which means leading to the weak expressive power to address security concerns. This results into model transformation from BPMN to another modelling language. As an input we take an extended version of BPMN (that we also call *Risk-aware BPMN*), which is described at the first stage of our research, and a goal-oriented modelling approach Secure Tropos, introduced in (Matulevičius et al., 2012). Secure Tropos supports the development of IS on early and late requirements analysis stages that give us an opportunity to see the business case from completely different side, concentrating on major system *goals*, *actors* and *dependencies* between them. We develop a set of transformation rules, created addressing the ISSRM domain coverage for both of the languages. The applicability of the rules is demonstrated on step by step transformation of confidentiality example from BPMN to Secure Tropos. To validate the transformation, we recreate the original Secure Tropos diagram from the example context and compare it with the resulted one. The validation is based on semantic completeness and correctness of two models sets. The goal of the validation is to find out, if transformed model provides sufficient coverage of security related concepts, or in other words will the transformed model have the solid informational background that can be understood by security or business analysts.

1.3 Scope

In current thesis we continue the research on security definition of business process modelling, and as it was defined previously (Altuhhova et al., 2012), we concentrate on BPMN 2.0 *descriptive* modelling level, *analytical* and *executing* modelling are out of scope (White., 2004). We demonstrate the applicability of developed extensions on the running example of Internet Store regarding to the *integrity*, *confidentiality* and *availability* of its valuable assets. Although the example is realistic, we are not attempting to apply it on the practice. The development or investigation of a practical tool for model transformation also remains out of scope.

1.4 Structure

The thesis is structured into three bigger parts: the Background, Contribution and Validation. The first part gives an overview on different techniques for security risk management, like CORAS, Automated Risk and Utility Management (known as AURUM), Goal-Risk Driven Assessment and Information Security Risk Management (ISSRM). We make a discussion and justify the choice of ISSRM for our research. Chapter 3 is dedicated to Business Process modelling and modelling approaches. Similarly to Security Engineering chapter structure we first make an overview of some modelling languages, such as UML Activity Diagrams, Yet Another Workflow Language (YAWL), Event-driven Process Chain, or EPC, and Business Process Model and Notation (BPMN), and explicate the idea for choosing BPMN. The last chapter in this part gives an overview on our previous study and related studies on BPMN applicability to express security. The second part, or Contribution, presents our investment regarding two research questions defined above. In Chapter 5 we propose the extensions to

BPMN approach and Chapter 6 in turn presents our attempt for model transformation from BPMN to Secure Tropos. The last part represents our validation to both steps of the study, including the language extensions and model transformation. In Chapter 7 and 8 we describe the experiment that was held to assess the understandability of the language extensions and give and give a discussion of quality of transformation. In Chapter 9 we provide our conclusions, including limitations of the study and future perspectives.

PART I. Background

CHAPTER 2. Security Engineering

“Security Engineering is concerned with lowering the risk of intentional unauthorized harm to valuable assets to level that is acceptable to the system’s stakeholders by preventing and reacting to malicious harm, misuse, threats and security risks.” (Firesmith, 2003)

The discipline of security engineering concentrates on tools, processes and methodologies that support analysing, designing and implementing new systems or adjusting existing system according to needs of its environment. The analysis of security should be performed through the whole software development process, starting from early requirements and going through design and implementation phases. An early consideration of security allows analysts and modellers to discover threats, its consequence and to develop different countermeasures. It also proposes sufficient security level design alternatives and even supports the decision for project abrogation in case if the risk is too high.

Security modelling is an important aspect of Security Engineering. It allows analysing the system security on early stages of its development, defining roles, permissions and valuable artefacts of the system, capturing the software or network architecture, the system trust boundaries and defining security policy. There exist different tools and languages to help modellers proposing suitable security solutions.

2.1 Security Risk Management

Security Risk Management is an analytical procedure that helps to identify system valuable assets, stakeholders and operations, as well as risk levels of undesirable events; provides logics and guidance to find and implement appropriate solutions for specific situations and mitigation strategies; offers measures, defined in order to lower the risk level and reduce the likelihood of undesired events. In purposes to perform security risk management, many different methodologies were developed. In this chapter we make a short overview of four security risk management methodologies, such as CORAS, AURUM, Goal-Risk driven assessment and ISSRM. We also provide a detailed view on Information System Security Risk Management approach and motivate the choice of selected methodology for the current research. The choice is based on comparison of above mentioned security risk management methodologies with respect to their domain, process, modelling languages being provided and means for estimation. The comparison and conclusions can be found in the end of running Chapter.

2.1.1 CORAS

The CORAS offers three artefacts, such as a method for asset-driven risk analysis, language and a tool. CORAS is a model-driven approach (Braber et al., 2007), which includes a systematic guidance for security risk analysis. The CORAS method doesn’t provide only an overview on how two fulfil the objectives, but also on how to apply different analysis techniques. The CORAS language, based on UML profile, proposes means for documentation, analysis and representation of security risks. It offers five kinds of diagrams for particular stages of risk analysis. The presentation and of risk analysis results can be realized by the CORAS tool, which is a graphical editor, supporting the CORAS language.

The guidelines for CORAS method consist of eight steps, starting from preparations for the analysis and finalized with risk treatment using treatment diagrams (Lund et al., 2010). Each step is divided into

subtasks with concrete objectives. First step is dedicated to initial preparations for the risk analysis. It includes setting the scope and objectives as well as informing the customer of its responsibilities. At step 2 the introductory meeting is organized in order to collect useful information from the customer’s discussion. It is important to know what assumptions that can be made, and what scope of overall analysis we are dealing with. The purpose of the next step is to present the targets from the analyst view; he identifies system valuable assets and performs a high-level analysis. It is a right time for corrections and clarifications of some issues and misunderstandings. Step 4 is focused on agreement on the target to be analysed; it includes scope, assumptions. Important aspect here is to agree on scales for consequences and likelihood. It is only finished when documentation is prepared by analyst and confirmed by customer. The objective of step 5 is to identify all the possibilities of threats, vulnerabilities and threat scenarios. For this purposes the structured brainstorming is used. All the findings are documented in CORAS threat diagram. Properly composed threat diagrams are then used to estimate likelihoods and consequences at Step 6. It is required to compute the risk values in order to decide if the risks are acceptable or should be treated in future. Step 7 concentrates on giving the first risk picture, including determination of what risks from identified must be considered for treatment. The risk evaluation criterion is also defined at this step. The objective of Step 8 is to identify treatment and address cost-benefits, before a final plan is composed.

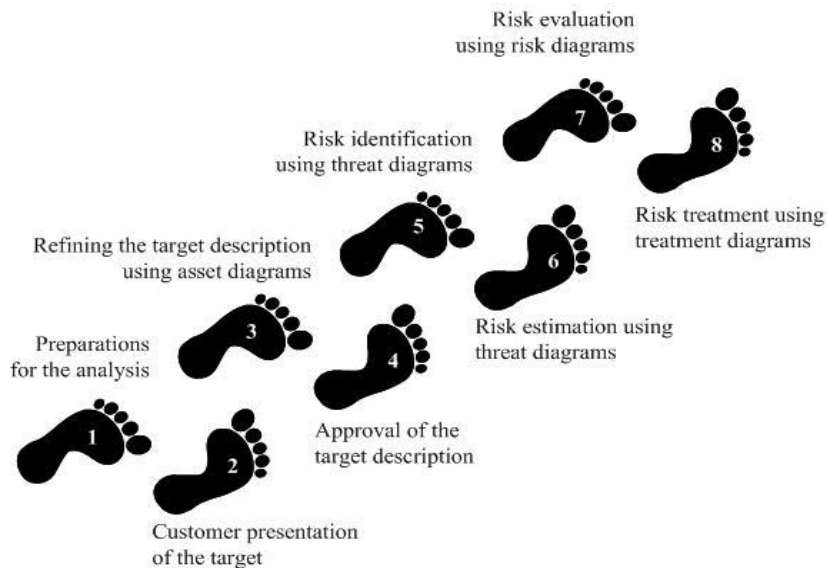


Figure 1. Eight steps of CORAS method guidance; adapted from (Lund et al., 2010)

2.1.2 Automated Risk and Utility Management

Automated Risk and Utility Management, AURUM, is a prototype that supports decision making according to organizational needs with respect to selection of security measures (Ekelhart et al., 2009). AURUM provides different modules regarding the general information security domain. The AURUM-Inventory module supports the system characterization phase, performing network scanning and providing solutions. For the threat determination AURUM uses Bayesian-network, a probabilistic graphical model that shows variables and their conditional dependencies. Having access to security

ontology and AURUM-Bayes module, AURUM-Risk module calculates the risk levels for different organizational assets.

The first step of methodology concentrates on defining the system boundaries. It takes into account document reviews, interviews and questionnaires. From the beginning AURUM guides user by inventory process and ensures formally correct infrastructure model. Moreover, it offers a comprehensive resource classification and allows defining new concepts in used classification. With clear resource definitions AURUM provides consistency in ontological model.

Step two is dedicated to threat identification. It is important to define potential threats and its possible sources; natural threats, human threats, environmental threats. The report on potential threats is produced and used as an input for a next phase, which stands for vulnerability identification.

The main objective of step three is to consider possible vulnerabilities in the field of management security, operational security and technical security. A mitigation control is created for each of defined vulnerabilities, the fact that the control is implemented means that vulnerability can be closed.

Next step of AURUM methodology involves control analysis. It gives an overview on what controls will mitigate the likelihood of a threat. AURUM uses the data collected at the system characterization stage and is able to conduct analysis of controls automatically.

In order to define the likelihood of a threat exploiting certain vulnerability within the system boundary, it is required to take into account the following aspects; the capability of threat resource, the origin of the vulnerability and effectiveness of existing controls. The Bayesian networks are used to determine the likelihood in specific organization.

Step 6 refers to impact analysis. It is necessary to understand the risk level and for the basis for suggestion the controls to implement. AURUM proposes automatic support to determine the impact of threats, using develop knowledge base.

The final step is dedicated to risk determination. At this step the risk is calculated by multiplying the values assigned to risk likelihood and ratings of impact.

After the overall risk was calculated, the list of control recommendations can be proposed. AURUM focuses on the selection of controls for all objectives. This ability is very important on the level of efficiency for the organization. Moreover, AURUM provides decision makers with intuitive interface that helps to find specific salvation for a problem. The methodology is based on modifications of lower and upper bounds for selected objective.

2.1.3 Goal-Risk driven assessment

The Goal-Risk driven assessment is another methodology in software engineering to manage security risks. In order to operate the concepts of goal, task and resource Goal-risk driven assessment uses Tropos software methodology. The conceptual representation of Goal-Risk model introduces three layers: asset, event and treatment layer (Asnar et al., 2011). It allows modelling and analyzing the organizational settings from early requirements to implementation. In purposes of introduction the specific concepts and relationships for risk analysis, Tropos goal modelling framework was extended. Conceptual representation of model includes three layers: one to introduce assets, another for events and

the last one for treatments. Mathematically, GR model is depicted as a set of nodes, having relations among them including special *impact* relation.

The asset layer introduces system valuable assets and strategic interests of stakeholders, the inter-relations between them, basing its graphical representation on Tropos goal model. The creation of a model starts with identification of top goals of system stakeholders. Each goal is then refined with AND- or OR-decomposition into sub-goals. The following decompositions allow branching the ways to fulfil the goal, providing alternatives. Next step is to create the relationships between goals. With the help of these relations, the interrelations between the asset layer and other layers can be also added.

The event layer characterizes events, concentrating on two properties: likelihood and severity. In the model, likelihood is defined as a property of an event and can hold the qualitative values like *likely*, *occasional*, *rare* or *unlikely*. Severity is modelled as a sign of an impact relation. Severity can be the following: *strong positive*, *positive*, *negative* or *strong negative*. Such view allows modelling the situation when impact has an influence on more than one goal. The creation of event layer starts with identification of events; it can be performed with different approaches. When it is done, events are decomposed into sub-events following the similar relations as in the asset layer. Dependency among events can be modelled with the use of contribution relations.

The treatment layer focuses on risk treatment, mitigation and countermeasures. The analysis of treatment layer can be realized with decomposition and contribution relations, similarly to assets and events. There are two variants of how the treatment impacts on the risk: it can reduce its likelihood or smooth its severity. Treatment methods are divided into following categories: removal or avoidance, prevention, attenuation and retention. The choice of how to elicit treatment depends on how it mitigate the risk on event layer.

Although, no process basing on step by step guidance is proposed by the Goal-risk driven assessment methodology, all the concepts and relations are structured in Meta-model. The model introduces the basic structures, such as: goals, tasks, resources and events, as well as relations between them: decomposition, contribution alleviation, means-end, needed-by. Impact is considered as a special relation because of the purposes of its use.

Current methodology also includes the risk assessment process. The latter consists of three steps: to find any alternative solutions, to measure the alternatives and evaluate them against relevant risks and finally, to assess the countermeasures in order to mitigate risks.

2.1.4 Information System Security Risk Management

Information System Security Risk Management (ISSRM) is practitioner-oriented methodological tool that helps organizations make decisions related to the security of Information Systems (Mayer, 2009; Dubois et al., 2010). The ISSRM application follows the general risk management process consisting of six steps. It starts with organizational context and assets identification. Then the determination of security objectives is performed. Basing on the required protection level for valuable assets, we turn to *confidentiality*, *integrity* and *availability*. The next step is dedicated to risk analysis and assessment, helping identification of potential risky scenarios and impacts they lead to. When the risk assessment is performed, it is time to take the risk treatment decision. The latter will result into security requirements definition, which in turn will be implemented into security controls. The important artefact of ISSRM

In order to compare different methodologies of security risk management we involve four criteria. The definition of *domain* stands for existence of Domain or Meta model that represents all the concept or groups of concepts that methodology uses, as well as relations between them. It can be realized with the help of class diagrams, like in case with ISSRM and Goal-Risk driven assessment. The next criterion called *process* describes the defined guidelines to perform security risk management step by step. The methodology can provide the concrete process or give the general guidance on how to apply the methodology to different situations. As well as a process, Security Risk Management methodology can provide the *language* developed in direct purposes of it. It can propose new and absolutely unique language as well as extend or use existing modelling languages. The last criterion is *estimation*. It stands for ability to provide any means, rules or values to estimate risks, impacts or cost-benefits.

The table below (see Table 1) gives an overview on how different methodologies cover these four aspects that we defined earlier. We can notice that the Domain model is provided by Goal-risk driven assessment methodology (Asnar et al., 2010) and ISSRM (Dubois et al., 2010). Both define basic concepts and relations with the help of Class diagram. The guiding process is introduced precisely in CORAS, AURUM and ISSRM. Goal-Risk driven assessment concentrates on general representation of three levels conceptual model representation; it defines the major purposes of each level. Considering the next criterion, the existence of applicable modelling language, CORAS introduces its own, specialized diagrammatic language for risk modelling. For graphical representation AURUM uses Bayesian networks, a probabilistic graphical model that represents variables and dependencies. In its turn Goal-risk driven assessment presents the Tropos extension. The language was extended with the purposes to present specific concepts, such as impact, likelihood, and severity. ISSRM propose an opportunity to choose the language to model with, it can be Secure Tropos (Matulevičius et al., 2008b), Misuse Cases (R. Matulevičius et al., 2008a) or some of other possibilities. With respect to estimation it can be said that CORAS process involves a step which dedicated to risk treatment and also addresses the cost-benefits. AURUM defines the estimation of risk by multiplying the values assigned to risk likelihood and ratings of impact. Considering the Goal-risk framework methodology, it Methodology includes the risk assessment process, consisting of three steps. ISSRM provides metrics for the risk level evaluation, threat likelihood and the potentiality of events, as well as cost benefits for risk treatment module (Mayer, 2009).

2.3 Why ISSRM?

ISSRM provides a domain model which presents concepts of three major groups: asset-related concepts, risk-related concepts and risk treatment-related concepts (Mayer, 2009). These concepts and relationships between them are presented in a form of UML class diagram. Precise definitions of concepts are given in documentation. The domain model of the methodological glossary helps to understand the scope of ISSRM. ISSRM follows the general risk management process, which consists of six steps, including the definition of organizational context, determination of security objectives, risk analysis and assessment, taking the risk treatment decision, defining the security requirements and finally the implementation of security controls. The process is iterative and should be followed as many times as it necessary to reach the acceptable level of risks. Although ISSRM doesn't define the concrete

language to be applied during the process, it can be confronted with a number of security-oriented modelling languages such as Misuse-Cases, Mal-activity Diagrams, KAOS, Secure Tropos and others. Finally ISSRM introduces metrics for evaluating risk and impact levels, the likelihood of treat and potentiality of event. It also involves evaluating the security need for objectives and cost benefits for risk treatment, security requirements and controls. Business assets are estimated with respect to its value.

Table 1. Artefacts provided by different risk management methodologies

| Criteria | CORAS | AURUM | Goal-Risk driven assessment | ISSRM |
|-------------------|--|--|--|---|
| Domain | - | - | All the concepts and relations are structured in the Meta-model | ISSRM Domain Model is divided into three groups of constructs related to assets, risks, risk treatment. |
| Process | Introduces eight-step process for risk management | The process describes a guidance that covers six steps of risk management | - | Follows the general risk management process consisting of six steps |
| Language | CORAS language is customised, diagrammatic language for risk modelling | Uses Bayesian-network, a probabilistic graphical model that shows variables and their conditional dependencies | Uses Tropos goal modelling framework extended for the purposes of methodology. | Doesn't provide any concrete language, but it is applicable with Secure Tropos, Misuse Cases, Kaos or EPC |
| Estimation | Defines treatment and addresses cost-benefits | Estimation assigned to risk likelihood and ratings of impact. | The risk assessment process, consisting of 3 steps | Mainly concentrates on risk and impact levels as well as on likelihood of threat events |

2.4 Security risk-oriented modelling languages

There exist a number of studies that are focusing on security risk management for IS requirements engineering, for example: Mal-activity Diagrams (Soomro and Ahmed, 2012), Misuse Cases (Chowdhury et al., 2012) and Secure Tropos (Matulevičius et al., 2012). In these studies modelling languages were aligned to ISSRM domain in purposes to understand how they deal with security. Each of these languages was then extended on semantic and syntactical levels. To investigate the possibility of transformation, which will be described in the following chapters, we have chosen one of security risk-oriented languages, Secure Tropos, which will be briefly introduced in the next section.

2.4.1 Secure Tropos

Secure Tropos (Matulevičius et al., 2012) is an extension of Tropos (Bresciani et al., 2004), which allows modelling security sensitive scenarios involving identification of *actors*, *goals* and *softgoals*, *plans*, *resources* and *believes* as well as *security constraints* and *threats* on early stages of system development. Early requirement analysis benefits the better understanding of organizational setting. In comparison to BPMN Secure Tropos doesn't cover the process flow, but focuses on security assets. It features security attack scenarios, highlighting security constraints and methods as well as business and IS assets, which are covered by protection mechanisms. The result of language alignment to ISSRM domain model can be found in **Table 4**, which demonstrates the part of security concepts covered by the language.

Secure Tropos introduces the following set of constructs. An *actor* is defined as an entity that expresses any intentions or strategic aims within the organization or its setting. A *goal* (sometimes *hardgoal*) is a representative of actor's strategic intentions or interests. A *softgoal*, unlike a *goal*, has a weak criterion to be decided whether it is satisfied or not, however it also represents the actor's aims or interests. A *plan* introduces a way to achieve the desired goal(s). Any informational or physical entities of the system can be represented by *resources*. A *belief* expresses the actor's knowledge of the world. The above mentioned concepts are introduced in both, Tropos and Secure Tropos, however the last introduces an additional set of concepts concerning security. A *security criterion* is a restriction in security that must be respected by organization and its actors. A *threat* is defined as circumstances that provoke the danger and loss for assets valuable to the system.

The number of models involved in the process of information system development and supported by Secure Tropos. The *security enhanced actor model* (SEAM) introduces actors, environment and dependency relationships between the components. The *security enhanced goal model* (SEGM) helps understanding how the actor's goals to be fulfilled, plans to be realized and availability of resources to be obtained. A *security reference diagram* covers the identification of security requirements.

To consider the models complete, different relations between the constructs of Secure Tropos should be introduced. A *dependency* relationship indicates who from the actors are depender, dependum and dependee. Dependency may be given between several actors. Secure dependencies establish security constraints, which have to be respected by actors and implemented by the dependee. *Decomposition* is used for decomposing abstract entities into more specific components. *Means-ends* in turn link a plan, resource or goal (mean) with a goal (end). *Contribution* describes a positive (+) or negative (-) impact one mean (e.g. plan, resource, goal) has on another. *Restricts* is a relationship that indicates which security constraint is applied on a certain asset. *Attack* relationship identifies the target of an attacker's plan.

Table 2. Secure Tropos and ISSRM alignment

| ISSRM concepts (R – relationship; C - concept) | | Constructs of Secure Tropos |
|--|-----|---|
| Assets | C | 1) <i>Actor, Hardgoal, Plan, Resource, Softgoal, Secure Goal</i> 2)Composition of the constructs (1) using dependency, means-ends, contribution and decomposition relationships |
| Business asset | C | |
| IS asset | C | |
| Supports | R | Relationships: dependency, means-ends, contribution and decomposition |
| Constraint of | R | Implicitly: In the <i>Secure Dependency</i> link Security constraint restricts (is a constraint of) a dependum Explicitly: Restricts link between <i>Security constraint</i> and <i>Plan, Resource, and Hardgoal</i> |
| Security objective | C | |
| Security criterion | C | <i>Softgoal, Security Constraint</i> (incl. decomposition), <i>Contribution</i> |
| Characteristics of | C | - |
| Vulnerability | C | not officially covered, but extension <i>vulnerability point</i> may be added to <i>Plan, Hardgoal</i> or <i>Resource</i> |
| Attack method | C | <i>Plan</i> |
| Impact/negates/harms | R/C | <i>Impacts</i> -relationship |
| Targets/Leads to (leads to a harm of IS assets) | R | - |
| Leads to (leads to a harm of Business assets) | R | - |
| Threat agent | C | <i>Agent</i> |
| Uses | R | Agent executes plan |
| Threat | C | <i>Goal, Plan</i> |
| Event | C | 1)Composition of an <i>Agent, Hardgoal, Plan</i> and <i>Vulnerability points</i> (incl. <i>target-</i> and <i>exploits</i> -relationships) 2) <i>Threat</i> |
| Risk | C | Composition of <i>Threat</i> and <i>Impacts</i> -relationship |
| Significance assessed by | R | - |
| Provokes | R | - |
| Decision to threat | R | - |
| Leads to | R | - |
| Implements | R | - |
| Controls | C | 1) <i>Actor, Hardgoal, Plan, Resource, Softgoal, Security Constraints</i> 2) Components constructed when combining constructs (1) using dependency, means-ends, contribution and decomposition relationships, mitigates-relationship |
| Security requirements and mitigates | C/R | |

CHAPTER 3. Business Process Modelling

Business process management enables to define, implement, improve and analyze the interactional flow within the organization correspondingly to its' goals and needs of its stakeholders. More specific it deals with interactions between people, working in the organization, parts of a system (i.e. applications, databases) or external communicating parties (i.e. business partners). In other words, it guides managing the organizational business processes. The approach on BPM implementation generally includes three basic steps. It starts with research, directed on the identification of business tasks involved into the process and the set of associating business rules. Resulted into the visual model, the business process now gives an overview of process flow and its participants. Step two: analytics continue the research, filling the model with information on process' external participants, their communication and exchange of recourses. In addition the model is updated with error and exception paths and requires finding the best way of handling. And finally, at the step three, model is implemented. It is then provided with additional business logic to perform the process functionality and tested to find out if something needs to be improved or changed.

The modelling aspect is an important part of the management process and in IS development. At the different stages of business process management models are the inputs and outcomes, so it considered being a valuable artefact in analysis. The visualization via modelling allows seeing the business process in details for its better understanding; it helps specify standard and optimised workflows of organisation. In addition, the graphical model opens an opportunity for marking the exceptional flows that leads to better error and exception handling.

3.1 Approaches to do BPM

We have chosen three more modelling languages in order to compare them between each other and to explain the choice of BPMN for analysis performed in this paper. The languages we involve into comparison are the UML Activity Diagrams, Yet Another Workflow Language (YAWL) and Event-driven Process Chain (EPC). The overview of all the languages, its major semantic aspects and simple examples are presented below. Conclusions of comparison and criteria can be found in last paragraph of this chapter.

3.1.1 UML Activity Diagrams

The tool that is frequently used for business process modelling is UML activity diagrams. The standardized general-purpose modelling language, UML, uses activity diagrams to model the workflow behind the system being designed (Börger et al., 2000). An activity diagram is a flowchart, which shows the flow of control between the sequential activities of a process. It is typically used for modelling the logic captured by a single use case or usage scenario or for modelling a detailed logic of a business rule.

The process depicted in activity diagram begins with initial node that symbolizes the default starting place and can be only one for the whole diagram. Action node represents an atomic action of the process flow. Activity nodes compose a flow of control in association with other activities and action states. Paths from one action or activity to another are represented with the help of so called transitions. Optionally, transition can be labelled by *event* or *guard*. The branching is also involved in UML activity

diagrams; it is realized with branch nodes that generally specify any alternative paths of a process flow. Fork and Join bars are used to split a single flow in two parallel control flows of the process. UML with its precise semantics allows depicting system dynamic behaviour and giving the opportunity to see the functionality of each system component.

The applied technique of building the UML activity diagram is demonstrated in example of order fulfilment process (see **Figure 4**). Let's imagine we have a system that receives an order for some good. The process of order fulfilment starts with initial node leading to the first activity *Receive order*. The process flow is split based on a decision if the order is approved (can be fulfilled) or not; activity *Fill order* continues the process flow in case if the order is approved. If the order cannot be fulfilled for some reasons it will be denied. The normal flow then is continued with activities *Send a bill* and *Ship order* that are performed in parallel. The following is realized with fork and join. After the order is fulfilled, the process flow leads to activity *Close order*. The same is performed if the order is denied, but missing the preceding part of a process. And finally activity flow is ended with the end node. **Figure 3** introduces basic shapes of UML activity diagrams.

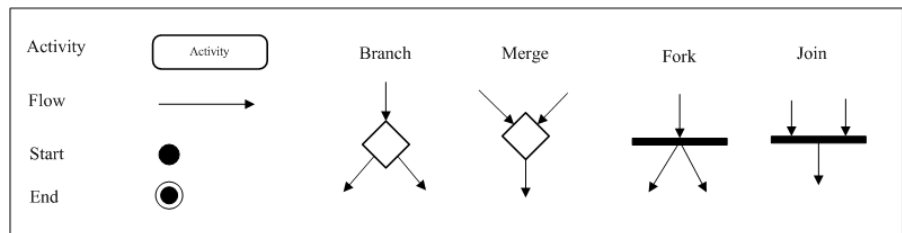
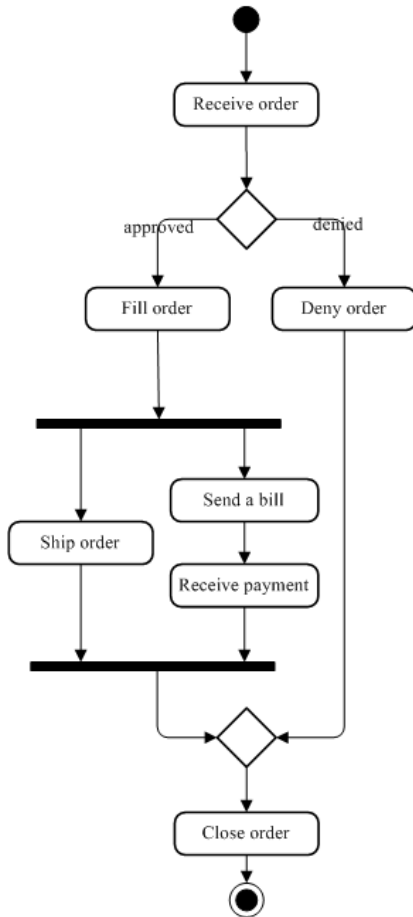


Figure 3. UML semantics

Figure 4. UML Activity Diagram:
Order fulfilment process

3.1.2 Yet Another Workflow Language

Another representative of the business process modelling tools is YAWL, Yet Another Workflow Language, which is java-based open-source workflow system (Aalst et al., 2005). Funded on a concise Petri nets, YAWL is able to support complex data, integration with organizational resources and external applications, process verification and process configuration.

YAWL has a formal semantics that benefit using it among other representatives of workflow modelling languages. A workflow specification in YAWL is formed with a set of so called EWF-nets, extended workflow nets. It has a hierarchical, tree-like structure, defined by *atomic* or *composite* tasks.

Each lower level of hierarchy is formed by a composite task of an ETF-net, which in turn contains tasks, atomic or composite, and conditions. Normally, each net has one input condition as well as one output condition. Tasks can have multiple instances that allow optioning: upper and lower bound should be identified, once the task is initiated.

In addition, YAWL has open interfaces based on Web standards that enable to plug-in existing applications and to extend the system. It also provides a graphical editor with built-in verification functionality that helps to detect errors automatically on early stages.

Figure 5 is a simple example of YAWL net that represents a process of order fulfilment. It has Initial input as a start element, which is followed by OR-split task *Receive order*, that points put the place of decision making basing on conditions. Condition *Approved* leads the process flow to next task *Fill order*. In case if order is not approved (Condition *not approved*), it is then cancelled. AND-split task *Fill order* slices the flow into two parallel actions; atomic tasks *Send a bill* and *Receive payment*, and the task *Ship order*. The flows are then united again into one with the help of an AND-join task and leded to the OR-join task *Close order*. The output condition is the one that end the process flow in YAWL. Basic YAWL graphical objects are represented in **Figure 6**.

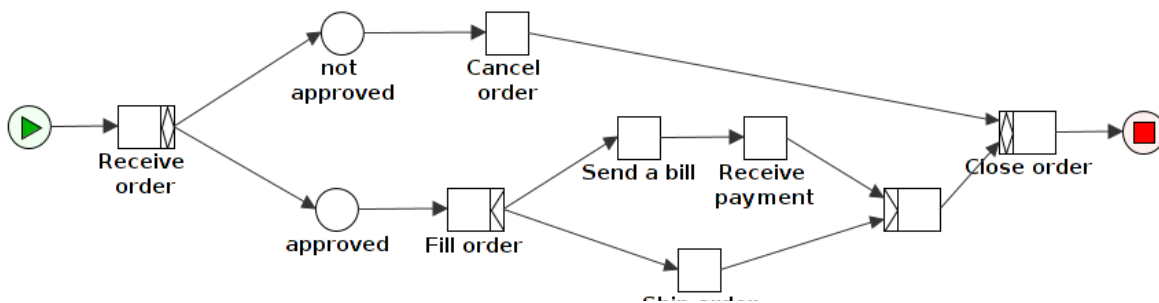


Figure 5. YAWL: order fulfilment process

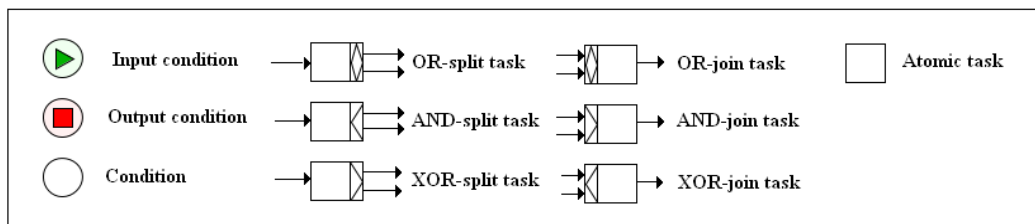


Figure 6. YAWL semantics

3.1.3 Event-driven Process Chain

An Event-driven Process Chain (EPC) is a type of flowchart that is used for business process improvement and also for laying out business process work flows, originally in conjunction with SAP R/3 modelling (Seel et al., 2005). The EPC is a base of the ARIS-framework and combines the different views towards the description of information systems in the control view on the conceptual level. EPC-models are created in order to plan, simulate and control the processes of private enterprise.

Processes, presented in models describe the use of functions and events as time-referring state changes. The process contains events and functions that describe passive states of activities. Functions and events are linked together with control flow. EPC model opens an opportunity to split and join the control flow; it is realized with the OR-, XOR- and AND-operators. In addition, control flow allows connecting two EPCs from different models together. Another important aspect of EPC modelling language is existence of resources that can be annotated to functions. The annotation describe the type of relation resource has to some function. Resources can be also annotated to other resources; the following is defined with relationship “resource structure”.

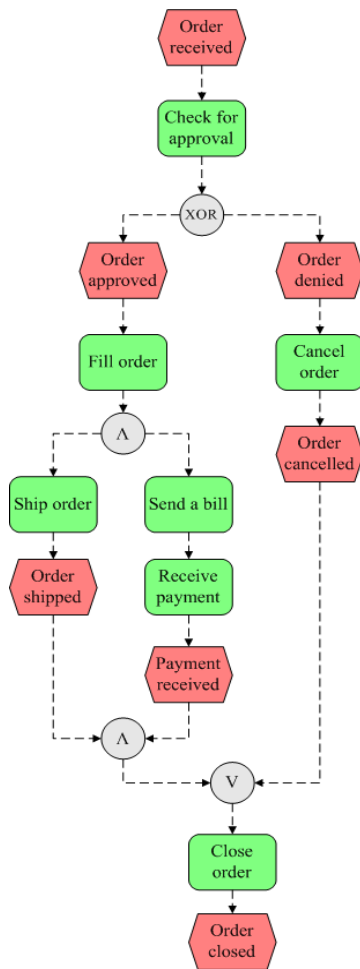


Figure 8. EPC example:
Order fulfilment

The EPC model depicted in *Figure 8* describes the simple order fulfilment process. The process flow starts with event Receive order. It is then followed by an EPC function Check order for approval. XOR-operator is used to split the flow into two paths guided by different decisions; if the order is approved it is then being filled (Function Fill order), otherwise the order is cancelled. Another type of operator, used in this example is AND-operator that unites the sets of functions performed in parallel (e.g. Ship order and Send a bill). After the payment is received and order is shipped, the order then can be closed; the function Close order is preceded by an OR-operator that specifies that the order is closed either in terms if it is filled or cancelled. Both paths lead to event Order closed, which ends the process flow. *Figure 7* gives an overview of basic EPC shapes used in the model.

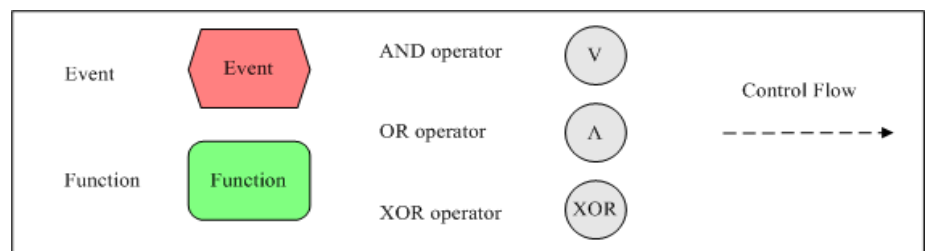


Figure 7. EPC semantics

3.1.4 Business Process Model and Notation

Business Process Model and Notation (BPMN) is a language for constructing business process models (Remco et al., 2007; Silver, 2009) It considered being business-friendly, because it is based on notions familiar from traditional flowcharting. At the same time, the notations are linked to a semantic model, which means that each shape used in the notation has a specific meaning, with defined rules and connections between objects. The key element of BPMN application is the Business Process Diagram. It is constructed of a set of graphical elements that were chosen to be distinguishable from each other and to utilize shapes that are familiar to most modellers. It describes a typical order of activities and what role or organizational unit performs or is responsible for the process.

The graphical objects of language are divided into four major groups, which are flow objects, containers, artefacts and flows. The business process diagram usually starts and ends with events; it can be triggered or not. In some cases Intermediate events can be used to end the some parts of the process, not the main process. The atomic activities of the process are represented by Tasks. Tasks and different types of gateways compose the process flow; it can be guided by decisions with the help of XOR-gateways or split into parallels by Parallel-Gateway. BPMN also allows involving system artefacts such as Data object and Data Store, and realise the communication between process participants, represented by Pools (Lanes in some cases), using the message flow.

Figure 8 represents the business process of order fulfilment. Process starts with event (non-triggered start event) and the task Receive order. The XOR-gateway then splits the flow into two; if the order cannot be approved the task Cancel order is performed, otherwise flow is continued with task Fill order and parallel activities Send a bill and Ship order. When the payment is received and order is shipped, task Close order can be performed. The same happens if the order was cancelled for some reasons. The process is completed and ended by default with End event. The legend of element used to build the BPMN order fulfilment process is presented in Figure 9.

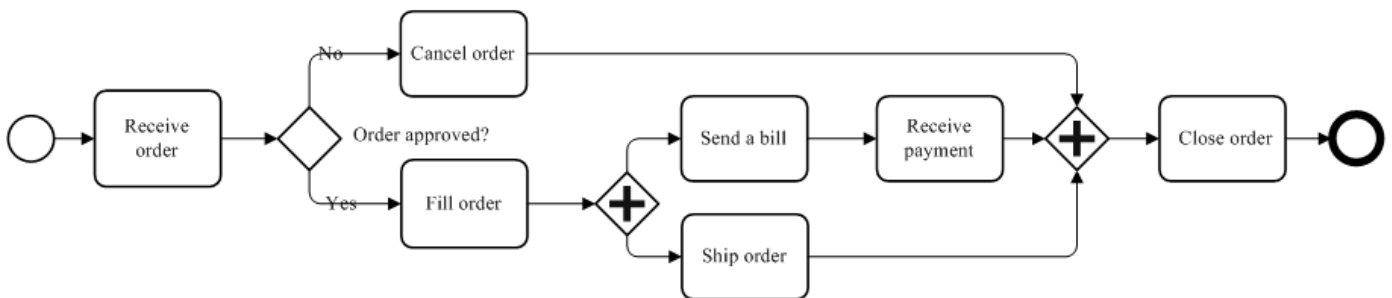


Figure 9. BPMN example: Order fulfilment process

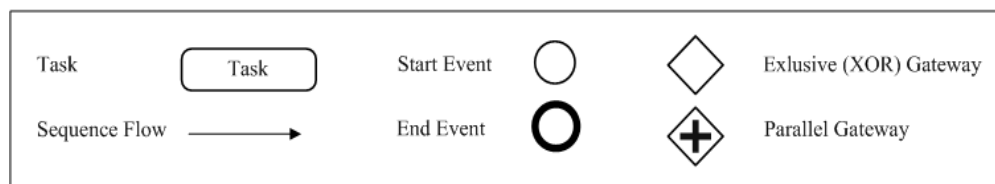


Figure 10. BPMN semantics

3.2 Comparison of the Languages

In this paragraph we introduce the reasons why we choose BPMN among other modelling languages. We compare BPMN with UML Activity Diagrams (Börger et al., 2000), EPC (Seel et al., 2005) and YAWL (Aalst et al., 2005). The criteria we decided to focus on is the *expressive capacity* of the language. We define it as an ability of the language to present as many semantic and syntactic aspects as it is possible. In current context term can be also described as a richness of language semantics to express different issues involved in business process; first of all, the process participants or stakeholders, their capabilities and skills. If we speak about business stakeholders, we should also pay attention to availability of communication between process participants. Then, we deal with data and information involved into the business process. The next important issue is the existence of resources that could be shared between process participants or just travel within the process flow. And finally, we examine the ability of the languages to describe processes, decisions and states (events). We analyse the expressive capacity of each language by defining the number of graphical objects that present one or another business issue. The more details, with respect to above mentioned, language is able to express the higher expressive capacity it has. The table below provides an overview on criteria and corresponding graphical objects in BPMN, EPC, YAWL and UML Activity Diagrams.

Table 3. Ability of the languages to express different business issues

| Business issues | BPMN | EPC | YAWL | UML Activity Diagrams |
|--------------------------------|---|---|---------------------------------------|--------------------------|
| <i>Stakeholders</i> | Pool, Lane | Organization unit | - | Swimlane |
| <i>Capabilities and skills</i> | Tasks (types User, Manual, Service) | Function, Organization Unit Assignment | Atomic task, Composite task | Activity |
| <i>Communication</i> | Data (Message) flow, Event (message- or signal-triggered) Data Association flow | Information flow | - | - |
| <i>Data and information</i> | Data Object, Data association flow | Information Resource, Information flow | - | - |
| <i>Resources</i> | Data Object, Data Store | Information Resource | - | - |
| <i>Process</i> | Sub-process, Parallel Gateway, Or-Gateway, Sequence flow | And operator, OR-operator, Path, Control flow | And-split and join, OR-split and join | Fork, Join, Control flow |
| <i>Decisions (Conditions)</i> | XOR-Gateway | XOR-operator | XOR-split and join, Condition | Branch, Merge |
| <i>Events (States)</i> | Start Event, End Event, Intermediate Event. | Event | Input Condition, Output Condition | Start, End |

Stakeholder is defined as a person, group, and organizational unit, part of a system or any other participant of business process that can be affected by organizational actions. BPMN supports representing system stakeholders with Pools, if we mean involving parties outside the organization, or Lanes, in case if stakeholders stand for different part of one system, e.g. workers of one organization. UML activity diagrams use Swimlanes for similar purposes. Each Swimlane defines external or internal participant of a process. EPC also has means to present system stakeholders; it is represented with a structure of Organizational Unit. It determines which person or organization within the structure is responsible for which Function. On the other hand, YAWL has no concrete structure to support business process stakeholders.

Stakeholders' capabilities and skills are the actions and activities performed by system stakeholders or tasks under their responsibility. BPMN provides opportunities to define capabilities through Tasks and labels such as Manual, User, or Service tasks, making an accent not only on responsibilities but division of automatic and manual tasks. EPC realizes relations between stakeholders and its capabilities with the help of Functions and Organizational Unit Assignment that connects the stakeholder and the Function it is responsible for. In its turn YAWL uses Atomic and Composite Tasks to define capabilities of process participants. UML Activity Diagram represents skills and capabilities of stakeholders with different Activities defined in Swimlanes.

Communication can be defined as an exchange of information, data, resources or different signals between process participants. BPMN supports the above mentioned issue by use of Events (e.g. message- or signal-triggered) and Message flow. Another way to depict the exchange of resources or data is to use Data Association Flow. It realizes the connection between Tasks and system resources. In case if two tasks are performed by different stakeholders and data is the outcome of one Task and the input of another, we can speak about communication. For similar purposes EPC uses Information Flow that creates communication and describes the exchange of Information Resource, which can be the outcome of one Function and the input for another. YAWL and UML Activity Diagrams provide no means for realizing the communication flow between process participants.

Data, information and resources are the important artefacts in business process. These terms cover all the material and immaterial assets, such as documents, databases, servers, processes and other, depending on context. To express data and resources BPMN uses Data Object and Data Store. It allows to define data sources such as databases, and also to involve any kinds of data sets. Both elements are related to Tasks and connected to it with Data Association Flow. EPC defined an Information Resource and Information Flow to deal with data and resources. YAWL, as well as UML Activity Diagrams, doesn't support any graphical representation of data, information or resources.

Every business organization involves many different processes to achieve its goals. Process is a sequential flow of activities performed by system stakeholders that leads to some end state or result. BPMN allows creating Sub-processes, using Parallel Gateways to fork the flow into some synchronous flows, involving OR-Gateways to give a preference to one of the forked flows. Similar to BPMN such opportunities are available in YAWL, EPC and UML Activity Diagrams.

Decisions are involved in almost every business process, because it has to provide optioning in majority of cases. BPMN as well as other three languages provide branching of the process flow and allow making the choice what path to follow. It is realized with so called XOR-Gateways in BPMN. In EPC and UML Activity Diagrams it is allowed not only splitting but also merging two or more flows into one if it is required.

It is also important to follow the states of business processes or to see results, end or intermediate, so that's why it can be useful to involve process states or events. Usually the process flow can be started with Events or so called Conditions, as well as it can be closed with End Events. BPMN provide triggers, such as Message-, Signal-, Cancel- and others, to define the type of Event. Besides it uses not only Start and End Event, but also Intermediate Events that allow to overview the intermediate results of Sub-processes. EPC uses Events as states of different stages of a business process. An Event shows what circumstances a Function works or which state a Function or process results in.

3.3 Why BPMN?

The choice of BPMN among other language is based on few aspects, mainly related to the fact that BPMN covers major part of business issues described above. First of all, It provides means to perform business communication, covering different issues; messaging (message-triggered events, message flow), the exchange of resources (data association flow), signals exchange (signal-triggered events). It allows involving such artefacts as Data Objects and Data Store to deal with various forms and volumes of data, which is very important in any type of business. Moreover, BPMN is able to define external participants of the process as well as internal ones. This property of the language broadens the view on process from the side of business relationships between participants or different organizations. Then, labelling and types are actively involved into the modelling; we can set tasks types dividing them into user, manual or system. And finally, BPMN defines the concepts of sub-process and intermediate event that allows making business process more granular and detailed. With all above mentioned we consider BPMN applicable for business process modelling in a way it is required by the research.

CHAPTER 4. BPMN and Security

In this Chapter we make an overview of our previous contribution regarding the alignment of BPMN 2.0 to ISSRM domain. We also observe some studies related to both, business process modelling with BPMN and security. And finally we make a discussion on the similarities and differences between our research and related works performed in the similar direction.

4.1 Previous work

There were number of papers published in recent years with a purpose to introduce the importance of information security and security in business processes. As the current aspects of system security becomes actual and vital for different types of organizations in modern technological world, in this Chapter we will observe the major findings with a focus to BPMN, considerations of security aspects in business processes and means of current modelling language to express these aspects.

First of all, we would like to present the results of our previous work (Altuhhova et al., 2012), dedicated to definition of secure business processes. More specifically, in our research we applied a structured approach in order to understand the key aspects of BPMN and the way they can be used to express secure assets, risk and risk treatment with constructs of BPMN. We align the BPMN constructs with key concepts of ISSRM domain model to investigate into question of applicability of BPMN to model security. We test current means of the language on running example of Internet Store. We conduct the analysis creating a table, which presents and summarize our major findings. The concise version of a table is presented below (See Table 3).

Thus the major contribution of this research is semantic alignment, we resulted into conclusions that BPMN does provide business analysts means for modelling security risks, but the graphical representation of these risks is limited with a sequence of different problems that can be grouped into three. First of all, it has to be noticed that the table is filled only partially, so the most common problem remains the absence of suitable constructs for ISSRM concepts. At this stage of analysis we could not define any constructs for *Asset*, *Security criterion*, *Risk*, *Impact*, *Event*, *Vulnerability*, *Risk* treatment and also for *Control*. Although none of these concepts found representation in BPMN concrete syntax, cases of *Event* and *Vulnerability* are different. The latter for example can be defined intuitively from the context of example and added to a textual annotation or description, the same way as *Security criterion* and *Impact*. Other aspects, such as *Asset*, *Risk* and *Event* remain undiscovered as well as missed in the BP Diagram. Moreover, in the Internet Store Example we deal with an overlapping semantics. The BPMN Task construct, for example, is used to express *Business asset*, *Attack method* and *Security criterion*, following situation leads to readability and comprehensibility problems. The same troubles are caused by use of Gateways, Flows and Events to express Security Risk Management aspects. Another problem is that some of the ISSRM concepts can be presented by several BPMN constructs. The following can be noticed for *Business asset*, *IS asset*, *Threat*, *Attack method* and *Security requirement*.

We claim that in general BPMN approach is not specifically directed to model the security, but to perform the business process modelling, and nevertheless, it should not lose its original purpose. On the other hand, though, we came to conclusion that the major set of constructs that benefits understanding of valuable business assets, risks, related to them and potential security requirements are covered by

BPMN with its current state, so the language requires some reasonable extensions. The scope of this research doesn't cover the determination of concrete extensions. However, we define some directions for potential and future research on that way.

Table 4. The alignment of BPMN constructs to ISSRM concepts

| | ISSRM Concepts | BPMN Constructs |
|---------------------------------|----------------------|--|
| Asset-related concepts | Asset | – |
| | Business asset | <i>Data object;</i> <i>Task, Gateway, Event, Sequence flow</i> |
| | IS asset | <i>Data store</i> <i>Pool, Lane</i> |
| | Security criterion | – |
| Risk-related concepts | Risk | – |
| | Impact | – |
| | Event | – |
| | Threat | A combination of constructs for <i>Threat agent</i> and <i>Attack method</i> |
| | Vulnerability | – |
| | Threat agent | <i>Pool</i> |
| | Attack method | <i>Task;</i> <i>Flows (e.g., Data flow with the label describing attack method;</i> <i>Data association flow with the label describing attack method);</i> |
| Risk treatment-related concepts | Risk treatment | – |
| | Security requirement | <i>Task, Gateway, Event,</i> <i>Sequence flow</i> |
| | Control | – |

4.2 Related Studies

Now we turn to other research papers that gained attention in the field of security in business process modelling. From the earlier researches, we should pay attention to the paper (Rodríguez et al, 2007) that raises the question of BPMN extensions with respect to Security Requirements modelling. Basing on the Model-driven Architecture (MDA) approach, authors propose the extensions on BPMN that allow involving security requirements considerations into business process modelling. They concentrate on the perspective of business analyst to investigate the understanding of current aspect. The analysis performed in this paper proposes developed Business Process Diagram (BPD) metamodel, where the relationships between core elements of BPMN are defined and where from comes the motivation to extend the syntax, abstract and concrete. First of all, they introduce innovations with respect to abstract syntax by involving such security-related concepts as non-repudiation, attack harm

detection, integrity, privacy and others. Then the concrete syntax of BPMN was extended by adding marks or indications to the graphical element of BPMN. In addition, the symbol of padlock is introduced and labelled with a capital letters that stand for security requirements. Authors believe that the presented extensions will enable business analyst to express security requirements from their perspective, which will lead to a high level of details captured from the security needs of different stakeholders and benefits the understanding of common goals.

In the next research (Menzel et al., 2009) authors direct the focus onto the BPMN enhancements towards trust modelling. They believe that multiple parameters have to be considered for sufficient security configurations. So they contribute to model-driven approach that enables annotating security intentions and ratings in business processes. More detailed, the metrics, described in the paper, allow giving a value to enterprise assets and concentrating on the level of security, the trust level for each participant in the process. The enterprise assets are presented using BPMN tasks, data objects, and communication links between tasks and participants. In addition, authors define how to enable trustworthy interactions, organisational trust, and security intensions through BPMN. Other proposed extension is a security policy model used to define specific security patterns for authorisation, authentication, integrity, and confidentiality.

Another paper (Mülle et al., 2011), which is motivated with the poverty of existing security vocabulary, concentrates on security constraints and security-specific user involvements. Authors contribute by integration of the security language, which supports above mentioned aspects, into BPMN 2.0. Each security unit of the language is represented as a structured text annotation. Each annotation is tied to a particular set of BPMN elements, such as tasks, lanes, message flows or others. The fact that defined security constrains are placed in BPMN artefacts doesn't create any contradictions at the syntax level. When transforming security constraints into the defined language, authors emphasize three more targets to observe: security policies, adjustment of the process flow, and parameter settings to apply security components. The components in turn create relations between users and tasks, apply security-specific policies to the process and help managing the security issues. Moreover, such an investigation into extensions of the open-source business-process-management system (BPMS) allows transforming traditional business diagram filled with security constraints to executable process.

Recently, in 2012, another paper has been explored. The research (Cherdantseva et al., Sept. 2012) contributes into definition of *SecureBPMN*, referring to the knowledge field of Information Assurance & Security (IAS). The latter covers the systematic management of Information Security countermeasures and is not limited with technical aspect of this question only; it also includes human-oriented aspects. The major contribution of this research is dedicated to consideration of IAS modelling capabilities in the focus for BPMN and to the perspective of developing the *SecureBPMN*, which will be the extended version of BPMN 2.0. The research method proposes to follow six steps on the way of development *SecureBPMN*. However, the scope of current paper doesn't cover all the steps of defined research method. Authors start with representing of the alignment of the BPMN with IAS ontology and the Multi-dimensional Model of IAS (MMIAS), which in turn was described in their previous research (Cherdantseva et al., Feb. 2012). As a result of this step, authors get an alignment table and make the

conclusion, that the security extensions for BPMN will not only use the existing elements of BPMN, but also require involving the new graphical elements to support effective security modelling.

4.3 Discussion

The main difference between our study and other researches (Rodríguez et al, 2007; Menzel et al., 2009; Mülle et al., 2011) at the first stage of current analysis is that we present a semantically grounded fine-grained analysis based on the well-established ISSRM domain model. On the contrary, authors of above mentioned works either concentrate on small number of security aspects in business processes or focus their analysis on the coarse-grained level. For example Rodríguez et al in his study emphasizes on necessity to include security requirements into business process modelling. We, in turn, cover the bigger range of security aspects, following the ISSRM process: asset identification, security objectives, risk analysis, risk treatment and security requirements. Menzel et al (2009) investigates into trust modelling, targeting the trustworthy operations and organizational trust concepts. The research is limited with only focus on security goals; authors don't consider any view on potential risks or its treatment. Mülle et al (2011) introduce the language that uses text annotations to present the security constraints and security-specific user involvements. However, some security aspects are still not covered.

Our previous work introduces the alignment between ISSRM concepts and the BPMN constructs, which allows analysts to understand current BPMN means to deal with security. We identify the horizons for potential BPMN extensions towards security. None of previously mentioned works provide a grounded analysis for proposed extensions or the reason why they target one or another security aspect. The conclusion of Cherdantseva et al. (2012) work points onto the necessity of extending the BPMN in terms of security modelling, which support our idea. In current work we go further and investigate into creation of such extensions at both, (concrete and abstract) syntax and the security risk-oriented semantics levels of BPMN. We introduce the new constructs for the language as well as define the meanings for existing ones in order to improve the ability of BPMN to express security concepts. That makes our research different from those works that provide no extensions or cover the limited number of security aspects. Rodríguez et al propose the extensions for abstract and concrete syntax introducing the symbol of padlock and labels for BPMN existing elements. Menzel et al extend the group of BPMN artefacts by adding the concepts of Organizational trust and Security Group. Mülle et al use the text annotations tied to BPMN elements to introduce the language structures expressing security policies, adjustment of the process flow, and parameter settings to apply security components. Cherdantseva et al. (2012) investigate into SecureBPMN but doesn't provide any extension at this stage.

The overview shows that some of these works do not present sufficient base for demonstrating the necessity of extending the BPMN or target only few aspects of IS security, meanwhile, others do not propose any extensions. We believe that our previous work (Altuhhova et al., 2012) and the current research, together cover the reasons *why* and *how* BPMN needs to be extended to consider security in business process modelling in full value.

PART II. Contribution

CHAPTER 5. Security Risk-oriented BPMN

In this chapter we introduce our major findings in solving problems described above (see Chapter 2). We provide extensions for the language, so that it can be used for purposes of modelling the security risks in business processes. At this stage of our research we propose the extensions on the BPMN 2.0 concrete syntax, abstract syntax and semantics levels.

5.1 Research Method

The research method used in current thesis describes four steps. We start with consideration of the literature on the risk management standards (AS/NZS 4360, 2004; ISO/IEC Guide 73, 2002), security-related standards (Common Criteria, 2005; Stoneburner *et al.*, 2002), security risk management methods (Alberts and Dorofee, 2001; Braber *et al.*, 2007) and software engineering frameworks (Firesmith, 2007; Haley *et al.*, 2008). We then analysed how current version of BPMN 2.0 supports ISSRM (Meyer, 2009; Dubois *et al.*, 2010). The process and the result of analysis were described earlier in previous contribution (Altuhhova *et al.*, 2012). Then, basing on that research, we identified the possible points for extension. We do not just point out the aspects that need to be improved but propose the ideas of how to improve them. The third step of the research method is to present the extensions. At this stage we introduce the extensions of BPMN semantics, abstract and concrete syntax. The main goal of the final step remains to demonstrate extensions in work, so it includes the presentation of illustrative examples. First two steps were accomplished earlier (Altuhhova *et al.*, 2012) and the performance of steps three and four is the objective of current thesis.

5.2 Extensions on Concrete Syntax

Although the need of language extension comes obviously from the results of our previous analysis (Altuhhova *et al.*, 2012), we followed the idea of keeping syntax simple, clear and using-friendly in order to avoid the congestion of BP Diagrams. Starting with ISSRM **Asset-related concepts** (see Table 1) we define the rules of expressing organization's valuable assets with combination of Events, Gateways and Tasks using the Sequence flow. We label Tasks with letter 'B' for defining the *Business asset*, and letters 'IS' – for *IS asset*. Use of labels is absolutely optional; if the following division for assets is not important, it can be skipped. Moreover, in some cases Tasks can be classified as automated or manual, which requires putting another label. So in order to keep the picture clear, emphasizing of this kind of information is allowed to be ignored. Another way to express *Business asset* is to use Data Object. It covers the cases when we deal with some valuable material or immaterial information, such as papers, documents or records in the database. *Business assets* are supported by *IS assets*, the latter in turn are depicted with the help of Data Stores and Containers: Pools, Lanes or both. *Support* relationship is described as follows: Container supports combination of Flow Objects by containing them. *Supports* is also the Sequence flow between or Data Association Flow between elements that play roles of *IS assets* and *Business assets*. To express the *Constraint of* relation we involved the new construct - the symbol of a lock, which is used in a combination with Association Flow and textual Annotation. The letter inside the Lock describes the *Security objective* of a *Business asset*. The Annotation gives additional information on *Security criterion*.

At the next stage we introduce the possibility to mark language elements with different colours: asset-related concepts are presented as usual (normally – black lines, no fill), we use red to highlight the risk-related concepts and blue – to show the risk treatment-related concepts. Such division helps to see the difference between the same BPMN constructs aligned to different ISSRM concepts. It also helps to avoid labelling methodology, which brings us to a risk of diagram overload with additional variables. Concentrating on ISSRM risk-related concepts, we open a new element which is a part of *Characteristics of* concept. In combination with Association Flow and Annotation, Vulnerability point is a property of constructs that describe *IS assets*, more concrete – Data Objects and Tasks. The vulnerability itself is described in the textual Annotation; red signals about the weakness of the system. When we speak about risks, we expect the existence of a *Threat agent*. If the attacker represents an independent unit, we use Pools to express this concept. In case if he is a part of a bigger system it can be represented by Lane, also coloured in red to differ from custom participants. The *Attack method* is a combination of Flow Objects and a Sequence Flow. It is built just the same way as a normal flow of a process, but using Tasks, Gateways and Events in red. It can be an independent process or a combination of constructs implemented into the normal flow of events, depending on what way the *Threat Agent* chooses to attack the system. *Uses* relationship is expressed with a Data Flow. This is usually the flow between Pool (*Threat agent*) and a Start Event of an attack process.

The next new element we introduce is an Unlock, which is a property of constructs that describe *Business assets* and present an *Impact*, *Harms* and *Negates* relationships. It is depicted as an opened lock and holds a letter of broken security criterion inside, correspondingly to *confidentiality*, *integrity* or *availability*. The *Impact* leads to a harm of valuable assets. There are two ways to express *Targets* and *Leads to* relationships; it depends on what assets we are dealing with. If the talk is about a harm of *IS assets*, we use Sequence Flow from Flow Objects that represent the *Attack method* to Flow Objects of *IS assets*, or Data Association Flow from Task to Data Store. In case of *Business assets* - Sequence Flow from Flow Objects that represent the *Attack method* to Flow Object expressing *Business assets*, or Data Association Flow from Task to Data Object. Such complex concepts as *Risk*, *Event* and *Threat* are described in the context of ISSRM model as a compositions from early defined simplex concepts. For example *Threat* is appear to be the combination of a *Threat agent* and his *Attack method*, so it is depicted as a combination of constructs for these two concepts. Rest of the risk – related concepts is defined just the same way (for more details see Table 2).

The idea of keeping the language simple and comprehensible influenced mostly on risk treatment-related concepts (see Table 3). We defined *Security Requirements* and *Mitigates* relationship as a combination of Flow Objects using Sequence Flow and also specified the colour of elements, which is blue. The reason why we only concentrated on this aspect is a wish to accentuate the process of risk treatment. We emphasized on showing the integration of tasks and activities that would help to keep the process safe from the harmful attacks. There are still no constructs for *Significance assessed by*, *Decision to threat*, *Controls* and all the relationships. But we consider this information to be excessive and inessential for being depicted within the process on the diagram. However, it should be mentioned in a textual annotation, description to a model or in the report for the whole picture.

Table 3. Extensions on BPMN Concrete Syntax: Asset-related Concepts



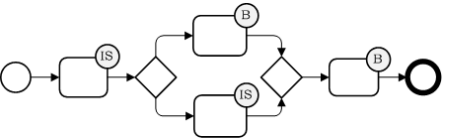



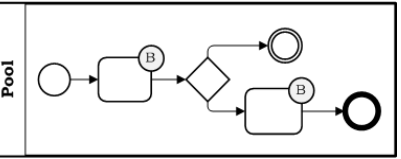
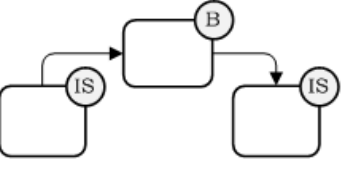
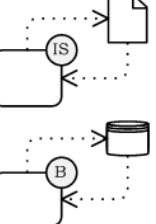



| ISSRM concepts | Constructs of BPMN | Concrete syntax |
|---------------------------|--|--|
| Assets | Combination of Flow Objects (Event, Gateway, Tasks) using sequence flow. For Business assets  For IS assets  |  |
| Business asset | Data object |  |
| IS asset | Data store Containers (Pool and Lanes) |   |
| Supports | Implicitly: Container (IS asset) supports combination of Flow Objects (Business assets) by containing them. |  |
| | Sequence flow between Flow Objects (IS assets) and Flow Objects (Business assets) |  |
| | Data Association Flow between Task (IS asset) and Data Object (Business asset) and between Data Store (IS asset) and Task (Business asset) |  |
| Constraint of | Lock and Association Flow that points from the Lock to an Annotation. Lock is a property of constructs that describe Business assets (Data Objects and Tasks) |  |
| Security objective | Is a property of a Lock that can have a value: c – confidentiality i – integrity a – availability |  |
| Security criterion | Annotation |  |

Table 4. Extensions on BPMN Concrete Syntax: Risk-related Concepts

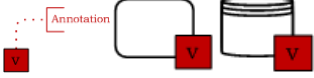

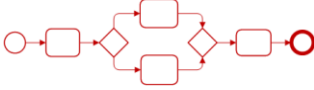

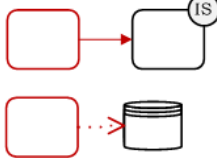
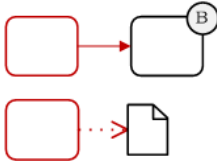


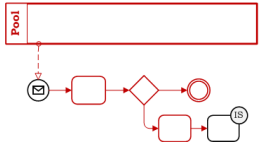
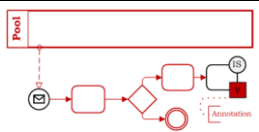
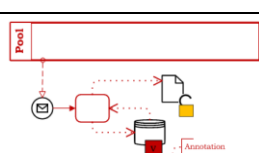
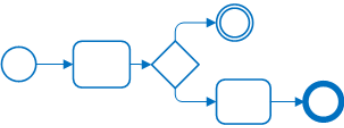
| ISSRM concepts | Constructs of BPMN | Concrete syntax |
|--|--|---|
| Characteristics of | Vulnerability point and Association Flow that points to Annotation. Vulnerability point is a property of constructs that describe IS assets, i.e. Data Object and Task |  |
| Vulnerability | Annotation |  |
| Attack method | Combination of Flow Objects (Event, Gateway, Task) using Sequence Flow |  |
| Impact/negates/harms | Unlock Unlock is a property of constructs that describe the Business assets |  |
| Targets Leads to (leads to a harm of IS assets) | Sequence Flow from Flow Objects (Attack method) to Flow Objects (IS assets). Data Association Flow from Task (Attack method) to Data Store (IS asset). Sequence Flow and Data Association Flow both correspond to Targets and Leads to (in this case it leads to the harm of the IS assets). |  |
| Leads to (leads to a harm of Business assets) | Sequence Flow from Flow Objects (Attack method) to Flow Objects (Business assets). Data Association Flow from Task (Attack method) to Data Object (Business asset). Leads to a potential harm of the Business asset. |  |
| Threat agent | Pool and Lane (Containers) |  |
| Uses | Data Flow |  |
| Threat | Combination of construct for Threat Agent and Attack method |  |
| Event | Combination of constructs for Threat and Vulnerability |  |
| Risk | Combination of Event and Impact |  |

Table 5. Extensions on BPMN Concrete Syntax: Risk Treatment-related Concepts

| ISSRM concepts | Constructs of BPMN | Concrete syntax |
|-------------------------------------|---|---|
| Significance assessed by | No construct | - |
| Provokes | No construct | - |
| Decision to threat | No construct | - |
| Leads to | No construct | - |
| Implements | No construct | - |
| Controls | No construct | - |
| Security requirements and Mitigates | Combination of Flow Objects using Sequence Flow |  |

5.3 BPMN Meta-Model

BPMN uses four major classes of constructs at the level of *descriptive modelling*: these are Flow Objects, Containers, Flows and Artefacts (see *Figure 1*). Flow Objects represent atomic units of a process with the help of Events, Tasks and Gateways. The start and the end of a process flow is indicated by the Event: it can be triggered or non-triggered. The Task describes any atomic activity performed in the process flow: sometimes it can also represent the collapsed sub-process. What concerns Gateways: it realises the control of the sequence flow: forks it basing on different decisions or helps to organize parallel activities.

The role of object holders is performed by Containers, these are Pools and Lanes. Pools represent the participants of the process as independent units, showing the message flow between them. The Pool can contain some number of Lanes, each represent different parts of one working system: a performer role on organizational unit.

The BPMN Artefacts’ group is represented by Data Store, Data Object and Annotation. Data Objects describe recourses that travel within the process flow: data can be produced by one activity and used as an input by another. To demonstrate how the data can be stored – we apply the construct of Data Store. Annotations give any additional textual information to the process or its components.

To extend the language functionality we bring in two more individual classes – Vulnerability Point and Lock. Vulnerability Point is a property of a Task or a Data Store and points out the place of a system weakness. Lock is used to express the constraint of valuable business with respect to security criterions such as *integrity*, *confidentiality* and *availability*. Lock has a value attribute which indicates what we are dealing with: constraint is expressed with a lock symbol, the broken security criterion is depicted with an unlock sign.

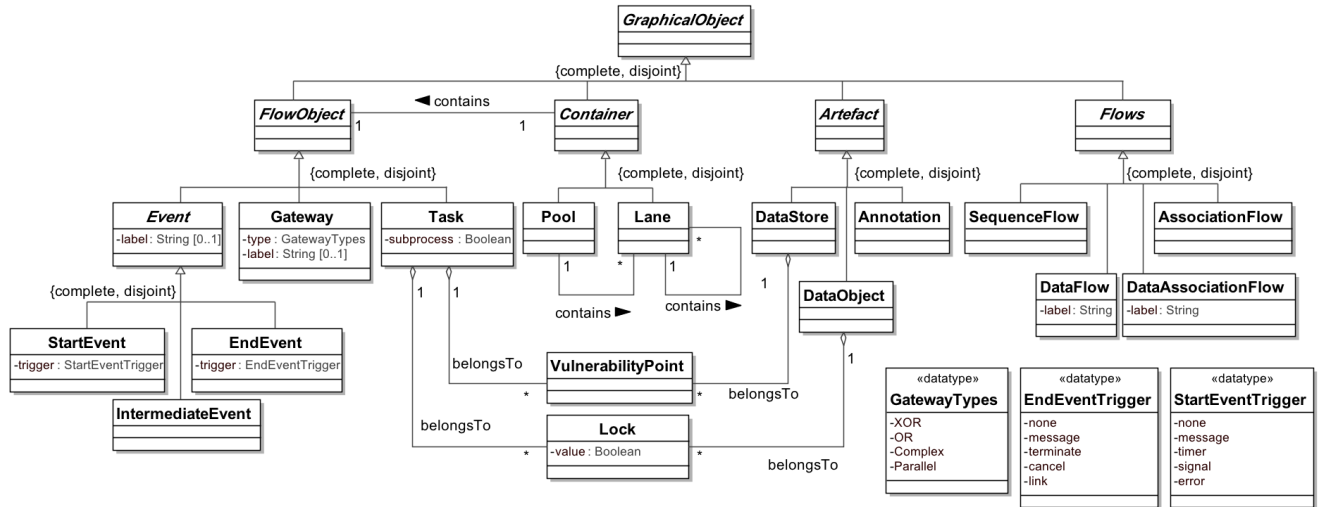


Figure 11. BPMN Extended Abstract Syntax: Concept Classification

Relationships between the Flow Objects are defined with the help of flows, including Sequence Flow, Data Flow and Data Association Flow (see Figure 2 and Figure 3). To link together BPMN Tasks, Gateways and Events within the single Pool we use Sequence Flow. Data Flow shows the communication between Pools, which depict the input and the output of process resources. And finally Data Association Flow links together Tasks and Artefacts, Vulnerability Point and the text Annotation and also the Lock with an Annotation.

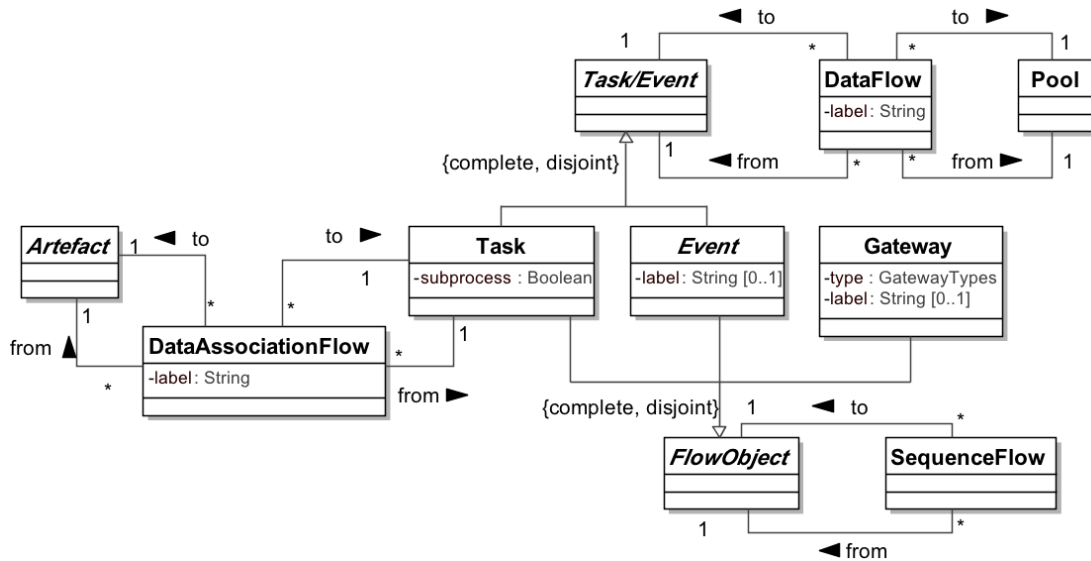


Figure 12. BPMN Extended Abstract Syntax: Relationships

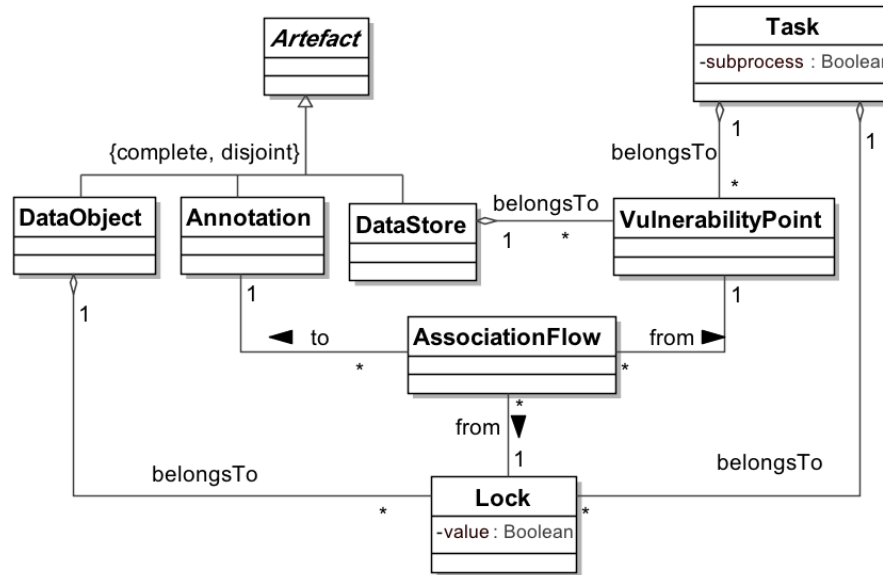


Figure 13. BPMN Extended Abstract Syntax: Relationships

5.4 Examples

In this Chapter we will introduce our running example modelled using extended BPMN. In order to investigate security risks in online Internet Store system, more detailed in message handling process, we will follow the ISSRM process (Meyer, 2009; Dubois et al., 2010).

5.4.1 Confidentiality

Context and asset identification. The process starts when potential User (Pool User in *Figure 14*) sends a message with a request to get the registration details to have right of using the Internet Store. After the message is sent, it is being accepted and registered by the system (Task Register received message). The administrator (Lane Administrator) accepts the message by opening it (Task Accept message), then he reads and writes the answer (Tasks Read message and Prepare answer). At the next step the answer with request to register on the webpage is being sent back to the user (Task Send out answer + message triggered End Event).

Determination of security objectives. The part of the process takes place on the system software, which helps to organize activities for accepting and sending the message back. The system database is also placed on the server side, so the purpose is to ensure the confidentiality of the stored data, i.e. usernames and passwords. In case of the *confidentiality* is revealed by the Violator, he can use the data for unintended purposes.

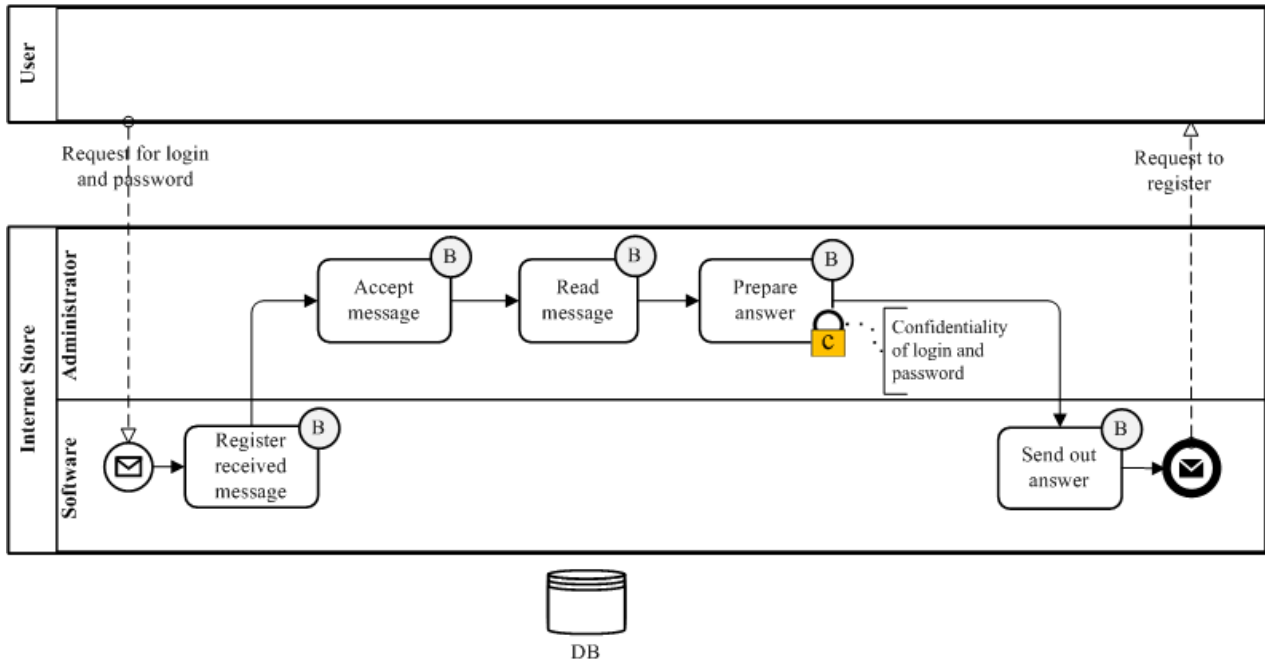


Figure 14. Message Handling Process: Asset Identification

Risk Analysis and assessment. *Figure 15* represents a potential risky scenario. Let's imagine that we have a third party this time, the violator (Pool Violator), who wants to skip the registration process and use the existing account data to enter the system. Just the same way as a custom user, the violator sends a message with request to the system administrator. But this time the spy malicious code is attached to the message (Message Flow Request for login and password (containing spy program)). As usually, the system registers the message, by keeping it on the server. Then administrator performs his usual activities. He opens the message, but this time the execution of the spy program starts in parallel. Just like in the normal flow, administrator goes on with reading the message and preparing the answer. Meanwhile, the malicious code initializes a new task; it extracts the data from the database (Task Extract username and password). By sending the inquiry to the database it gets the username and password of some existing user. The information then is attached to the answer message (Data Association Flow and Data Object Login and password), which is sent back to violator. We identified a number of weaknesses in the current system. First of all, registration of messaged is accomplished without any previous scanning (Vulnerability point + Annotation Message is handled without scanning). Moreover, the database control is not reliable; it can be accessed without proper rights (Vulnerability point + Annotation DB access is not controlled). And finally, the outgoing traffic control is not implemented, which leads to loss of valuable data (Vulnerability point + Annotation Outgoing traffic is not monitored). Combination of elements that represent the ISSRM *Threat agent* and *Attack method* (e.g. message containing a spy program, Tasks Start spy program and Extract username and password) forms a Threat. The latter leads to an impact: the *confidentiality* of usernames and passwords (Unlock with 'c' on Data Object).

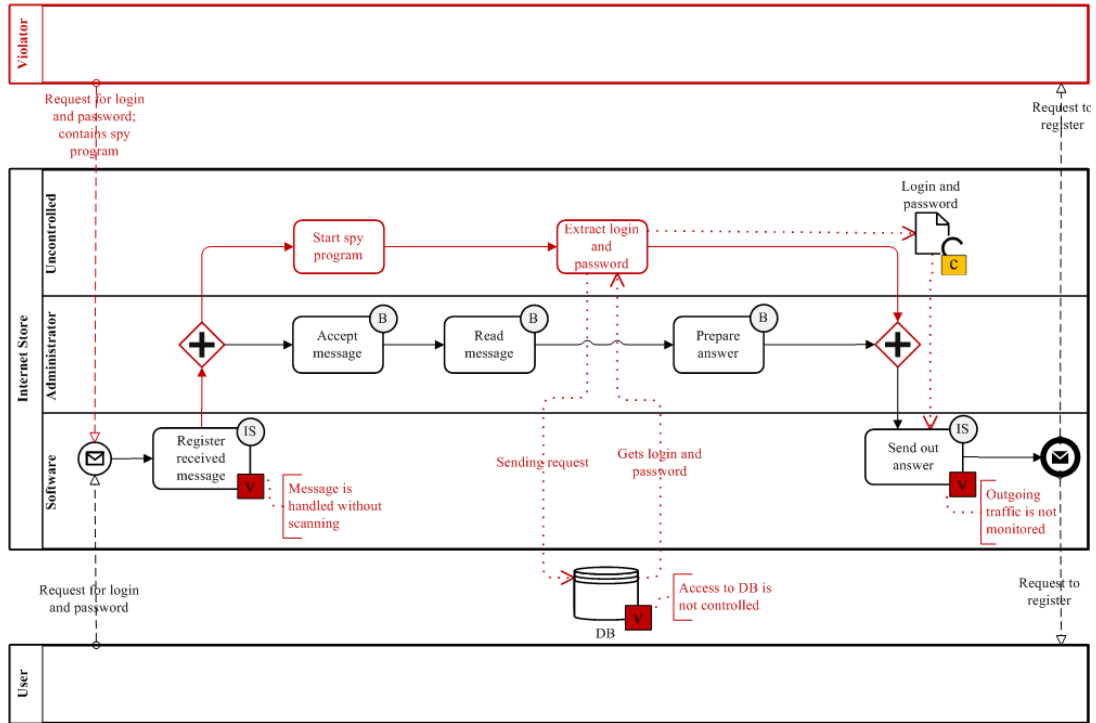


Figure 15. Message Handling Process Including Security Risk Attack

Risk treatment involves deciding how the identified security flows could be mitigated. In current situation we choose a *risk reduction decision* i.e., actions to lessen the probability of the negative consequences.

Security requirements definition. We implement the activity of message scanning to reduce the probability to receive an infected message. If the message scanning (Task Scan incoming message + Gateway Secure?) doesn't show any kind of insecurity, it's being registered into the system. Otherwise the process is cancelled and message is not accepted. The next security requirement is the control of database access and occurring activity (Task Control DB activity). If there is a try of unauthorized access during the process, it terminates. And the last required implementation is the activity of the traffic control (Task Control outgoing traffic) in order to get an information about the exchanged resources. If the check shows a problem with traffic, the operation is stopped and the process is cancelled (Cancel triggered End Event).

Control implementation. At current stages of system analysis we don't propose any control implementation; it remains for the future system development phase.

5.4.2 Availability

Context and asset identification. In this example we introduce new situation, the process of service delivery, performed in the Internet store (see Figure 17). Let's imagine we have a User (pool User), who sends a request for some services (message flow Sending request) that Internet Store provides. We do not talk about what services exactly user is requesting due to the reason we are interested in the process

flow and not concentrating on details. The normal flow includes receiving the request by a server (task Listen for request), if the check reports some problems, server sends an error message to the user (error triggered event + message flow Invalid request). Otherwise, the server provides requested service (task Provide services), which is a valuable asset of the Internet Store system.

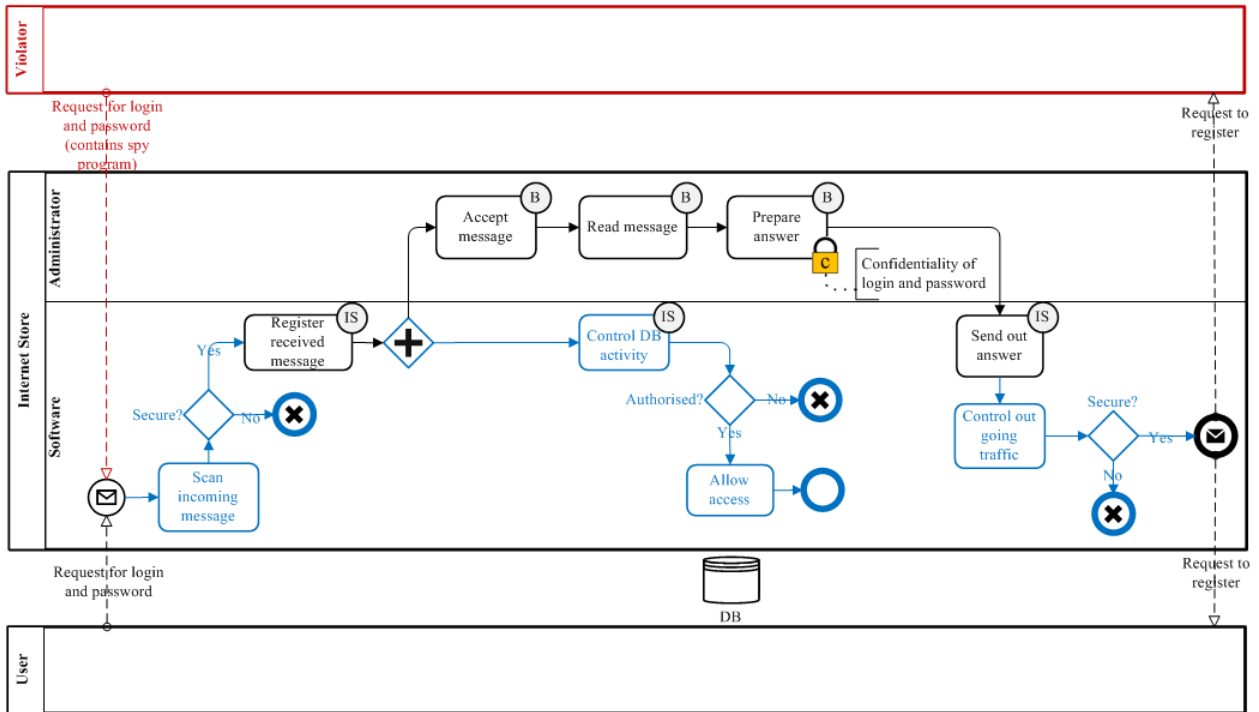


Figure 16. Message Handling Process Including Security Requirements.

Determination of security objectives. In current example we are emphasizing on the availability of the services provided by the server of the Internet store system.

Risk Analysis and assessment. The process including security risk is presented in Figure 18. We model the risky scenario by involving the attacker (pool Violator), who uses DoS (Denial of Service) attack to harm the system. He sends multiply requests to the system (message flow Sending multiply requests) provoking the inability of server to listen and cope with this number of requests at one time, what in turn leads to the server being hanged (end event Server is hanged) and services – unavailable for users.

Risk treatment. For the risk treatment we decide to take a decision of risk reduction that hopefully will reduce the probability of such a risk.

Security requirements definition. *Figure 19* represents the process of service providing including definition of security requirements. We implement the activity of controlling the incoming requests (task Control incoming request) which includes filtering and sorting and will prevent server from hanging. So the services will stay available for users.

Control implementation. We don't propose any control at this stage of system analysis. It remains for development stages.

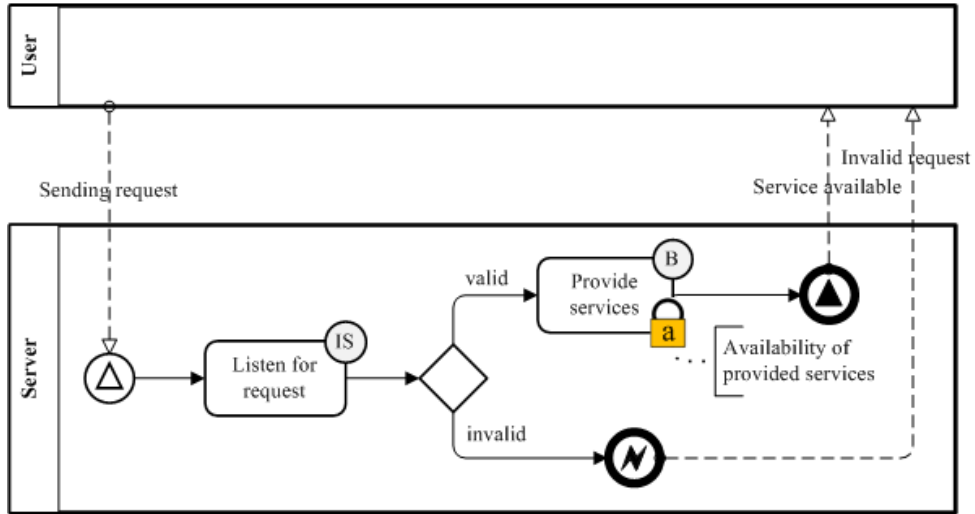


Figure 17. Service providing process; assets and security objectives

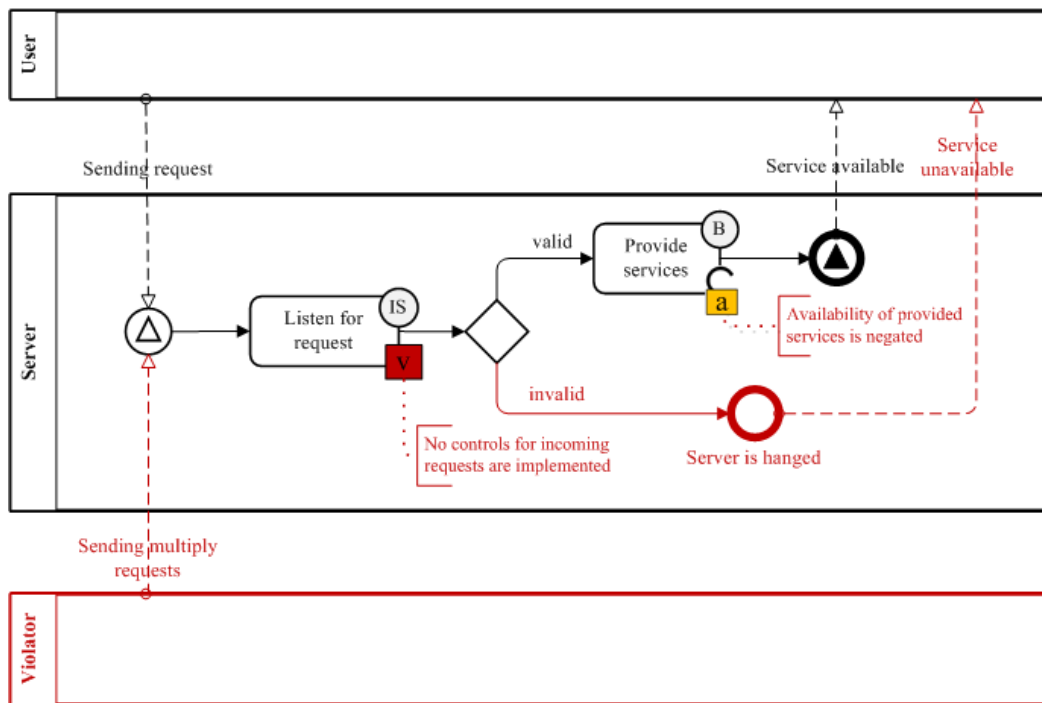


Figure 18. Service providing process; security risk

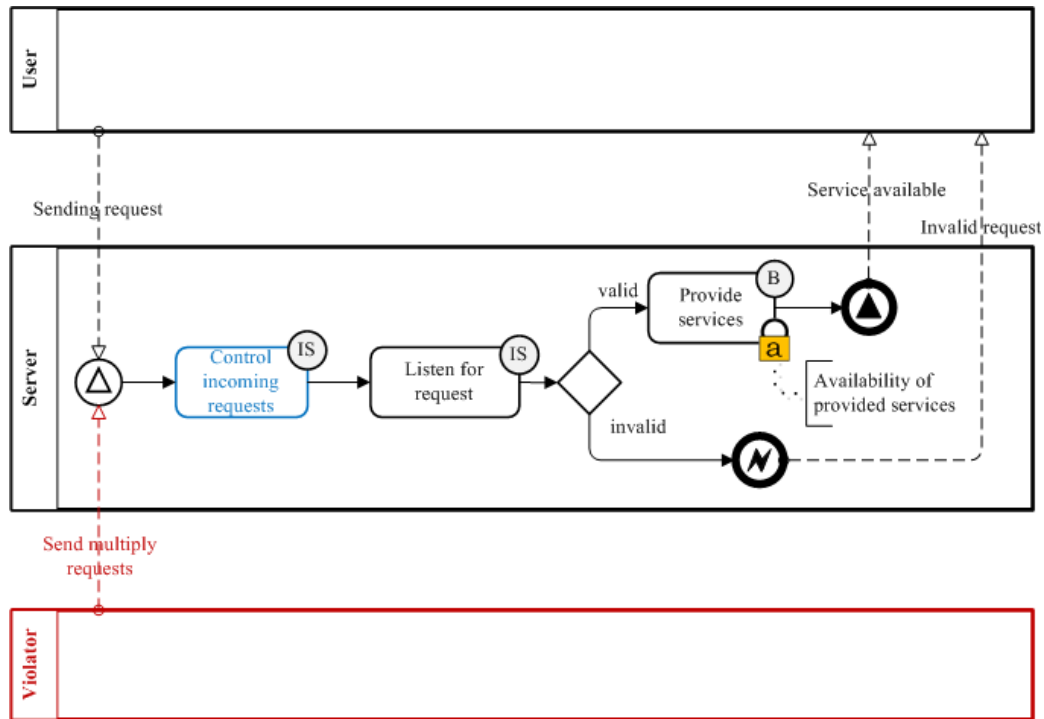


Figure 19. Service providing process; risk treatment

5.4.3 Integrity

Context and asset identification. Lets imagine the following situation when User (pool User) wishes to use the Internet Store system and sends a request for registration details to administrator. Administrator in his turn accepts and reads the message (task Accept message, Read message) and sends an answer (task Send answer) with a guidelines of how to perform the registration. The valuable asset of this process is a message handling process itself.

Determination of security objectives. It is considered that the process should include all the above mentioned activities and be performed exactly that way how it is depicted in *Figure 20*. So the security criterion we focus on is integrity of the message handling process.

Risk Analysis and assessment. *Figure 21* presents the potential security risk scenario. Let's assume that there exists a violator (pool Violator) who attempts to use the system like a custom user. He sends the similar request for registration, but this time it contains a malicious code (message flow Registration inquiry + spy malicious code). The execution of a spy program starts after message is accepted. The program extracts data from DB and then data is attached to outgoing message to be sent back to Violator (task Send answer, message flow Demand for registration + stolen data).

Risk treatment. For the risk treatment decision we choose risk reduction to lessen the probability of risk.

Security requirements definition. To reduce the probability of described risks some security requirements were introduced. More detailed view on this situation is presented in *Figure 16*.

Control implementation. Just like it was mentioned in previous example, the identification of the controls is postponed for later stages of system development.

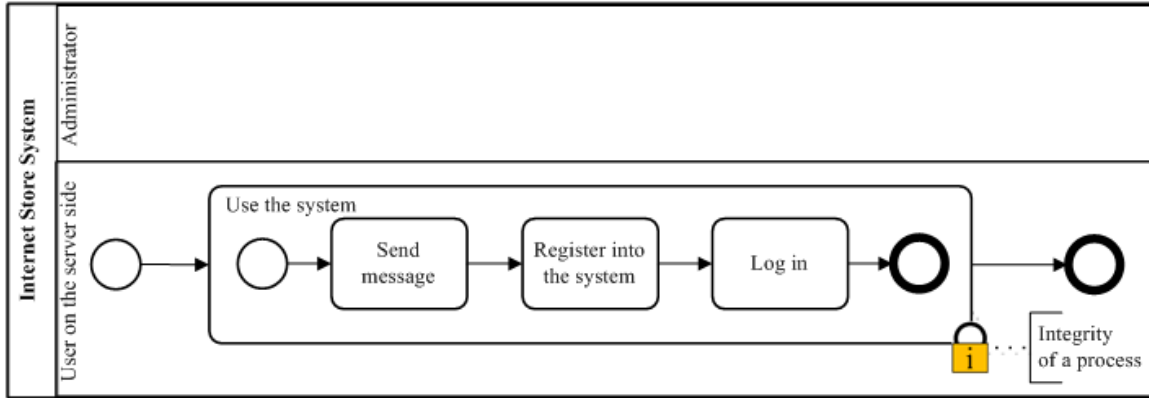


Figure 20. Process of logging in; assets and security objectives

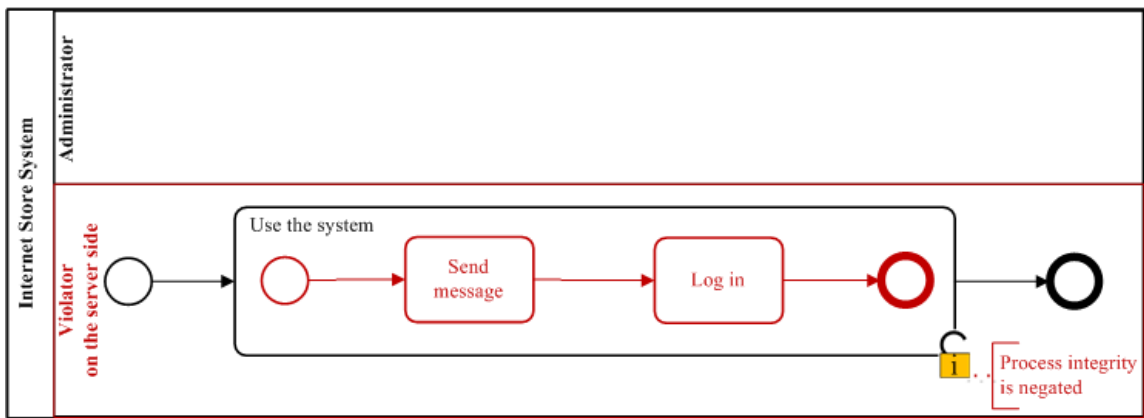


Figure 21. Process of logging in; security risk definition

5.5 Summary

In this Chapter we presented the overview of research method, the extensions on language concrete and abstract syntax and the resources developed in the process of this part of study. We result into three tables (*Table 3-5*) that give us a visual demonstration of ISSRM domain coverage after the BPMN syntax extension. We can see that Asset-related concepts and Risk-related concepts can be sufficiently addressed by security risk-oriented BPMN: major part of security concepts is now covered. We also demonstrate the applicability of proposed extensions of Internet Store example, covering three scenarios: integrity, confidentiality and availability analysis.

Chapter 6. Transformation rules from BPMN to Secure Tropos

In this chapter we introduce model transformation opportunity from extended BPMN to Secure Tropos. The goal of current investigation is to allow security and business analyst to elicit risk in a systematic way from different point of view, broaden the horizons for specifying security requirements, reason what is the trade-off, and benefit the security requirements regarding the analyzed system. In other words, we investigate into providing the stakeholders new opportunities for secure system design. As an input for this stage of analysis, we take models previously designed with the help of extended version of BPMN, covering confidentiality, integrity and availability. We establish a comprehensive set of transformation rules from BPMN to Secure Tropos. And finally we analyze the completeness of transformed model with respect to security risk management. More specifically, we make the transformation alignment that helps defining what part of the model can be transformed and pointing onto losses of any valuable artefacts.

6.1 Transformation rules

Analysing the opportunities for transformation of models from BPMN to Secure Tropos, we designed a set of comprehensive transformation rules (Step by step transformation example is presented in the *Appendix A*). However, not everything can be transformed from one language to another, so number of manual changes have to be performed. As an input for transformation process we take models, designed in earlier chapters with the help of extended version of BPMN 2.0. The set of defined rules is presented below.

6.1.1 Rules for asset definition

TR1. Define the stakeholders. Transform BPMN *Pools* and *Lanes* to Secure Tropos *Actors*. BPMN *Pool/Lane* that represents a process participant (e.g. job, name, system) is simply transformed to *Actor*. If BPMN *Lanes* are used as a representation of different functional parts of one working system, there is no need to transform each part, transforming the general naming of described system (often *Pool* name) will be essential in Secure Tropos.

TR2. Another characteristic of Secure Tropos is ability to define dependencies between *Actors*. For dependencies definition transform each BPMN *Task* that represent any activity of communication between participants (e.g. send, receive, request, get, or any other of that kind) to Secure Tropos *Plan*, which will become a dependum. Dependencies can be also represented by BPMN *Message flow* with the same mission: to send something, request or receive. Transform *Message flow* to Secure Tropos dependums.

TR3. Next step is to add a direction to previously defined dependencies; to decide who is a depender and who is a dependee. A depender in BPMN is usually somebody who performs the action; a dependee is the one who depends on this action. In case with the transformation of *Message flow*, the direction of dependency is specified with direction of observed *Message flow*; if it goes out from the *Pool* that means we are dealing with a depender, and conversely.

TR4. Define the *Security constraint* that will regulate the dependency between *Actors*. *Security constraint* that depender expects to be satisfied, is transformed from BPMN Security objective.

TR5. Transform BPMN *Tasks* to Secure Tropos *Plans* and add them to the corresponding Actor's boundary; add only *Plans* that are under responsibility of the observed Actor.

TR6. Next step is dedicated to completion of Secure Tropos Model with *Goals*. *Goals* can be formed from already transformed BPMN *Tasks*, meaning from *Plans* added on the previous step, or from BPMN *Tasks*. Transform Secure Tropos *Goals* in the following form of event: Task Request item list to *Goal* Item list requested.

TR7. Duplicate the *Security constraint* to the boundary of an *Actor*. Add *Restricts* relationship between *Security constraint* and the *Goal*. This relationship should correspond to the union of BPMN *Tasks* restricted with a *Lock* defining *Security criterion*.

TR8. Define the *Softgoal* and add it to the *Actor's* boundary. The *Softgoal* should comply with *Security objective*, which is a property of the BPMN *Lock*. Add *Contribution* relationship between *Security constraint* and the *Softgoal*.

TR9. Add *Resources* to Secure Tropos model. *Resources* are transformed from BPMN *Data Object* or *Data Store*, in other words BPMN *Artefacts* can be transformed to Secure Tropos *Resources*.

As it was mentioned before, Secure Tropos model should be complemented with some manual additions. Define the rest of relationships, such as *Decomposition* and *Means-ends*.

6.1.2 Rules for risk identification

Next stage of transformations is creating Secure Tropos Risk model from BP Diagram. The latter comes as an input.

TR10. Start from defining the impact of the risk described in BPMN model. Add Secure Tropos *Threat* and the *Impacts* relationship between *Threat* and corresponding *Security constraint*.

TR11. Basing on the rule **TR1**, define the attacker as a new Secure Tropos *Actor* with its boundary.

TR12. Transform BPMN *Tasks* (in red) to Secure Tropos *Plans* that represent an attack method. The following rule is based on **TR5**. It can be also noticed that we do not define any dependencies between Agent and other Actors, so BPMN *Message flow* from the *Pool* that represents a threat agent can be transformed to Secure Tropos *Plan*.

TR13. Define the main goal of the attacker from BPMN diagram; it can be formed from Task or Event (usually some undesirable end event). Once it is defined, transform it to Secure Tropos *Goal*.

TR14. Next step is to detect the *Vulnerability point* from BP Diagram. Secure Tropos *Vulnerability point* should be added to the *Plan (Goal)* or *Resource* that correspond to *Task* or *Artifact* that carried the vulnerability in BPMN model. Put *Exploits* relationship between *Plan* (representing attack method) and *Vulnerability point*. Add the *Attacks* relationship between the *Plan* and *Resource* being attacked.

Define the rest of relationships, such as *Decomposition* and *Means-ends* in the boundary of an attacker.

6.1.3 Rules for risk treatment solution

As an input we take asset model in Secure Tropos. Transformation is based on BP Diagram, describing treatment options.

TR15. Transform BPMN *Tasks* and *Gateway* structures that represent security requirements into Secure Tropos *Plans* and *Goals*, basing on rules **TR5**, **TR6**. Add (*S*)-labels to all defined *Plans*, *Goals* that will emphasize security requirements.

TR16. Define additional *Security constraint(s)*. It is possible through transforming the BPMN *Gateway* structures that represent security requirements control.

TR17. Add *Satisfies* relationship between defined *Goal(s)* and existing *Security constraints*.

TR18. Collapse the risk scenario to an *Impact*. Add *Mitigates* relation between *Security constraint* and the *Impact*.

Add missing relationships (*Decomposition*, *Means-end*) manually to complete the model.

6.2 Transformation examples

6.2.1 Availability

As an input for transformation we receive **Figure 22**, which represents message handling process in Internet Store and modelled with extended version of BPMN. Following the rules **TR1-TR3** we define *Actors* and dependencies between them. We decide upon the dependums and emphasize such Secure Tropos *Plans* as Request item list, Provide (display) item list, Select goods from Internet Store that are transformed from BPMN *Tasks*. The decision is based on understanding that current activities represent interaction between process participants as defined in the rule **TR2**. The direction of dependencies is defined as follows; as User requests item list and selects goods, he (she) is a depender, and on the contrary – a dependee on the *Plan* Provide item list. Then we define the *Security constraint* by transforming the Annotation Availability of item list to Secure Tropos *Security constraint* and adding it to the corresponding dependency and to the boundary of Internet Store (see **TR7**). The security objective depicted in the *Lock* is transformed to *Softgoal* and connected to defined security constraint with *Contribution* relationship. BPMN *Tasks* located in the Internet Store *Pool* are transformed to Secure Tropos *Plans* and *Goals* with respect to rules **TR5** and **TR6**. BPMN *Data Object* Item list is transformed to *Resource* basing on the rule **TR9**. The result of transformation is presented in **Figure 23**. It is also important that Secure Tropos presents goal-oriented model not the process flow, which means that not everything can be transformed from BPMN. To make the model complete, we add necessary relationships, and here we rely mostly on intuition and example context. We start with defining the major goal and put *Means-end relationship* between the *Goal* and corresponding *Plan*. We also put

Decomposition between Plans and Goals, defining the hierarchy. The version of complete model is presented in *Figure 24*.

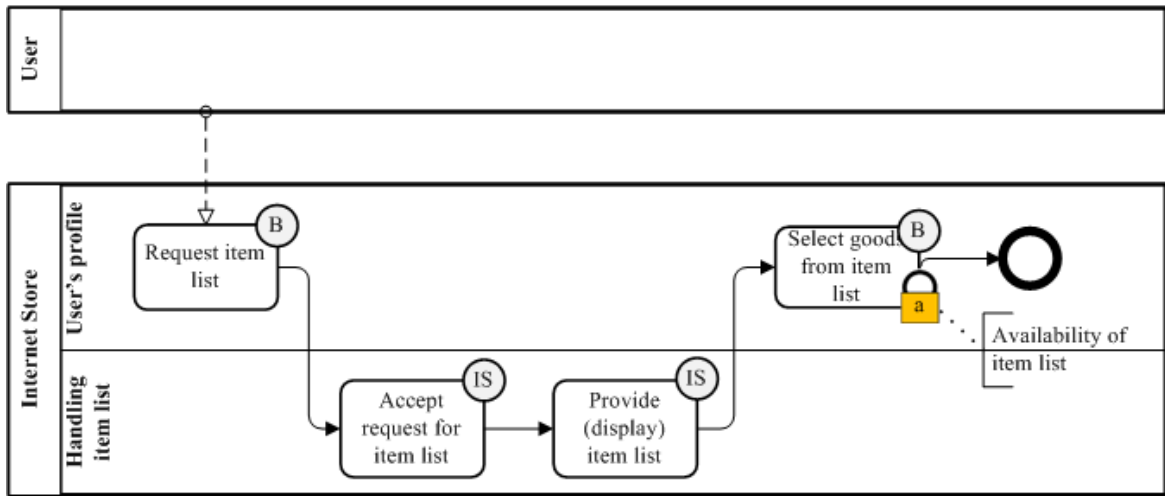


Figure 22. BPMN: Availability analysis - Message handling process assets

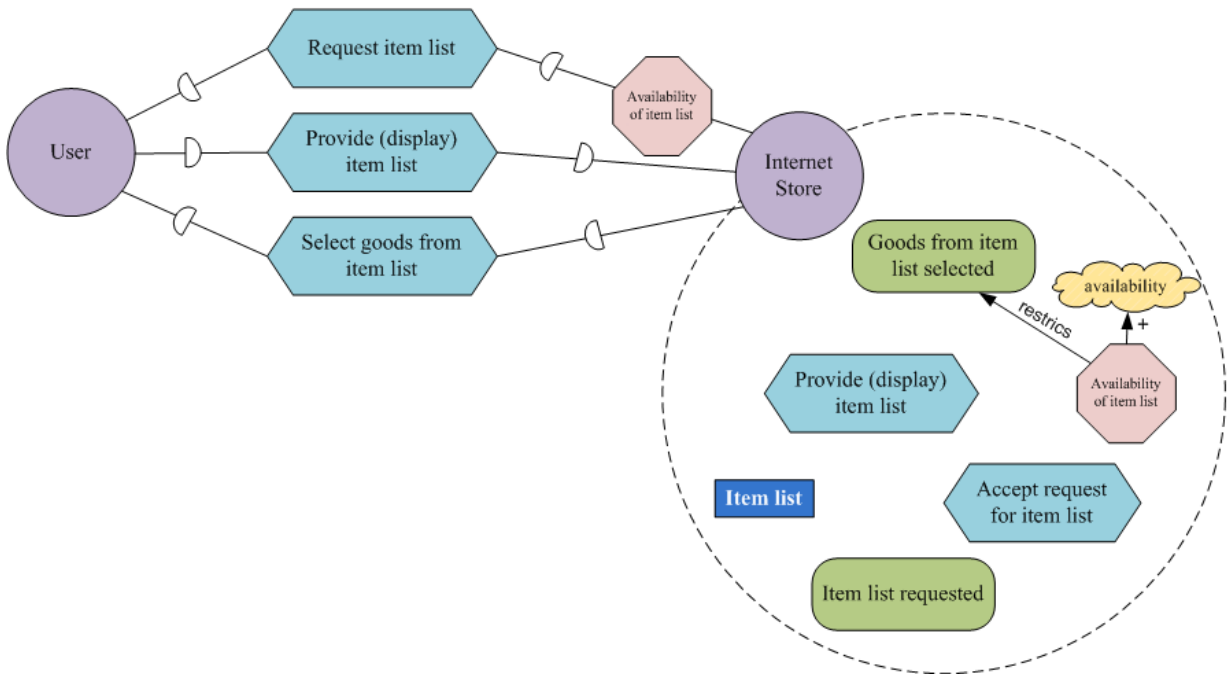


Figure 23. Secure Tropos: Availability analysis – assets

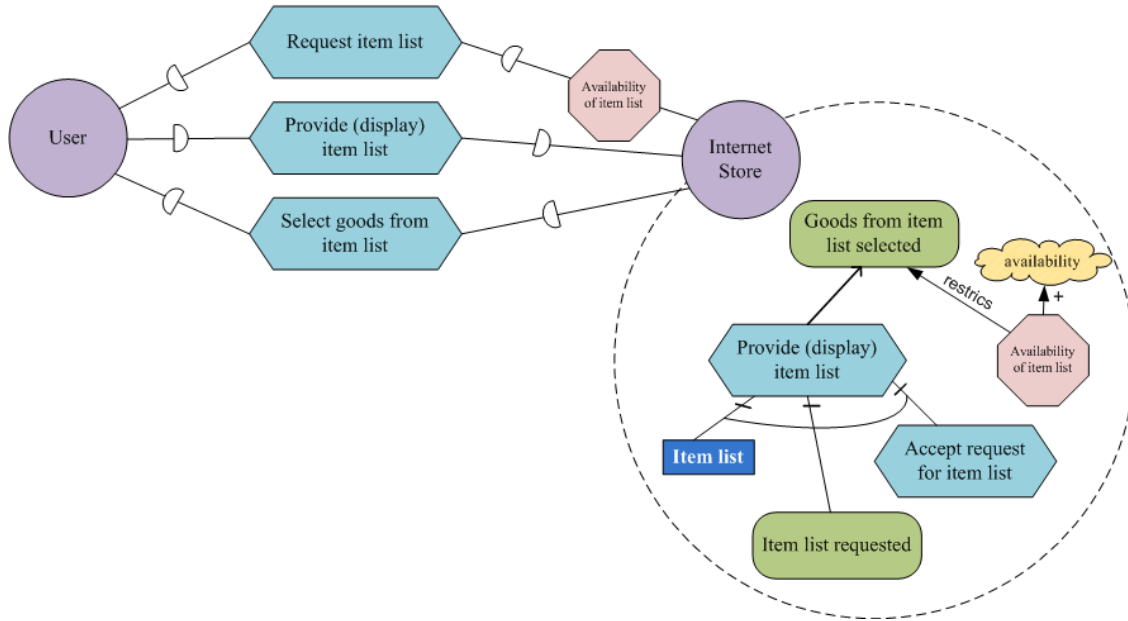


Figure 24. Secure Tropos: Availability analysis - assets (complete)

Next step is dedicated to risk definition and we take **Figure 25** as an input. We continue working with asset model (**Figure 24**) and add an *Agent Violator* and a new boundary. Following the rule **TR12** we define the attack method by adding the *Plan Request item list* multiple times and transforming the *error-triggered End Event* to the *Goal Internet Store is not capable to handle all requests*. Next we add a *Vulnerability point* to the *Plan Accept request for item list* (complying with the BP Diagram). We also add *Exploits* and *Attacks* relationships as it is defined in rule **TR14**. The end result of transformations is presented in **Figure 26**, after some manual add-ons, the final model can be interpreted like depicted in **Figure 27**. However in Secure Tropos there exist an opportunity to represent the defined risk using a *Threat* construct, the following representation is based on the rule **TR10** and can be seen in **Figure 28**.

The final step for Availability example transformation is to describe the treatment scenario – as an input we take BPMN model presented in **Figure 29**. Basing on the asset model (**Figure 24**) and the rule **TR15** we add the *Plan Check for abnormal request* to Internet Store boundary. We handle the structure for Gateway Normal? and define the *Goal Abnormal request terminated*. On this basis, we identify the new *Security constraint* Only normal request allowed (See rule **TR16**). However this security constraint is not transformable from BP Diagram, we consider it to be essential for Secure Tropos model. We add *Satisfies* relationship between the *Goal* and *Security constraint*. The result for above described transformation is presented in **Figure 30**. Manually we add relationships, and following the rule **TR18** define an impact and *Mitigates* relationship. The final result is presented in **Figure 31**.

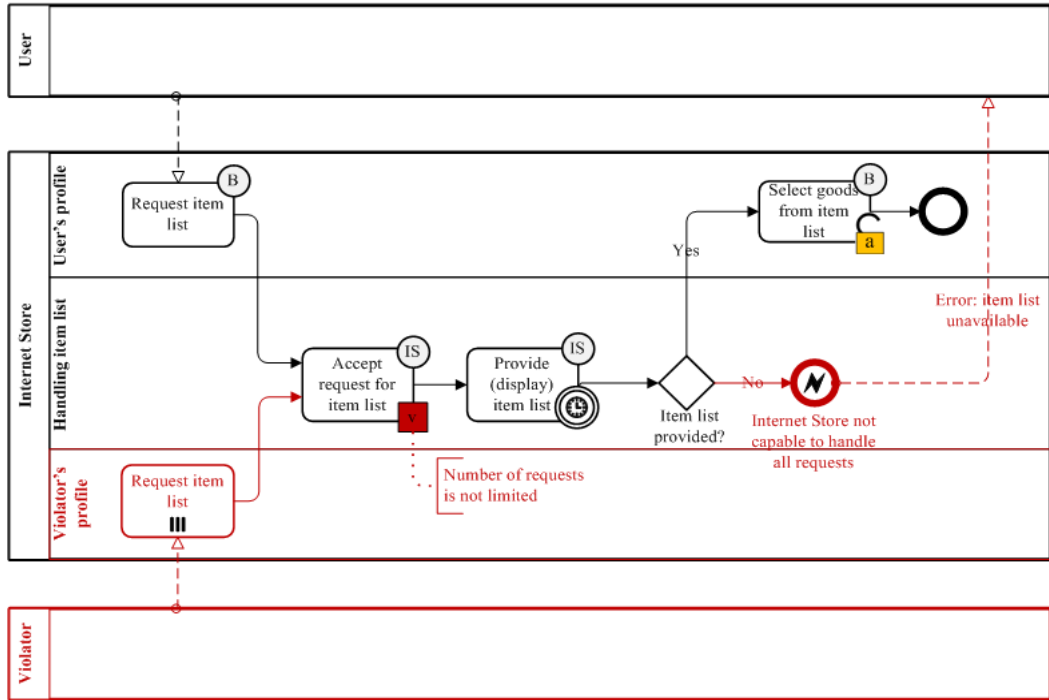


Figure 25. BPMN: Availability analysis - Denial of service risk

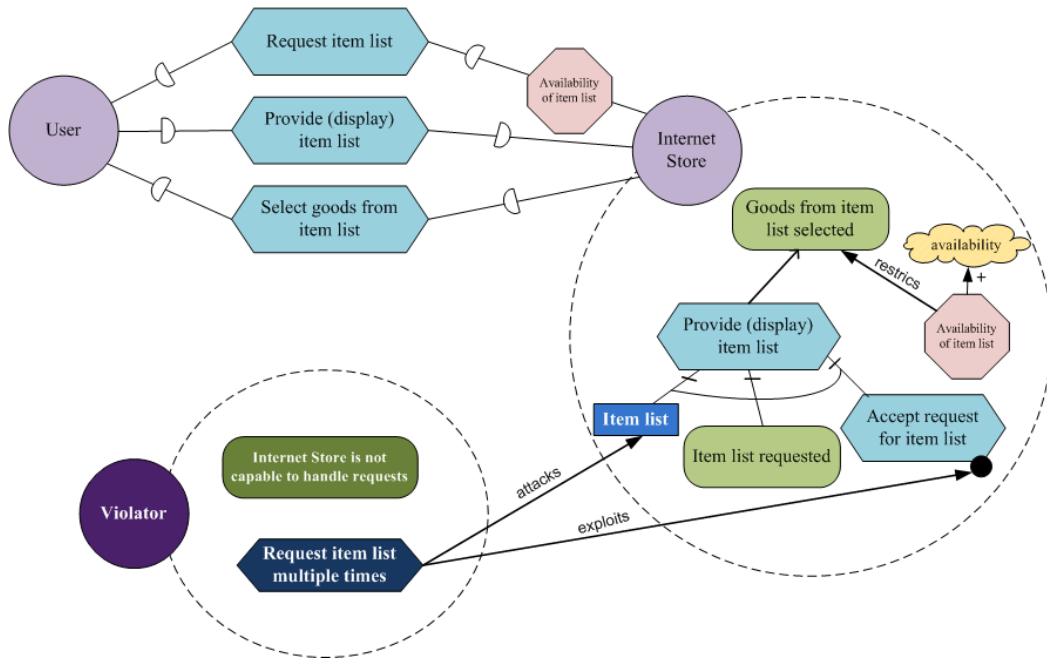


Figure 26. Secure Tropos: Availability analysis – Risk

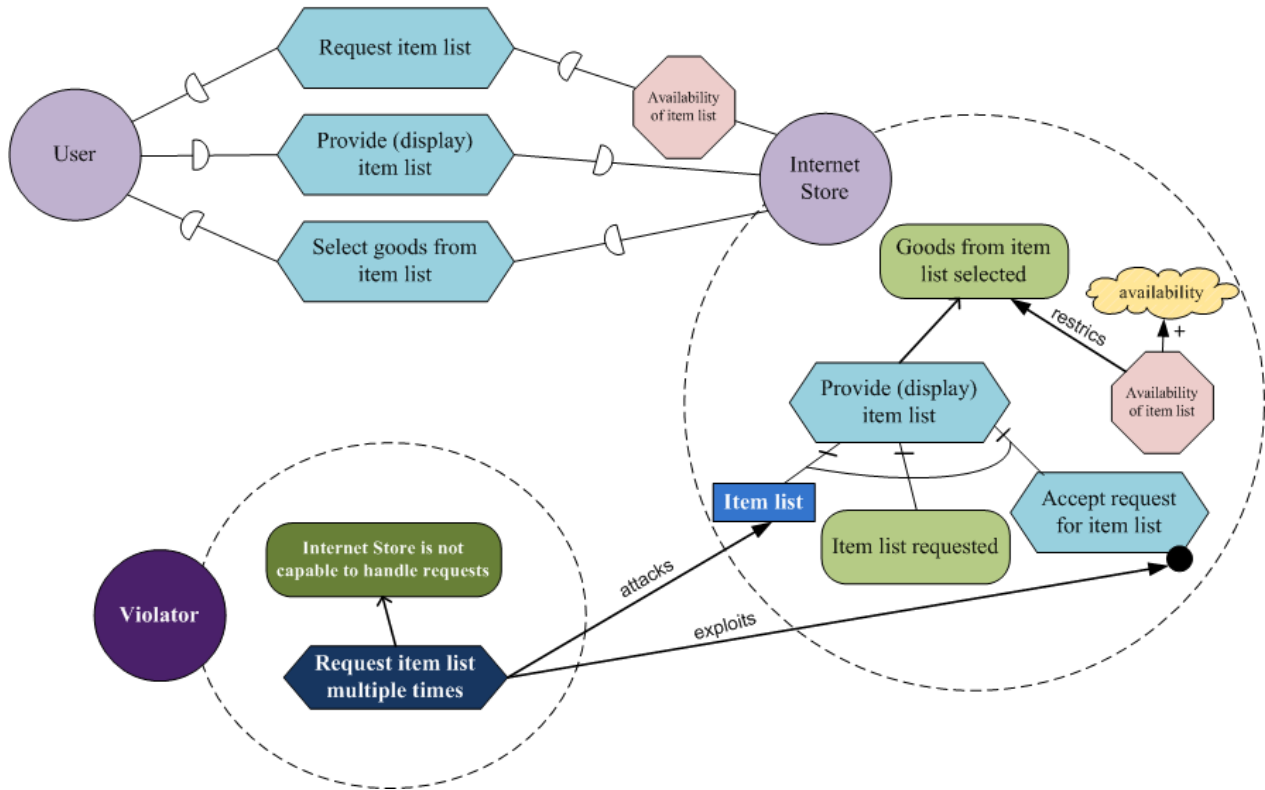


Figure 27. Secure Tropos: Availability analysis – Risk (complete)

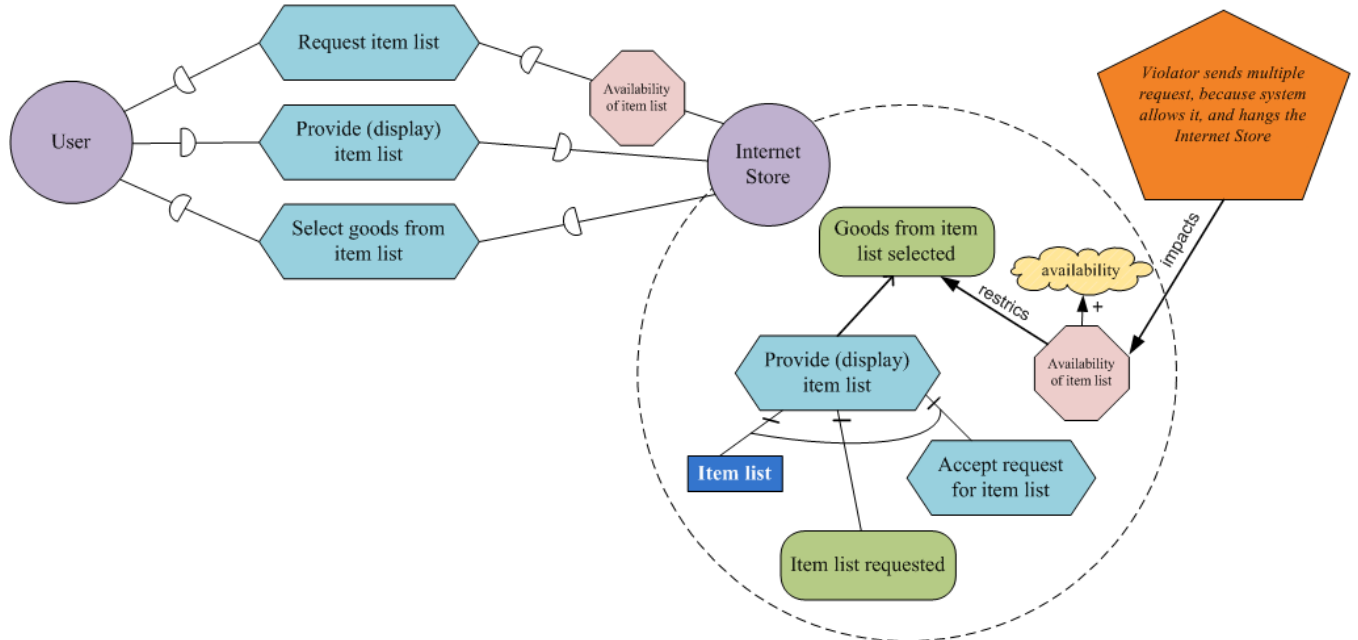


Figure 28. Secure Tropos: Availability analysis – Risk (Threat construct)

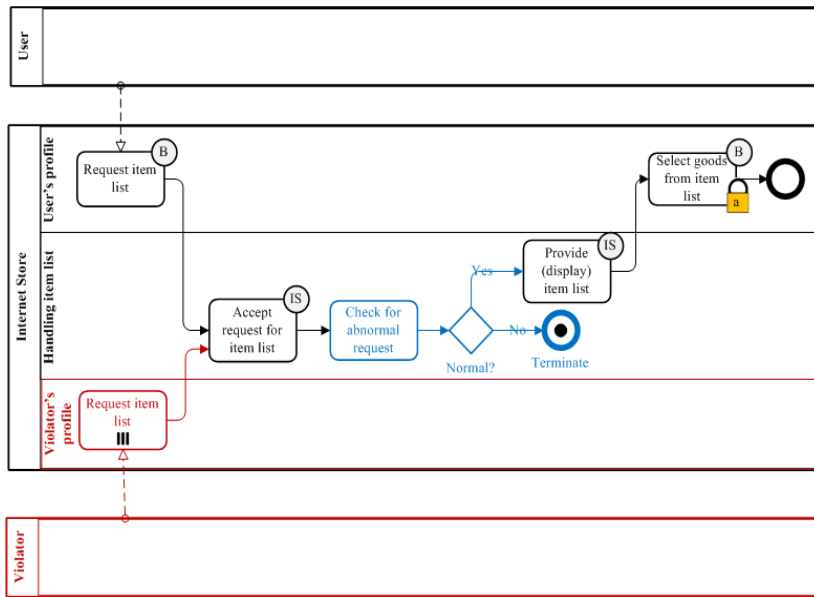


Figure 29. BPMN: Availability analysis - security requirements

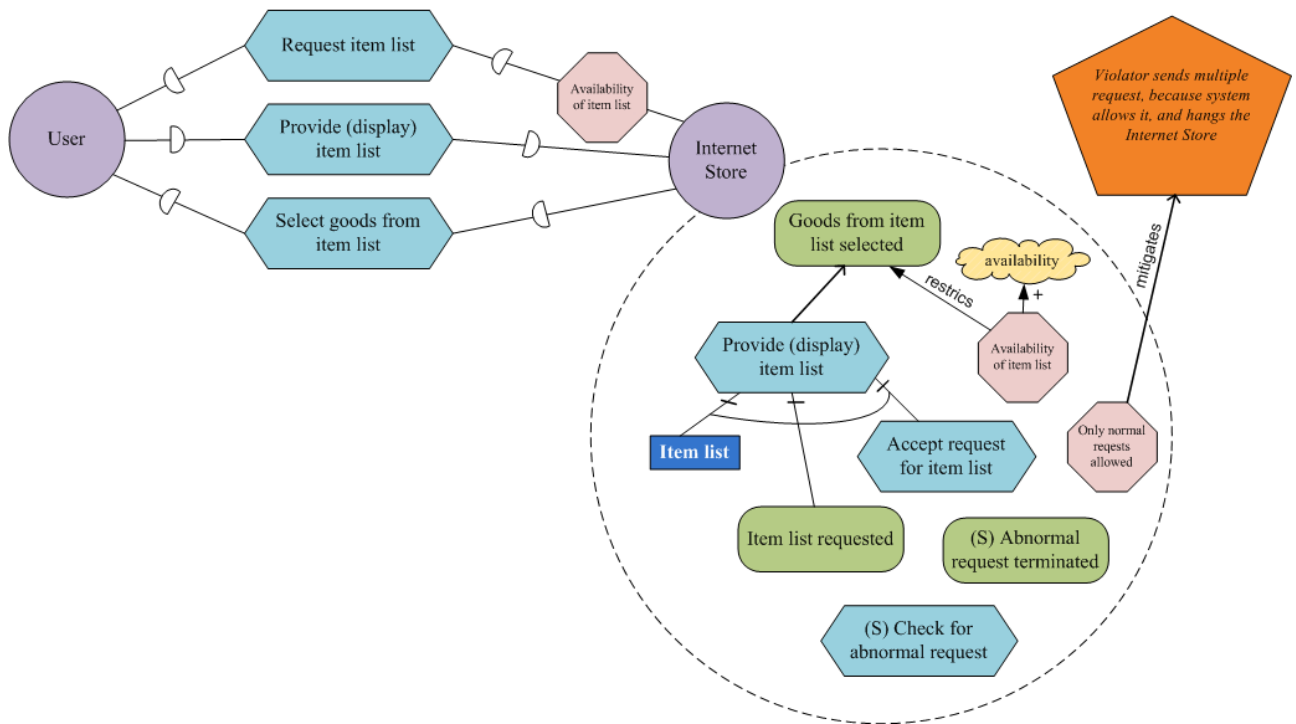


Figure 30. Secure Tropos: Availability analysis – security requirements

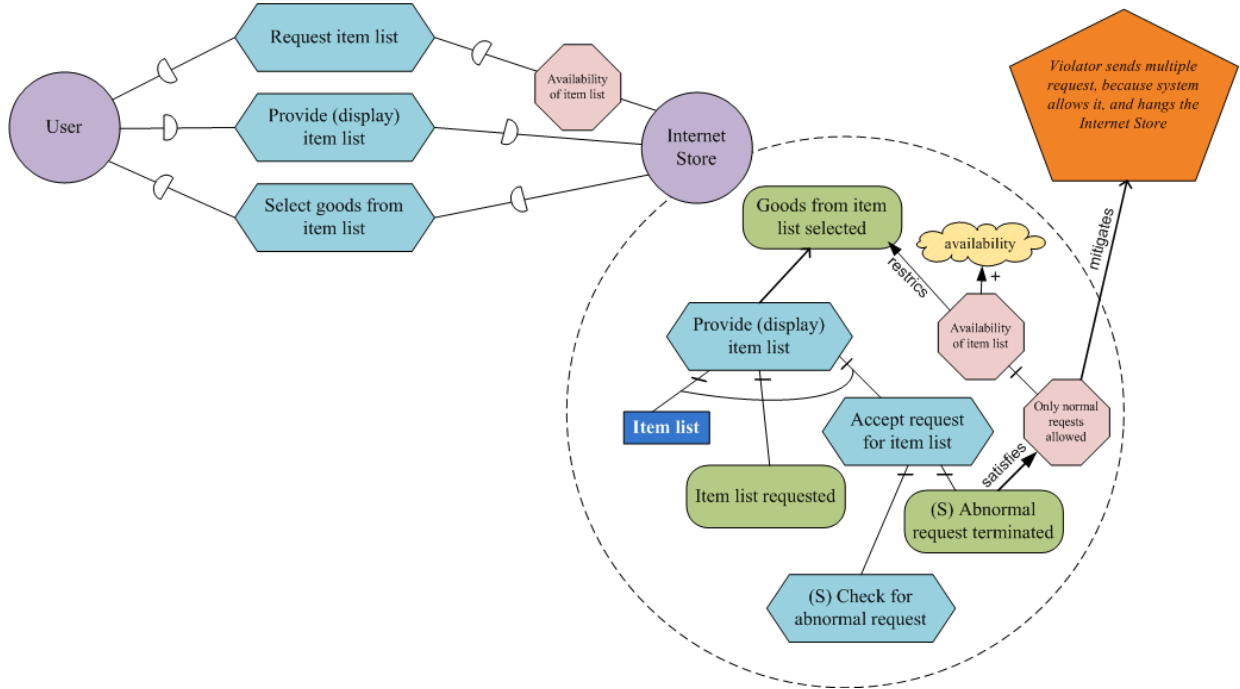


Figure 31. Secure Tropos: Availability analysis – security requirements (complete)

6.2.2 Confidentiality

The transformation of confidentiality example starts from observation of two BP diagrams: one represents the message handling process in the Internet Store (see **Figure 32**) and another depicts user registration process (see **Figure 33**). As both of the diagrams have value with respect to confidentiality analysis, we transform output from both diagrams to create a Secure Tropos model.

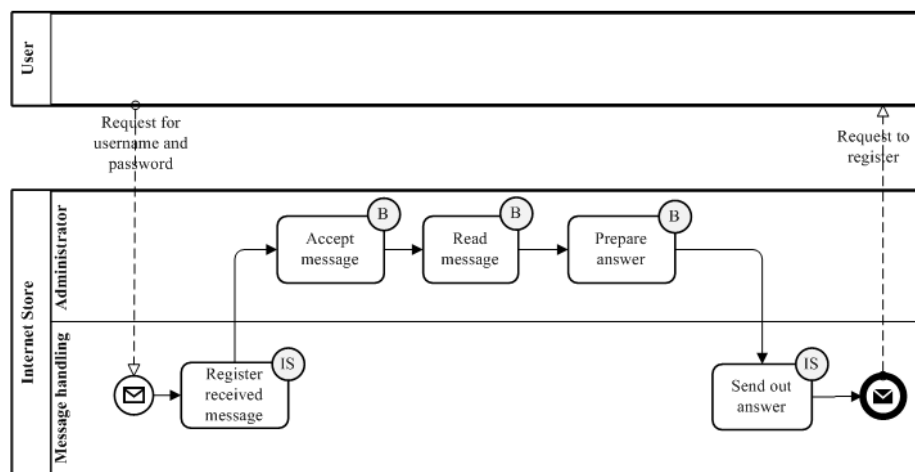


Figure 32. BPMN: Confidentiality analysis – Handle request message

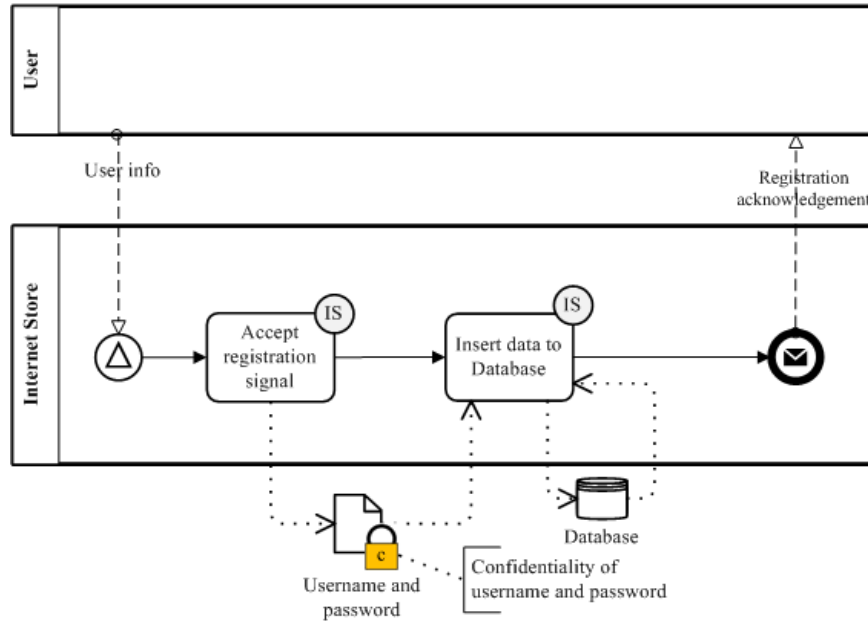


Figure 33. BPMN: Confidentiality analysis – User registration process

BPMN *Pools* are transformed to Secure Tropos *Actors* according to transformation rule **TR1**. To define dependencies between User and Internet Store, we follow the rules **TR2** and **TR3** and transform BPMN *Message flows* to the following Secure Tropos *Plans*: Request for username and password and Request to register (from **Figure 32**), as well as Send user info and Send registration acknowledgement. The defined dependencies are regulated by Confidentiality of username and password *Security constraint*, which is transformed from corresponding BPMN *Security objective*. Next we define *Plans* and *Goals* from corresponding BPMN *Tasks* according to **TR5** and **TR6**. We duplicate the *Security constraint* into *Actor's* boundary and continue transformation with adding the *Softgoal*, translated from BPMN *Lock* and *c-label* inside. We supplement the model by adding supporting relationships with respect to rules **TR7** and **TR8** and the following Secure Tropos *Resources*: User info and Database (transformed from corresponding *Data Object* and *Data Store*). At this stage we receive a final asset model in Secure Tropos (see **Figure 34**). After performing some manual operations, similar to ones described in availability example, we receive the version of complete Secure Tropos model (see **Figure 35**).

As an input for creation of Secure Tropos risk model we take **Figure 36** that represents security risk in message handling process, modelled with extended BPMN. We add an *Agent* and his boundary to define the Violator. According to **TR12**, we transform BPMN *Tasks* Start spy program and Extract username and password to Secure Tropos *Plans*. *Message flow* Request for username and password is also transformed to Secure Tropos *Plan*. Next we decide on the main goal of threat agent: in current case it is described in the message flow Request to register + username and password copied from database. We transform it to Secure Tropos *Goal* Username and password are copied from database.

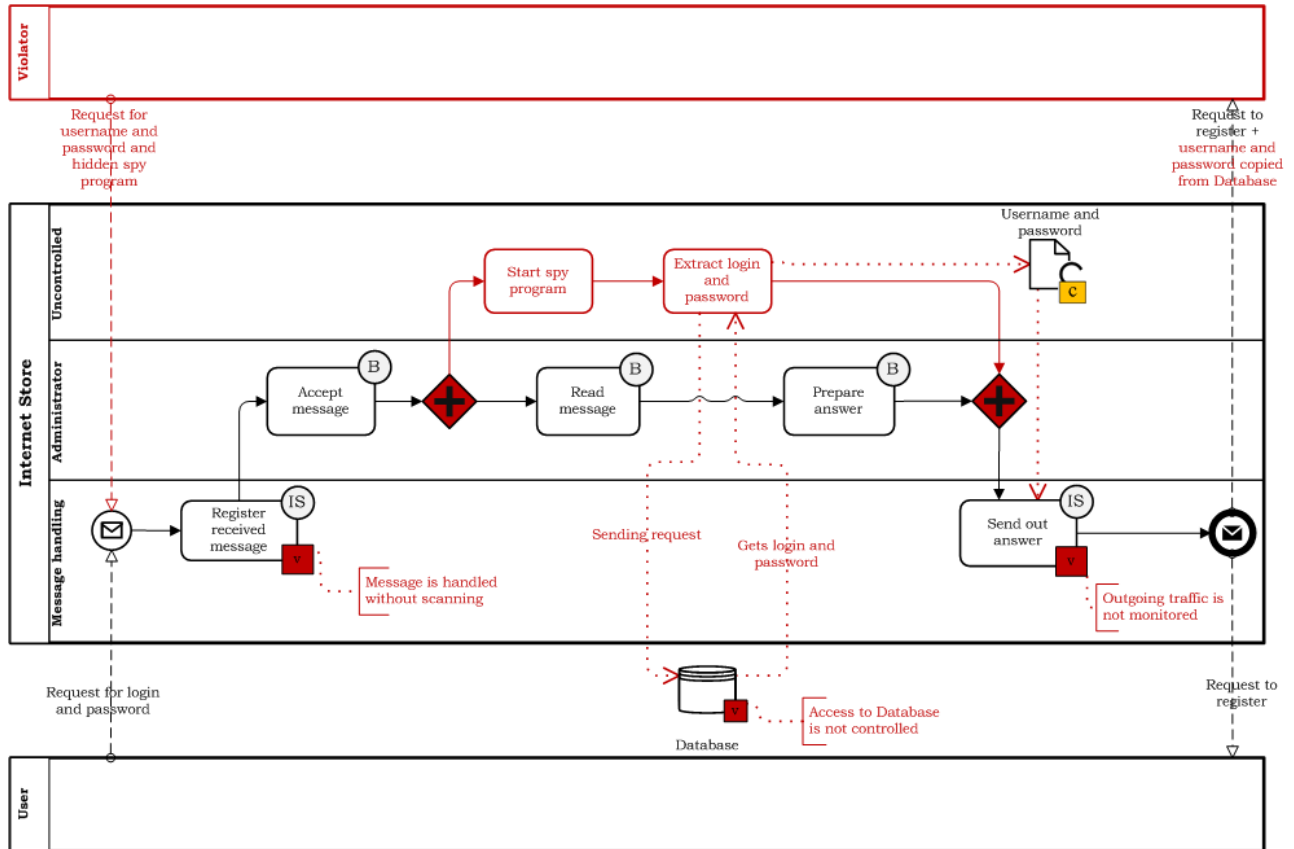


Figure 36. BPMN: Confidentiality analysis – Security Risk

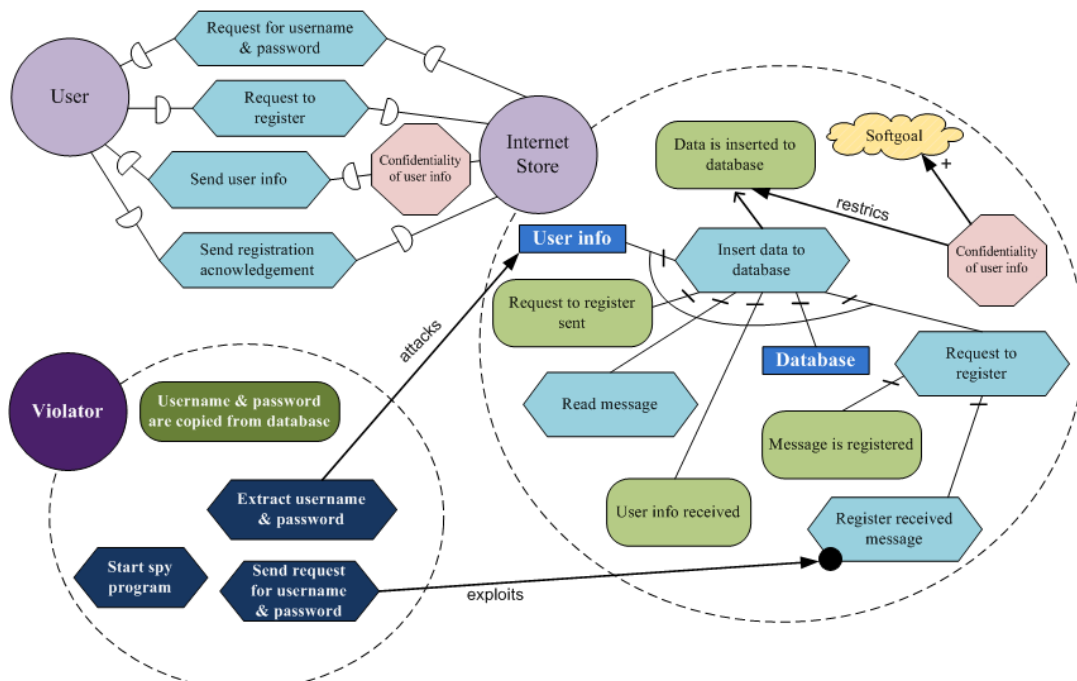


Figure 37. Secure Tropos: Confidentiality analysis – risk model

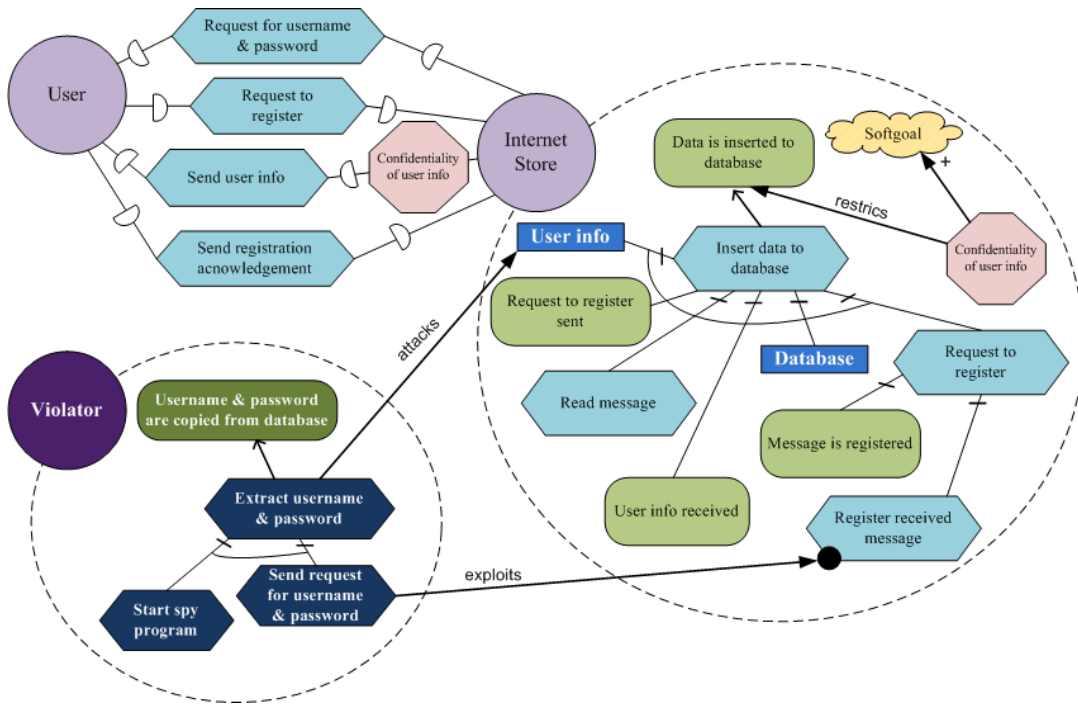


Figure 38. Secure Tropos: Confidentiality analysis – risk model (complete)

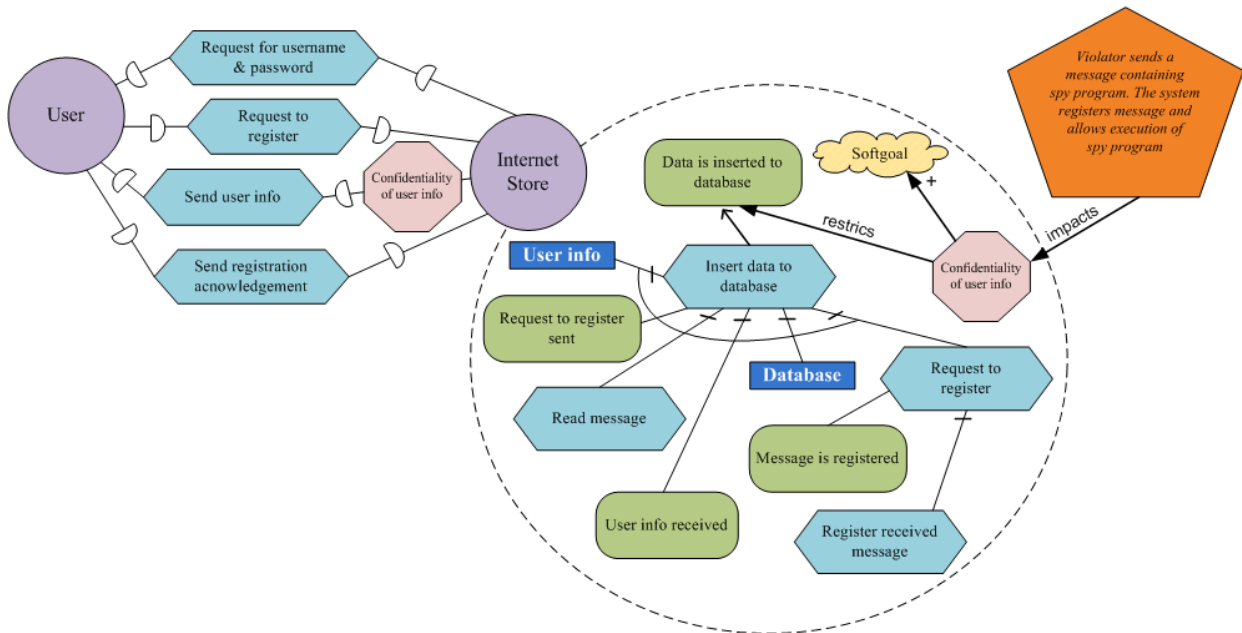


Figure 39. Secure Tropos: Confidentiality analysis – risk model (Threat construct)

Model that is taken as an input for treatment scenario is **Figure 40**. Transformation of BP Diagram starts with definition of security requirements from BPMN *Tasks* to Secure Tropos *Plans* and *Goals* following the rule **TR15**. Additional *Security constraint* Register only trusted messages is defined from BPMN Gateway structure controlling the safeness of incoming message. Model is supplemented with *Satisfies* relationships with respect to **TR17**. In the final version of Secure Tropos model (**Figure 41**),

the risk scenario is collapsed to an *Impact* and completed with *Mitigate* relationship. After some manual add-ons, relationships between security requirements representing *Goals* and *Plans* are defined and can be seen in *Figure 42*.

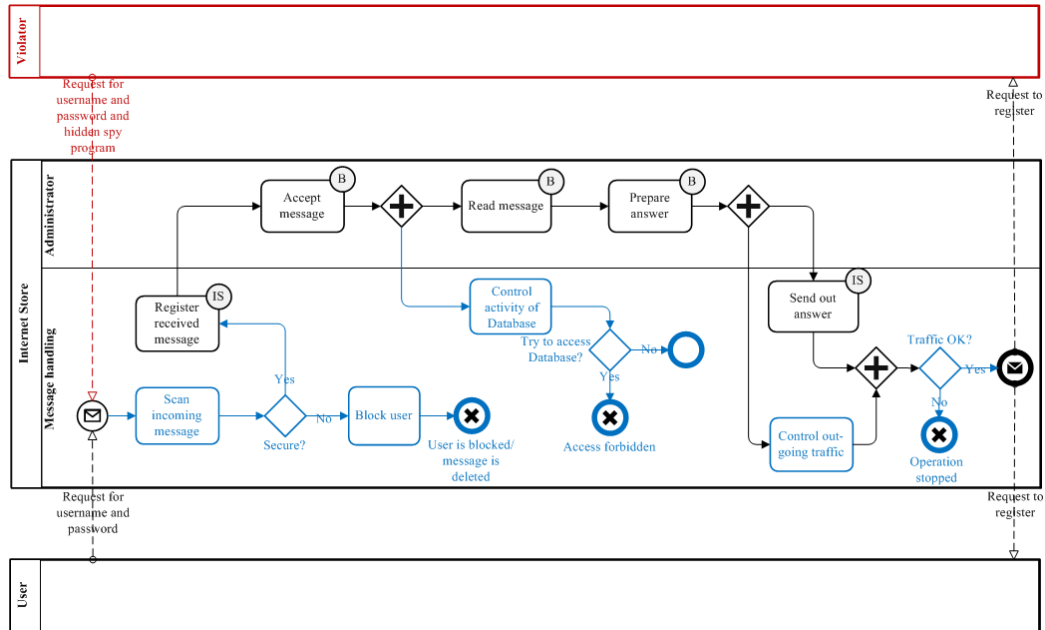


Figure 40. BPMN: Confidentiality analysis – Security Requirements

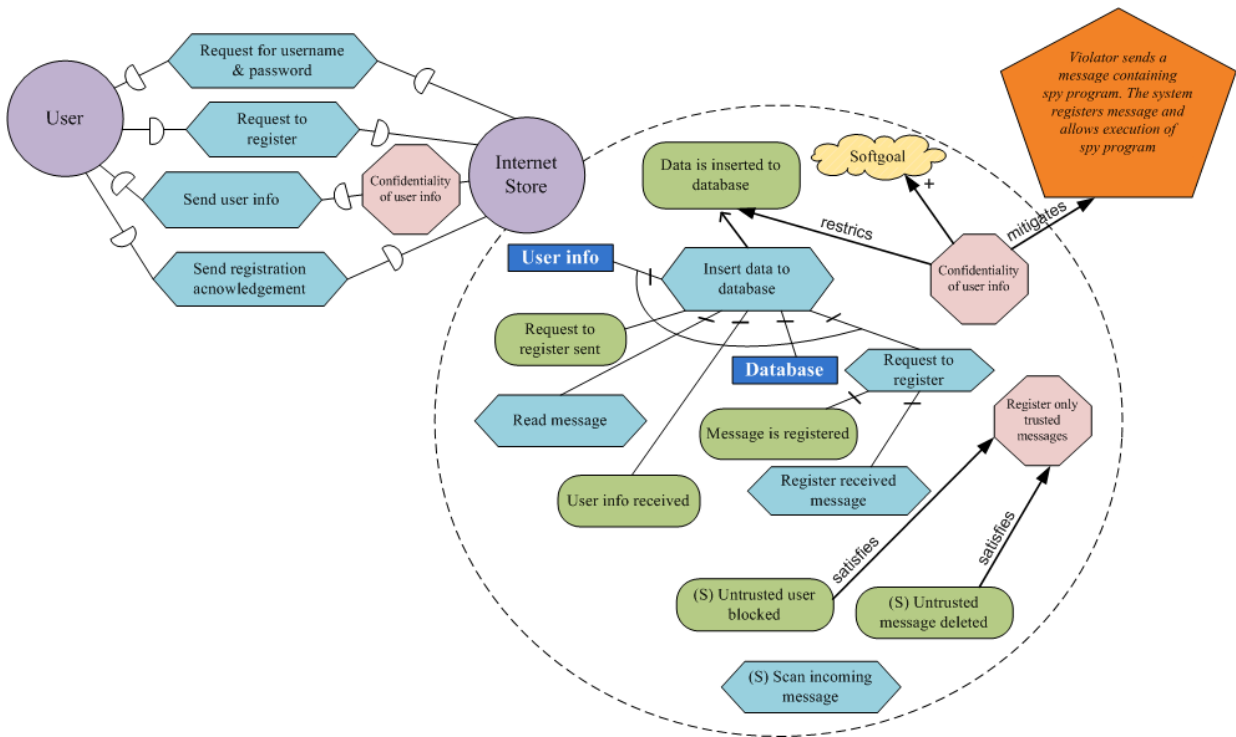


Figure 41. Secure Tropos: Confidentiality analysis – Security Requirements

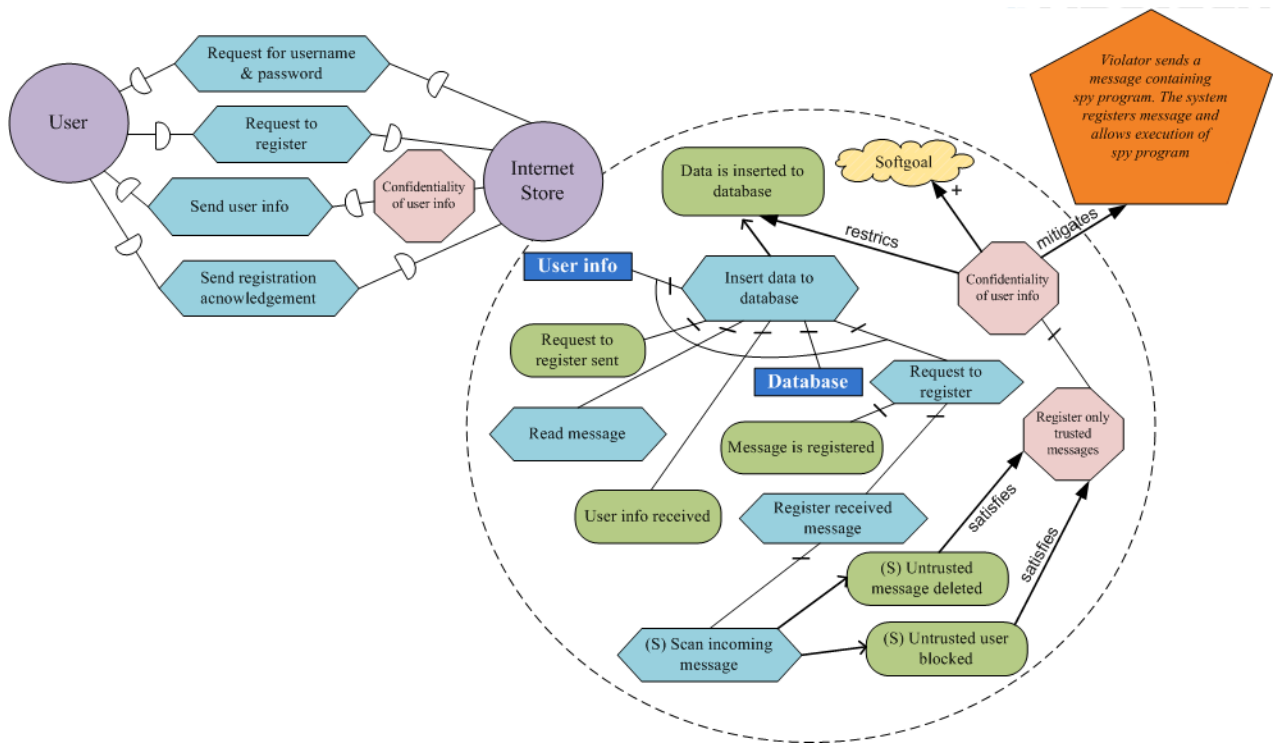


Figure 42. Secure Tropos: Confidentiality analysis – Security Requirements (complete)

6.2.3 Integrity

The BPMN process for transformation is presented in **Figure 43** and **Figure 44**. Just like in two previous examples we start with identification of Actors and dependencies between them. According to transformation rules **TR2** and **TR3** we translate all the Message flows in BP Diagram (Request for username and password, Request to register, User info and others) to Secure Tropos *Plans* and basing on the directions of the flows we define dependencies. The main purpose of presented process is that Internet Store would be used properly, with respect to that context we transform *Task Use Internet Store* (from **Figure 43**) to Secure Tropos *Plan* and *Goal* (Use Internet Store, Internet Store is used properly). We add *Security constraint* and *Softgoal* transformed from the BPMN *Lock* and *Annotation*. We also define the *Resource* User info. After all the necessary relationships are added and manual work is done we receive the complete Secure Tropos asset model in **Figure 46** (model without manual work is presented in **Figure 45**).

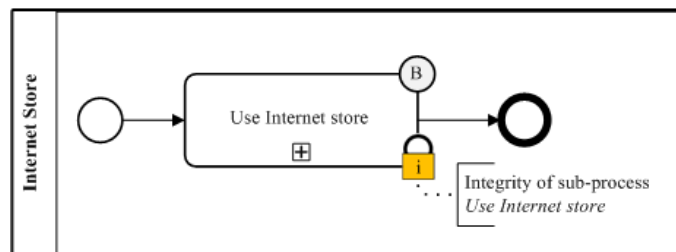


Figure 43. BPMN: Integrity analysis – asset identification (collapsed process)

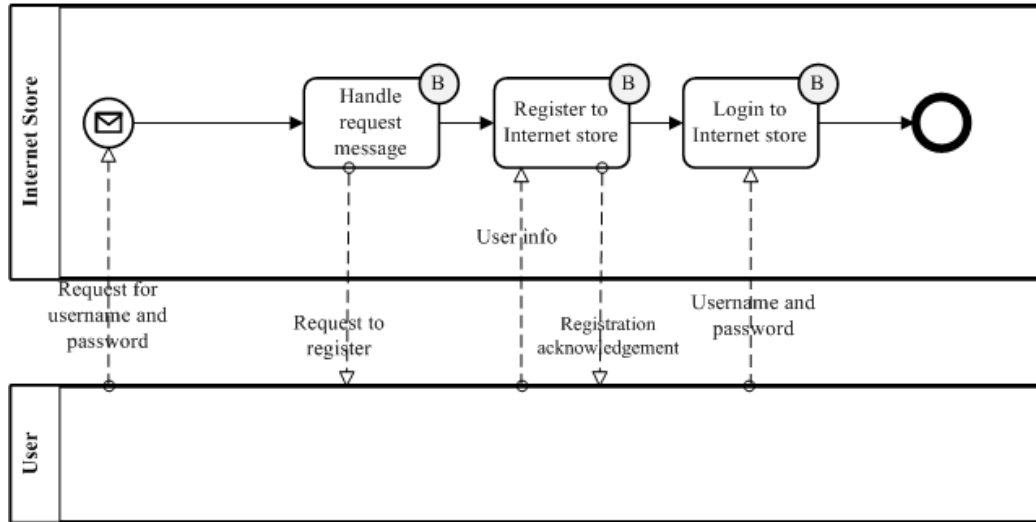


Figure 44. BPMN: Integrity analysis – asset identification (expanded process)

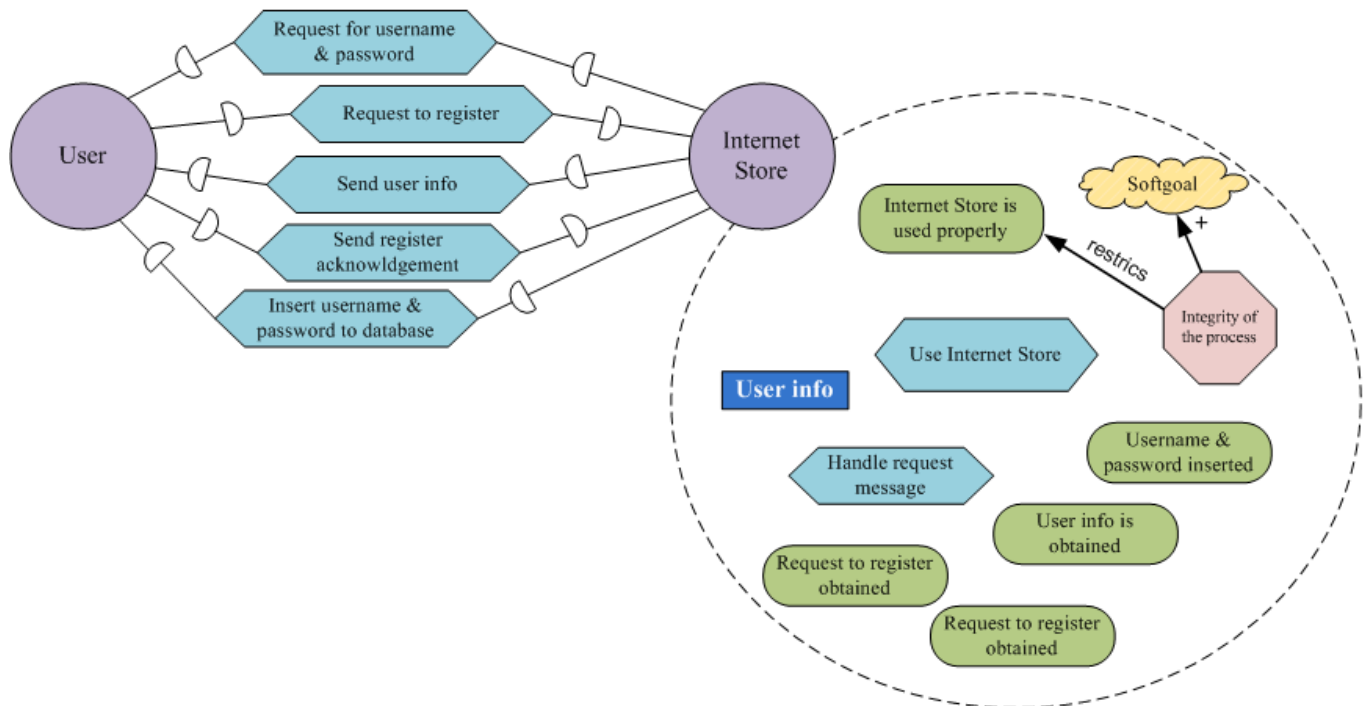


Figure 45. Secure Tropos: Integrity analysis – asset identification

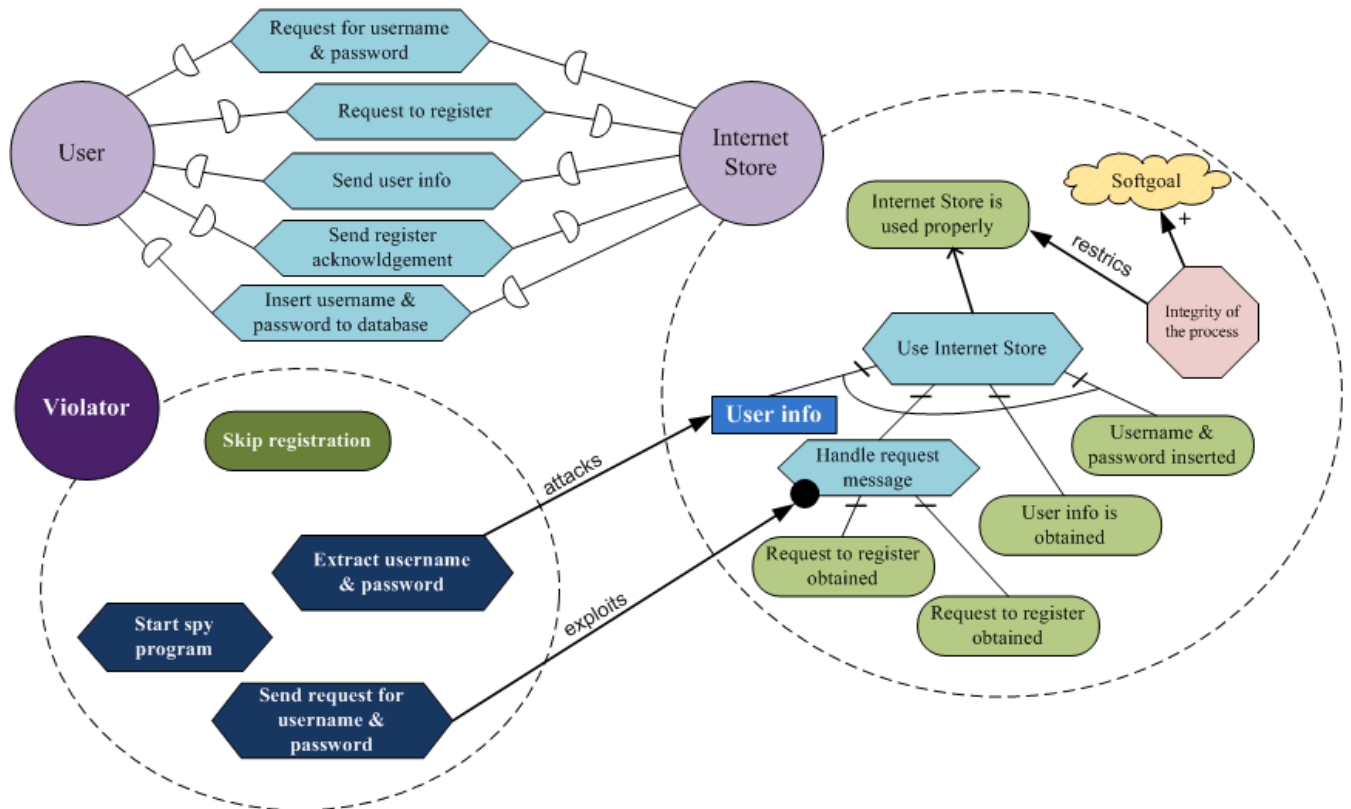


Figure 48. Secure Tropos: Integrity analysis – security risk

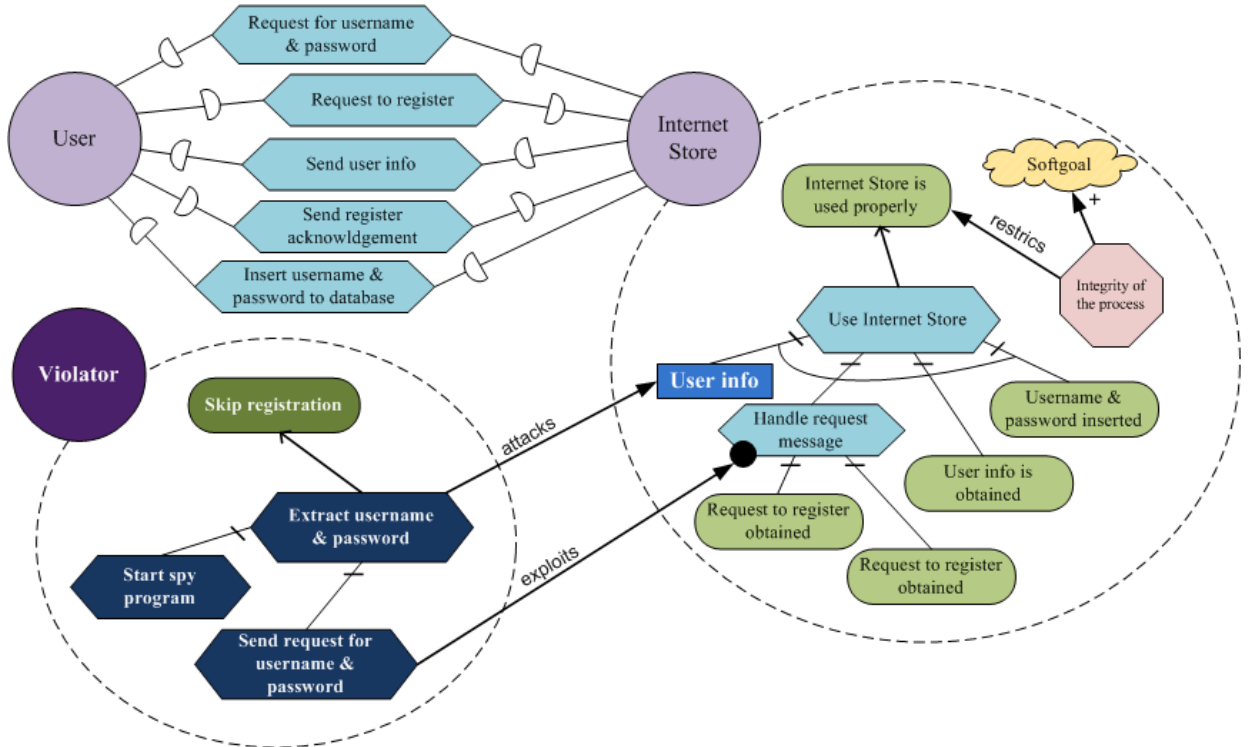


Figure 49. Secure Tropos: Integrity analysis – security risk (complete)

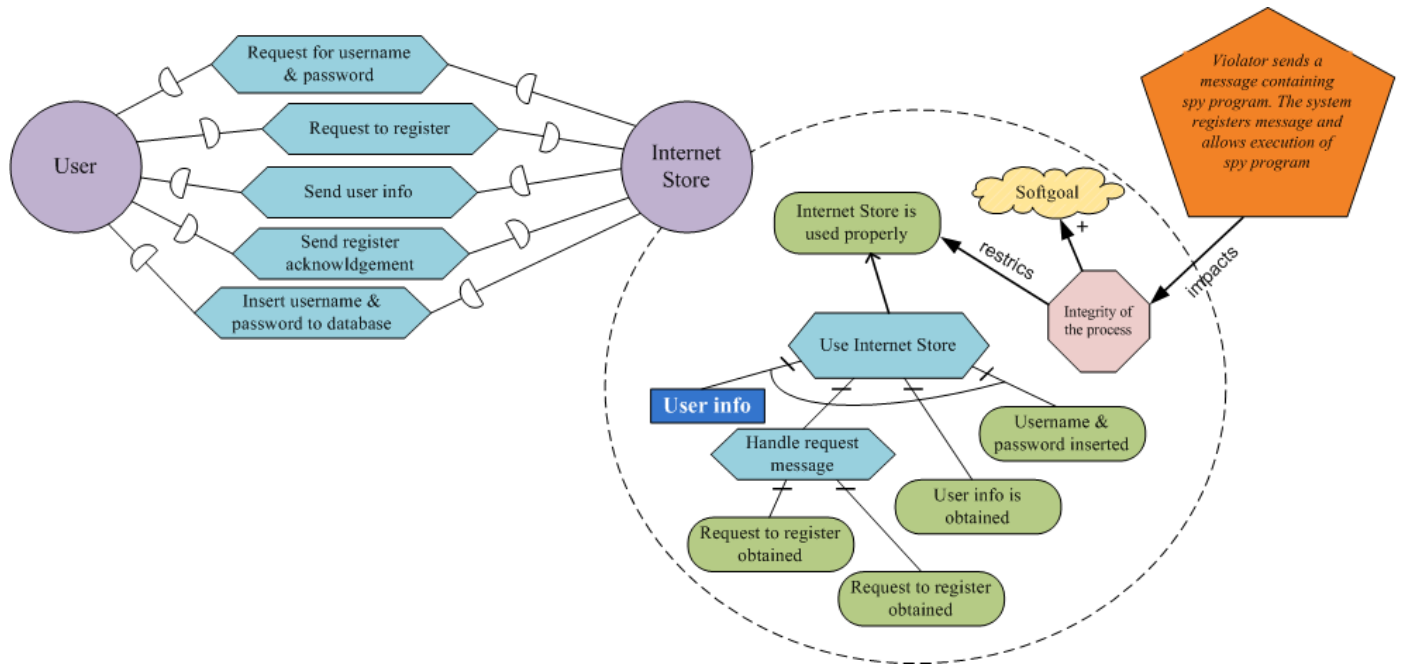


Figure 50. Secure Tropos: Integrity analysis – security risk (Threat construct)

The treatment decision for current example is also described previously as it is based on Confidentiality example (see *Figure 41* and *Figure 42*). The final result including manual add-ons is presented in Figure 51.

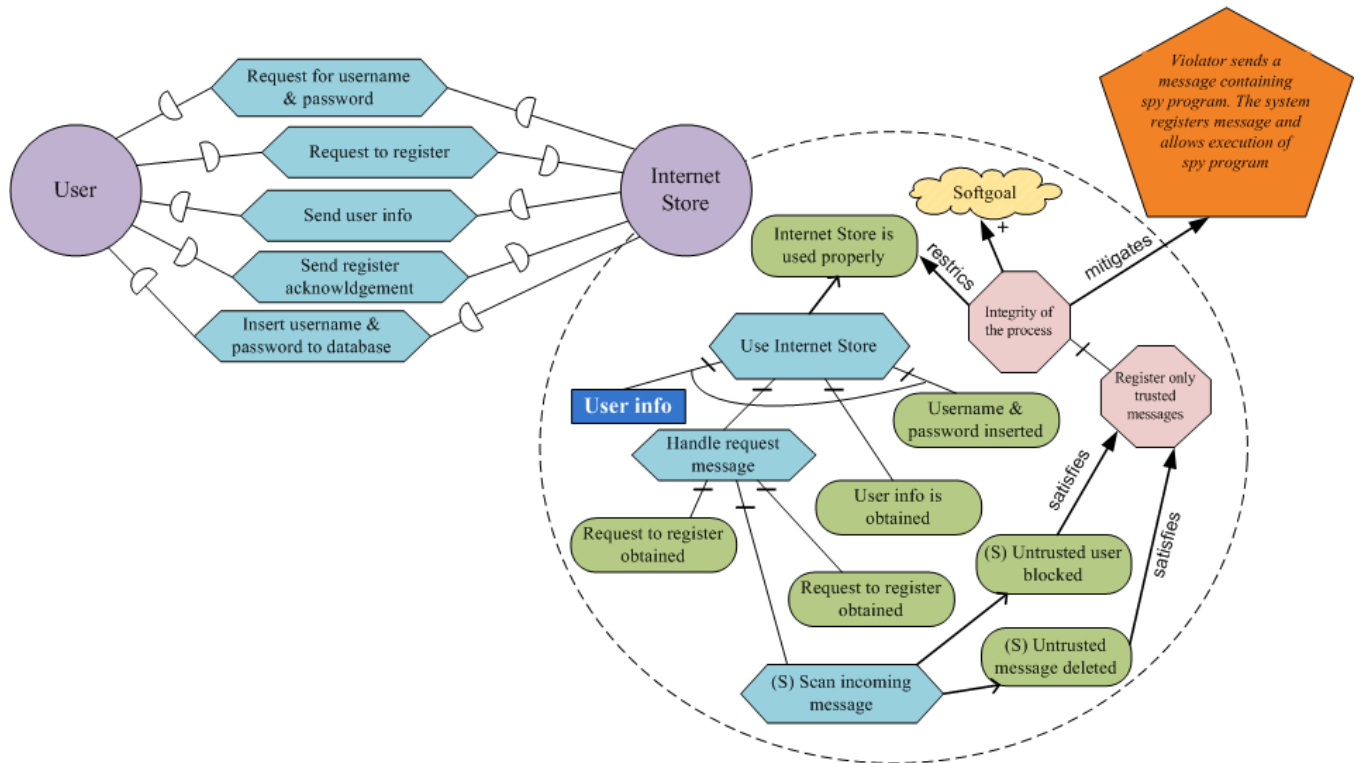


Figure 51. Secure Tropos: Integrity analysis – risk treatment

Table 5. BPMN to Secure Tropos constructs transformation mapping: Asset-related concepts

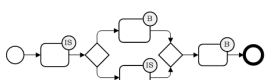




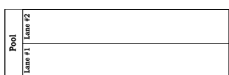

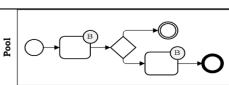
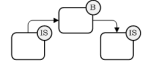
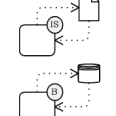
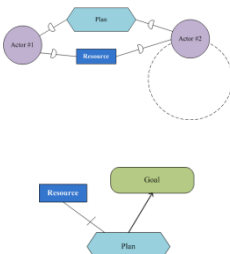






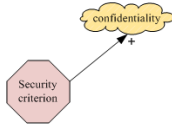
| ISSRM (Concepts, Relationships) | BPMN | | Secure Tropos | |
|---------------------------------------|--|---|---|---|
| Asset | Combination of flow objects, connected with sequence flow |  | Combination of Actor, Goal, Plan, Resource objects, combined using <i>Dependency</i> , <i>Contribution</i> , <i>Means-ends</i> and <i>Decomposition</i> links |  |
| Business asset | Data object |  | Resource |  |
| IS asset | 1. Data store 2. Containers (Pools, Lanes) |   | |  |
| <i>Supports</i> | 1. Containers (Pools, Lanes) 2. Sequence flow 3. Data association flow |    | 1. Dependency 2. Means-ends, Decomposition |  |
| <i>Constraint of</i> | Lock + Association flow |    | Security criterion and Restricts relationship | |
| Security objective | Label in the lock: i-integrity, a-availability, c-confidentiality |  | Softgoal |  |
| Security criterion | Annotation |  | Softgoal, Security criterion, Contribution |  |

Table 6. BPMN to Secure Tropos constructs transformation mapping: Risk-related concepts

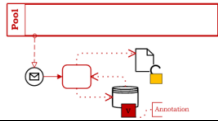

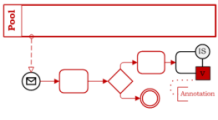
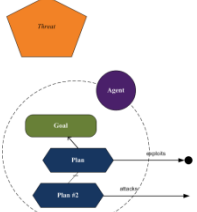
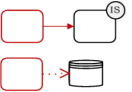

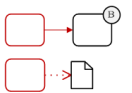



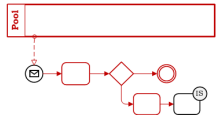
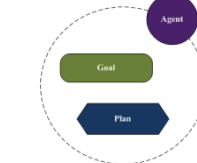
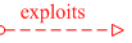


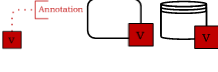







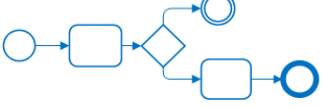
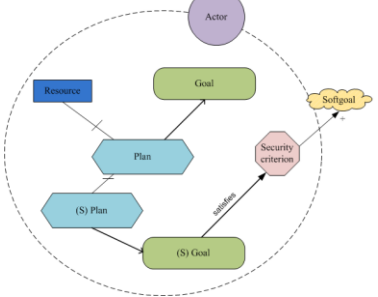
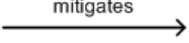
| ISSRM (Concepts, Relationships) | BPMN | Secure Tropos |
|--|--|--|
| Risk | Combination of Event and Impact  | Combination of Threat and <i>Impacts</i> relationship  |
| <i>Significance assessed by</i> | - | - |
| Event | Combination of Threat and Vulnerability  | 1) Threat 2) Combination of Agent, Goal, Plan, Targets, Exploits and Vulnerability point  |
| <i>Targets/leads to</i> (leads to harm of IS assets) | Sequence flow, Data association flow  | <i>Attacks</i> relationship  |
| <i>Leads to</i> (leads to harm of business assets) | Sequence flow, Data association flow  | <i>Attacks</i> relationship leading to Resource  |
| Impact/ <i>negates</i> | Unlock  | Combination of Threat, <i>Impacts</i> relationship and Security criterion  |
| Threat | Combination of Threat agent and Attack method  | Combination of Agent, Plans, Goals  |
| <i>Exploits</i> | Exploits relationship  | <i>Exploits</i> relationship  |
| Vulnerability | Annotation  | - |
| <i>Characteristics of</i> | Vulnerability point + association flow  | Vulnerability point (presented on Goal, Plan, Resource)  |
| Threat agent | Pool, Lane  | Agent  |
| Attack method | Combination of flow objects  | Plan  |
| Uses | Data flow  | Agent executes Plan  |

Table 7. BPMN to Secure Tropos constructs transformation mapping: Risk treatment-related concepts

| ISSRM (Concepts, Relationships) | BPMN | | Secure Tropos | |
|---------------------------------------|---|---|--|---|
| Risk treatment | - | - | - | - |
| <i>Decision to treat</i> | - | - | - | - |
| <i>Leads to</i> | - | - | - | - |
| Security requirements | Combination of Flow objects + sequence flow |  | Combination of Plan, Goal, Resource, Security constraint, Softgoal and relationships: <i>Contribution, Means-end, Decomposition</i> |  |
| <i>Mitigates</i> | | | <i>Mitigates</i> relationship |  |
| <i>Implements</i> | - | - | - | - |
| Controls | - | - | - | - |

6.2.4. Summary

The summary of transformation results is conducted into *Table 5 - 7*. From the tables and demonstration of model translation from one language to another, it can be summarized that transformation of BP diagram to Secure Tropos model is possible. The major part of security concepts can be easily translated as it is seen from tables below. However complete transformation remains impossible. First of all, definition of relationships in Secure Tropos can be performed only manually, from the context of running example. The reason for this comes from difference of nature of two modelling languages: BPMN depicts process flow and helps to see the sequence of performed activities; Secure Tropos in turn gives the static picture of environment, and benefits defining hierarchy and decomposition relations between objects, definition of *actors* and *goals*. Secondly, there is no opportunity to represent Vulnerability in Secure Tropos, which leads to loss of some part of valuable information in security analysis. More detailed assessment on the quality of model transformation will be given in the Validation part.

PART III. Validation

Chapter 7. Validation of BPMN extensions

This chapter is dedicated to validity of BPMN extensions proposed earlier in current work. Chapter will cover the research method that we used to validate understandability of Security-oriented BPMN, description of experiment we held in frames of this research and analysis of resulted dataset.

7.1 Description of the experiment

It was decided that understandability validation should be held in form of an exercise. The research experiment was organized in Tartu University among student of Software Engineering and Informatics. Majority of the participants are the first year master's students, who have received a degree in Computer science, which means person has an understanding of basic software engineering processes, risk management and system modelling. In total, ten persons were examined with completely identical questions regarding the same example.

For this case study we designed three different flows of one example and recreated it with the help of Security-oriented BP diagrams. These models include normal scenario, potential risk scenario and proposal for risk treatment. Participants of the experiment were asked to observe the models and extract the asset-, risk- and risk treatment-related issues, following the basis of the ISSRM process, which guidance were provided as an additional material. Apart from defining the contextual information of the example, it was also assumed that understanding of correspondence between this extracts to visual constructs can be defined. This is a reason behind the division oh hypothesis formulation and as a consequence also the scope of assessment. First of all we are purposed on receiving the picture of understandability of applied semantics: if person can extract valuable information from the model and define what are the security assets, threat agent, vulnerabilities etc. And secondly we are interested to see if person is able to understand the applicability of the language extensions: meaning to match extracted information with visual representation.

Participants received a question sheet with three models and tables with corresponding fill-in fields. They were also provided an additional material that introduced the concrete and abstract syntax of designed extensions and gave an overview on basic steps of ISSRM process. The formulation of exercise was to fill in the gaps in two table columns, according the requested information.

7.2 Introducing the variables and assessment

As the independent variables we introduce the scale for evaluation of the correctness of given answers to assess the understandability percentage for above mentioned aspects. Each correct answer gives one point, which means that information is understood for more than 75%, although we assume that some infelicities are possible to make in frames of given answer. In case if no answer is given or given answer is not correct, it is assessed with 0 points, which means understanding of information is less than 25%. Other cases are covered in the *Table 8*, which is located below. The correct extraction of information for both, asset- and risk treatment concepts gives maximum 3 points for example context identification and 3 more points for the correct definition of visual constructs. Risk-related concepts give maximum 7 points for each column; it means 14 points in total. Once the answers are checked for correctness, the understandability percent can be counted. We validate the understandability of used

language by finding the average from the right answers (points) for each model and transforming the number to percentage basing on the total possible sum.

$$\textit{Understandability} (\%) = \frac{\textit{AvP}}{\textit{MaxP}} * 100, \text{ where}$$

AvP is average number of points gained for identification of either a visual construct or its meaning in the example context. Another opportunity is to assess the understandability of the whole diagram, and for each diagram separately (e.g. asset-related model). *MaxP* stands for maximum possible number of points in concrete set of questions. Data conducted from exercise sheets is represented in Appendix B. The major extracts will be highlighted below.

Table 8. Scale for understandability assessment

| Scale | Criteria |
|-------|--|
| 0 | No answer or completely wrong answer |
| 0,25 | ¼ part of the answer is right: person captures the relatively small idea in the right direction; person captures only small part of the right answer or proposes this idea in a row with wrong ideas. |
| 0,5 | Only half of the answer is right: person captures just a part of the right idea, it can be presented separately or in a row with a wrong idea in relation of 1 to 1. |
| 0,75 | Person captures the idea of correct answer, but represents the small deviation from the right direction; person suggests right idea with less than 3 small infelicities; person suggests the ¾ of the right answer or number of the right answers. |
| 1 | Person gives completely right answer or the right answer with a small infelicity |

7.3 Data interpretation

Answers, which were evaluated according to the above described scale, were conducted together in the separate spreadsheet file. Once the data was structured and organized, we can define the understandability percent for each concept separately or for the whole diagram.

Asset model. The following concepts were expected to be correctly extracted from Asset model: IS Asset, Business Asset and Security criterion (see *Table 9*). Identification of business asset from the model was performed successfully: nine of ten participants defined it correctly, which gave the value of 90% for understandability. However, the security criterion presentation was not so obvious for majority of the participants; it gained only 20% in understandability for its visual representation. The overall asset model semantics is understandable for 68%, and visual representation of the model – for 53%.

Risk model. Understandability of Risk model included the correct definition of Risk, Impact, Event, Threat, Vulnerability, Threat agent and Attack method. The overview of data set shows that the percentage of understood semantics for such concepts as Risk, Impact, Threat, Threat Agent and Attack method is high, which means it is equal or more than 80 (see *Table 10*). However the understandability for visual representation of the same concepts varies from 47% to 100%; it shows that some of the

constructs are more essential and natural to detect from the diagram (or example context) than others. It also points on the idea that definition of semantics comes easier; sometimes semantic meaning can be (or even should be) extracted not from the model itself, but from the example situation. So the definition of semantics aspect remains more flexible. Understandability percent for the remaining concepts stays between 50% and 70% for semantics as well as for visual representation. The overall model is visually understandable for 65% and contextually for 80%.

Risk treatment model. The lowest percent of understandability is presented in data extracted from Risk treatment model (*Table 11*). We believe that the reason behind that is a lack of resources investigated into this aspect in frames of our current analysis; definition of extensions for risk treatment modelling remained out of scope. Risk treatment and Security criterion concepts gained 22% in understandability of semantics and higher percent for graphical representation correspondingly 55% and 44%. The concept we defined in frames of designing the language extensions is Security requirement, which was semantically understood for 100%, visually but for 78%. Risk treatment related model semantics is understandable for 48%, visual representation – for 59%.

Table 9. Asset model’s understandability (in %)

| Person ID | Business asset | | IS asset | | security criterion | |
|------------|-----------------|---------------|-----------------|---------------|--------------------|---------------|
| | <i>semantic</i> | <i>visual</i> | <i>semantic</i> | <i>visual</i> | <i>semantic</i> | <i>visual</i> |
| 1 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 | 1,00 |
| 2 | 0,00 | 0,00 | 0,50 | 0,50 | 0,00 | 1,00 |
| 3 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 | 0,00 |
| 4 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 | 0,00 |
| 5 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 | 0,00 |
| 6 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 | 0,00 |
| 7 | 1,00 | 1,00 | 1,00 | 0,50 | 0,00 | 0,00 |
| 8 | 1,00 | 1,00 | 0,50 | 0,50 | 0,00 | 0,00 |
| 9 | 1,00 | 1,00 | 0,50 | 0,50 | 1,00 | 0,00 |
| 10 | 1,00 | 1,00 | 0,50 | 0,50 | 0,00 | 0,00 |
| avg | 0,90 | 0,90 | 0,55 | 0,50 | 0,60 | 0,20 |
| max | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| % | 90 | 90 | 55 | 50 | 60 | 20 |

Table 10. Risk model's understandability (in %)

| Person ID | Risk | | Impact | | Event | | Threat | | Vulnerability | | Threat Agent | | Attack method | |
|------------|------------|------------|------------|------------|------------|------------|------------|------------|---------------|------------|--------------|------------|---------------|------------|
| | <i>sem</i> | <i>vis</i> | <i>sem</i> | <i>vis</i> | <i>sem</i> | <i>vis</i> | <i>sem</i> | <i>vis</i> | <i>sem</i> | <i>vis</i> | <i>sem</i> | <i>vis</i> | <i>sem</i> | <i>vis</i> |
| 1 | 0,75 | 1,00 | 1,00 | 0,00 | 0,50 | 0,00 | 0,50 | 0,50 | 0,00 | 0,00 | 1,00 | 1,00 | 0,50 | 0,00 |
| 2 | 0,75 | 0,00 | 1,00 | 0,00 | 0,75 | 0,00 | 1,00 | 1,00 | 1,00 | 0,00 | 1,00 | 1,00 | 0,50 | 0,00 |
| 3 | 1,00 | 1,00 | 1,00 | 1,00 | 0,75 | 0,25 | 1,00 | 0,00 | 0,50 | 0,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 4 | 1,00 | 1,00 | 1,00 | 0,50 | 0,50 | 0,25 | 0,00 | 0,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 5 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,50 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 6 | 1,00 | 1,00 | 0,00 | 0,75 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 7 | 1,00 | 0,25 | 0,50 | 0,00 | 0,00 | 0,00 | 0,50 | 0,00 | 0,00 | 0,00 | 1,00 | 1,00 | 0,00 | 0,00 |
| 8 | 0,75 | 0,75 | 1,00 | 0,75 | 0,50 | 0,50 | 1,00 | 1,00 | 0,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,00 |
| 9 | 0,75 | 1,00 | 1,00 | 0,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 10 | 0,25 | 0,25 | 1,00 | 0,75 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| avg | 0,83 | 0,73 | 0,85 | 0,48 | 0,70 | 0,50 | 0,80 | 0,65 | 0,60 | 0,60 | 1,00 | 1,00 | 0,80 | 0,60 |
| max | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| % | 83 | 73 | 85 | 48 | 70 | 50 | 80 | 65 | 60 | 60 | 100 | 100 | 80 | 60 |

Table 11. Treatment model's understandability (in %)

| Person ID | Risk treatment | | Security requirement | | Security criterion | |
|------------|-----------------|---------------|----------------------|---------------|--------------------|---------------|
| | <i>semantic</i> | <i>visual</i> | <i>semantic</i> | <i>visual</i> | <i>semantic</i> | <i>visual</i> |
| 1 | 1,00 | 0,00 | 1,00 | 1,00 | 0,00 | 0,00 |
| 2 | 0,00 | 1,00 | 1,00 | 1,00 | 0,00 | 1,00 |
| 3 | 0,00 | 0,00 | 1,00 | 1,00 | 0,00 | 0,00 |
| 4 | 1,00 | 0,00 | 1,00 | 0,00 | 0,00 | 0,00 |
| 5 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 6 | 0,00 | 1,00 | 1,00 | 1,00 | 0,00 | 1,00 |
| 7 | 0,00 | 0,00 | 1,00 | 1,00 | 0,00 | 0,00 |
| 8 | 0,00 | 0,00 | 1,00 | 1,00 | 0,00 | 0,00 |
| 9 | 0,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| 10 | 0,00 | 1,00 | 1,00 | 0,00 | 0,00 | 0,00 |
| avg | 0,22 | 0,56 | 1,00 | 0,78 | 0,22 | 0,44 |
| max | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| % | 22 | 55 | 100 | 77 | 22 | 44 |

7.3 Threads to validity

Our proposal for BPMN extensions contains a certain degree of subjectivity, which is a major threat to validity in current part of the study. We assume that at least three aspects of that research are influenced. First of all we can speak about the subjectivity of the input for the above described experiment; the study on BPMN alignment to security and the proposed extensions are performed by two researchers only, which means the solution can be interpreted, aligned and designed differently from the perspective of another knowledge domains, experience and backgrounds. On the other hand, experiment involves participation of ten different people. However, the similar knowledge backgrounds and relatively small number of students involved into the research also raise the question of subjectivity. And finally, the scale for results analysis and data interpretation cannot be objective for the similar reasons.

7.4 Data analysis results

The validation of extensions' understandability from two points of view shows that majority of security risk-related concepts are understandable and can be successfully extracted from the models with respect to language semantics as well as to its visual representation. However during the analysis of experiment resources (exercise sheets, data set) we discovered some difficulties that potentially had an influence on the understandability percentage. For instance the practice shows that use of the same graphical construct for presentation of different ISSRM concepts (despite the fact that different colours were used) makes questions and misunderstandings. We believe that it has an influence on the definition of such concepts as Security criterion and Vulnerability. Although contextually these concepts are not easy to mess up, the representation of both with the help of BPMN Annotation highlights the risk of wrong interpretation. Another problem is that some information is not presented in the models, but required to be extracted from the context, such as Risk treatment for example. We did not define any visual construct for definition of Risk treatment; however it can be extracted logically basing on the actions defined for the security requirements. Identification of such information may require additional experience and skills in working with this kind of analysis. Obviously it was the reason behind such a low percentage of correctly stated answers. And finally, it is not easy to define the border between what can be logically extracted from the example, and what is out of scope. With respect to this issue, we faced the problem of internal information involvement, for example some attempts to define Controls. To conclude, we believe that proposed extensions are understandable and applicable to model security. However the improvement of representation abilities and review of scope remains for the future work.

Chapter 8. Transformation quality assessment

In this chapter we will give the evaluation to the quality of transformed models. We will emphasize on comparison between Secure Tropos model, created essentially from the example context and the model received from BPMN via translation based on transformation rules described in Chapter 6. Our analysis is based on predefined assessment criteria, which will be introduced further in this chapter. We also give an overview of research method and main evaluation goals.

8.1 Research method

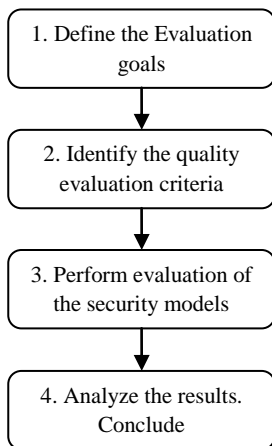


Figure 52. Quality assessment process (adapted from Matulevičius et al, 2011)

Our research method is based on the approach presented in (Matulevičius et al, 2011). More specifically we adapt four steps from the introduced process to form our research method. The first step is dedicated to definition of evaluation goals. As it is stated in the above mentioned paper, it covers the understanding of security needs, learning of the quality and scope of security models and also the aspect we are concentrating on - comparison of security models according to assessment criteria. After major goals are defined, we move to the identification of the quality assessment criteria, which will be introduced later in the next section. We use the example models that were developed in frames of this study to perform the evaluation. Once the evaluation is complete, we perform the analysis of the results and lead to conclusions, where we decide, whether our evaluation goals are satisfied or not. The steps of our research method are presented in *Figure 52*.

8.2 Evaluation goals

First of all, we are interested in giving the assessment and learning about the quality of two models: (i) Secure Tropos model created from the textual context of familiar example (confidentiality analysis in the Internet Store) and (ii) model transformed from BP Diagram to Secure Tropos. In this purposes we will compare these models with respect to its informational value, more specifically we will operate with the numbers of security concepts used in one or another model aligning it with ISSRM. And finally we are aimed on giving the assessment on the transformation results; what value has the transformed model in comparison to the model originally created in Secure Tropos.

8.3 Quality evaluation criteria

For evaluation we have chosen three types of quality criteria, models will be assessed with respect to its semantic and syntactic quality. The semantic quality stands for the correspondence between the model and the semantic domain it was aligned to, in our case we deal with ISSRM. We assess semantic quality by defining the semantic completeness and semantic correctness of the security model. The *semantic completeness* stands for the percentage of ISSRM concepts covered by the semantics of the language used to create the model. It is calculated as follows: number of ISSRM security concepts used in the model will be divided by total number of security concepts defined in the scope of this research.

The *semantic correctness* will give a picture on correlation of used security-related statements to the overall model semantics. In other words it requires the separation of the informational resources of the model between ones that represent data, and those which stand for security.

Assessment of syntactic quality of the model is based on definition of correspondence between the model and used modelling language. We assess models with respect to its syntactic validity and completeness. First assessment criteria defined the check that grammatical expression used to create model belongs to the modelling language. In this case we address the numbers of syntactically valid and syntactically invalid statements. The number of syntactically invalid statements influences on the quality of the model in the following proportionality; the more syntactically invalid statements are discovered in the model, the lower is the quality of it. The syntactic completeness in turn refers to the fulfillness of the grammar structures, meaning that all the parts are presented within the construct. Concerning the number of syntactically complete and incomplete statements, the assessment criterion is applied similarly to the previous and adapts the same proportionality regarding the quality of the model.

8.4 Analysis of evaluation results

In this section we will give an assessment based on the comparison between two sets of models. First of all, we address Secure Tropos models designed in Chapter 6; models that were transformed from BPMN to Secure Tropos according to the transformation rules (see section 6.1). In order to avoid the overload of this chapter we will not duplicate the models, so please turn back to *Figures 35, 38, 39, 42*. The second set of models (see *Figures 53-56*) was originally created with Secure Tropos from the same example context that was used previously for BPMN confidentiality analysis example creation. These models represent so called “perfect model” view, which means we assume first of all the correctness and appropriateness of the used language, and secondly, we believe that these models sufficiently and also correctly address the ISSRM domain alignment. So it will be taken for base for future assessment.

We start our evaluation with analysis of semantic quality for two sets of models. To present the semantic completeness we count the number of security concepts in each model with respect to ISSRM domain coverage (see *Table 12*). Basing on (Matulevičius et al., 2012), ISSRM domain represents 7 asset-related, 13 risk-related and 7 risk treatment-related concepts and relations, each of this values stands for 100% coverage. Our analysis shows the following: the semantic completeness of transformed model is 100% for asset related concepts, 77% for risk-related and 14% for risk treatment-related concepts. Semantic completeness of Secure Tropos models is correspondingly 100%, 92% and 43%. According to our estimations, transformed asset model as well as original Secure Tropos asset model is 100% semantically correct, that tells about the success of transformation. More specifically, we assume that we do not lose any valuable information regarding system assets when transforming a model from BPMN to Secure Tropos. The difference in percentage of semantic correctness in risk models indicated majorly because of various ways for Vulnerability concept representation. In our research we didn't find an opportunity to present the vulnerability in Secure Tropos in the transformation process. However authors of (Matulevičius et al., 2012) propose to use vulnerability point to express vulnerability itself, we consider that it doesn't give sufficient information of the system weakness, but just gives a signal. We respect the authors' proposal and assume that for perfect model there exist a construct for

Vulnerability. Moreover, in the previous stages there was no defined construct for Significance assessed by relationship for both Secure Tropos and BPMN, so it was not covered by semantics of the languages, which reduces the percent of semantic completeness. Regarding the risk treatment model, it can be mentioned that major part of concepts were not represented in BPMN nor in Secure Tropos, however the original Secure Tropos model still covers the higher percentage of ISSRM domain; in addition to Security requirements concept, Mitigates relationship and Controls concept can be also addressed.

Next we assess the semantic correctness of the models by counting the number of constructs used for expression of one or another security concept. This part of the research is presented in the diagrams below (*Figure 57*) for a better view on differences in numbers we were able to discover. Two columns of the diagrams (*green, orange* colours on the plot) stand for two types of models that were discussed before. Numbers above the columns are the number of constructs, used to express ISSRM concepts. The data shows that the number of construct for two sets of models is nearly the same, but there is a tendency that model originally created in Secure Tropos (in green) uses less constructs to represent the security concepts such as Asset, Security requirements and also Supports relationship (see *Figure 57*). The only exception is Vulnerability concept, which was mentioned already before in semantic completeness analysis. Such kind of resource division seems logical to us, because we assume that in a row with some advantages, transformation will cause the overload of the model with extra element that can be considered as excess or waste. However the major idea remains the possibility for analysts to read and understand the model and also to extract necessary information. At that point, the difference in numbers of used constructs is not large; we are not dealing with overlapping semantics neither in transformed models, nor in original Secure Tropos model.

8.5 Threads to validity

The major limitation of this part of research is the scope. Due to the lack of time resource, we were able to apply the transformation rules to confidentiality example only. The transformation is realised in only one direction; reverse transformation from Secure Tropos to BPMN is not performed and remains for the future research. We see the need to investigate into transformation of other examples, and we also would like to involve a group of people that could try on practice the transformation according to the rules we defined. Unfortunately we didn't find the opportunity before with respect to time stamps and other resources.

Another threat to validity is also subjectivity; the set of transformation rules, examples and analysis were performed by only two researchers. We believe that this part of validation could be also performed in a form of experiment to investigate the model understandability and modifiability. However, the design and organization of this experiment remains for the future research and we decide to limit the validation with analysis on above selected criteria, defining the semantic quality of the models.

Table 12. Assessment on model semantic completeness (1-construct exists, 0-no construct)

| | Transformed Models BPMN→SecureTropos | Secure Tropos models |
|--|---|---------------------------------|
| Asset | 1 | 1 |
| Business asset | 1 | 1 |
| IS asset | 1 | 1 |
| <i>Supports</i> | 1 | 1 |
| <i>Constraint of</i> | 1 | 1 |
| Security objective | 1 | 1 |
| Security criterion | 1 | 1 |
| Asset model: Semantically complete (%) | 100 | 100 |
| Risk | 1 | 1 |
| <i>Significance assessed by</i> | 0 | 0 |
| Event | 1 | 1 |
| <i>Targets/leads to(leads to harm of IS assets)</i> | 1 | 1 |
| <i>Leads to (leads to harm of business assets)</i> | 1 | 1 |
| <i>Impact/negates</i> | 1 | 1 |
| Threat | 1 | 1 |
| <i>Exploits</i> | 1 | 1 |
| Vulnerability | 0 | 1 |
| <i>Characteristics of</i> | 1 | 1 |
| Threat agent | 1 | 1 |
| Attack method | 1 | 1 |
| <i>Uses</i> | 1 | 1 |
| Risk model: Semantically complete (%) | 76,9 | 92,3 |
| Risk treatment | 0 | 0 |
| <i>Decision to treat</i> | 0 | 0 |
| <i>Leads to</i> | 0 | 0 |
| Security requirements | 1 | 1 |
| <i>Mitigates</i> | 0 | 1 |
| <i>Implements</i> | 0 | 0 |
| Controls | 0 | 1 |
| Risk treatment model: Semantically complete (%) | 14,3 | 42,8 |

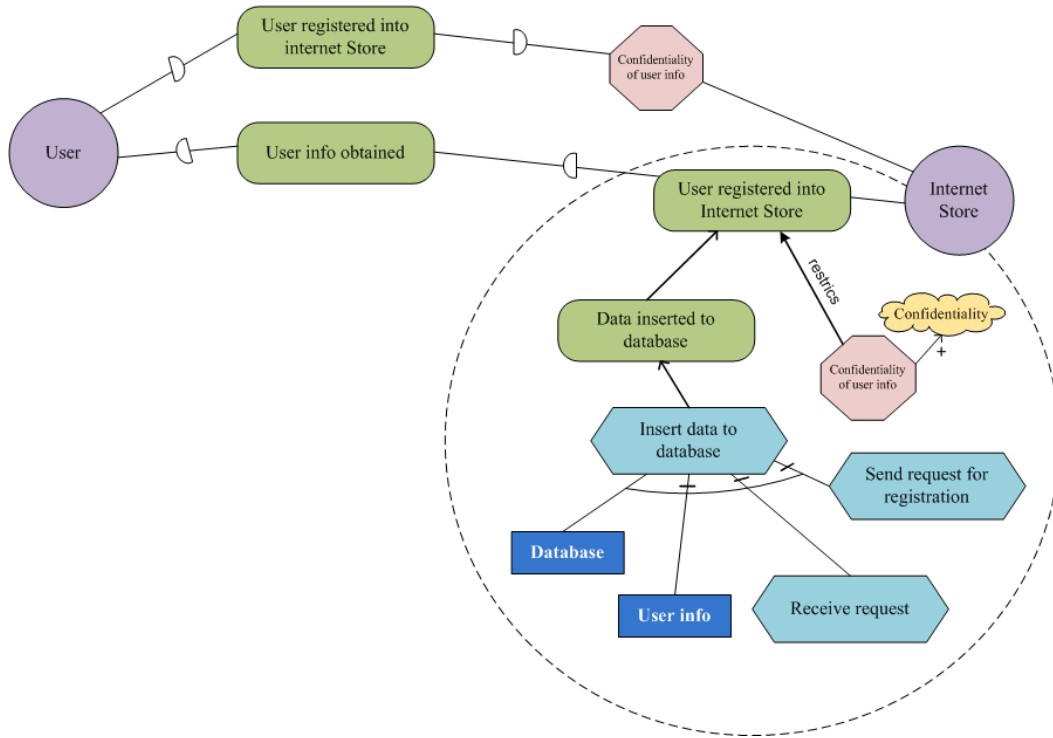


Figure 53. Secure Tropos: Asset identification – confidentiality (created originally in Secure Tropos)

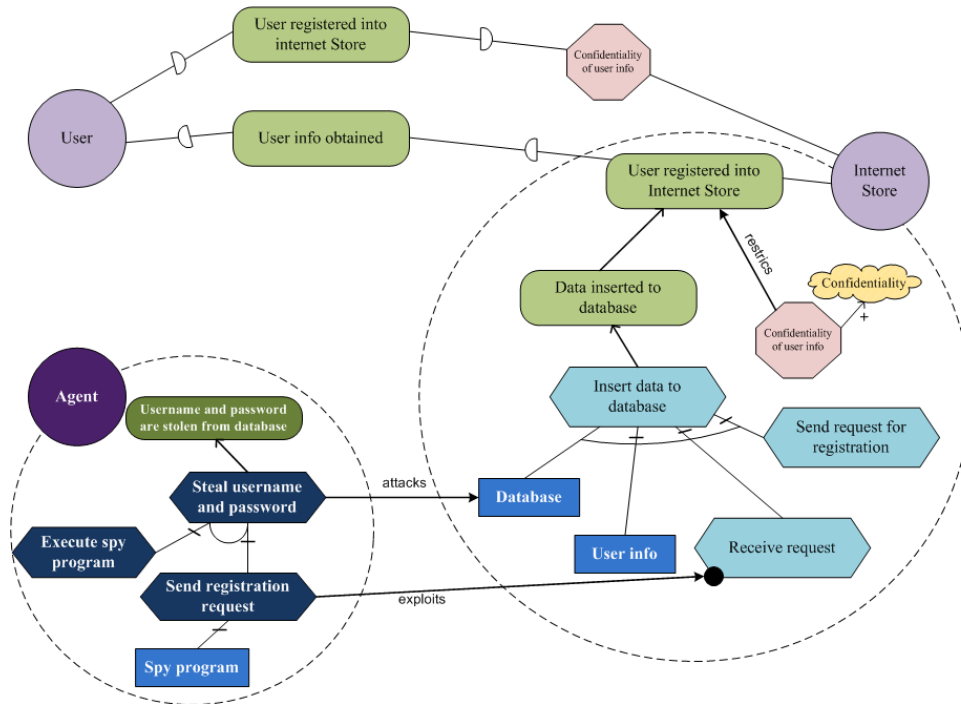


Figure 54. Secure Tropos: Risk identification – confidentiality (created originally in Secure Tropos)

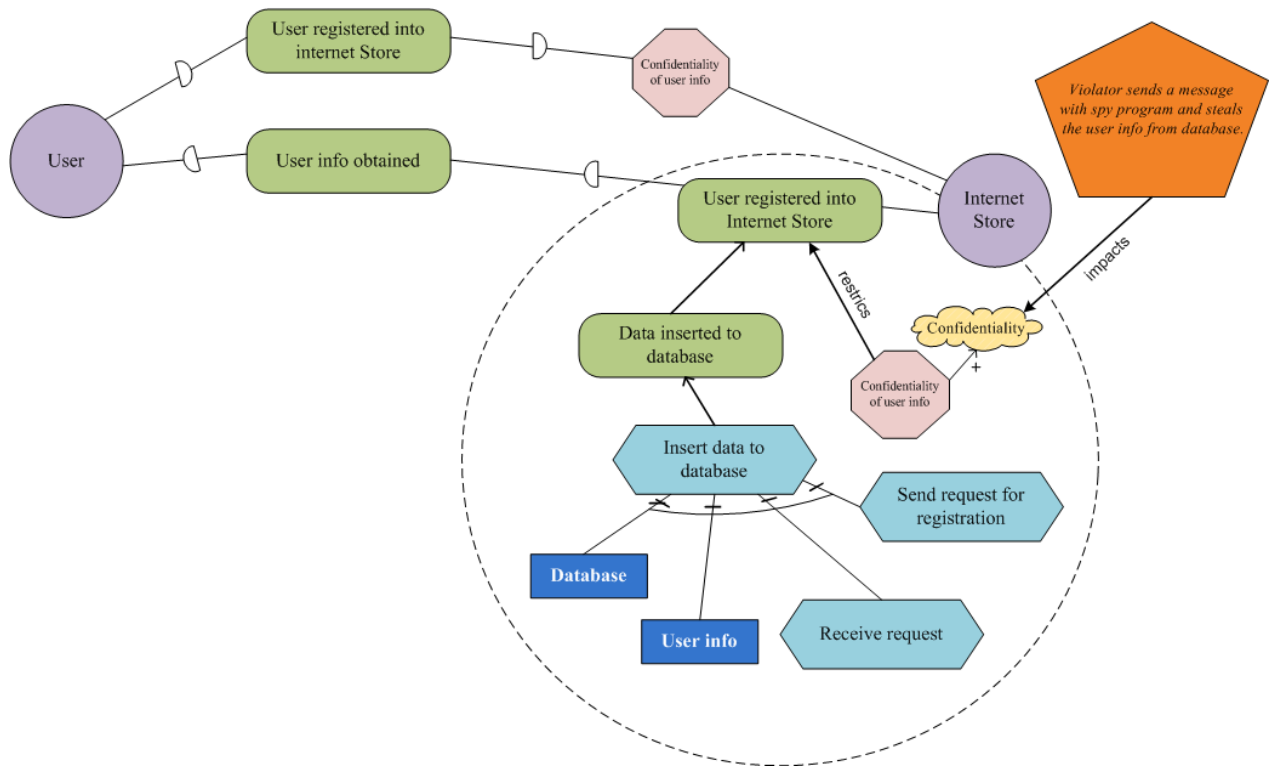


Figure 55. Secure Tropos: Risk identification (collapsed to Impact concept) – confidentiality (created originally in Secure Tropos)

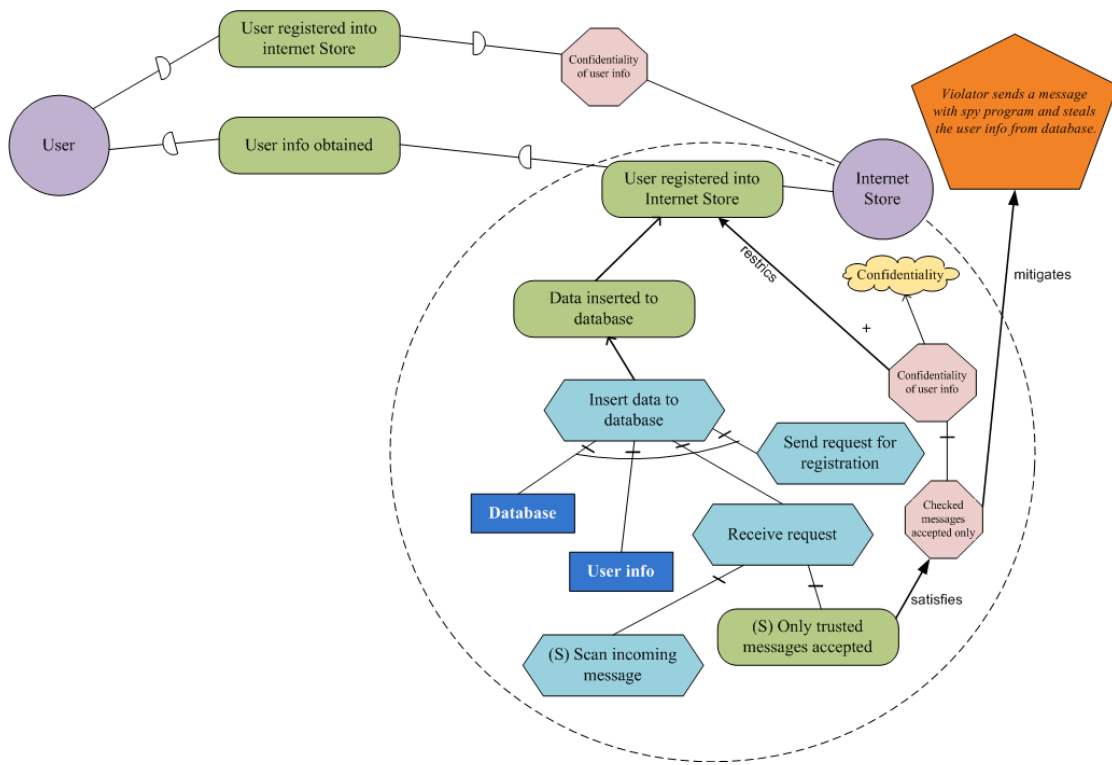


Figure 56. Secure Tropos: Risk Treatment – confidentiality (created originally in Secure Tropos)

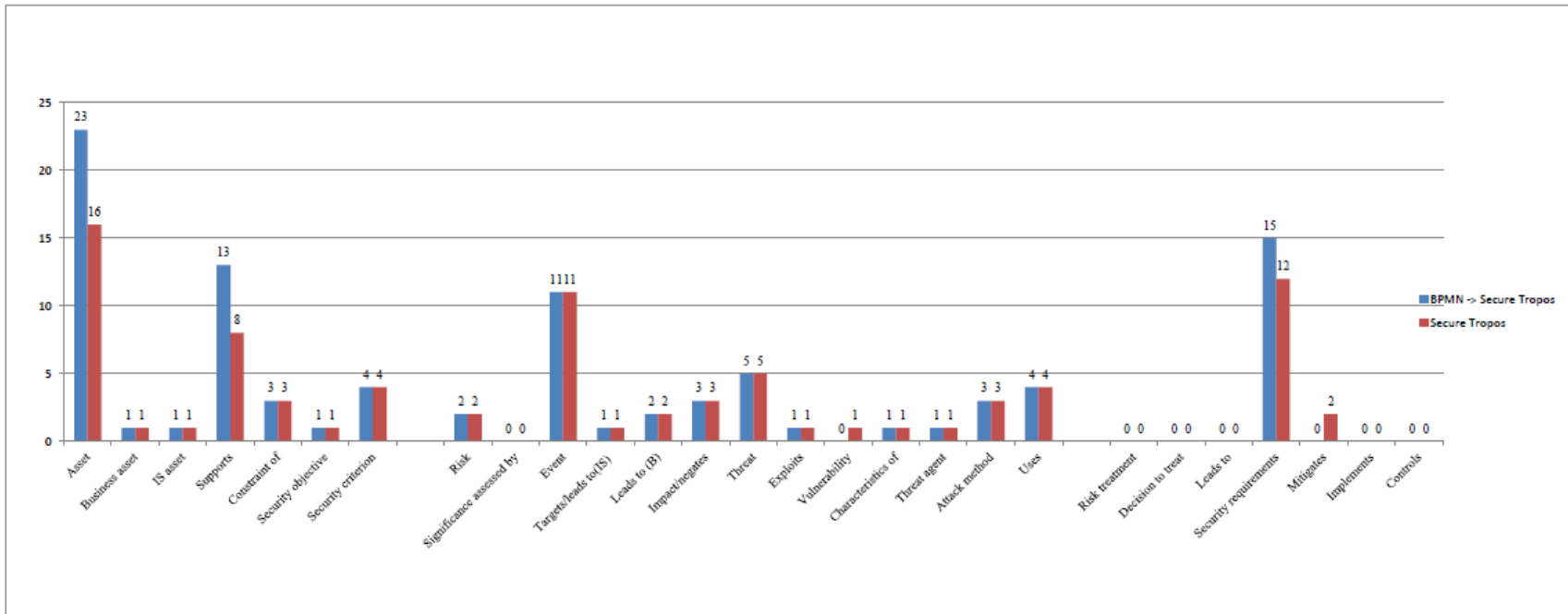


Figure 57. Semantic correctness of models (number of elements used to express ISSRM concepts)

8.6 Summary

It can be summarized that percentage for semantically complete statements is higher for risk and risk treatment models, however taking into account the above described reasons for such a difference, we can assume that transformation is completely possible and will bring the value compatible with the perfect model. We believe that above performed analysis confirms that asset model can be successfully transformed from BPMN to Secure Tropos without any losses. Risk and risk treatment models have small loss of informational value, regarding such concepts as Vulnerability that could not be transformed from BP diagram and risk treatment-related concept Controls, which was not initially defined for BPMN. It was also noticed that transformed model use more resources for representation of some security concept, but the difference in numbers is insufficient and the main objective was still directed to learning the general possibility for model transformation, which is available according to our assessment.

Chapter 9. Conclusions

9.1 Limitations

In this study we are dealing with a limitation of the scope. The study is concentrated on the BPMN *descriptive modelling*, remaining other levels like *analytical* and *executable modelling* for the future research. We also concentrate on the familiar Internet Store example, limiting the business view of language applicability.

The use of BPMN extensions is demonstrated on examples including all the three major security criteria, confidentiality, availability and integrity, however only one criterion is covered by transformation example. In addition we have chosen only one transformation scenario, which represents the translation of model from BPMN to Secure Tropos. We acknowledge the importance to investigate the other possible transformations, such as a reverse transformation from Secure Tropos (Matulevičius *et al.*, 2008b; 2012), as well as different combinations of other security oriented modelling languages like Mal-Activity diagrams (Chowdhury *et al.*, 2012), KAOS extensions to security (Mayer, 2009), misuse cases (Matulevičius *et al.*, 2008a; Soomro and Ahmed, 2012) or others.

9.2 Conclusions

The idea for current study was guided by two major research questions. One of them was the proposal of extensions for BPMN with respect to security engineering. Basing on our previous analysis (Altuhhova *et al.*, 2012) we defined the ability of BPMN 2.0 to express different security concepts and made a conclusion that existing language doesn't cover the ISSRM domain sufficiently. In this work we defined additional constructs and relationships that can be used to cover the bigger part of security domain with respect to informational value of the models. We succeeded to cover the majority of asset- and risk-related concepts, but risk treatment-related concepts were not covered with a scope of current research. To investigate this research question we started with overview of existing security managements techniques, such as CORAS (Lund *et al.*, 2010), Automated and Utility Manager (Elkhart *et al.*, 2009), also Goal-Risk driven Assessment (Asnar *et al.*, 2011) and finally ISSRM (Dubois *et al.*, 2010) in order to perform the comparison between these techniques and explain the decision for ISSRM to be used in our research. The next step was dedicated to introduction on business processes and resource for its modeling. We observe four modelling languages, including UML (Börger *et al.*, 2000), YAWL (Aalst *et al.*, 2005), EPC (Seel *et al.*, 2005) and BPMN (Silver, 2009) and provide a comparison to reason the choice of BPMN for the research. We then provide a short analysis of our previous results (Altuhhova *et al.*, 2012) and give an overview on related studies of BPMN and security. As it was mentioned above, we introduce the extensions to BPMN language on semantic, concrete and abstract levels, covering the ISSRM domain with missing concepts. The use of proposed extensions is illustrated and examined on running examples and validated in a form of experiment. The validation results show that

On the other hand, we emphasized the possibility for transformation of security models from BPMN to Secure Tropos, which allows switching the view from the process flow to a static goal-oriented model. For that purposes we created an Internet Store example with the help of extended

BPMN and transformed the model into Secure Tropos, following the set of predefined transformation rules. To assess the quality of resulted model, we created Secure Tropos model from original context and used it for the comparison. We addressed the semantic completeness and correctness of two Secure Tropos model sets. The percentage for semantic completeness is high for both, transformed and original Secure Tropos models, regarding asset- and risk related concepts. The low percentage for risk-treatment related concepts is justified with the fact that neither Secure Tropos nor BPMN alignment with ISSRM was focused on risk treatment-related concepts. The small difference in percentage is caused by the absence of one (or maximum two) security concepts in transformed model, which, like we concluded, doesn't have a big impact on model informational value regarding security concerns. With respect to semantic correctness, it can be summarized that numbers of concepts representing one or another aspect are slightly different for some ISSRM concepts, but majority of concepts have close numbers of constructs used to express one or another ISSRM concept. So we concluded that (i) model can be successfully transformed from BPMN to Secure Tropos, (ii) the informational value of two models is slightly different, however it can be said that difference in percentage is not essential and doesn't have a serious impact.

9.3 Future work

As we already discussed in the work limitation section, we would like to investigate into two other levels of BPMN, *analytical* and *executable modelling* that possibly will open a new and interesting view on business process modelling and security analysis in business processes. Next step will be the further study on transformation possibilities. We would like to perform the reverse transformation for Secure Tropos and BPMN, and also investigate into some other security oriented languages from that perspective. Hopefully it will require the additional validation, so planning and designing of the experiment to validate the quality and importance of transformed models will be also held in future.

The most enthusiastic idea for the future work still remains a creation of a modelling tool, supporting extended BPMN as well as number of other security oriented languages, and allowing generating the models from different points of view, or transforming model from one language to another.

Modelleerimiskeele laienduste väljatöötamine turvalisuse riskide juhtimiseks äriprotsesside analüüsimisel

Olga Altuhhova

Magistritöö

Kaasaegsed infosüsteemide arendamise meetodikad hõlmavad erinevaid tehnilisi äriprotsesside modelleerimise meetmeid. Äriprotsesside modelleerimiseks kasutatav keel (BPMN¹) on tänapäeval muutunud üheks standartseks meetmeks, mis edukalt rakendatakse infosüsteemide loomisel ning edasi arendamisel selleks, et ettevõtete äriprotsesse kirjeldada ja modelleerida. Vaatamata sellele, et BPMN on hea tööriist, mille abil on võimalik ettevõtte äriprotsesse mõistma ja esitama, see ei võimalda äriprotsesside modelleerimisel adresseerida süsteemi turvalisuse aspekte. Autor leiab, et see on BPMN nõrk külg, selle pärast, et turvalise infosüsteemi arendamiseks on oluline nii äriprotsesse kui ka süsteemi turvalisust vaadeldada tervikuna. Käesolevas magistritöös autor töötab välja BPMN 2.0 keele jaoks uusi elemente, mis edaspidi peavad võimaldama adresseerima turvalisuse temaatika süsteemi modelleerimisel. Autori pakutud lahendus põhineb BPMN modelleerimiskeele seostamisel turvalisuse riski juhendamise meetodikaga (ISSRM). Antud magistritöös rakendatakse struktureeritud lähenemine BPMN peamiste aspektide analüüsimisel ja turvalisuse riskide juhtimiseks uute elementide väljatöötamisel, selleks ühildades BPMN ning ISSRM-i kontsepte.

Magistritöös on demonstreeritud väljatöötatud lisaelementide kasutus, selgitatud kuidas antud elementidega laiendada BPMN võimaldab väljendada ettevõtte varasid (*assets*), nendega seotuid riske (*risks*) ja riskide käsitlust (*risk treatment*). See on analüüsitud internetkaupluse varade konfidentsiaalsuse, terviklikkuse ja kättesaadavuse näitel. Autor on veendunud, et BPMN laienemine turvalisuse kontseptide osas ja antud töö raames tehtud konkreetset ettepanekud aitavad infosüsteemide analüütikutele mõistma kuidas süsteemi turvalisust arendada nii, et läbi äriprotsessi tuvastatud olulisemate ettevõtte varade turvalisus oleks infosüsteemis käsitletud ning tagatud. Autori poolt antud käsitus on vaadeldud ka laiemas mõttes, nimelt, BPMN keelele pakutud laienemisega avaneb perspektiiv äriprotsesside ja turvalisuse mudeleite koosvõimele ning BPMN-i teiste modelleerimise meetodikatega, nagu ISSRM või Secure Tropos, integreerimisele.

¹ Business Process Model and Notation

References

1. Alberts C. J., & Dorofee A. J. (2001). OCTAVE Method Implementation Guide Version 2.0. Carnegie Mellon University. Software Engineering Institute, Pennsylvania.
2. Altuhhova O., Matulevičius R., Ahmed N.,: Towards Definition of Secure Business Processes. M. Bajec and J. Eder (Eds.): CAiSE 2012 Workshops, LNBIP 112, pp. 1-15, Springer-Verlag Berlin Heidelberg (2012)
3. AS/NZS 4360 (2004). Risk management. SAI Global.
4. Asnar, Y., Giorgini, P., Mylopoulos, J.,: Goal-driven risk assessment in requirements engineering, Requirements Engineering, pp 101-116, Springer (2011)
5. Börger, E., Cavarra, A., Riccobene, E.: An ASM Semantics for UML Activity Diagrams. In: Proceedings of the 8th AMAST 2000, pp. 293-308. Springer, Heidelberg (2000)
6. Braber, F., Hogganvik, I., Lund, M. S., Stølen, K., & Vraalsen, F. (2007). Model-based Security Analysis in Seven Steps—a Guided Tour to the CORAS Method. BT Technology Journal, vol. 25(1) (pp.101–117).
7. Bresciani P., Perini A., Giorgini P., Fausto G. and Mylopoulos J., “*TROPOS: an Agentoriented Software Development Methodology*”. Journal of Autonomous Agents and Multi-Agent Systems, Volume 25, pages 203–236, 2004
8. Cherdantseva Y., Hilton J., Rana O.: SecureBPMN – a New Approach to Achieving Synergy between Information Security and Business Process Modelling. (Feb. 2012)
9. Cherdantseva Y., Hilton J., Rana O.:Towards SecureBPMN – Aligning BPMN with the Information Assurance & Security Domain, Lecture Notes in Business Information Processing Volume 125, 2012, pp 107-115 (Sept. 2012)
10. Chowdhury M. J. M., Matulevičius R., Sindre G., & Karpati P. (2012). Aligning Mal-activity Diagrams and Security Risk Management for Security Requirements Definitions. In Proceedings of REFSQ 2012, LNCS 7195 (pp 135-139). (in press)
11. Common Criteria, (2005). Common Criteria for Information Technology Security Evaluation, version 2.3, CCMB-2005-08-002.
12. Dubois, E., Heymans, P., Mayer, N., Matulevičius, R.: A Systematic Approach to Define the Domain of Information System Security Risk Management. In: Intentional Perspectives on Information Systems Engineering. pp. 289-306. Springer (2010)
13. Ekelhart, A., Fenz, S., Neubauer, T.: AURUM: A Framework for Information Security Risk Management, Proceedings of the 42nd Hawaii International Conference on System Sciences (2009)
14. Firesmith, D. G.: Common Concepts Underlying Safety, Security, and Survivability Engineering, Technical Note CMU/SEI-2003-TN-033 (2003)
<http://www.tse.org.tr/turkish/belgelendirme/ortakkriter/ccpart2v2.3.pdf>
15. ISO/IEC Guide 73. (2002). Risk management - Vocabulary - Guidelines for use in standards. International Organization for Standardization, Geneva
16. Jurjens J. (2005) Secure Systems Development with UML. Berlin Heidelberg. Springer-Verlag.
17. Lund, M.S., Solhaug B., Stølen, K.: Model-Driven Risk Analysis. The CORAS Approach, Springer (2010)

18. Matulevičius R., Lakk H. and Lepmets M., “*An approach to assess and compare quality of security models*”. ComSIS, Volume 8 No 2, Special Issue, (2011)
19. Matulevičius R., Mayer, N., Mouratidis, H., Dubois, E., Heymans, P., & Genon, N. (2008b). Adapting Secure Tropos for Security Risk Management during Early Phases of the Information Systems Development. In Proceedings of CAiSE’08, (pp. 541-555). Springer
20. Matulevičius, R., Mayer, N., Heymans, P.: Alignment of Misuse Cases with Security Risk Management. In: Proceedings of ARES 2008, pp. 1397–1404. IEEE (2008a)
21. Matulevičius, R., Mayer, N., Mouratidis, H., Martinez, F.H., Heymans, P., Genon, N.: Adapting Secure Tropos for Security Risk Management in the Early Phases of Information Systems Development. In: Bellahsene, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 541–555. Springer, Heidelberg (2008b)
22. Mayer, N.: Model-based Management of Information System Security Risk. Doctoral Thesis, University of Namur (2009)
23. Menzel M., Thomas I., Meinel C.: Security Requirements Specification in Service-oriented Business Process Management. ARES 2009, 41-49 (2009)
24. Mülle J., Stackelberg S., Bohm K.: A Security Language for BPMN Process Models. Karlsruhe Reports in Informatics (2011)
25. Remco, M., Dijkman, R.M., Dumas, M., & Ouyang, C. (2007). Formal Semantics and Analysis of BPMN Process Models using Petri Nets. In Journal Information and Software Technology. Elseiver.
26. Seel, C., Vanderhaeghen, D.: Meta-Model Based Extensions of the EPC for Inter-Organisational Process Modelling. In: Proceedings of the 4th GI-Workshop EPK 2005 – Geschäftsprozessmanagement (2005)
27. Silver, B.: BPMN Method and Style: A Levels-based Methodology for BPMN Process Modeling and Improvement using BPMN 2.0, Cody-Cassidy Press (2009)
28. Soomro, I., & Ahmed, N. (2012) Towards Security Risk-oriented Misuse Cases. In Proceedings of the of Business Management Workshops, BPM 2012 workshops, LNBIP, vol 132, (pp. 673-684)
29. Soomro, I., & Ahmed, N. (2012) Towards Security Risk-oriented Misuse Cases. In Proceedings of the of Business Management Workshops, BPM 2012 workshops, LNBIP, vol 132, (pp. 673-684)
30. Stoneburner, G., Goguen, A., & Feringa, A. (2002). NIST Special Publication 800-30: Risk Management Guide for Information Technology Systems. National Institute of Standards and Technology, Gaithersburg.
31. van der Aalst, W.M.P., ter Hofstede, A.H.M. : YAWL: Yet Another Workflow Language. Information Systems, 30(4), pp 245–275 (2005)
32. White, S.A.: Introduction to BPMN, IBM, [http://www.bpmn.org/Documents/Introduction to BPMN.pdf](http://www.bpmn.org/Documents/Introduction%20to%20BPMN.pdf) (2004).

Appendix A

The following Appendix is a part of “An Extension of Business Process Model and Notation for Security Risk Management” study. It demonstrates the step by step model transformation from security risk-oriented BPMN to Secure. It covers transformation of three different business cases, considering Internet Store assets availability, integrity and confidentiality. The Appendix is addressed in Chapter 6, and represents the extended part of a study described in the chapter, focusing strictly on rules. Description regarding the examples contexts is available in the thesis.

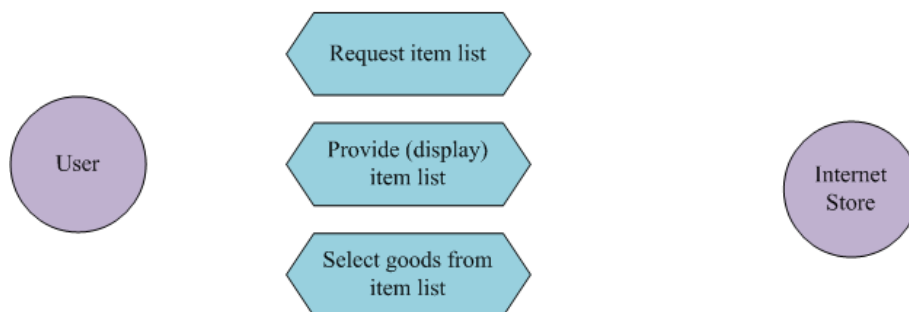
A1. Availability example transformation

A.1.1 Availability: Assets

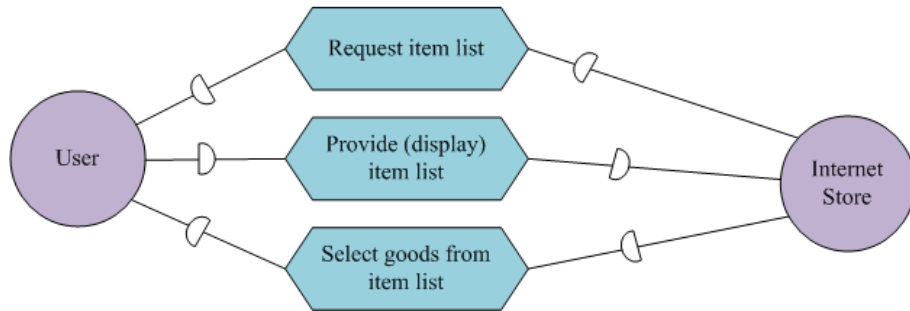
TR1. Define the stakeholders. Transform BPMN *Pools* and *Lanes* to Secure Tropos *Actors*. BPMN *Pool/Lane* that represents a process participant (e.g. job, name, system) is simply transformed to *Actor*. If BPMN *Lanes* are used as a representation of different functional parts of one working system, there is no need to transform each part, transforming the general naming of described system (often *Pool* name) will be essential in Secure Tropos.



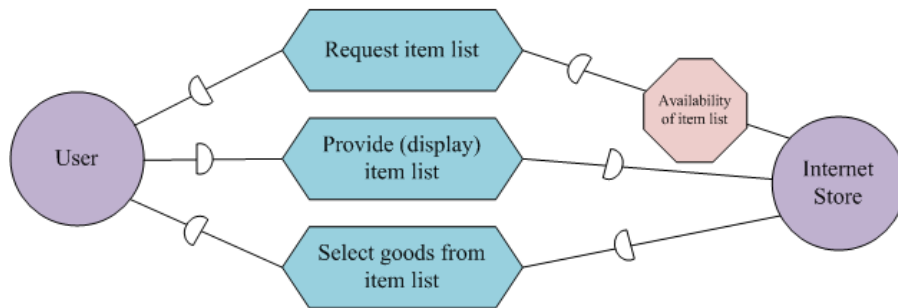
TR2. Another characteristic of Secure Tropos is ability to define dependencies between *Actors*. For dependencies definition transform each BPMN *Task* that represent any activity of communication between participants (e.g. send, receive, request, get, or any other of that kind) to Secure Tropos *Plan*, which will become a dependum. Dependencies can be also represented by BPMN *Message flow* with the same mission: to send something, request or receive. Transform *Message flow* to Secure Tropos dependums.



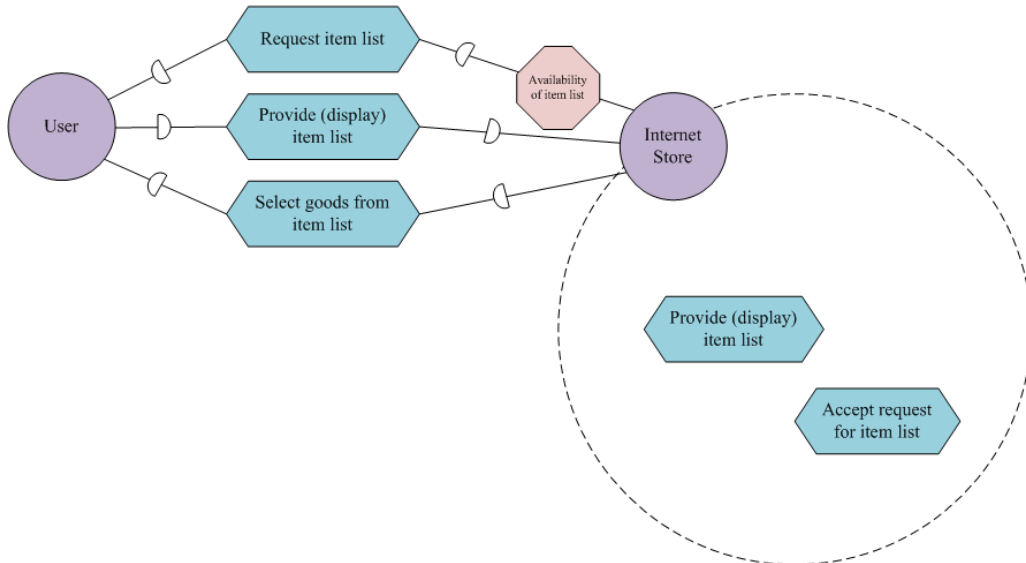
TR3. Next step is to add a direction to previously defined dependencies; to decide who is a depender and who is a dependee. A depender in BPMN is usually somebody who performs the action, a dependee is the one who depends on this action. In case with the transformation of *Message flow*, the direction of dependency is specified with direction of observed *Message flow*; if it goes out from the *Pool* that means we are dealing with a depender, and conversely.



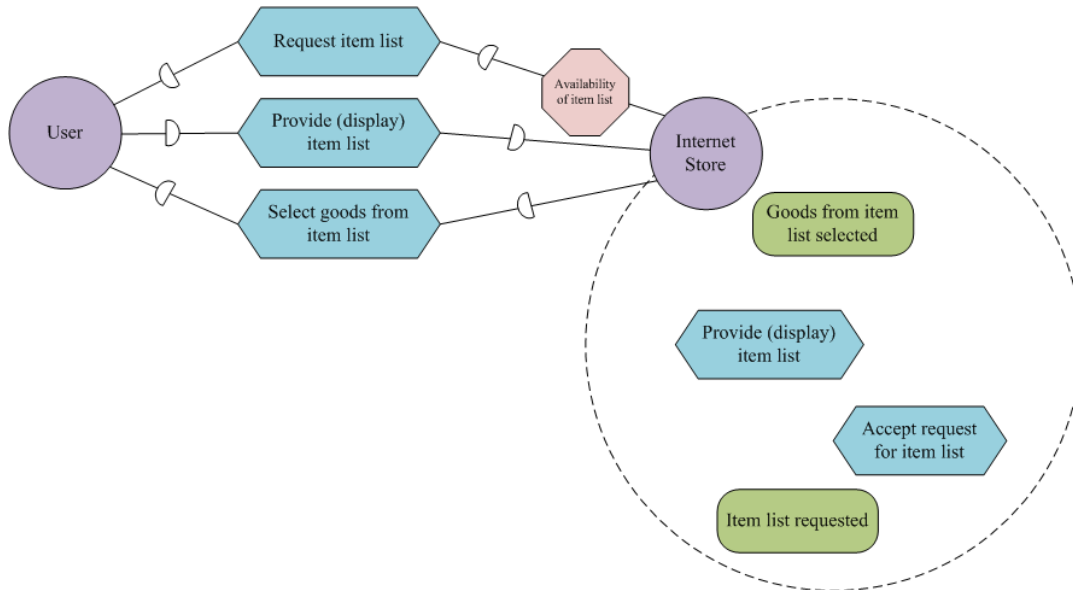
TR4. Define the *Security constraint* that will regulate the dependency between *Actors*. *Security constraint* that depends expects to be satisfied, is transformed from BPMN Security objective.



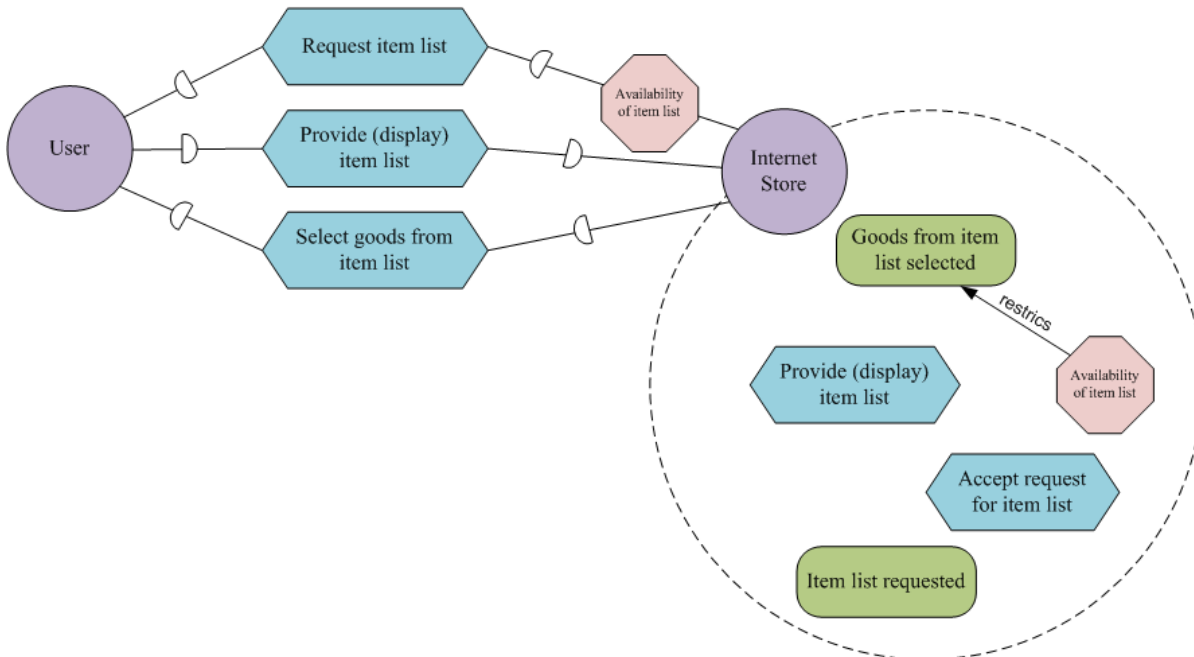
TR5. Transform BPMN *Tasks* to Secure Tropos *Plans* and add them to the corresponding Actor's boundary; add only *Plans* that are under responsibility of the observed Actor.



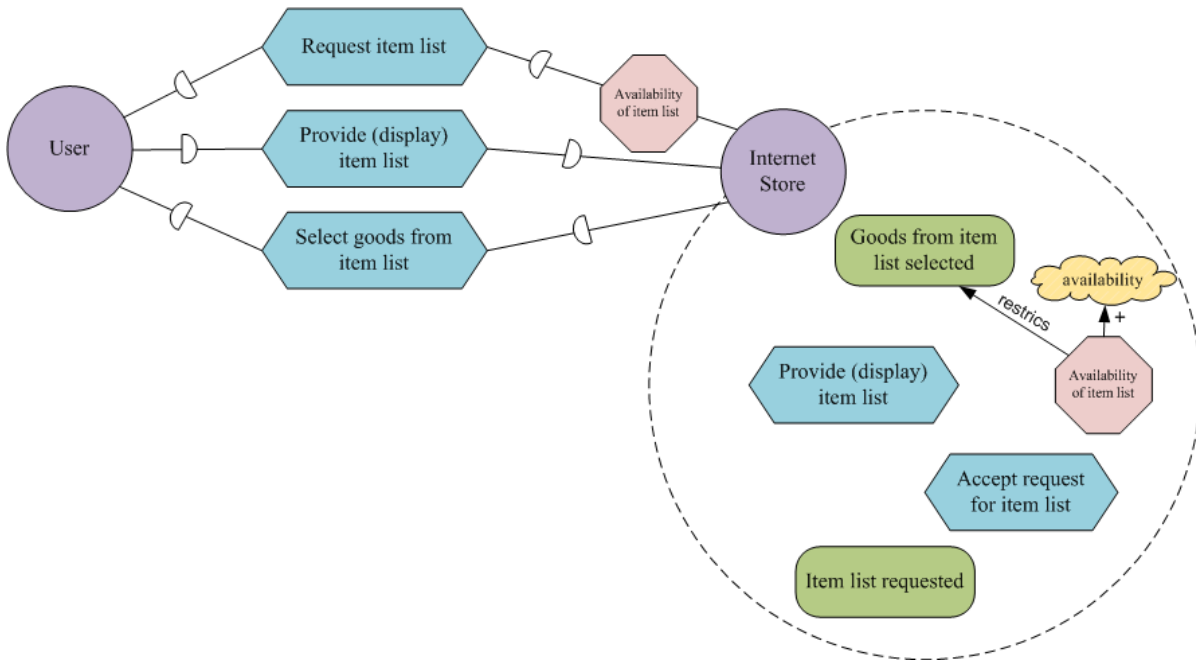
TR6. Next step is dedicated to completion of Secure Tropos Model with *Goals*. *Goals* can be formed from already transformed BPMN *Tasks*, meaning from *Plans* added on the previous step, or from BPMN *Tasks*. Transform Secure Tropos *Goals* in the following form of event: Task Request item list to *Goal* Item list requested.



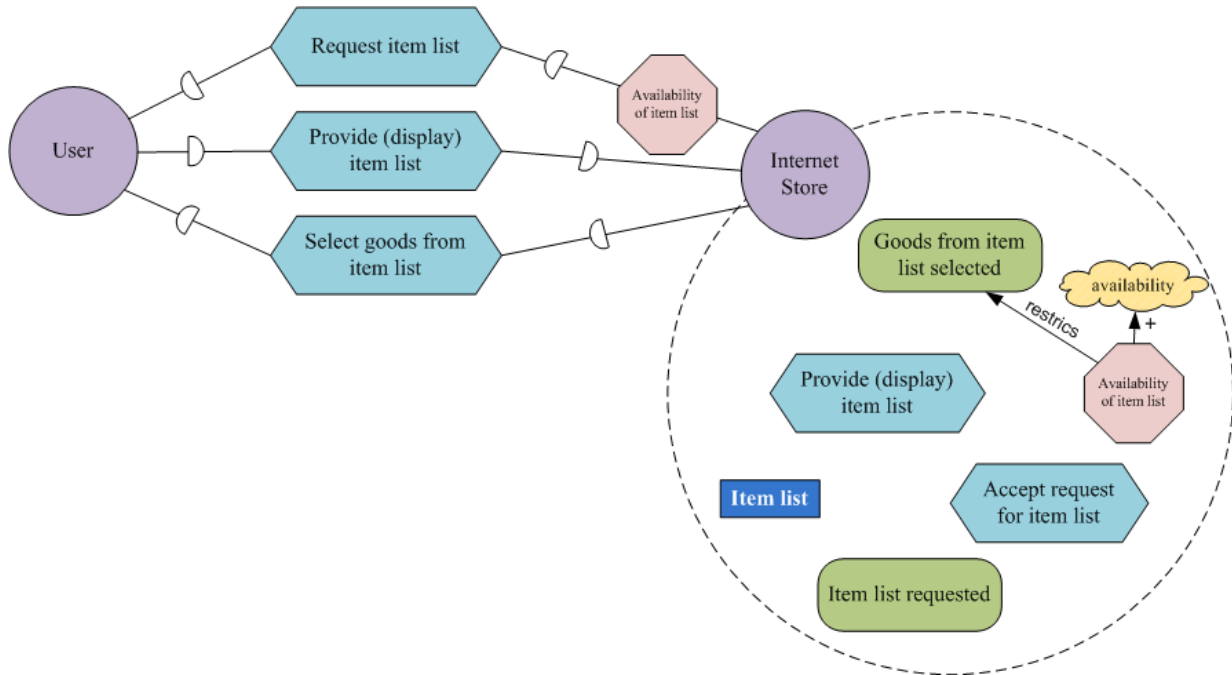
R7. Duplicate the *Security constraint* to the boundary of an *Actor*. Add *Restricts* relationship between *Security constraint* and the *Goal*. This relationship should correspond to the union of BPMN *Tasks* restricted with a *Lock* defining *Security criterion*.



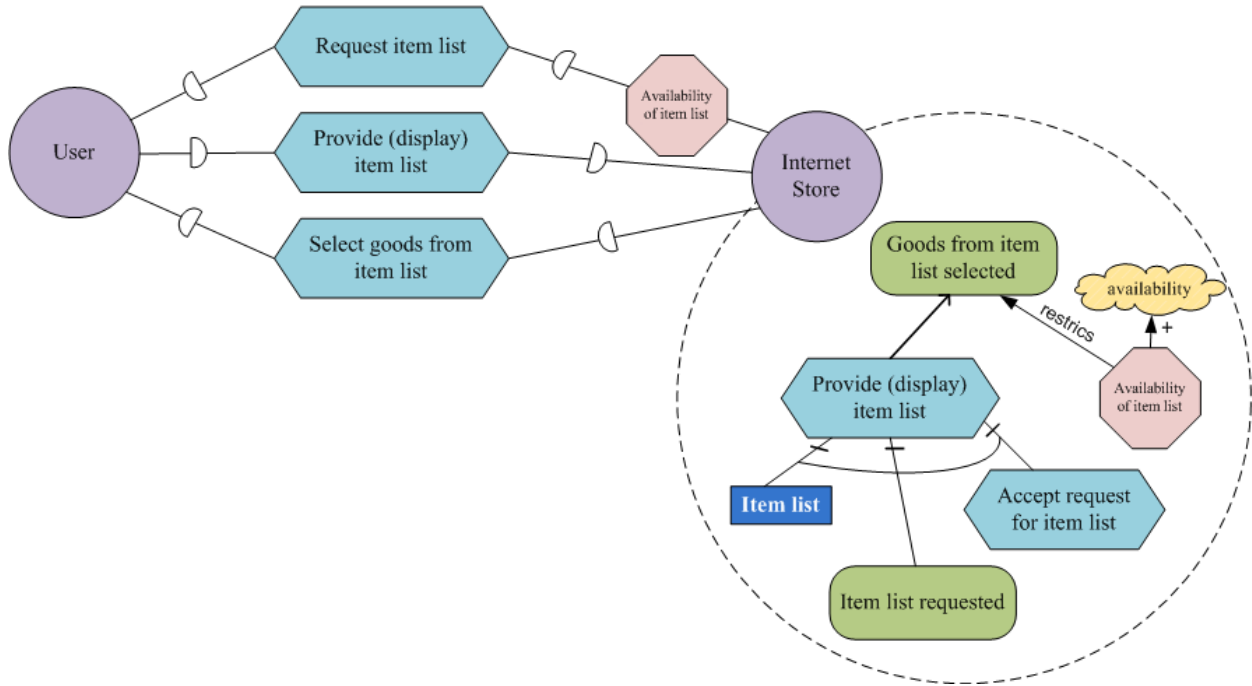
TR8. Define the *Softgoal* and add it to the *Actor's* boundary. The *Softgoal* should comply with *Security objective*, which is a property of the BPMN *Lock*. Add *Contribution* relationship between *Security constraint* and the *Softgoal*.



TR9. Add *Resources* to Secure Tropos model. *Resources* are transformed from BPMN *Data Object* or *Data Store*, in other words BPMN *Artefacts* can be transformed to Secure Tropos *Resources*.



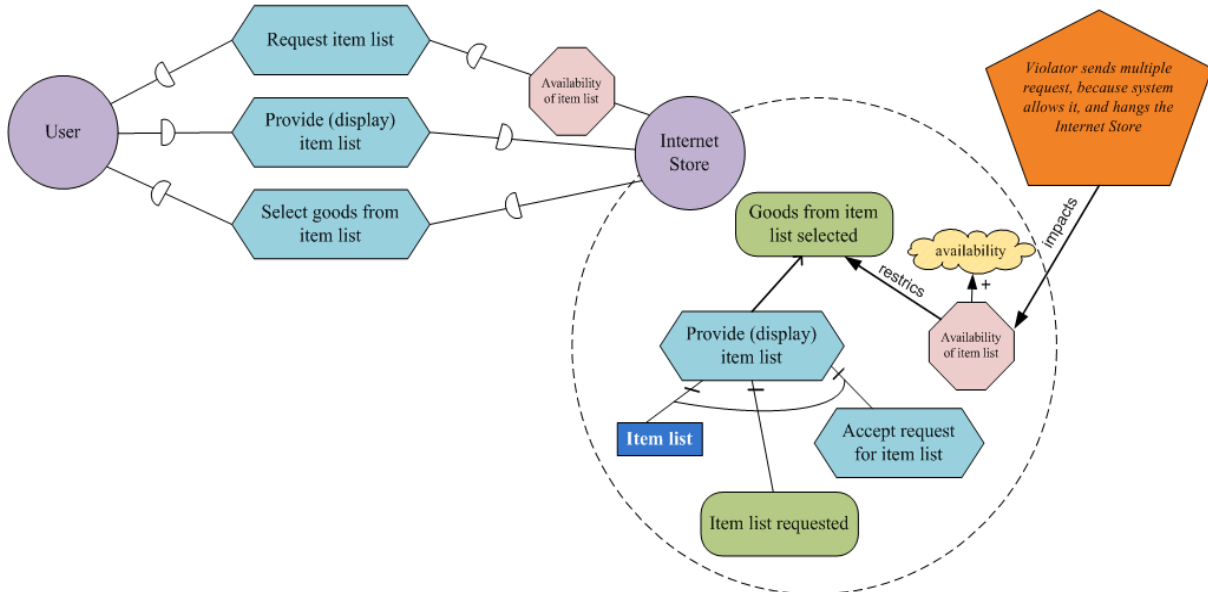
Tropos model should be complemented with some manual add-ons. Define the rest of relationships, such as *Decomposition* and *Means-ends*.



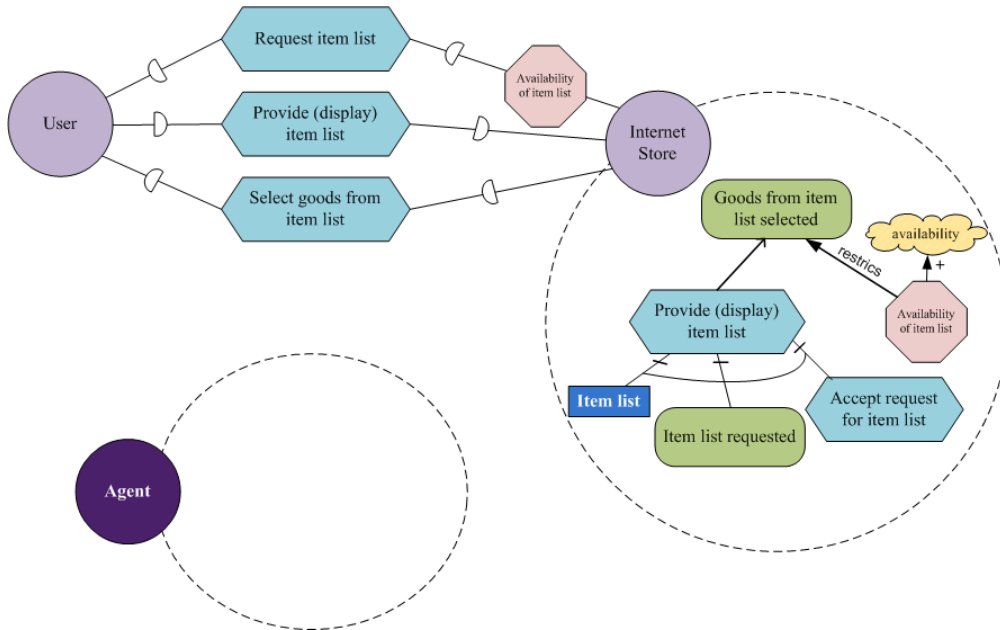
A.1.2 Availability: Risk

Next stage of transformations is creating Secure Tropos Risk model from BP Diagram. The latter comes as an input.

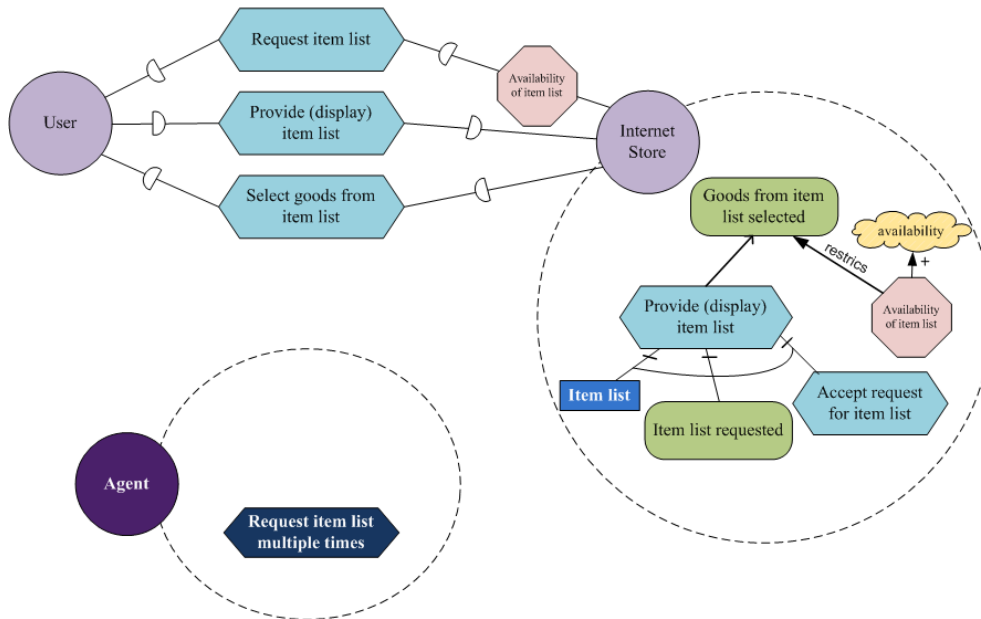
TR10. Start from defining the impact of the risk described in BPMN model. Add Secure Tropos *Threat* and the *Impacts* relationship between *Threat* and corresponding *Security constraint*.



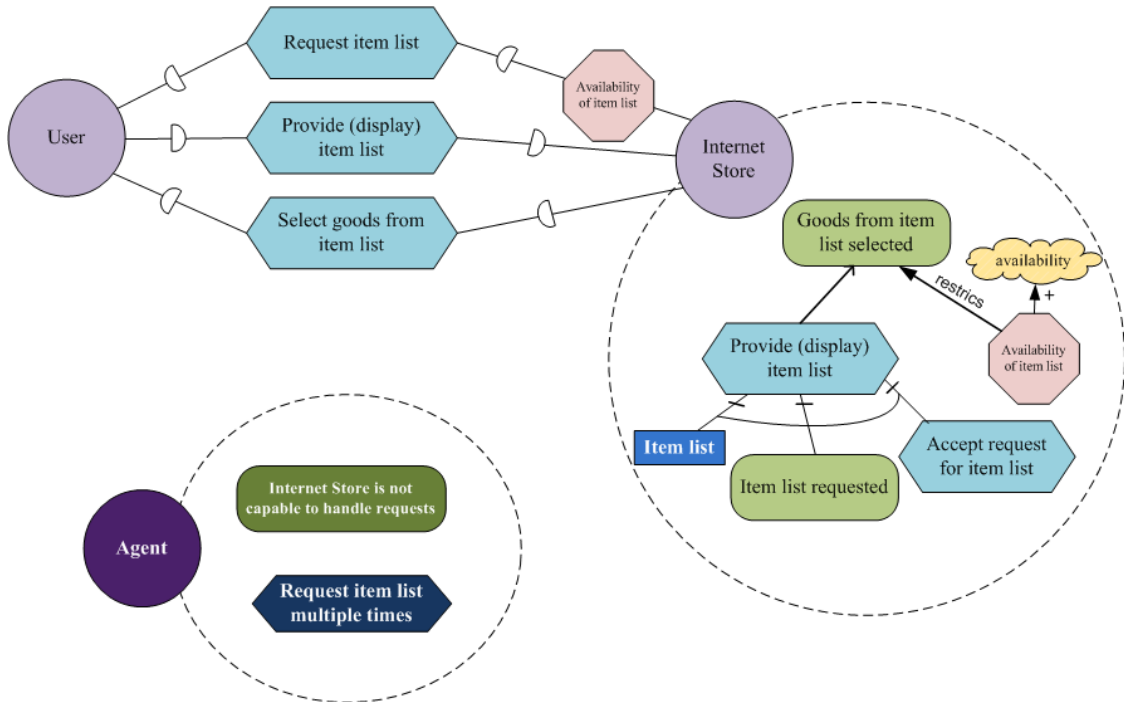
TR11. Basing on the rule **TR1**, define the attacker as a new Secure Tropos *Actor* with its boundary.



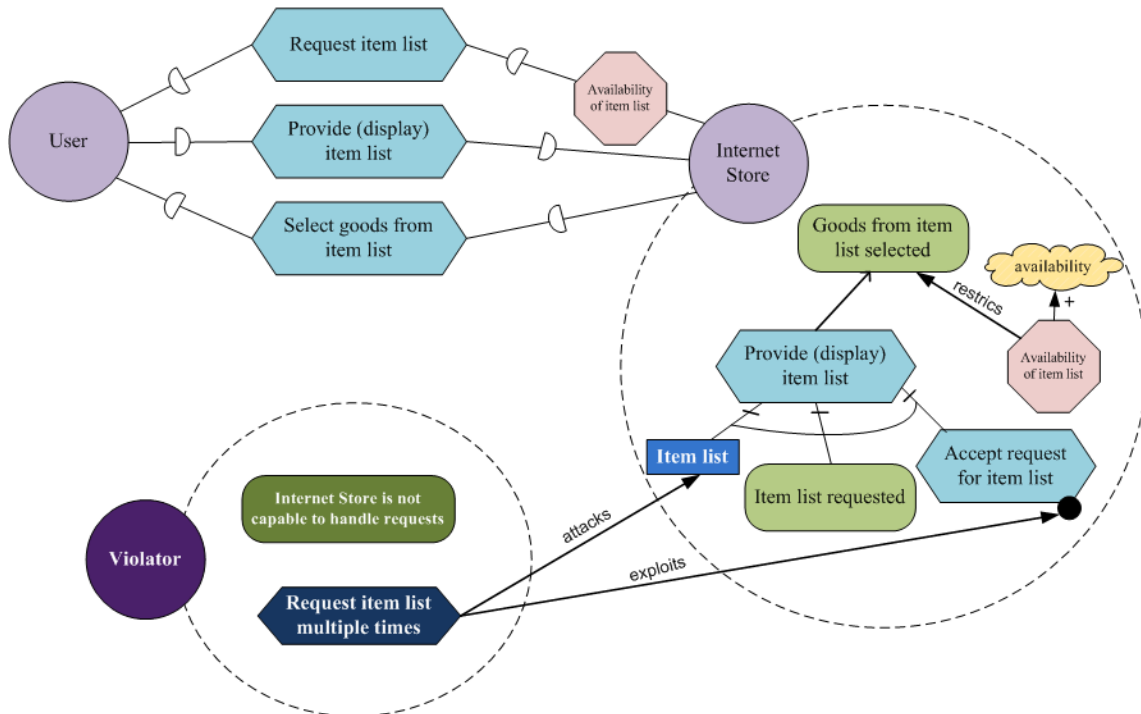
TR12. Transform BPMN *Tasks* to Secure Tropos *Plans* that represent an attack method. The following rule is based on **TR5**. It can be also noticed that we do not define any dependencies between Agent and other Actors, so BPMN *Message flow* from the *Pool* that represents a threat agent can be transformed to Secure Tropos *Plan*.



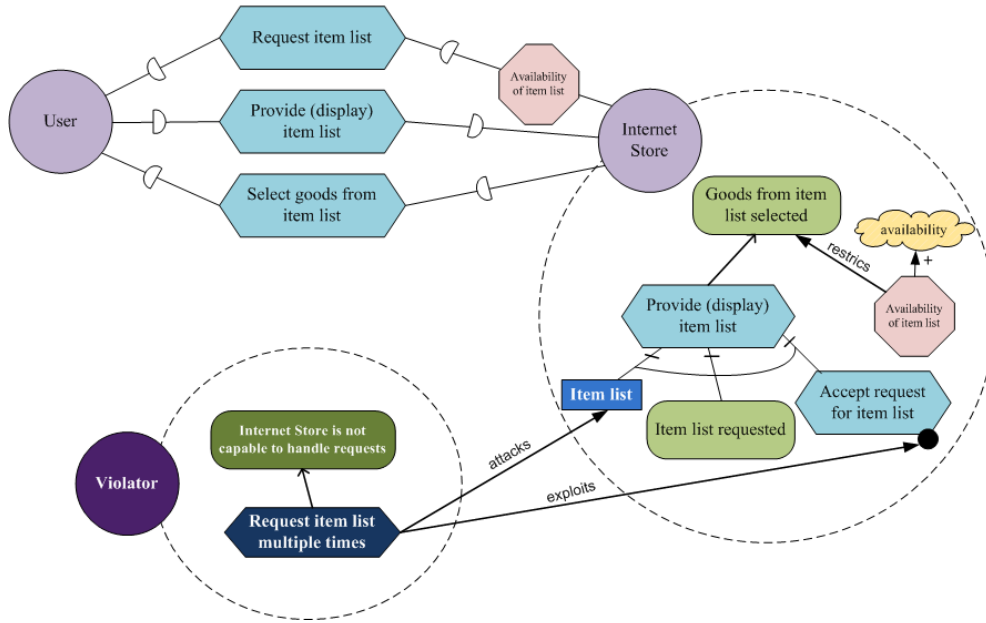
TR13. Define the main goal of the attacker from BPMN diagram; it can be formed from Task or Event (usually some undesirable end event). Once it is defined, transform it to Secure Tropos *Goal*.



TR14. Next step is to detect the *Vulnerability point* from BP Diagram. Secure Tropos *Vulnerability point* should be added to the *Plan (Goal)* or *Resource* that correspond to *Task* or *Artifact* that carried the vulnerability in BPMN model. Put *Exploits* relationship between *Plan* (representing attack method) and *Vulnerability point*. Add the *Attacks* relationship between the *Plan* and *Resource* being attacked.



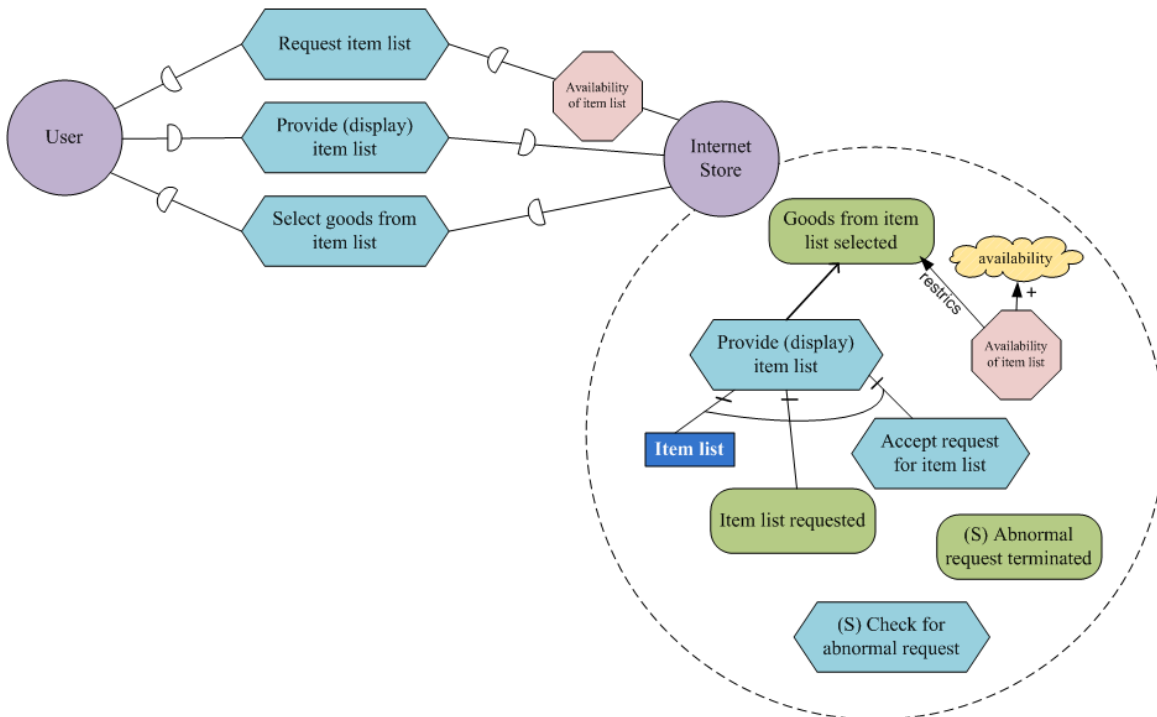
Define the rest of relationships, such as *Decomposition* and *Means-ends* in the boundary of an attacker.



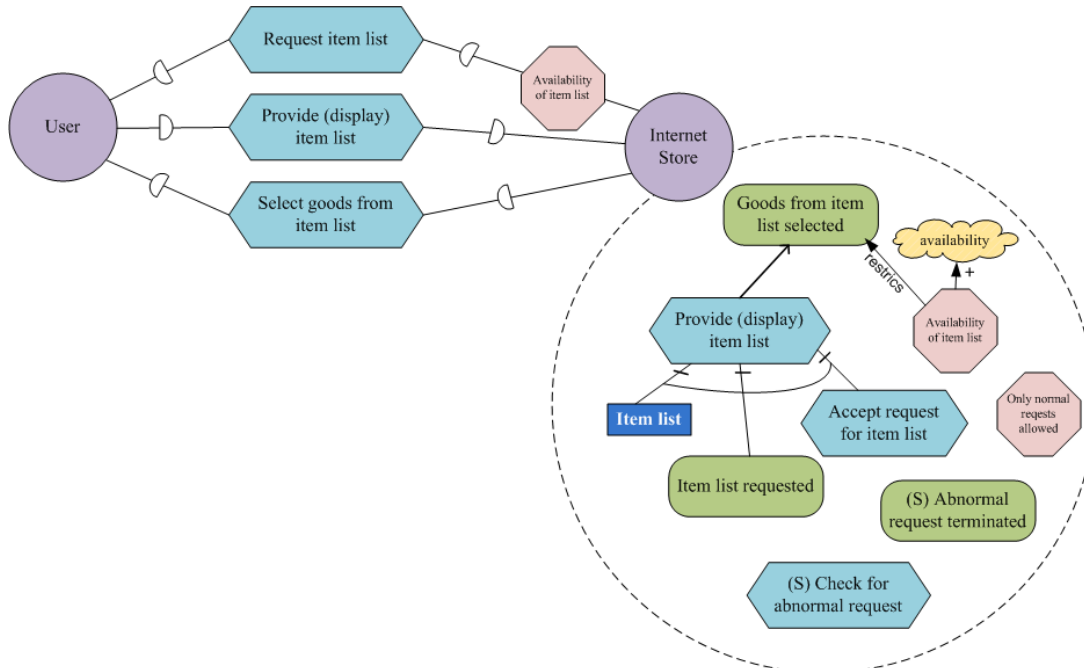
A.1.3 Availability: Risk Treatment

As an input we take asset model in Secure Tropos. Transformation is based on BP Diagram, describing treatment options.

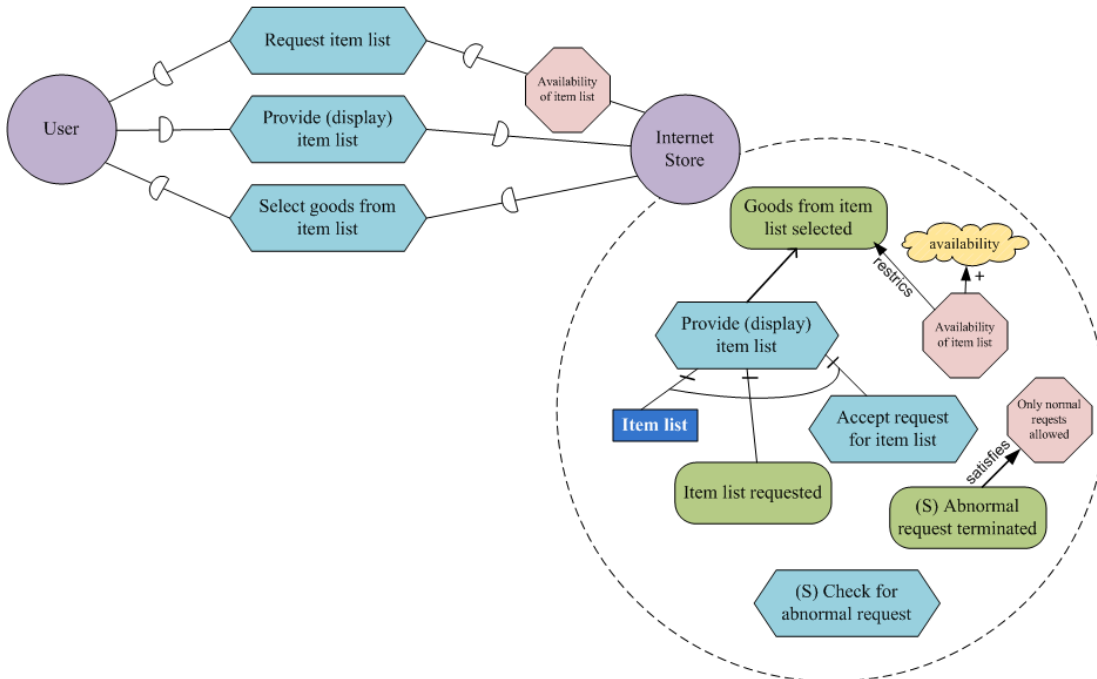
TR15. Transform BPMN Tasks and Gateway structures that represent security requirements into Secure Tropos Plans and Goals, basing on rules **TR5**, **TR6**. Add (S)-labels to all defined Plans, Goals that will emphasize security requirements.



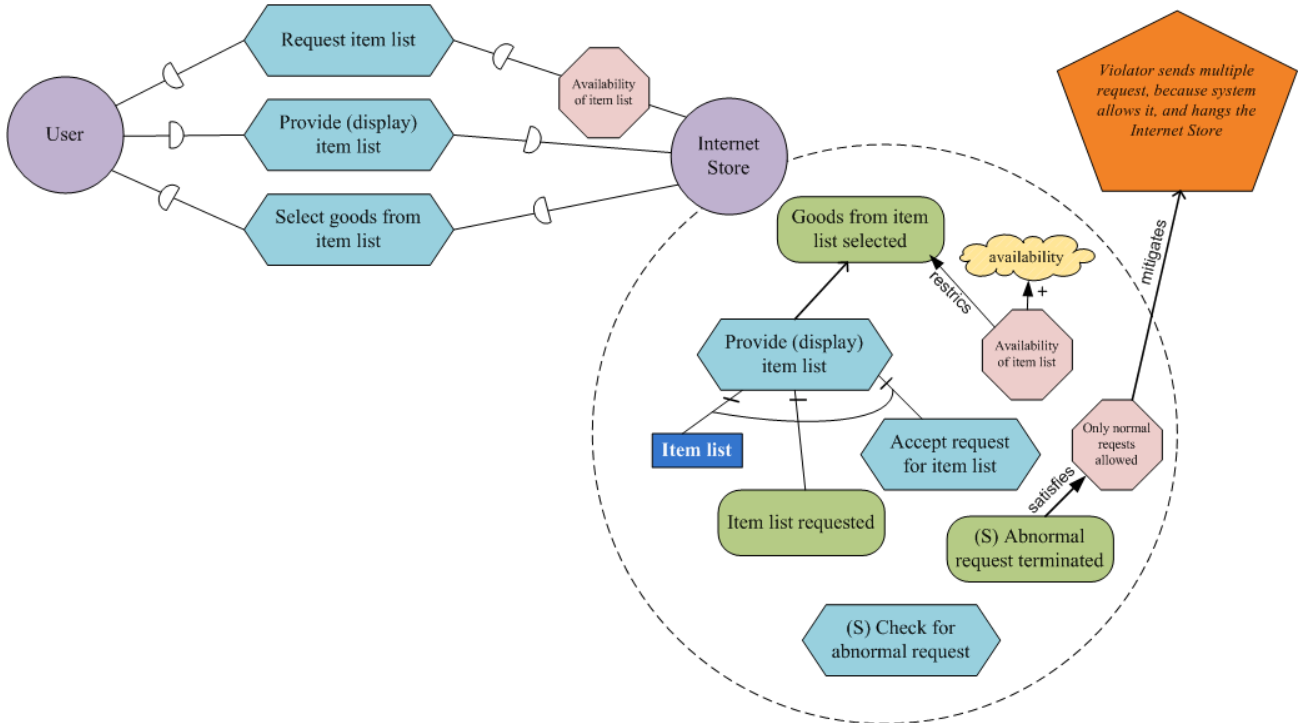
TR16. Define additional *Security constraint*(s). It is possible through transforming the BPMN Gateway structures that represent security requirements control.



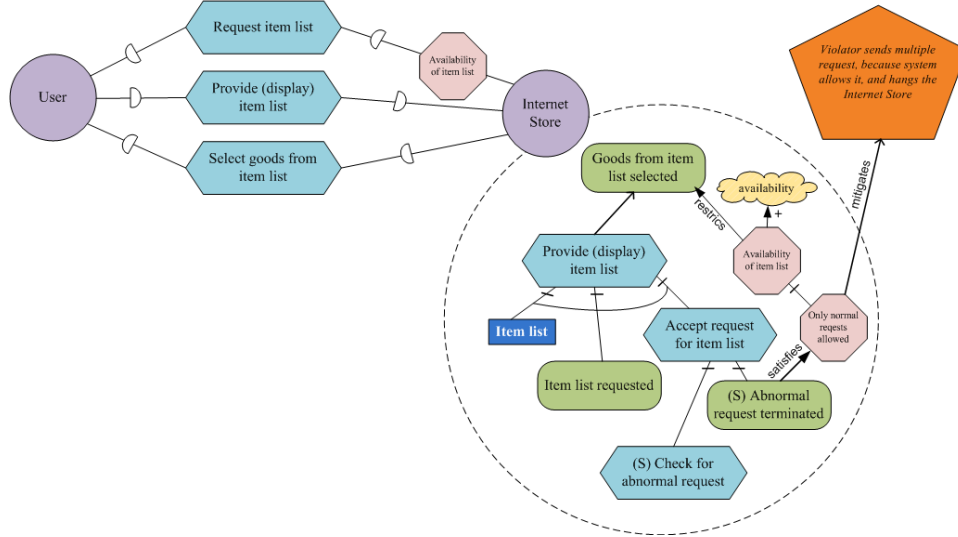
TR17. Add *Satisfies* relationship between defined *Goal*(s) and existing *Security constraints*.



TR18. Collapse the risk scenario to an *Impact*. Add *Mitigates* relation between *Security constraint* and the *Impact*.



Add missing relationships (*Decomposition, Means-end*) manually to complete the model.



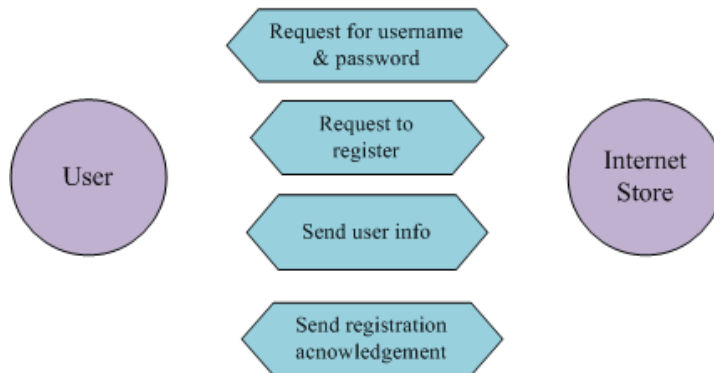
A2. Confidentiality example transformation

A 2.1 Confidentiality: Assets

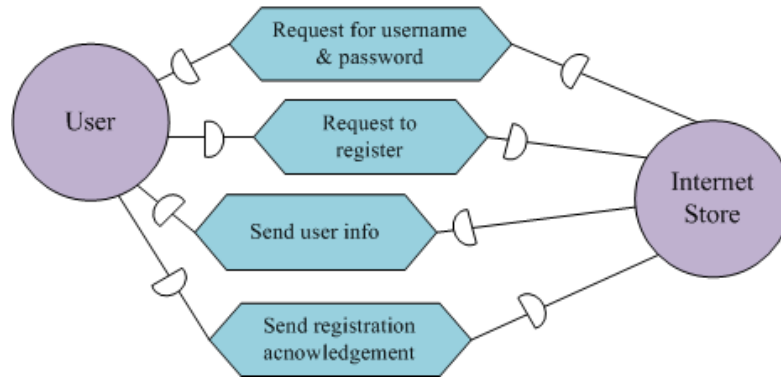
TR1. Define the stakeholders. Transform BPMN *Pools* and *Lanes* to Secure Tropos *Actors*. BPMN *Pool/Lane* that represents a process participant (e.g. job, name, system) is simply transformed to *Actor*. If BPMN *Lanes* are used as a representation of different functional parts of one working system, there is no need to transform each part, transforming the general naming of described system (often *Pool* name) will be essential in Secure Tropos.



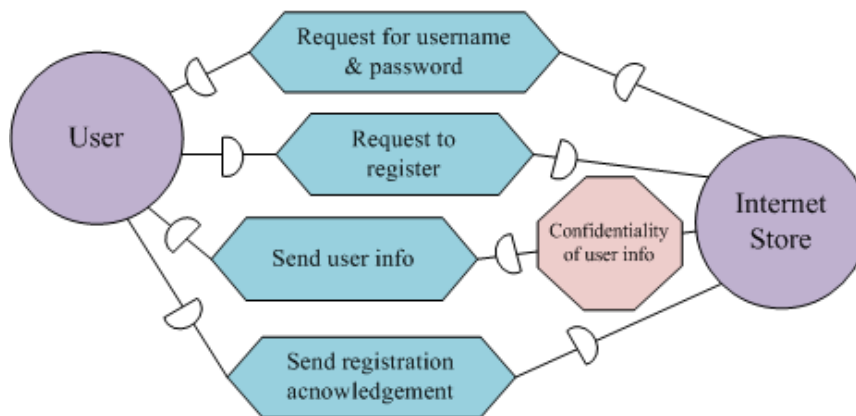
TR2. Another characteristic of Secure Tropos is ability to define dependencies between *Actors*. For dependencies definition transform each BPMN *Task* that represent any activity of communication between participants (e.g. send, receive, request, get, or any other of that kind) to Secure Tropos *Plan*, which will become a dependum. Dependencies can be also represented by BPMN *Message flow* with the same mission: to send something, request or receive. Transform *Message flow* to Secure Tropos dependums.



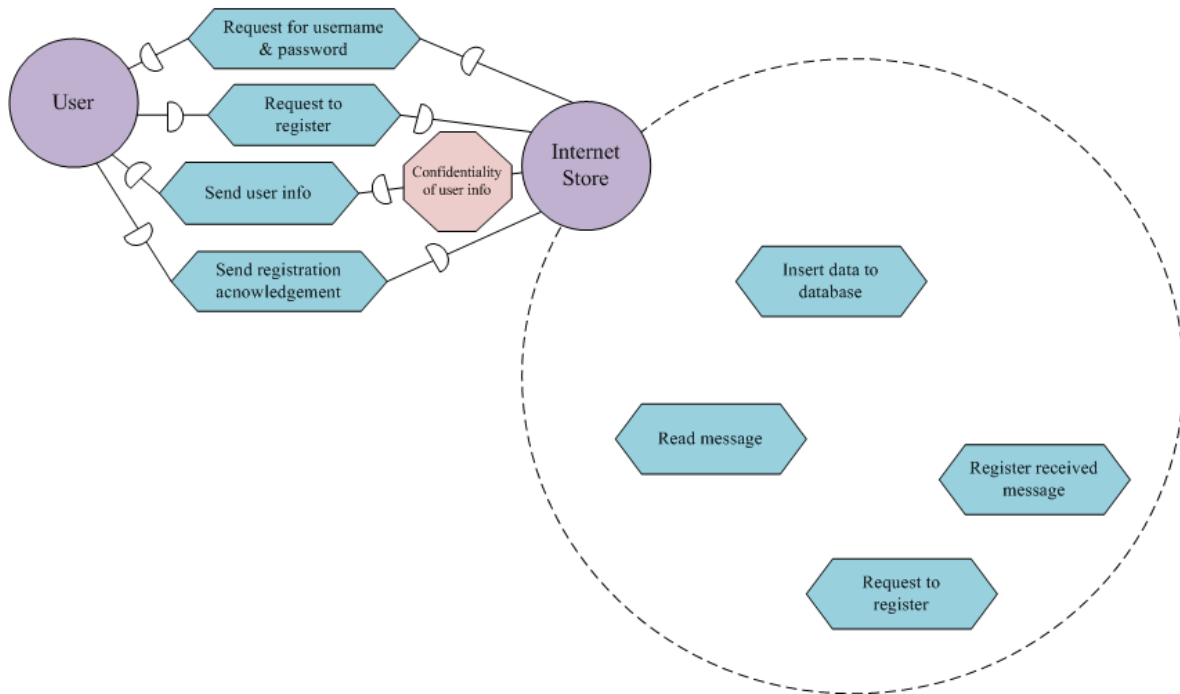
TR3. Next step is to add a direction to previously defined dependencies; to decide who is a depender and who is a dependee. A depender in BPMN is usually somebody who performs the action, a dependee is the one who depends on this action. In case with the transformation of *Message flow*, the direction of dependency is specified with direction of observed *Message flow*; if it goes out from the *Pool* that means we are dealing with a depender, and conversely.



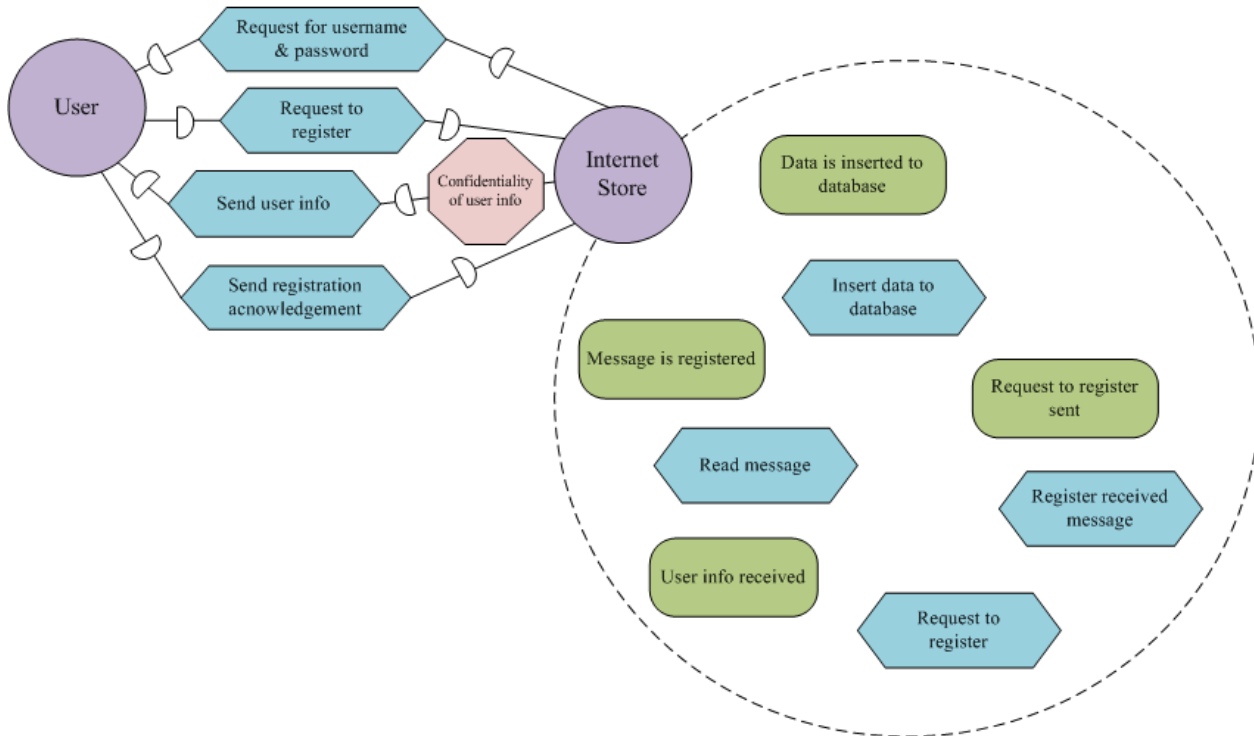
TR4. Define the *Security constraint* that will regulate the dependency between *Actors*. *Security constraint* that dependers expects to be satisfied, is transformed from BPMN Security objective.



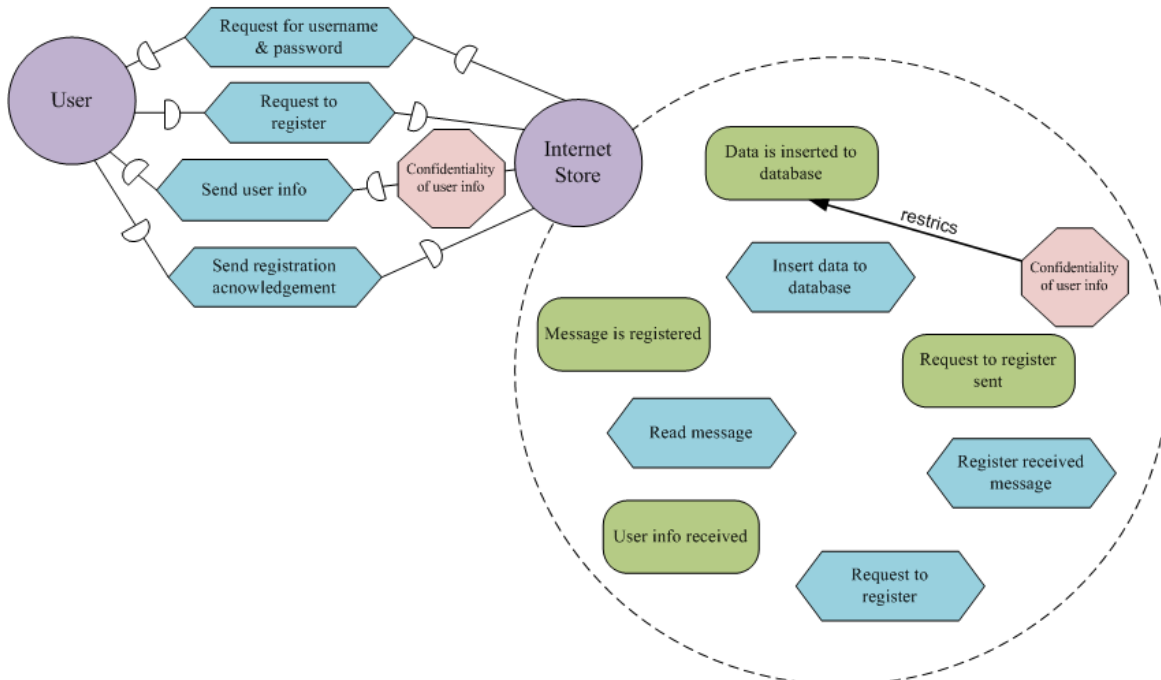
TR5. Transform BPMN *Tasks* to Secure Tropos *Plans* and add them to the corresponding Actor's boundary; add only *Plans* that are under responsibility of the observed Actor.



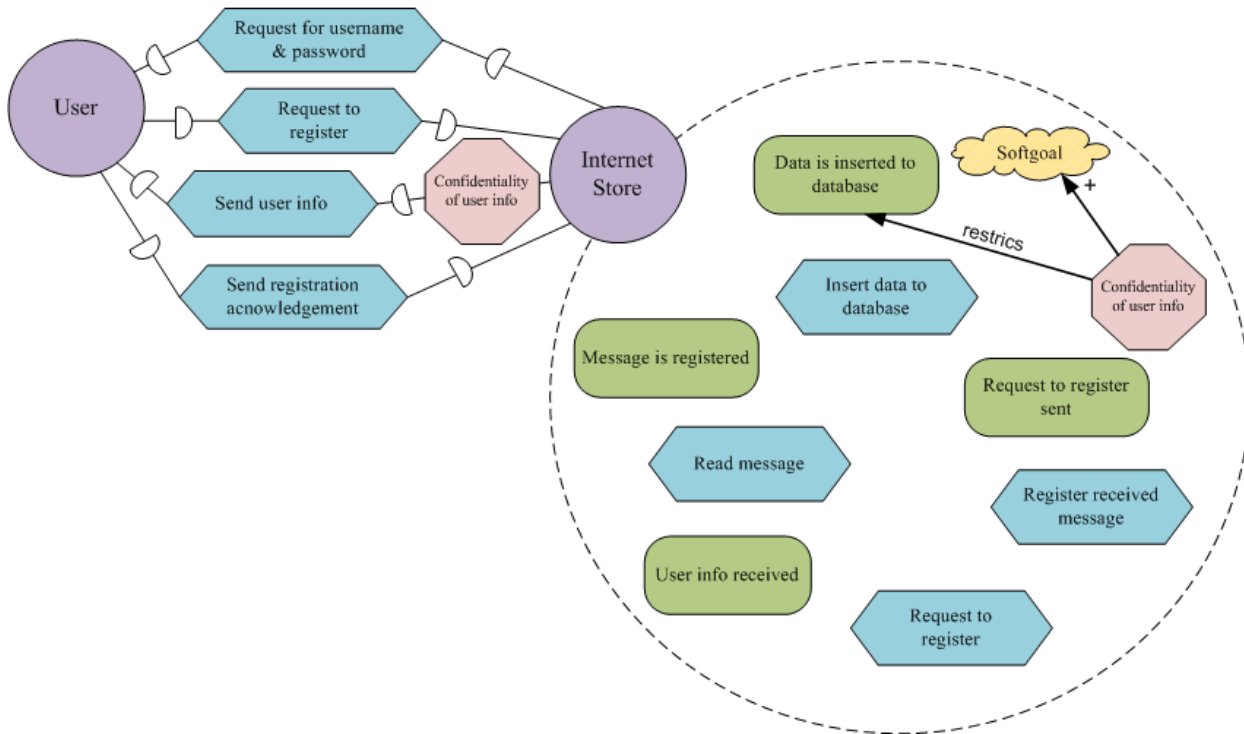
TR6. Next step is dedicated to completion of Secure Tropos Model with *Goals*. *Goals* can be formed from already transformed BPMN *Tasks*, meaning from *Plans* added on the previous step, or from BPMN *Tasks*. Transform Secure Tropos *Goals* in the following form of event: Task Request item list to *Goal* Item list requested.



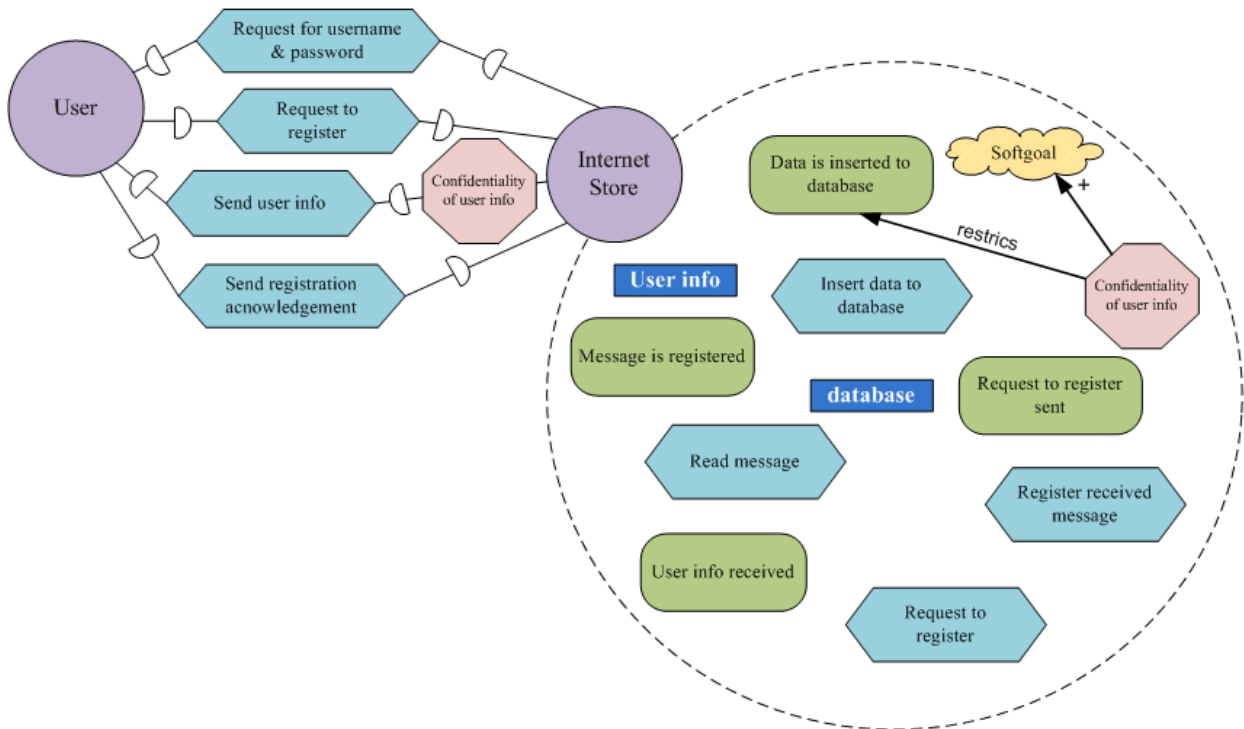
TR7. Duplicate the *Security constraint* to the boundary of an *Actor*. Add *Restricts* relationship between *Security constraint* and the *Goal*. This relationship should correspond to the union of BPMN *Tasks* restricted with a *Lock* defining *Security criterion*.



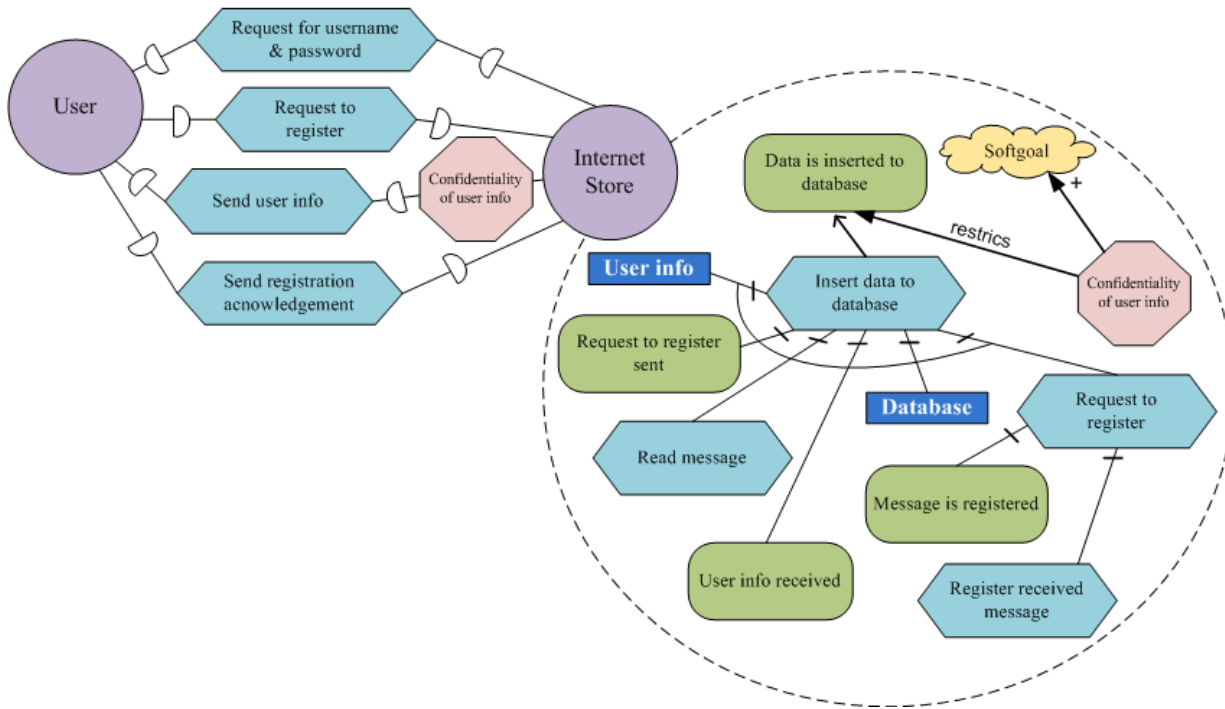
TR8. Define the *Softgoal* and add it to the *Actor's* boundary. The *Softgoal* should comply with *Security objective*, which is a property of the BPMN *Lock*. Add *Contribution* relationship between *Security constraint* and the *Softgoal*.



TR9. Add *Resources* to Secure Tropos model. *Resources* are transformed from BPMN *Data Object* or *Data Store*, in other words BPMN *Artifacts* can be transformed to Secure Tropos *Resources*.



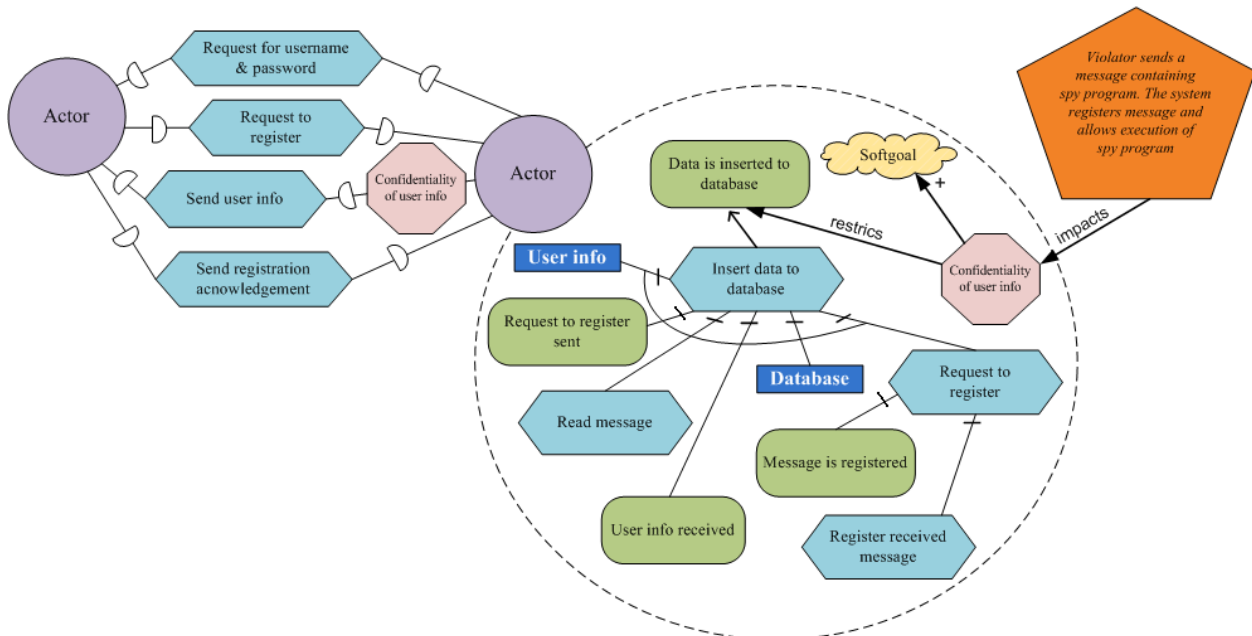
As it was mentioned before, Secure Tropos model should be complemented with some manual add-ons. Define the rest of relationships, such as *Decomposition* and *Means-ends*.



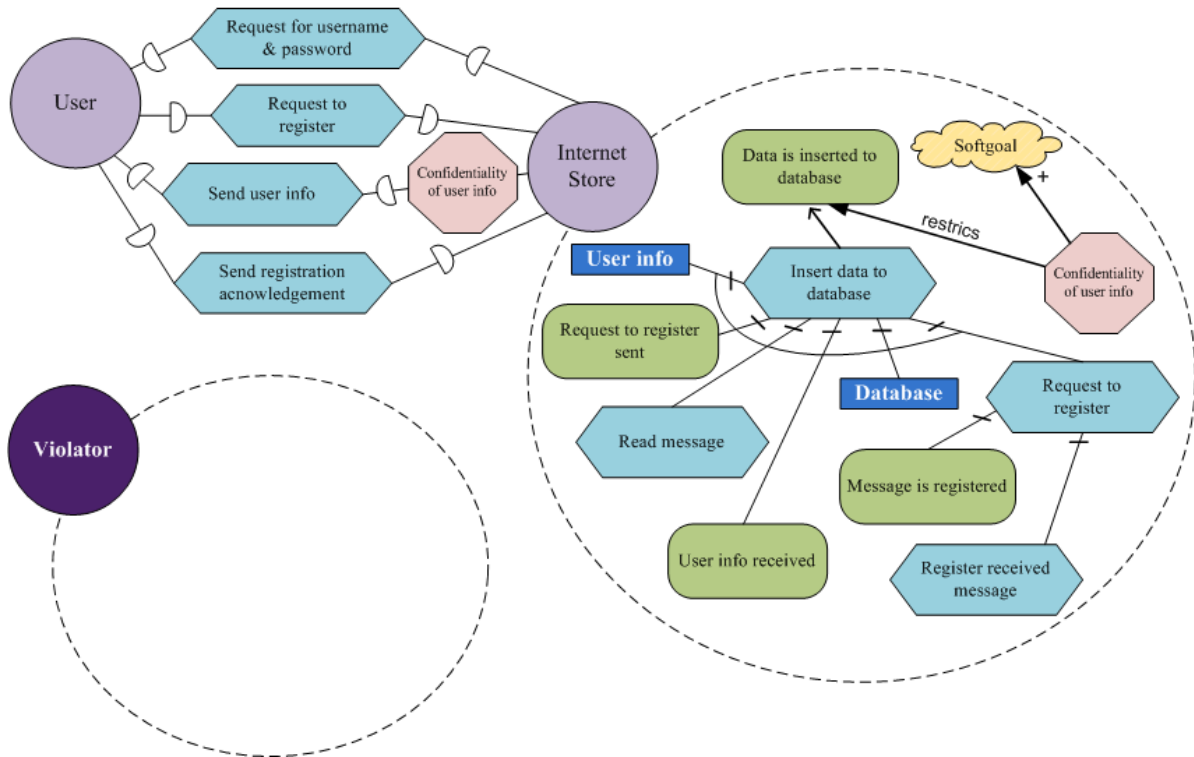
A 2.2 Confidentiality: Risk

Next stage of transformations is creating Secure Tropos Risk model from BP Diagram. The latter comes as an input.

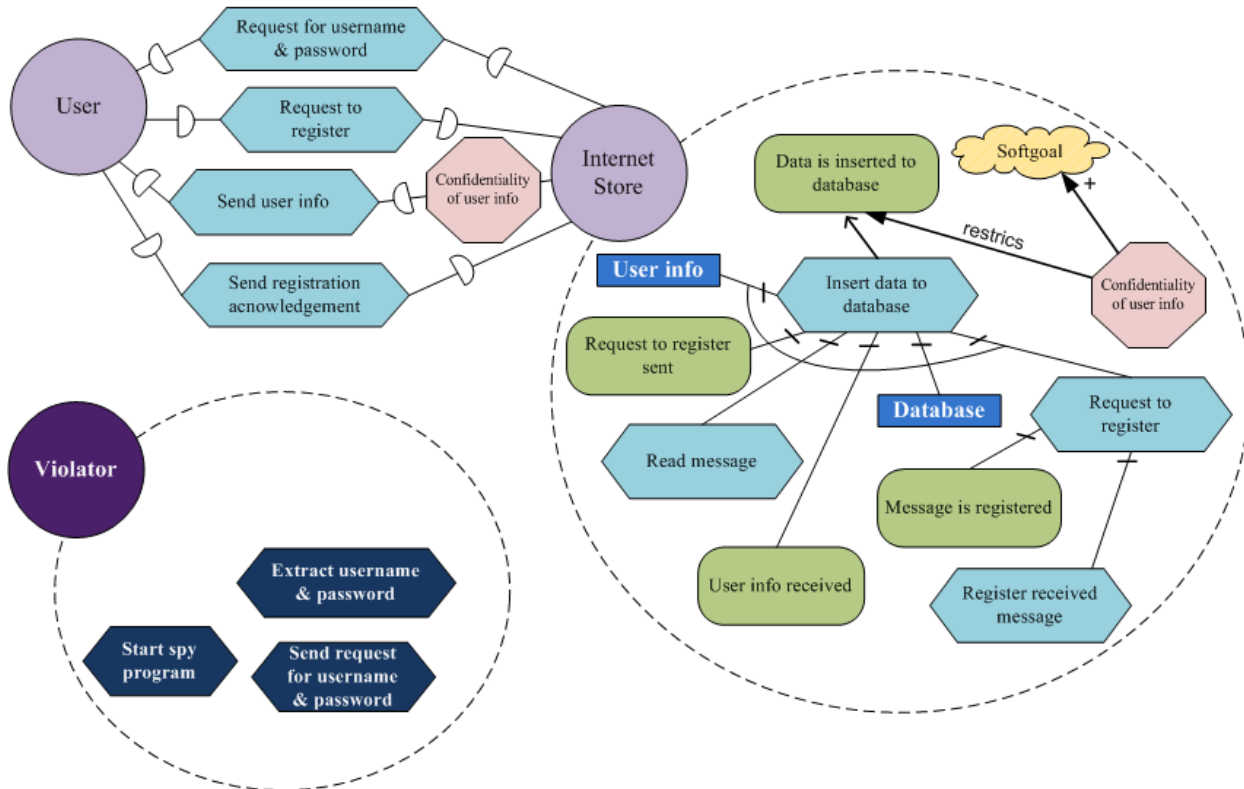
TR10. Start from defining the impact of the risk described in BPMN model. Add Secure Tropos *Threat* and the *Impacts* relationship between *Threat* and corresponding *Security constraint*.



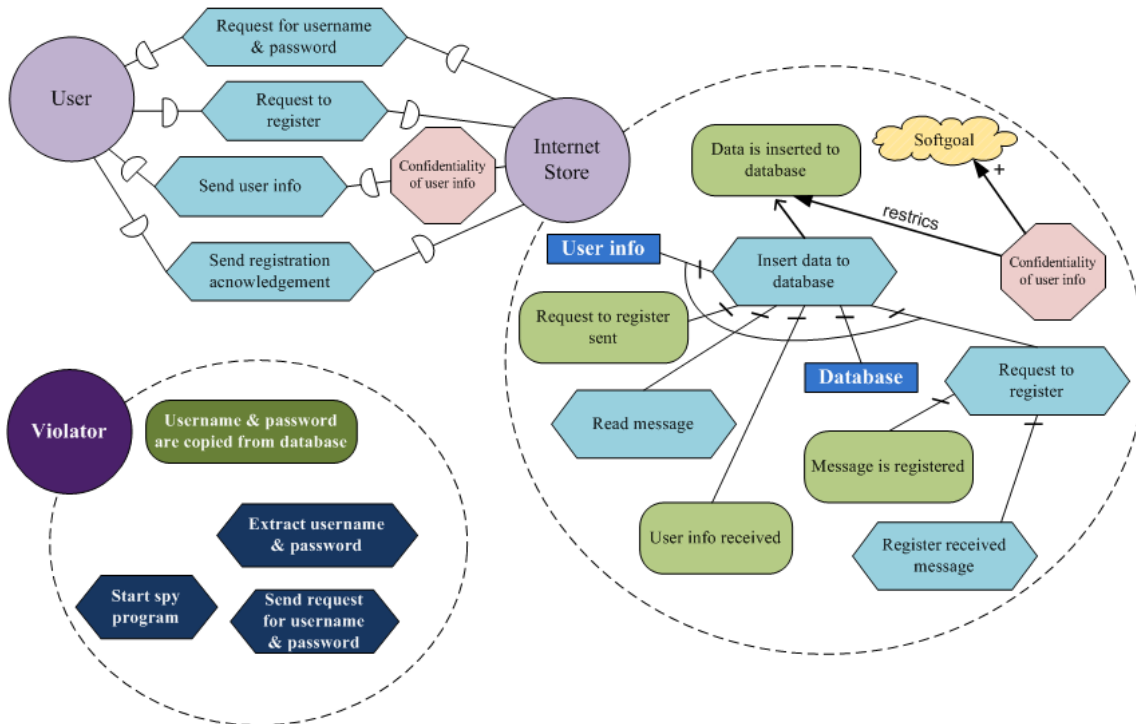
TR11. Basing on the rule **TR1**, define the attacker as a new Secure Tropos *Actor* with its boundary.



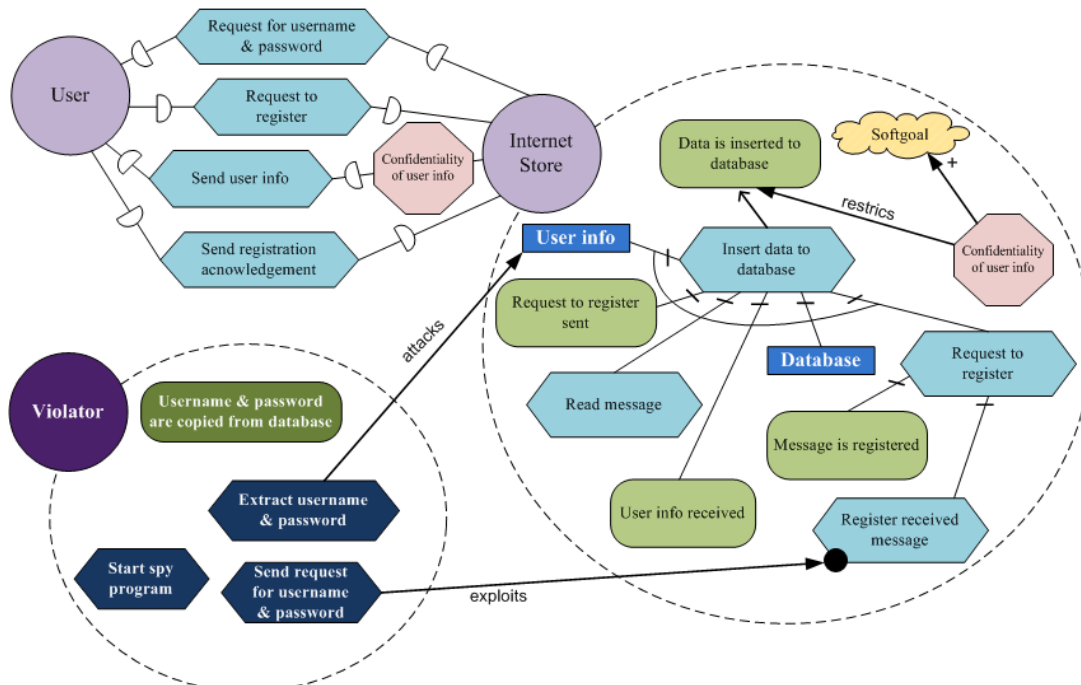
TR12. Transform BPMN *Tasks* (in red) to Secure Tropos *Plans* that represent an attack method. The following rule is based on **TR5**. It can be also noticed that we do not define any dependencies between Agent and other Actors, so BPMN *Message flow* from the *Pool* that represents a threat agent can be transformed to Secure Tropos *Plan*.



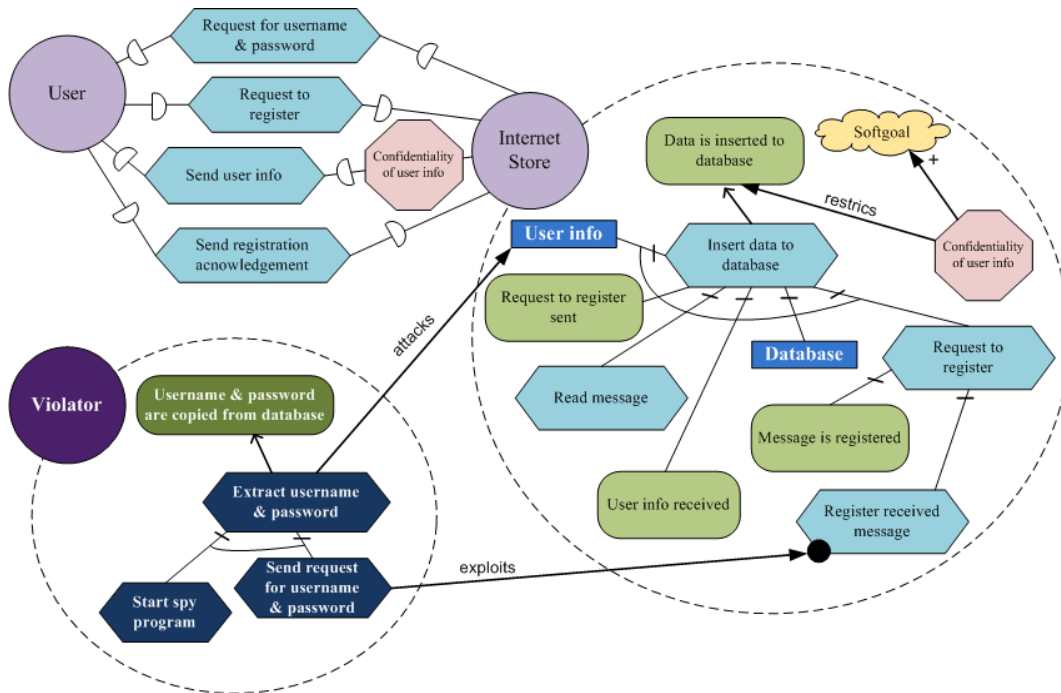
TR13. Define the main goal of the attacker from BPMN diagram; it can be formed from Task or Event (usually some undesirable end event). Once it is defined, transform it to Secure Tropos *Goal*.



TR14. Next step is to detect the *Vulnerability point* from BP Diagram. Secure Tropos *Vulnerability point* should be added to the *Plan (Goal)* or *Resource* that correspond to *Task* or *Artifact* that carried the vulnerability in BPMN model. Put *Exploits* relationship between *Plan* (representing attack method) and *Vulnerability point*. Add the *Attacks* relationship between the *Plan* and *Resource* being attacked.



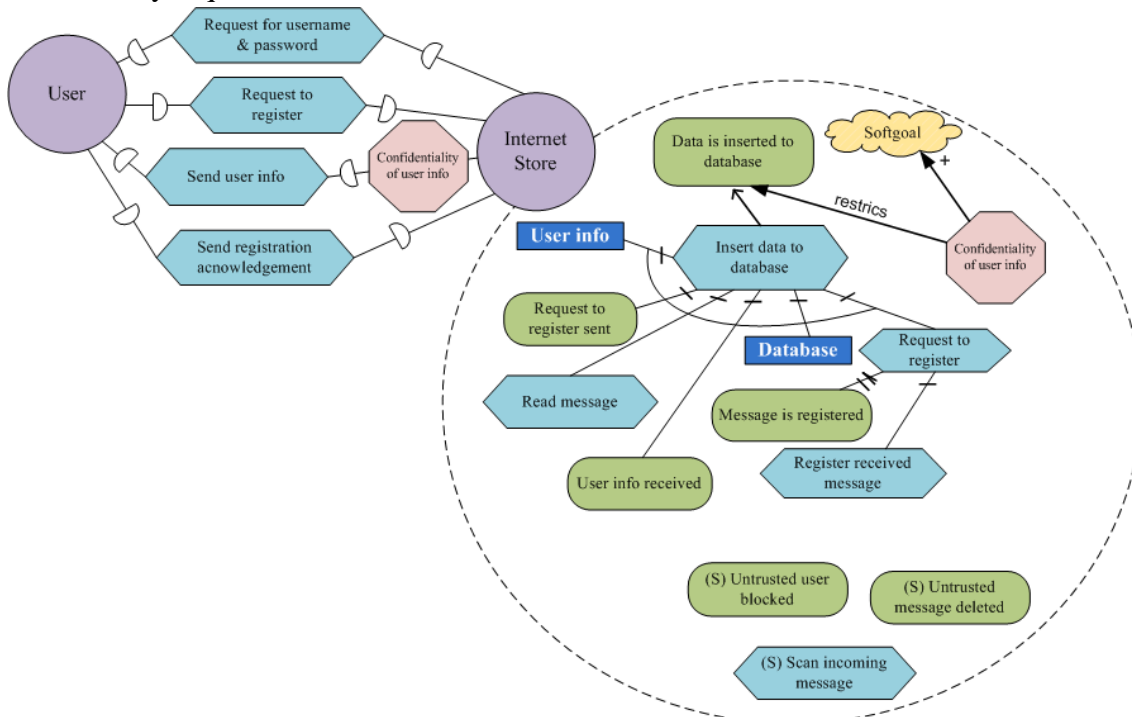
Define the rest of relationships, such as *Decomposition* and *Means-ends* in the boundary of an attacker.



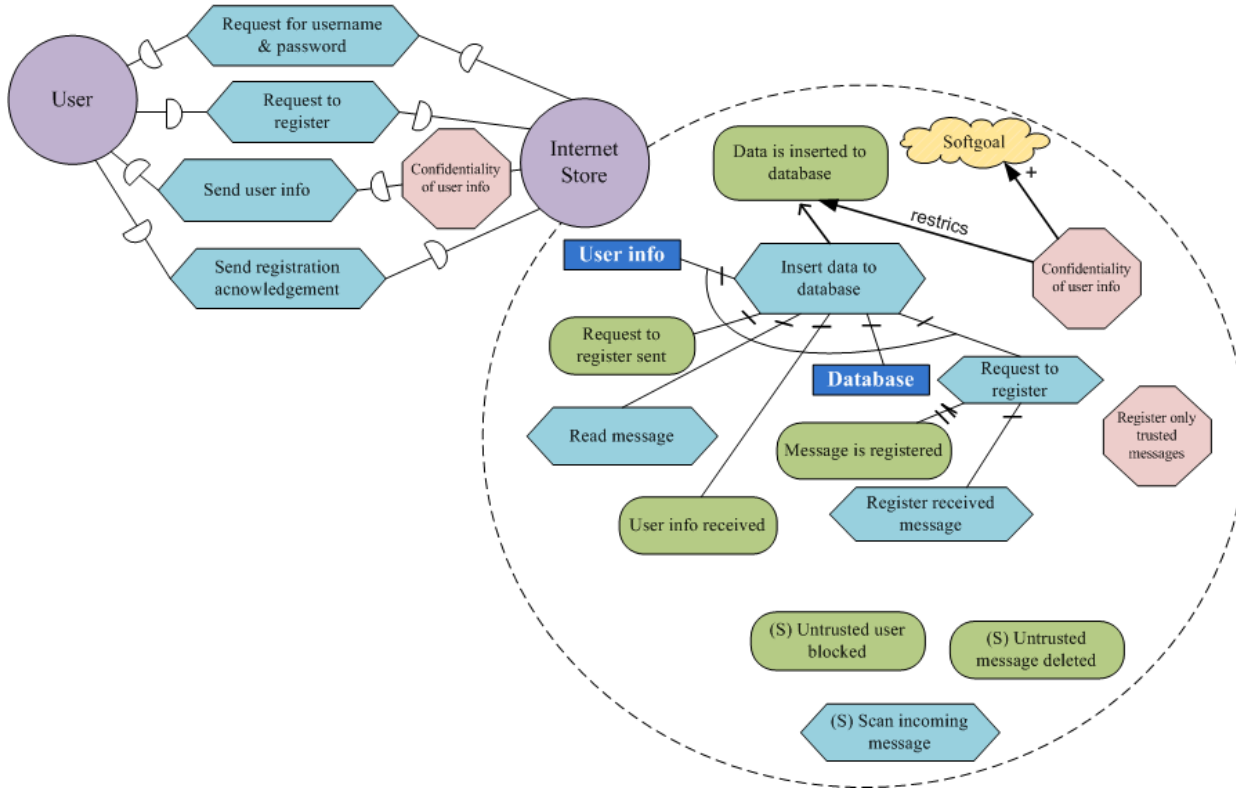
A 2.3 Confidentiality: Risk Treatment

As an input we take asset model in Secure Tropos. Transformation is based on BP Diagram, describing treatment options.

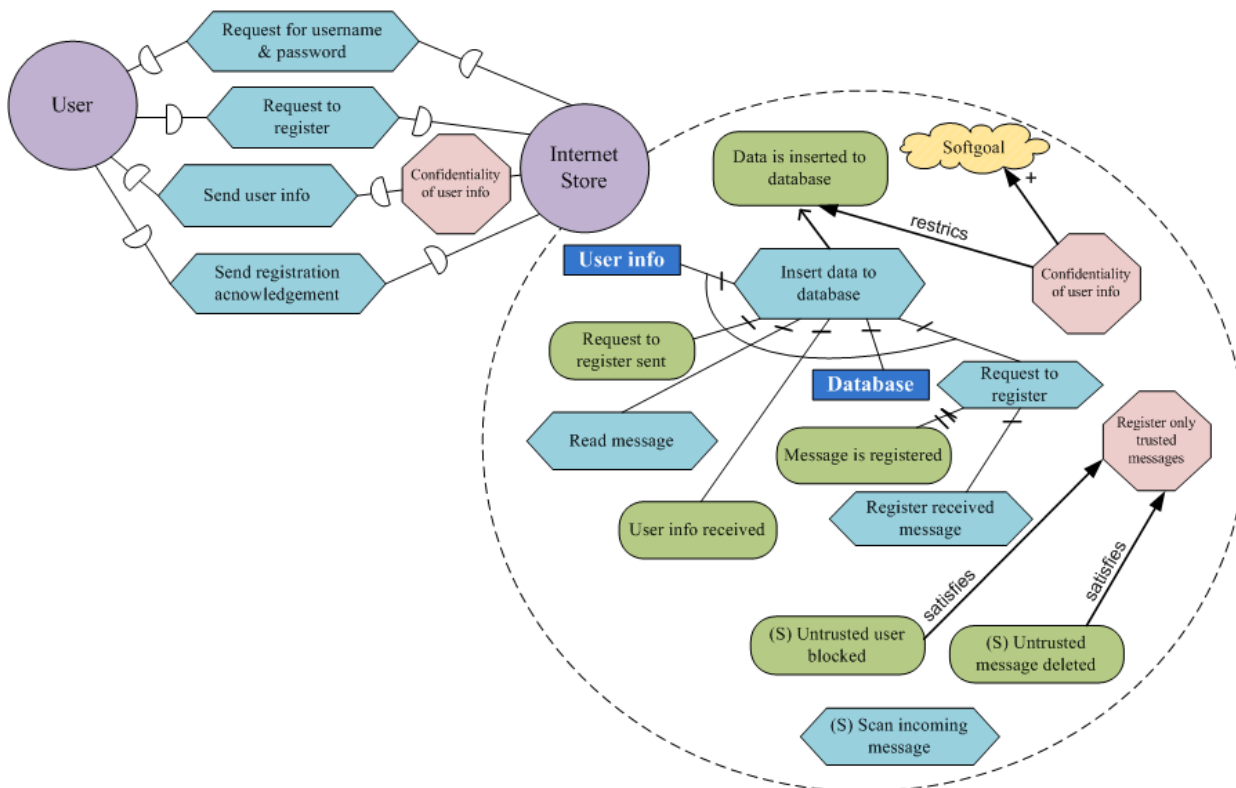
TR15. Transform BPMN Tasks and Gateway structures that represent security requirements into Secure Tropos Plans and Goals, basing on rules **TR5**, **TR6**. Add (S)-labels to all defined Plans, Goals that will emphasize security requirements.



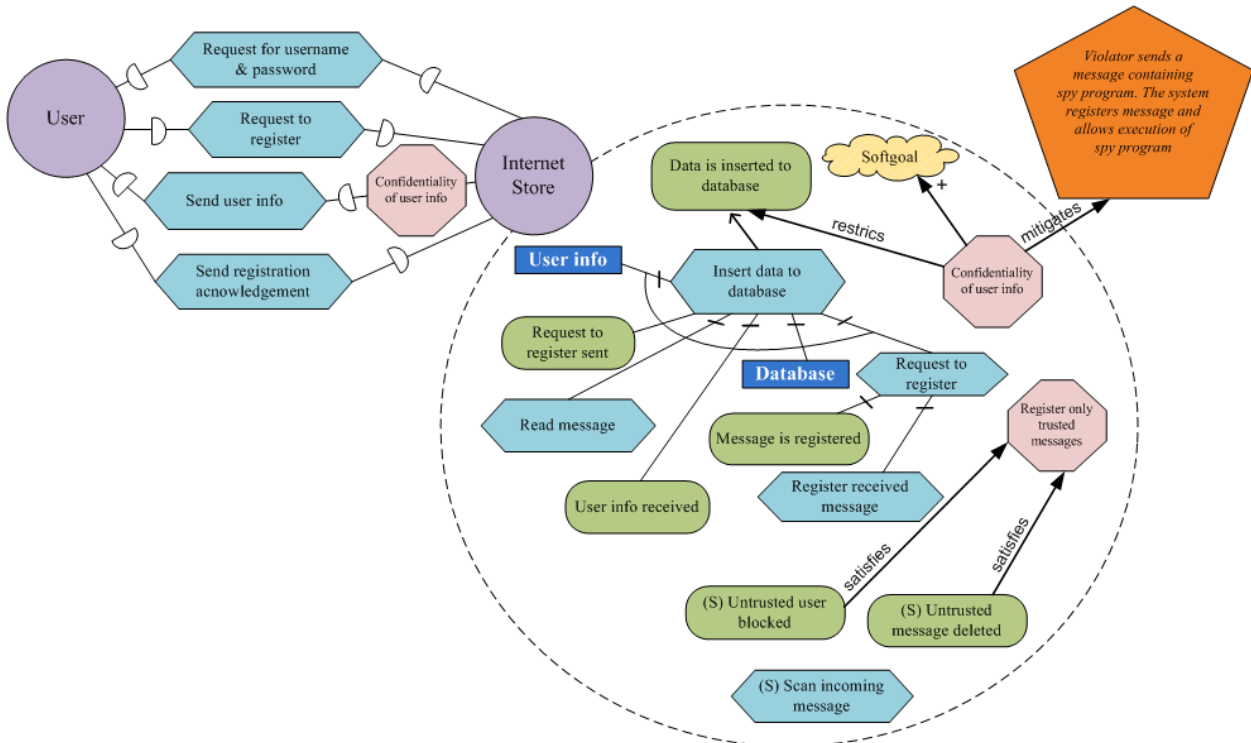
TR16. Define additional *Security constraint*(s). It is possible through transforming the BPMN Gateway structures that represent security requirements control.



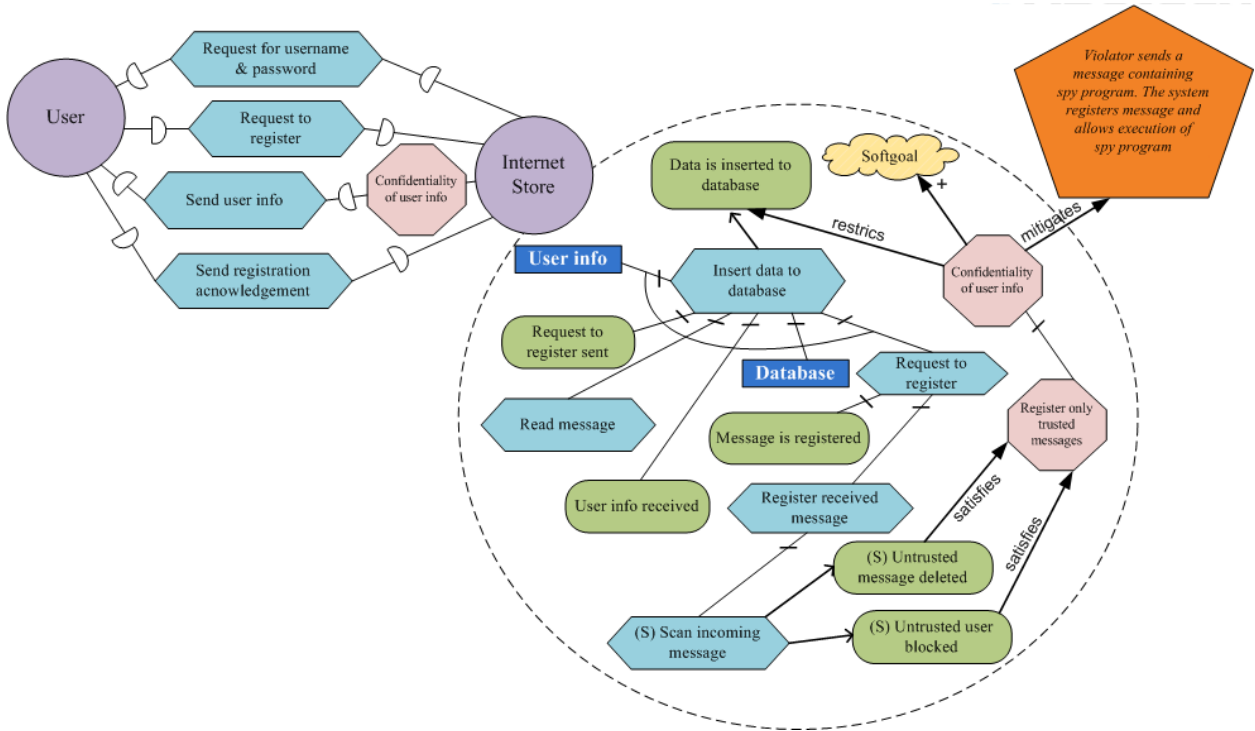
TR17. Add *Satisfies* relationship between defined *Goal*(s) and existing *Security constraints*.



TR18. Collapse the risk scenario to an *Impact*. Add *Mitigates* relation between *Security constraint* and the *Impact*.



Add missing relationships (*Decomposition*, *Means-end*) manually to complete the model.



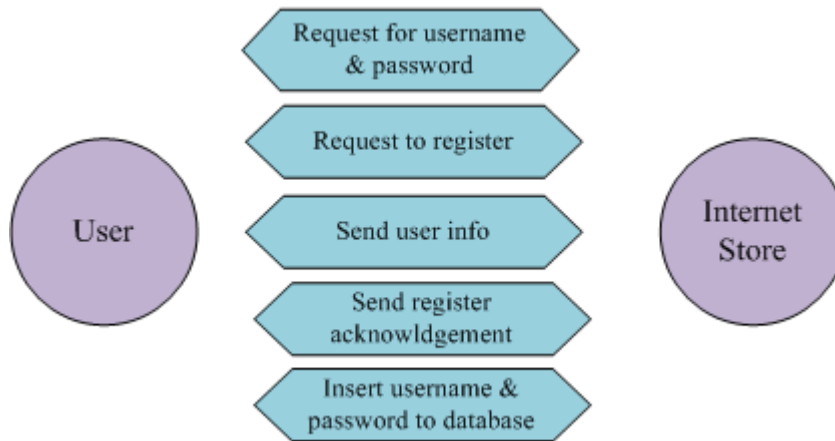
A3. Integrity example transformation

A 3.1 Integrity: Assets

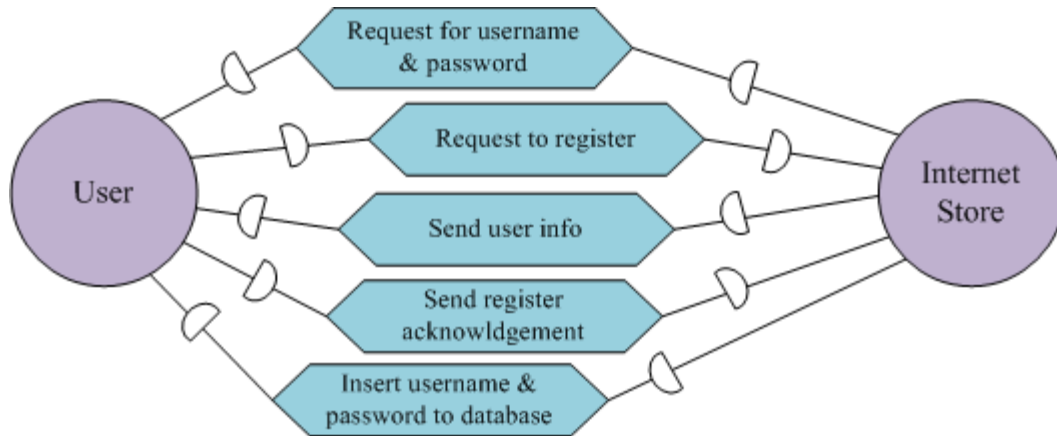
TR1. Define the stakeholders. Transform BPMN *Pools* and *Lanes* to Secure Tropos *Actors*. BPMN *Pool/Lane* that represents a process participant (e.g. job, name, system) is simply transformed to *Actor*. If BPMN *Lanes* are used as a representation of different functional parts of one working system, there is no need to transform each part, transforming the general naming of described system (often *Pool* name) will be essential in Secure Tropos.



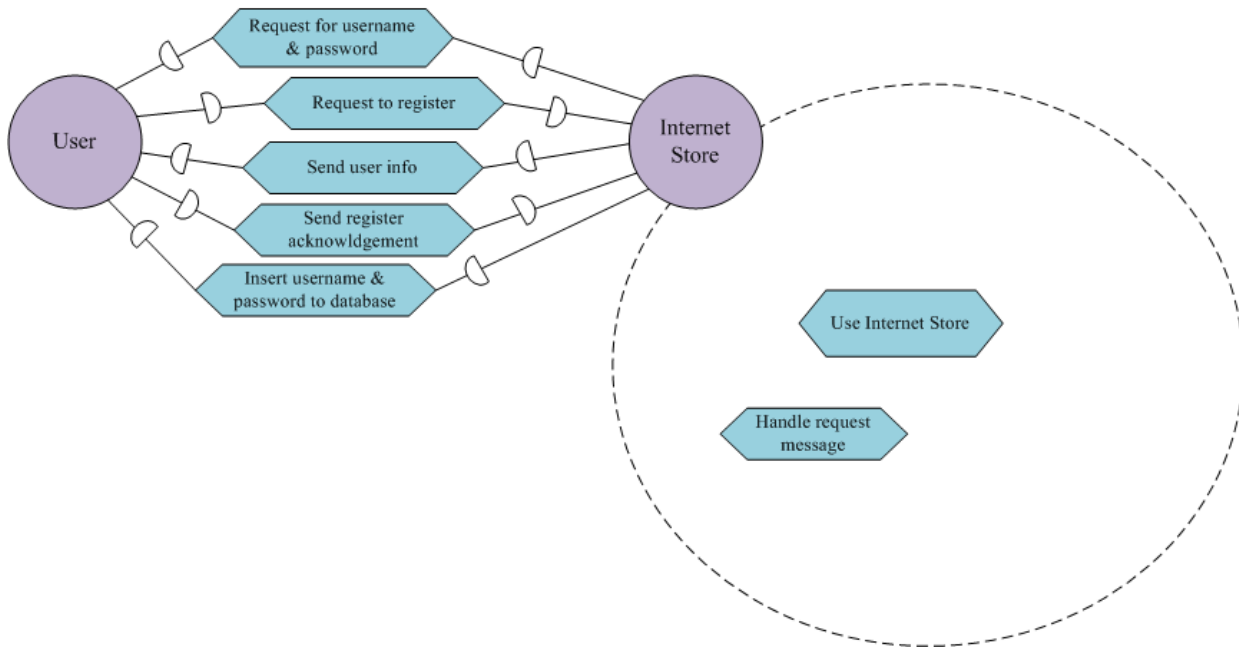
TR2. Another characteristic of Secure Tropos is ability to define dependencies between *Actors*. For dependencies definition transform each BPMN *Task* that represent any activity of communication between participants (e.g. send, receive, request, get, or any other of that kind) to Secure Tropos *Plan*, which will become a dependum. Dependencies can be also represented by BPMN *Message flow* with the same mission: to send something, request or receive. Transform *Message flow* to Secure Tropos dependums.



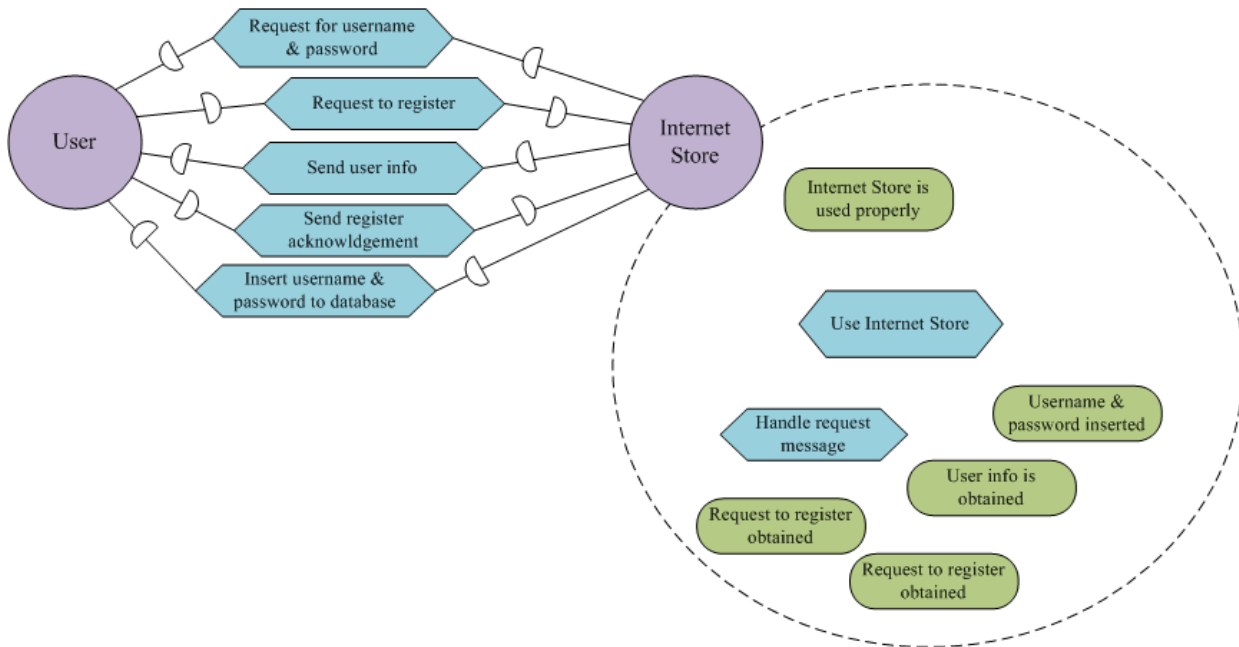
TR3. Next step is to add a direction to previously defined dependencies; to decide who is a depender and who is a dependee. A depender in BPMN is usually somebody who performs the action, a dependee is the one who depends on this action. In case with the transformation of *Message flow*, the direction of dependency is specified with direction of observed *Message flow*; if it goes out from the *Pool* that means we are dealing with a depender, and conversely.



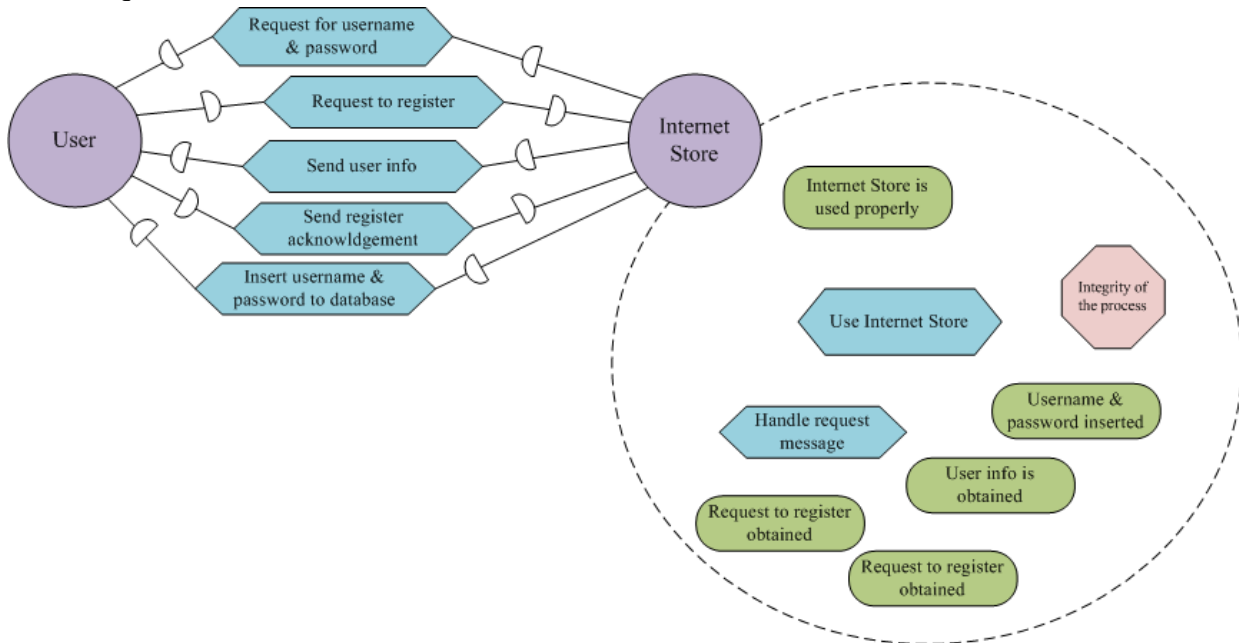
TR4. Define the *Security constraint* that will regulate the dependency between *Actors*. *Security constraint* that depender expects to be satisfied, is transformed from BPMN Security objective.



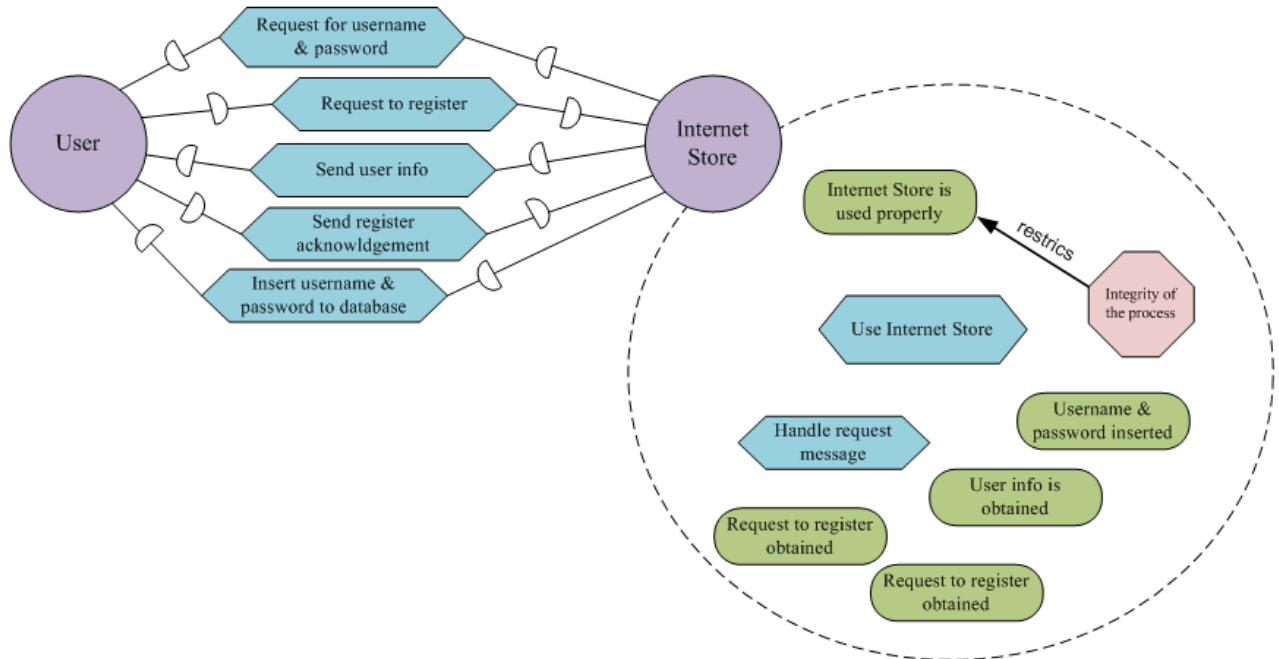
TR5. Transform BPMN *Tasks* to Secure Tropos *Plans* and add them to the corresponding Actor's boundary; add only *Plans* that are under responsibility of the observed Actor.



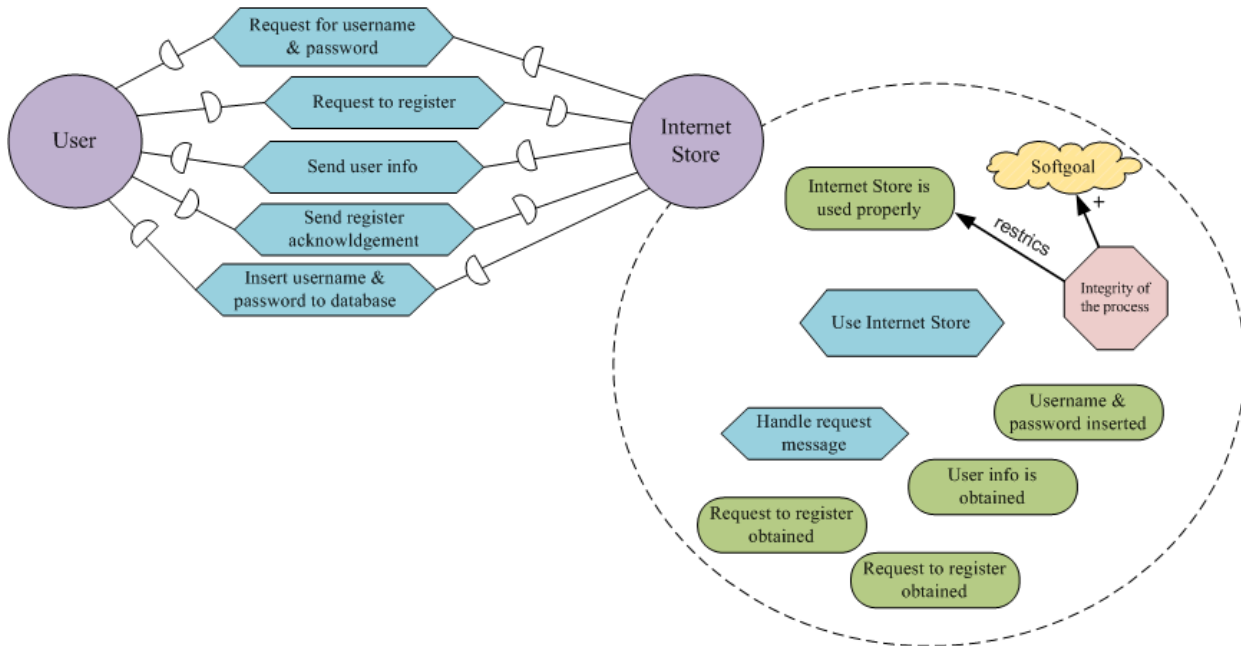
TR6. Next step is dedicated to completion of Secure Tropos Model with *Goals*. *Goals* can be formed from already transformed BPMN *Tasks*, meaning from *Plans* added on the previous step, or from BPMN *Tasks*. Transform Secure Tropos *Goals* in the following form of event: Task Request item list to *Goal* Item list requested.



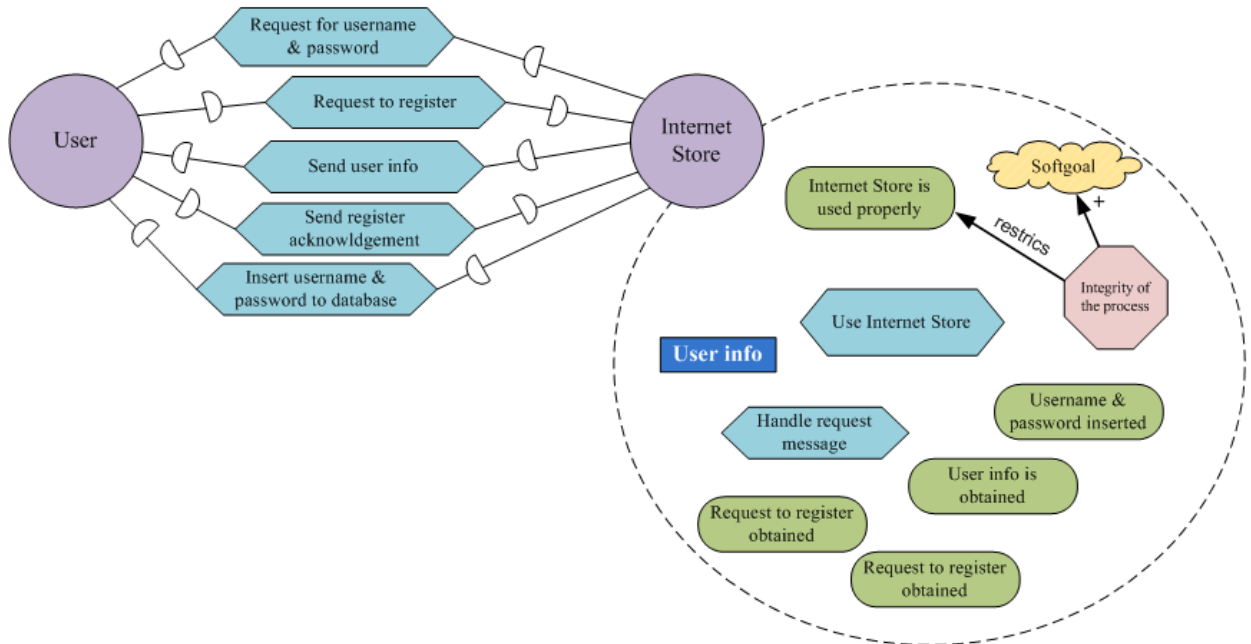
TR7. Duplicate the *Security constraint* to the boundary of an *Actor*. Add *Restricts* relationship between *Security constraint* and the *Goal*. This relationship should correspond to the union of BPMN *Tasks* restricted with a *Lock* defining *Security criterion*.



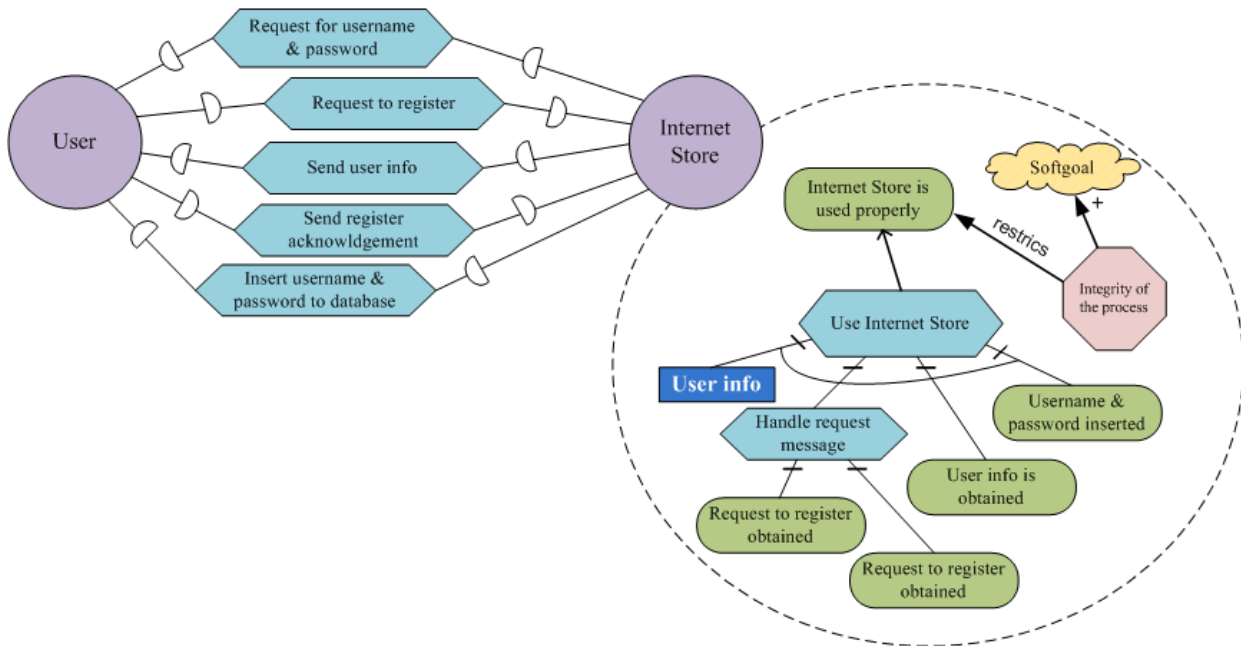
TR8. Define the *Softgoal* and add it to the *Actor's* boundary. The *Softgoal* should comply with *Security objective*, which is a property of the BPMN *Lock*. Add *Contribution* relationship between *Security constraint* and the *Softgoal*.



TR9. Add *Resources* to Secure Tropos model. *Resources* are transformed from BPMN *Data Object* or *Data Store*, in other words BPMN *Artifacts* can be transformed to Secure Tropos *Resources*.



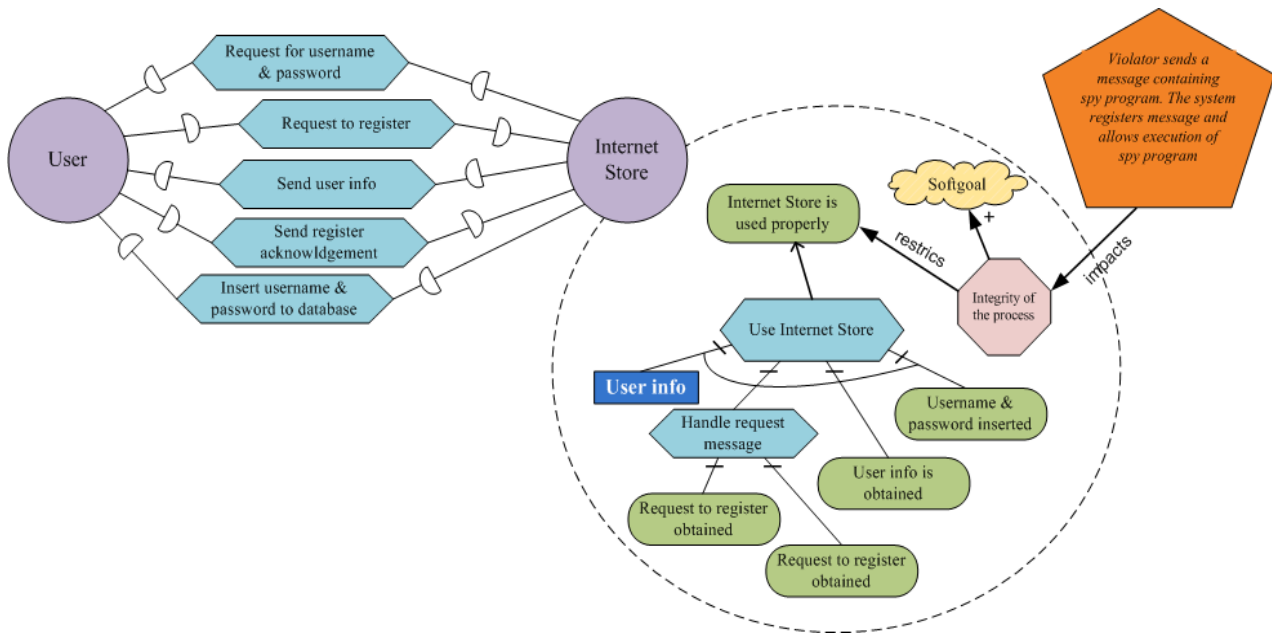
As it was mentioned before, Secure Tropos model should be complemented with some manual add-ons. Define the rest of relationships, such as *Decomposition* and *Means-ends*.



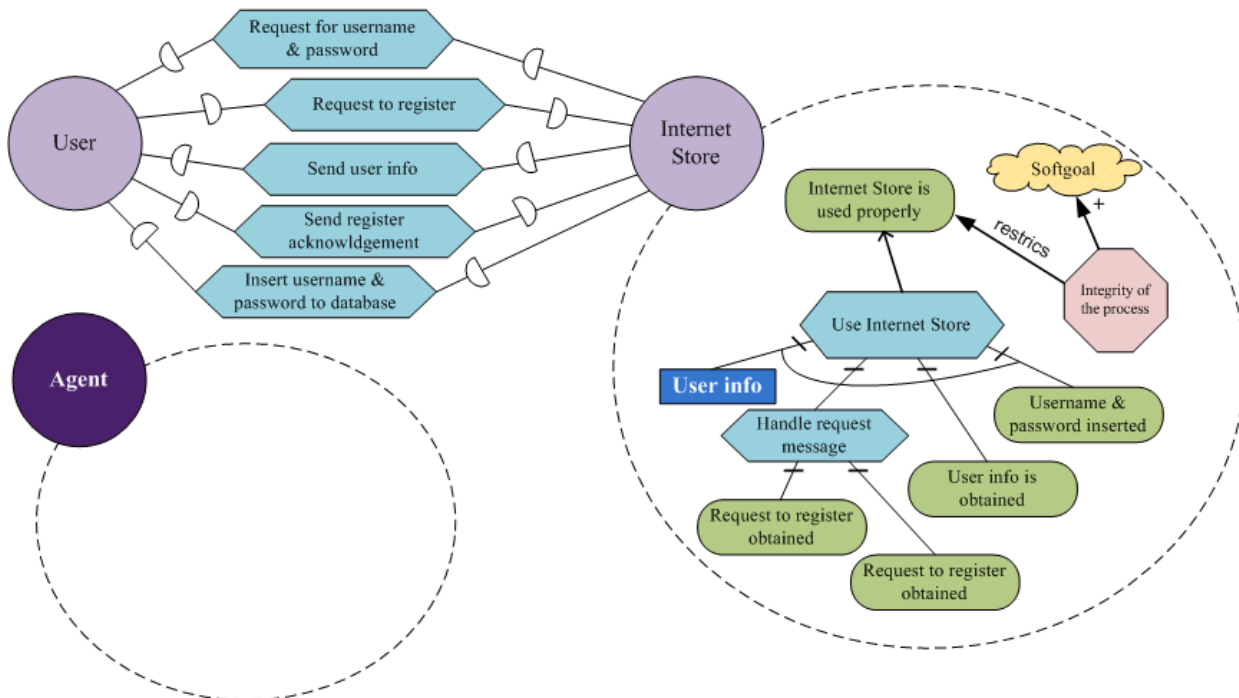
A 3.2 Integrity: Risk

Next stage of transformations is creating Secure Tropos Risk model from BP Diagram. The latter comes as an input.

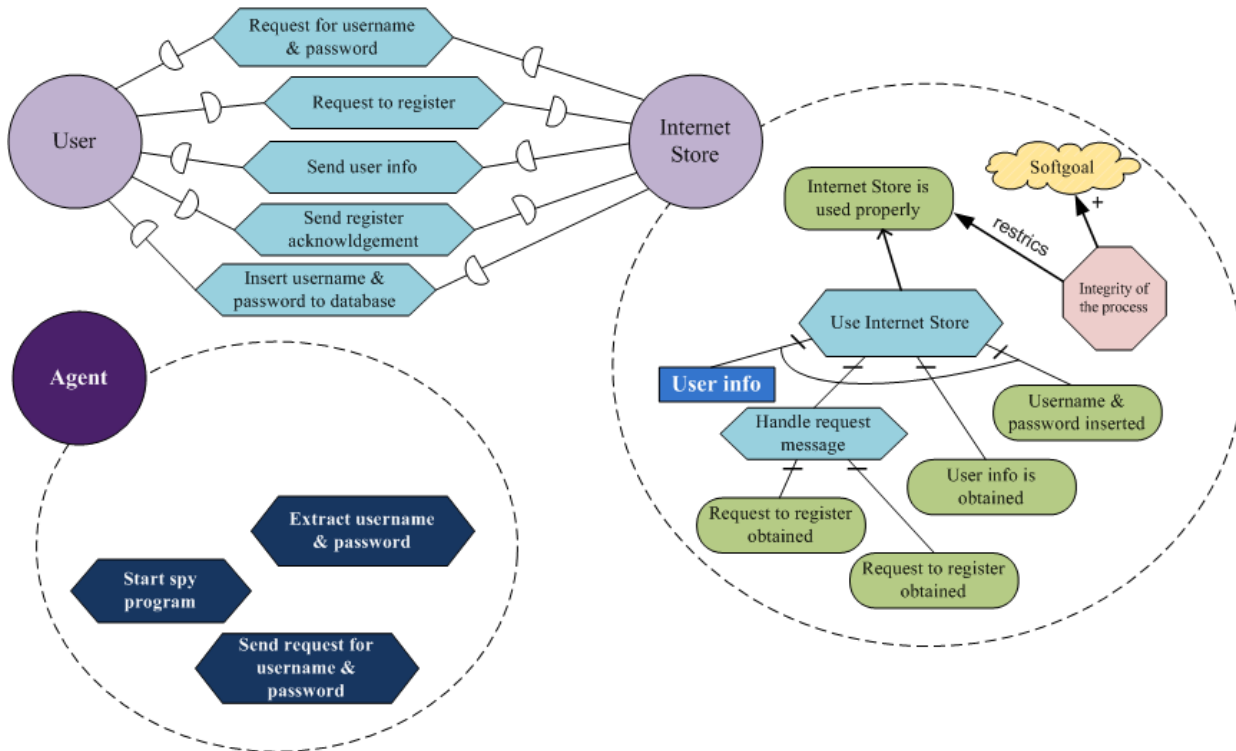
TR10. Start from defining the impact of the risk described in BPMN model. Add Secure Tropos *Threat* and the *Impacts* relationship between *Threat* and corresponding *Security constraint*.



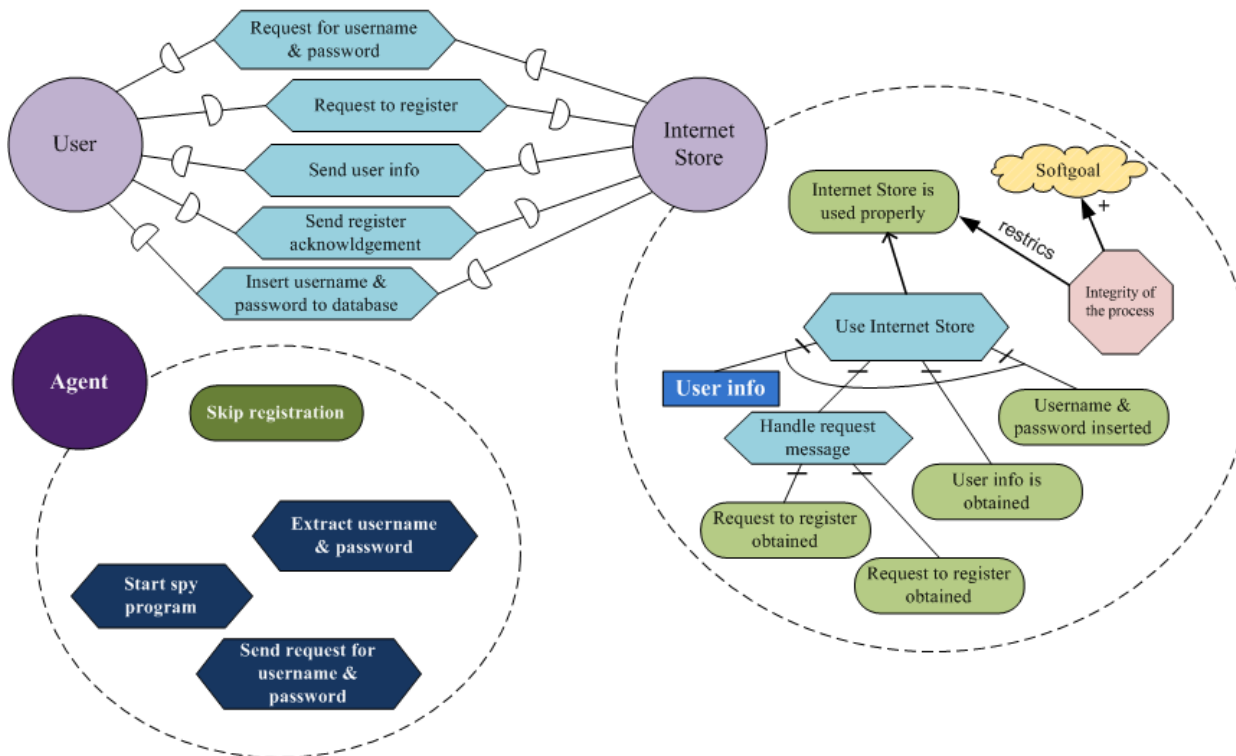
TR11. Basing on the rule **TR1**, define the attacker as a new Secure Tropos Actor with its boundary.



TR12. Transform BPMN Tasks (in red) to Secure Tropos Plans that represent an attack method. The following rule is based on **TR5**. It can be also noticed that we do not define any dependencies between Agent and other Actors, so BPMN Message flow from the Pool that represents a threat agent can be transformed to Secure Tropos Plan.

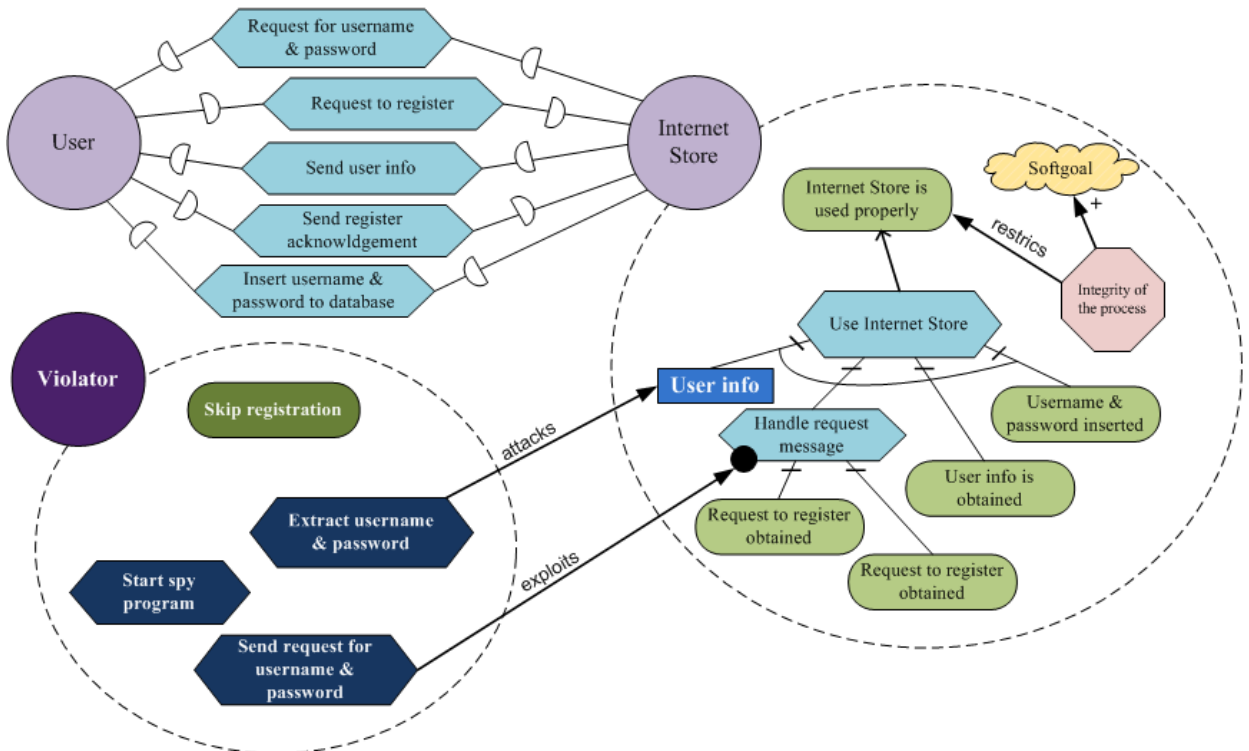


TR13. Define the main goal of the attacker from BPMN diagram; it can be formed from Task or Event (usually some undesirable end event). Once it is defined, transform it to Secure Tropos *Goal*.

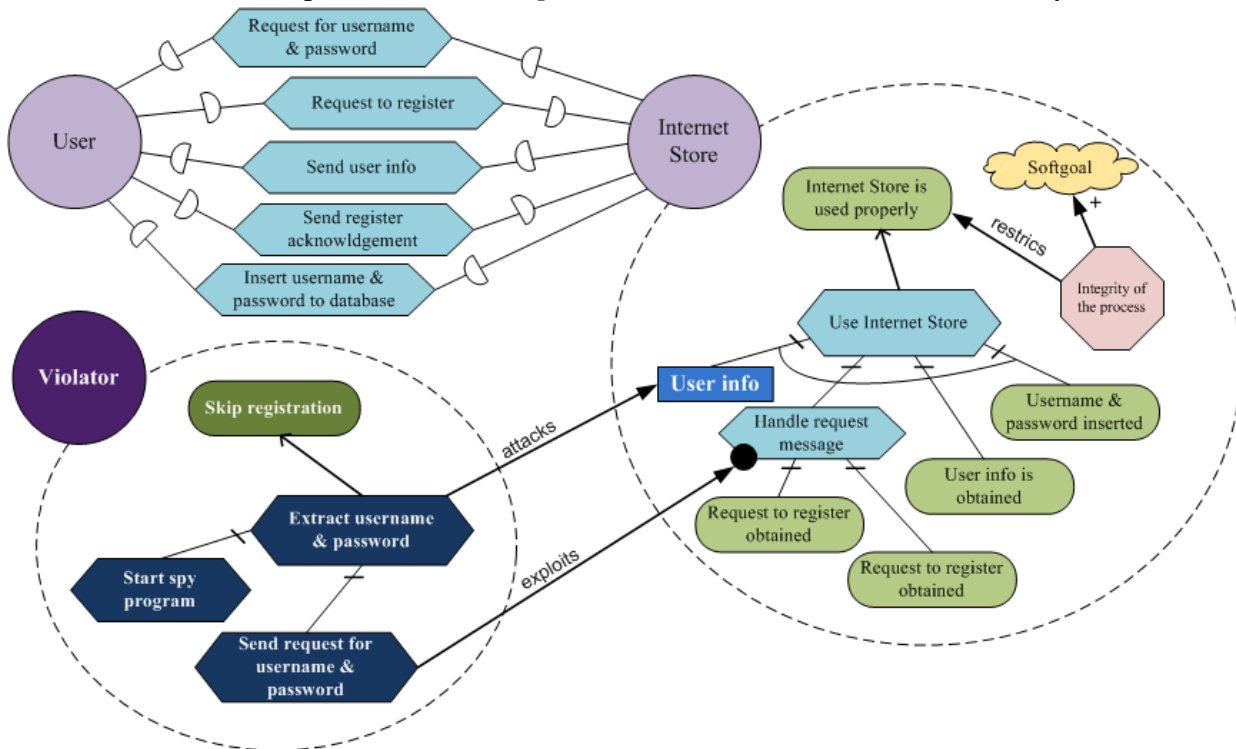


TR14. Next step is to detect the *Vulnerability point* from BP Diagram. Secure Tropos *Vulnerability point* should be added to the *Plan (Goal)* or *Resource* that correspond to *Task* or *Artifact* that carried the

vulnerability in BPMN model. Put *Exploits* relationship between *Plan* (representing attack method) and *Vulnerability point*. Add the *Attacks* relationship between the *Plan* and *Resource* being attacked.



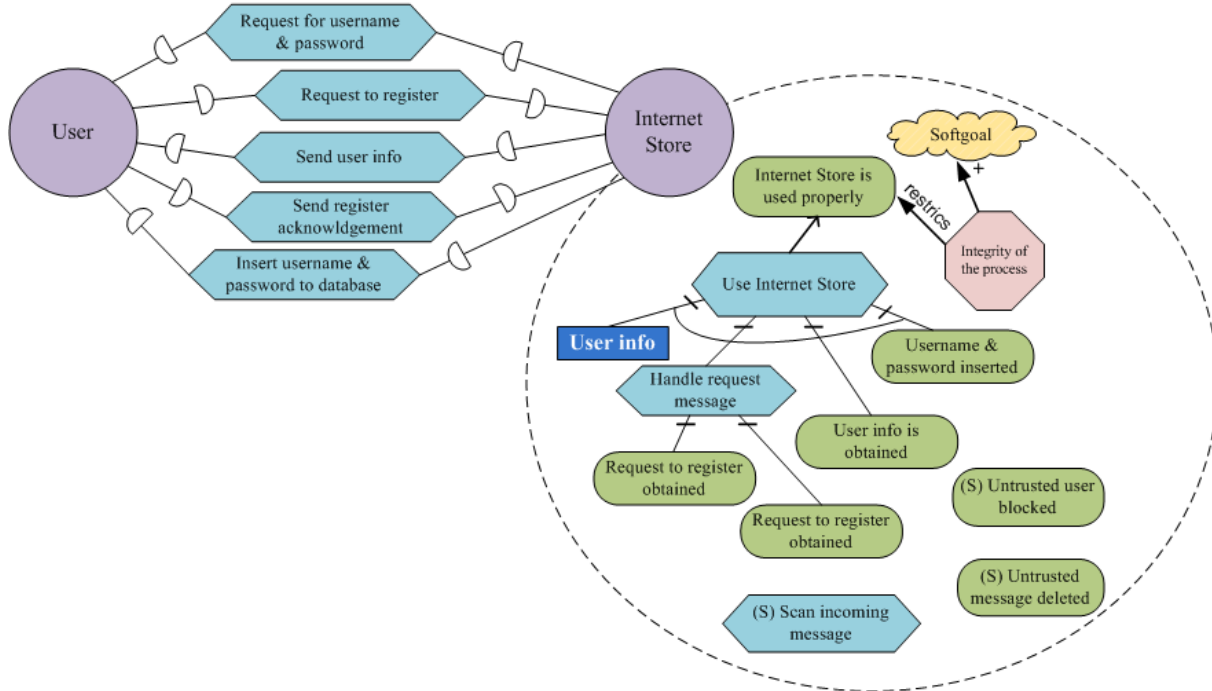
Define the rest of relationships, such as *Decomposition* and *Means-ends* in the boundary of an attacker.



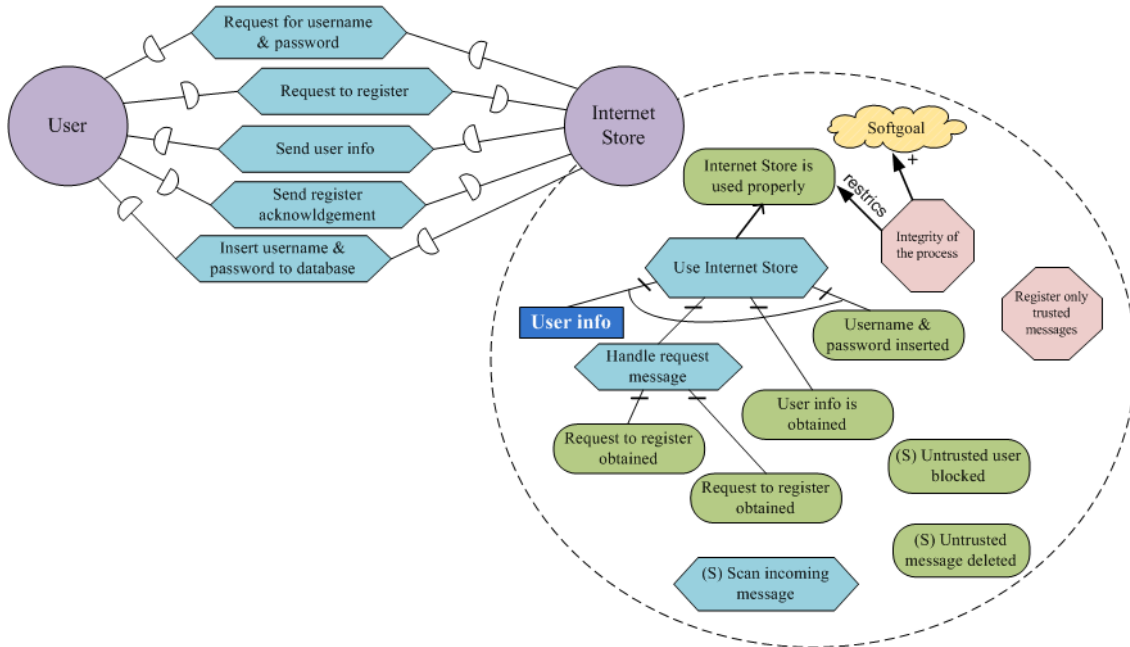
A 3.3 Integrity: Risk Treatment

As an input we take asset model in Secure Tropos. Transformation is based on BP Diagram, describing treatment options.

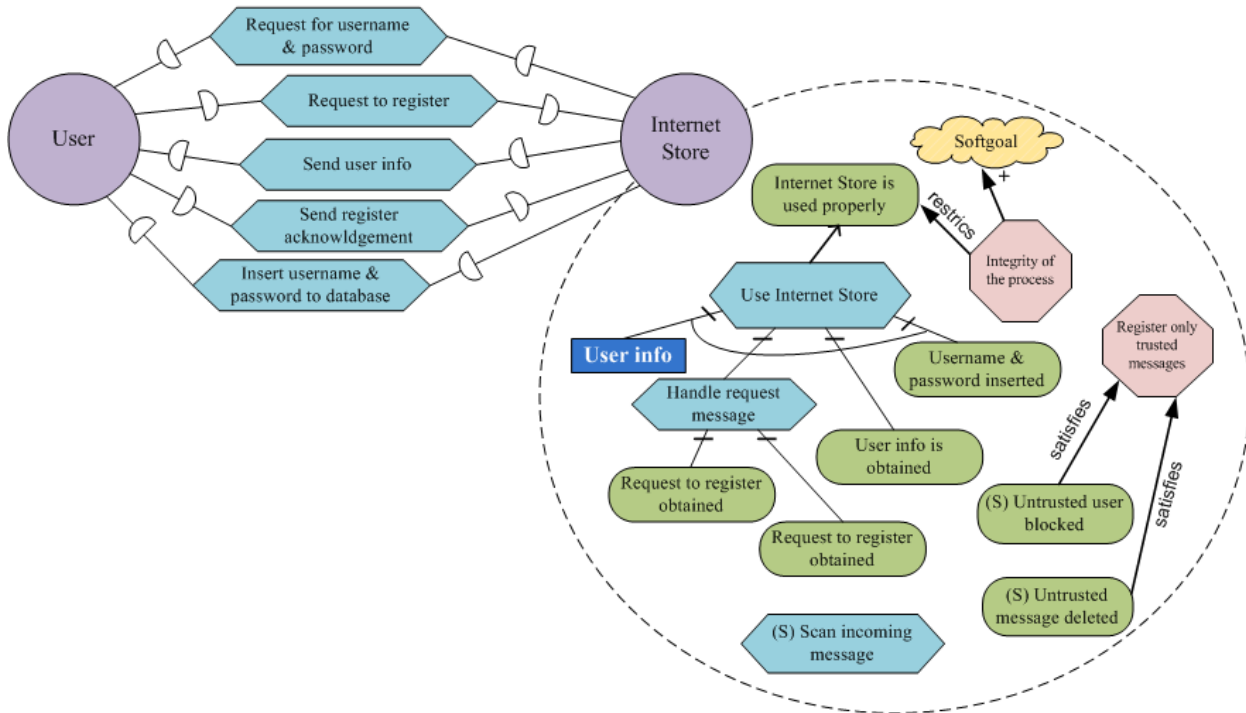
TR15. Transform BPMN Tasks and Gateway structures that represent security requirements into Secure Tropos Plans and Goals, basing on rules **TR5**, **TR6**. Add (S)-labels to all defined Plans, Goals that will emphasize security requirements.



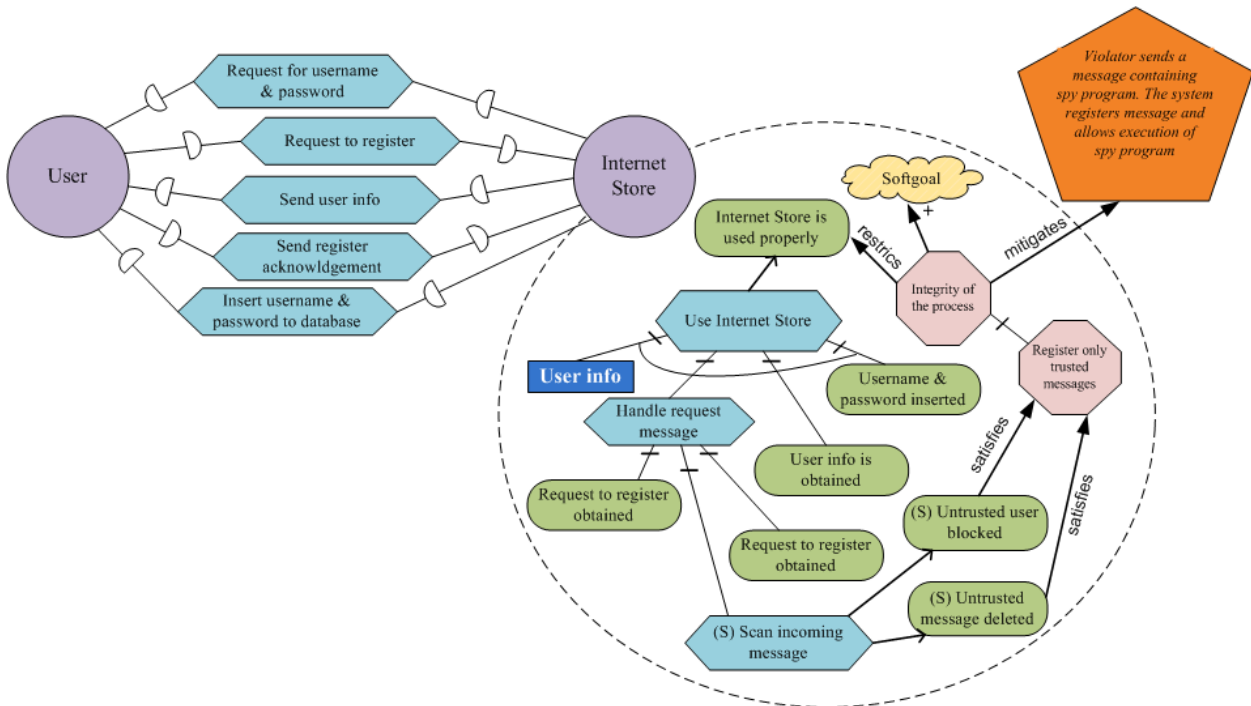
TR16. Define additional Security constraint(s). It is possible through transforming the BPMN Gateway structures that represent security requirements control.



TR17. Add *Satisfies* relationship between defined *Goal(s)* and existing *Security constraints*.



TR18. Collapse the risk scenario to an *Impact*. Add *Mitigates* relation between *Security constraint* and the *Impact*. Add missing relationships (*Decomposition*, *Means-end*) manually to complete the model.



Non-exclusive licence to reproduce thesis and make thesis public

I, Olga Altuhhova, (date of birth: 17.05.1988),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
 - 1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

An Extension of Business Process Model and Notation for Security Risk Management,
supervised by Raimundas Matulevičius,

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

Tartu/Tallinn/Narva/Pärnu/Viljandi, **20.05.2013**