An Approach for Modeling Information Systems Security Risk Assessment

Subhas C. Misra[1], Vinod Kumar[2] and Uma Kumar[2]

[1] Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6

[2] Carleton University, 1125 Colonel By Drive, Ottawa, Ontario, Canada K1S 5B6

Abstract. In this paper, we present a conceptual modeling approach, which is new in the domain of information systems security risk assessment. The approach is helpful for performing means-end analysis, thereby uncovering the structural origin of security risks in an information system, and how the root-causes of such risks can be controlled from the early stages of the projects. The approach addresses this limitation of the existing security risk assessment models by exploring the strategic dependencies between the actors of a system, and analyzing the motivations, intents, and rationales behind the different entities and activities constituting the system.

1 Introduction

Security is considered to be a very important issue while developing complex information systems. Security risk assessment is a systematic process that includes identification of the risk, determining the consequences of the risk and managing risks [1,9]. In this paper, we present an approach which is new in the domain of modeling information systems security risk assessment, based on the concept of analysis of the strategic dependencies between the actors of a system [3,8].

Before presenting our approach, we first review the relevant previous work in this area. Several risk assessment methodologies tailored towards specific domains are available in the existing literatures [1]. For instance, SEISMED is a methodology that provides a set of guidelines on IT security risk analysis for health care IT personnel and ODESSA is a methodology that provides health care data security [1]. COBIT project addresses the good management practices for security and control in IT for world-wide endorsement by various organizations [5]. CRAMM is a risk analysis methodology that was developed with an aim of providing a structured approach to manage security for computer systems [2]. The most influential of the above for our work is the CORAS methodology that bases itself on UML-based modeling [14].

In contrast to the previously proposed modeling techniques in the domain of security risk assessment, we use the concept of modeling intentional relationships using i* [3] between actors of a project. Similar studies have been conducted in the areas of requirements engineering [6], process verification and validation [8], trust in information systems (e.g., [8]). Our modeling technique adopts ideas of two methodologies –

C. Misra S., Kumar V. and Kumar U. (2005).
An Approach for Modeling Information Systems Security Risk Assessment.
In *Proceedings of the 3rd International Workshop on Security in Information Systems*, pages 253-262
DOI: 10.5220/0002561702530262

CORAS and actor-dependency framework. In contrast to CORAS, our proposed method described in our paper uses agent-oriented approach and allows modeling security by modeling dependencies between actors of the system.

In this paper, we show how we can use the concepts of modeling intentional dependencies between actors to explore the structural origins of risks in an information security assessment project. Although the concept of actor-dependency is not new, the way we use these concepts and *extend* them to analyze security risks is *novel* to our work, and requires ingenuity of the modelers. In this paper, we have discussed a methodology that can help one to model and explore the strategic security risks, the actual causes and intentions of the different actions one can undertake, the risks involved in undertaking those actions, and finally, model how one can control those risks from the inception of a project.

i* framework is an agent-oriented modeling methodology that was developed for modeling intentional dependencies among strategic actors, who can be social agents, organizations or other active units. Actors depend on one another for achieving goals, performing tasks and furnishing the resources. Using this approach one can model various hard-to-formalize phenomena such as goal, belief, ability, and so on. However, i* is insufficient for risk assessment. In this paper we alter some modeling elements of i* and show how the altered framework can be used to perform risk assessment. *Contribution*s of our work are the development of a new methododology to represent assets, vulnerabilities and security features using i* elements and the introduction of different new diagrammatic elements for representing *risk value* and *asset value*.

Our current work has been inspired by the work done in [9] to model security using Tropos. The paper [9] introduces several new security-related concepts and shows how the concepts can be represented diagrammatically. We adopt the security feature concept introduced in this paper. However the paper on security modeling with Tropos does not cover such concepts as assets, security risk, asset and risk value that are central for security risk assessment.

In our work we referred to a widely recognized standard for security risk management named the Australian/New Zealand AS/NZS standard [12]. The risk management process described in AS/NZS consists of five stages: identify context, identifying risks, analyze and evaluate risks, identify and document treatment. We will show how to use our extended i* framework to accomplish the entire cycle of the risk management process.

2 Modeling Dependencies between Strategic Actors

The concepts associated with modeling actor dependencies have their roots in Requirements Engineering (RE). RE methodologies can be used to model organization goals, processes, relationships, and actors. In order to perform very good quality risk assessment one is required to understand the organization clearly.

In this section, we briefly discuss the actor dependency concept using i* (see, for example, [3], [6], [8] and [13] to learn more about this area of research, and its applicability to various domains). Although i* is a “brain-child” of software RE research, it can be used as a powerful tool to model organizational tasks, processes, actors and goals. In order to model, and solve this problem, two actor-dependency diagrams are used: the Strategic Dependency Model (SD), and the Strategic Rationale Model (SR). In the interest of brevity, only brief introductions of SD and SR are provided. Interested readers are referred to the literatures mentioned in Section 1 for learning further details. SD diagrams are used to model dependencies between actors, while SR diagrams are used to model internally why each actor has those dependencies. All dependencies comprise of a “depender”, a “dependee”, and a “dependum”. “Depender” depends on a “dependee” to get “dependum”. There can be different nature of dependencies in SD diagrams: goal dependency, task dependency, resource dependency, and softgoal dependency. Fig.1 is an example of a SD model. It represents the dependencies between actors of a Card Payment System.

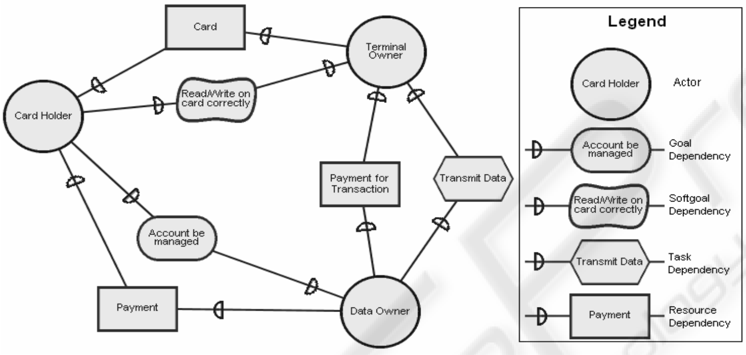


Fig. 1. SD Diagram for a Card Payment System

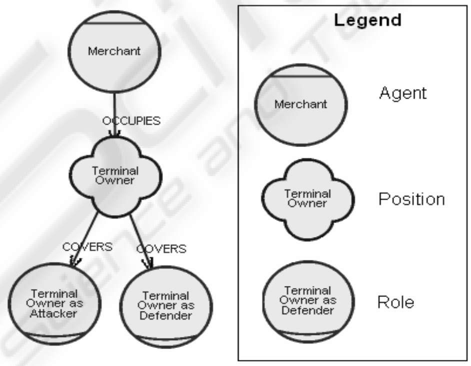


Fig. 2. Example of Different Types of Actors

Actors can be modeled as a generalized relationship among agents, position and role [7]. In general, agents represent physical manifestation of actors. Agents occupy a *position* in SD diagrams. In fact, a *position* is a generalization of an agent. Fig.2 shows an example of different types of actors. Without much elaboration, we present the SR diagram in Fig.3.

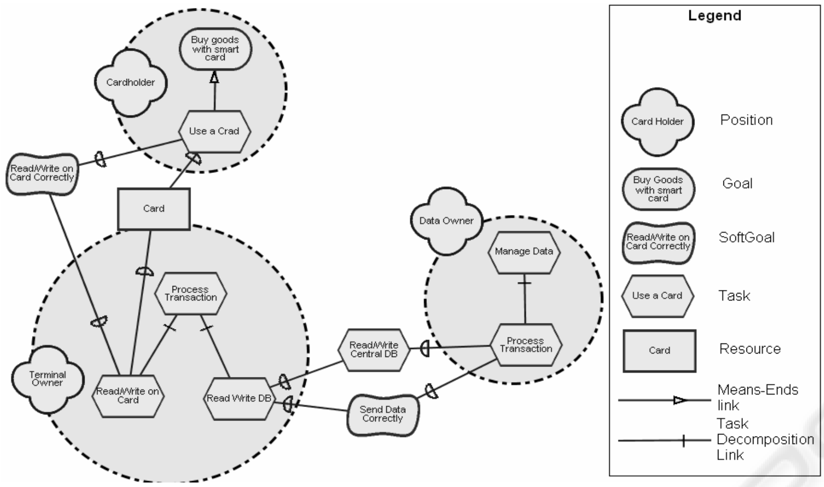


Fig. 3. SR diagram for a Card Payment System

3 Security Risk Assessment

Prior to this work there was no method of using i* framework for security risk assessment. We slightly extend i* framework by altering some i* elements and describe how risk assessment can be done using the extended framework.

Before we describe how to extend i* for risk assessment, we introduce basic concepts of risk management and introduce phases of risk management process according to AS\NZS -- widely referred standard of security management.

3.1 Security Risk Components

Asset: Asset is anything that has value and therefore it has to be protected. Examples of assets are information/data, physical equipment, people and their knowledge, quality of service, image and reputation of company. Information assets have to be preserved in context of confidentiality, integrity, and availability.

Asset value: Asset values are used to describe the importance of assets to a business. Asset value should reflect potential impact of unwanted incidents that can happen to the asset. Asset value can be discreet (in this case it is an element of some finite set, for example {HIGH, MEDIUM, LOW}), or continuous (a number from some predetermined segment, for example [0,1]).

Security Feature: When we deal with information security assets, the assets should be protected in the context of privacy, availability or integrity. Privacy, availability and integrity are different security features of an asset. Harm provided to different security features of the same asset can lead to different consequences to a business. Thus, we can assign different values to different security features.

Threat: A threat is a potential cause of a harm of an asset. A threat leads to reduction of asset value. In information security settings, a threat leads to the loss of confidentiality, integrity or privacy of information. Threats may be acts of nature (i.e., floods, fires, earthquakes), acts of unqualified personal, hackers, and so on As we consider an agent-oriented modeling technique, we will deal only with threats who can be considered as actors acting in an improper way. Threat actors may act in an improper way both intentionally or accidentally.

Vulnerability: Vulnerability is a weakness of the system through which a threat can harm an asset. It is a condition that can allow a threat to harm an asset more frequently or more intensively. For example, if a threat is a hacker who intends to access a website illegally, the possible vulnerabilities may be lack of firewall, firewall is configured incorrectly, or software security patches are not up-to date. To gain some impact on an asset, a threat should exploit vulnerability.

Unwanted incident (Attack): In the settings of agent-oriented approach, unwanted incidents (or attacks) are malicious actions of actors that could harm assets.

Risk: Security risk is a potential that a threat will exploit vulnerability to make some harm to an asset. We assume that a risk will be associated to an attack. The potential is measured by a risk value. Risk value should be determined from a value of all assets that are affected by the attack, and from potential frequency of the attack.

3.2 Risk Management Process

AS\NZS is a widely recognized risk management standard. The standard specifies a risk management process. The process is interactive and its every iteration consists of several stages. The stages are:

- Identification of context, where the target system is described and assets are identified.
- Identification of risks, where threats, vulnerabilities and possible unwanted incidents (attacks) are identified.
- Analysis and evaluation of risks, where frequency of attacks and risks values are identified, and risks are prioritized.
- Identification and documentation of treatment

We will show how to extend the i* framework so that it can be applied on every stage of the risk management process.

Identification of context: In this stage of the risk management process, the target system under assessment is described and assets are identified. To describe the system being assessed, we use the SD diagram of i*. We identify actors of the system and intentional dependencies among the actors. We put the actors and their dependencies in the SD diagram.

The next task is to identify assets. According to CORAS, an asset is everything that has value for stakeholders of the project. As agent-oriented modeling approach does not involve the concept of stakeholder. We may view an asset as anything that has value for one of the actors of the system. The actor, who considers that the asset has value for him, has a goal to preserve the asset. If an asset has value for several actors of the system, then all the actors should have a goal to preserve the asset. Goals associated with the preservation of assets are depicted by internal goal or softgoal elements in the SR diagram. Owner of a credit card can have two assets: card password and account information. Fig.4 shows how the assets are depicted as internal goals of Card Holder in SR diagram. The goals associated with preservation of attacks are subgoals of a generalized goal "Preserve Assets"

The next step is to identify security features that should be protected. The security features are depicted as internal goals and sofgotals of an actor. We use softgoal to depict security feature when it is hard to formalize the criteria that the security feature is protected. Security features are linked with goals associated with assets by the mean-ends dependencies. Fig. 5 shows the security features of assets of a Cardholder. The next step is to assign values to assets and security features. If a security feature corresponding to an asset is identified, we assign a value only to the security feature. The assigned value should reflect importance of an asset or security feature to the business.

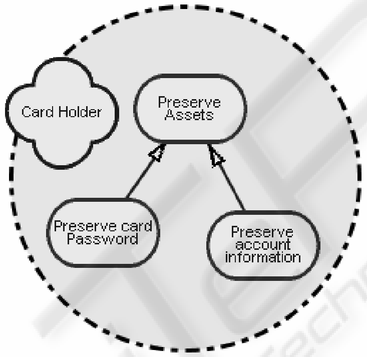
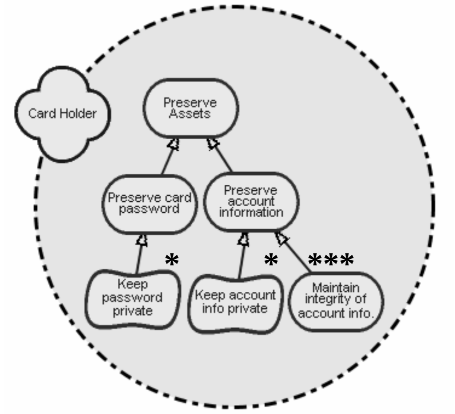


Fig. 4. Assets depicted as internal goals in the SR diagram

If we want to assign value to an asset or a security feature, we assign a value to the corresponding goal or softgoal. We represent values on the top of i* goal and softgoal elements. If we use discreet values, we represent values as star marks ("*"). The example is shown in Fig.5.

Identification of risks: At this step of the risk management process, threats, vulnerabilities and possible unwanted incidents (attacks) should be identified. So far, we identified and put in SR and SD diagrams elements that are related to the normal operation of the system. Now we need to consider potential threats and attacks and determine further dependencies between actors. We depict the newly determined security-related external dependencies in SR diagram. Fig. 6 shows security-related dependencies between Card Holder and Terminal Owner.


Card Holder
Preserve Assets
Preserve card password
Preserve account information
*
*
***
Keep password private
Keep account info private
Maintain integrity of account info.

*

Fig. 5. Security features of assets of a Cardholde

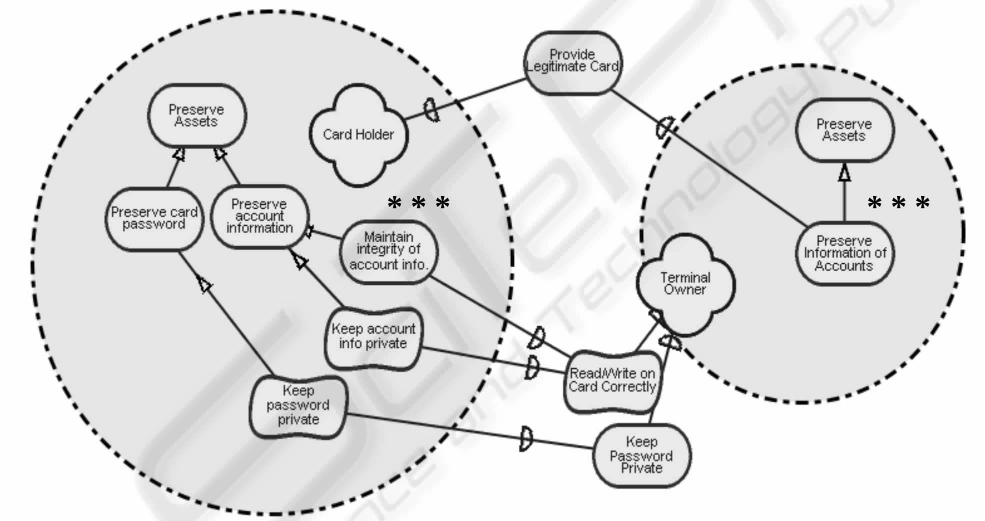


Fig. 6. Security-related dependencies between Card Holder and Terminal Owner

The next task is to identify threats. As we use an agent-oriented modeling technique, threats are treated as actors acting in an improper way. As mentioned before, since they may act in an improper way intentionally or accidentally, in both cases, the improper action is called an attack. Virtually any actor of the system may act as an attacker. For malicious actors we introduce a special role: "Actor as an attacker". Attack is depicted in SR diagram as an internal task of an attacker.

To conduct an attack, an attacker must exploit vulnerabilities. Thus exploitations of vulnerabilities are subtasks of the task associated with the attack and they are linked with task decomposition links. We note task elements corresponding to the exploitation of vulnerabilities by letter “V” on the right of the element (Fig. 7).

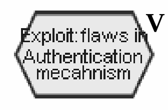


Fig. 7. Representation of task elements corresponding to the exploitation vulnerabilities

If one actor depends on another actor who is an attacker, the attacker may provide a number of attacks to make the dependency not viable. Making a dependency not viable is depicted by a contribution link connecting to dependum. The links originate from the tasks associated with the attacks. For attacks that make dependency totally unviable, link is labeled as a *break* link. This can be seen in the example in Fig.8.

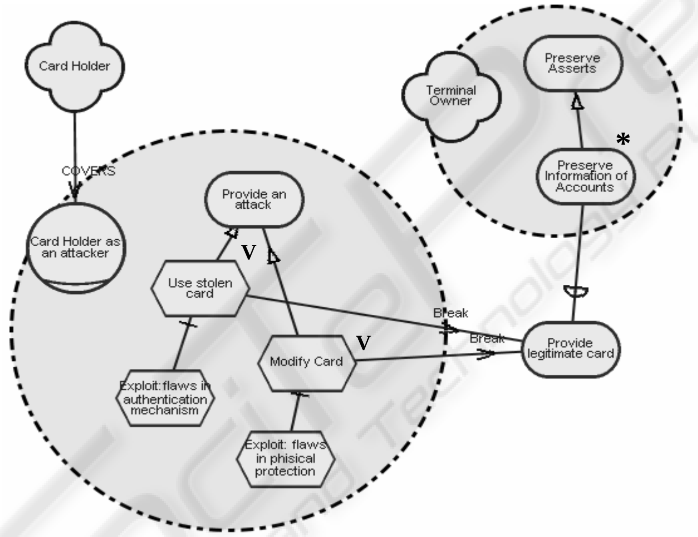

Card Holder
COVERS
Card Holder as an attacker
Provide an attack
V
Use stolen card
V
Modify Card
Exploit:flaws in authentication mechanism
Exploit: flaws in phisical protection
Break
Break
Provide legitimate card
Terminal Owner
Preserve Asserts
*
Preserve Information of Accounts

Fig. 8. Representation of attacks that make the dependency totally unviable. The link is labeled as a *break* link.

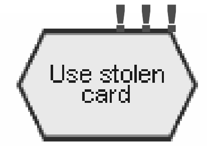


Fig. 9. Depiction of risk values

Risk Analysis and Evaluation: The goal of this step of the risk management process is to determine values of risks. In our approach, we have seen that a risk is associated with an attack. To estimate a risk value, we need to estimate possible frequency of the attack associated with the risk. Then we need to take into consideration values of all the assets that can be affected by the attack.

If we use discreet risk values, we can represent risk values by the exclamation mark ("!") at the top of a task element associated with an attack. For example, three exclamation marks depict that risk value is high, whereas one exclamation mark depict that the risk value is low. Fig. 9 shows an example of how risk value is depicted. If continuous risk values are used, the values can be depicted as numbers on the top of the corresponding task element.

Identification and documentation of treatment: To protect their assets, in other words, to reduce security risks, actors may want to provide countermeasures against possible attacks. An actor who provides defensive measures plays a role "Actor as a defender". The countermeasures are represented as internal tasks of defending actors. The aim of treatment measure is to fix some vulnerability, and thus to reduce the impact or the frequency of the attack. To show that a treatment measure is aimed at fixing vulnerability, we draw a negative contribution link connecting the task representing the treatment measure with the task representing exploitation of the vulnerability. Fig.10 illustrates how treatment measures are depicted in SR diagram.

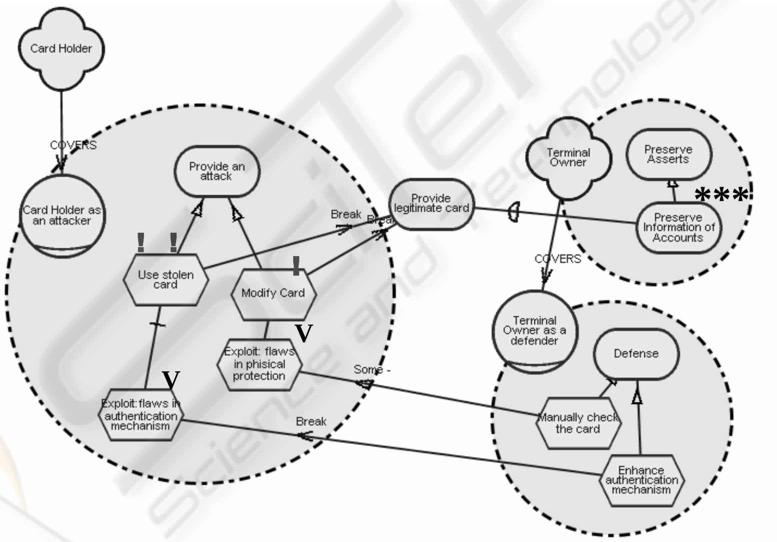

Card Holder
COVERS
Card Holder as an attacker
Provide an attack
! !
Use stolen card
!
Modify Card
V
Exploit: flaws in phisical protection
V
Exploit:flaws in authentication mechanism
Break
Break
Break
Some -
Provide legitimate card
Terminal Owner
Preserve Asserts
***
Preserve Information of Accounts
COVERS
Terminal Owner as a defender
Defense
Manually check the card
Enhance authentication mechanism

Fig. 10. Depiction of *treatment* measures in an SR Diagram

4 Conclusion

We have outlined a new technique for modeling information system security risk analysis using the concept of actor-dependency, and extending its scope to the domain of security risk management. The technique can reason about the opportunities, vulnerabilities, changes, and risks that are associated with information systems security, and can incorporate prominently the issues related to security risk.

i* was not originally designed for studying security risks. We have leveraged the concept of actor-dependency of i*, extended it to address its limitations for use in security risk management We recommend further research to design a new language for modeling actor-dependencies in security risk management.

References

1. Aagedal, J.O., Braber, F. D., Dimitrakos, T., Gran, B.A., Raptis, D., Stolen, K.: Model-Based Risk Assessment to Improve Enterprise Security. In *Proceedings of the Fifth International Enterprise Distributed Object Computing Conference (EDOC 2002),* September 17-20, Lausanne, Switzerland (2002)
2. Barber, B., and Davey, J.: The use of the CCTA Risk Analysis and Management Methodology (CRAMM) in health information systems. In: Medinfo 92. Amsterdam: North Holland (1992) 1589-1593.
3. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J. *Non-Functional Requirements in Software Engineering*, Kluwer Academic Publishers (2000)
4. Common Criteria Organization: Common Criteria for Information Technology Security Evaluation. http://www.commoncriteria.org, accessed: 2004 (2002)
5. Control Objectives for Information and Related Technology, 2002. “COBIT”. http://www.isaca.org/ct-denld.htm (2002)
6. Donzelli, P., Bresciani, P.: An Agent-Based Requirements Engineering Framework for Complex Socio-Technical Systems. In *Proceedings of SELMAS 2003*, Portland (2003)
7. Dubois, E., Yu, E. and Petit, M.: From Early to Late Formal Requirements: a Process Control Case Study. In *Proc. 9th International Workshop on Software Specification and Design*, April 16-18, Ise-Shima, Japan (1998) 34-42.
8. Gans, G., Jarke, M., Kethers, S., Lakemeyer, G., Ellrich, L., Funken, C., Meister, M.: Requirements Modeling for Organization Networks: A (Dis)Trust-Based Approach, In *Proc. 5th IEEE International Symposium on Requirements Engineering*, Toronto (2001)
9. Mouratidis, H., Giorgini, P., Manson, G., Philip, I.: A Natural Extension of Tropos Methodology for Modeling Security. In *Proceedings of the Agent Oriented* Methodologies *Workshop (OOPSLA 2002)*, November, Seattle-USA (2002)
10. Reactive System Design Support: RSDS. http://www.kcl.ac.uk., Sandia National Laboratories (2002)
11. Schechter, S.E: Computer Security & Risk: A Quantitative Approach. Ph.D. Thesis, Computer Science, Harvard University (2004)
12. Standards Australia: AS/NZS 4360: Risk Management. AS/NZS 4360 (1999)
13. Sutcliffe, A.G. and Minocha, S: Linking Business Modeling to Socio-technical System Design, In Proceedings of CaiSE’99 (1999) 73-87.
14. Vraalsen, F., Braber, F.D., Hogganvik, I., Lund, S., Stolen, K: The CORAS Tool-Supported Methodology. SINTEF Report, Report # STF90A04015, February, Norway (2004)