Author(s): Matulevičius, Raimundas; Mayer, Nicolas; Mouratidis, Haralambos; Dubois, Eric; Heymans, Patrick; Genon, Nicolas.
Article title: Adapting Secure Tropos for Security Risk Management during Early Phases of the Information Systems Development
Year of publication: 2008
Citation: Matulevicius, R. et al. (2008) 'Adapting Secure Tropos for Security Risk Management during Early Phases of the Information Systems Development' In: Dubois, E; Pohl, K. (Eds) CAiSE 2008, LNCS 5074 pp 541-555
Link to published version: http://dx.doi.org/10.1007/978-3-540-69534-9_40
DOI: 10.1007/978-3-540-69534-9_40

Adapting Secure Tropos for Security Risk Management during Early Phases of the Information Systems Development

Raimundas Matulevičius, Nicolas Mayer, Haralambos Mouratidis, Eric Dubois, Patrick Heymans, and Nicolas Genon

PReCISE, Computer Science Faculty, University of Namur, Belgium
{rma, phe, nge}@info.fundp.ac.be
CRP Henri Tudor - CITI, Luxembourg
{nicolas.mayer, eric.dubois}@tudor.lu
School of Computing and Technology, University of East London, UK
H.Mouratidis@uel.ac.uk

Abstract. Security is a major target for todays information systems (IS) designers. Security modelling languages exist to reasoning on security in the early phases of IS development, when the most crucial design decisions are made. Reasoning on security involves analysing risk, and effectively communicating risk-related information. However, we think that current languages can be improved in this respect. In this paper, we discuss this issue for Secure Tropos, the language supporting the eponymous agent-based IS development methodology. We analyse it and suggest improvements in the light of an existing reference model for IS security risk management. This allows checking of Secure Tropos concepts and terminology against those of current risk management standards, thereby improving the conceptual appropriateness of the language. The paper follows a running example, called eSAP, located in the healthcare domain.

Key words: Risk analysis, information systems, security, Secure Tropos, information systems security risk management.

1 Introduction

Information systems (ISs) undoubtedly play an important role in today's society and more and more are at the heart of critical infrastructures. ISs are also facing an increasing complexity because of their interoperability with other systems and of their operation in open, distributed and mobile environments. In such contexts, secure issues are vital and are still reinforced in many sectors with the introduction of new regulations like Basel II [1] or SOX [2]. Risk management is considered as central by IS professionals. These activities do not only support security officers in the handling of security vulnerabilities but they also provide a framework in terms of which the return on investment of the security solutions are evaluated against the economic and business consequences of

not implementing them. Today exists more than 200 risk management methods with the challenge to select the most adequate approach. Through our analysis [3] [4] we have identified some important points for possible improvements. Firstly, elements are related to the nature of the artefacts produced with such methods. These artefacts are largely informal and typically consist of natural language documents, complemented with tables and ad hoc diagrams for structuring the information. The powerful abstraction mechanisms and visualisations offered by conceptual modelling techniques are thus underexploited. Secondly, a drawback of methods is that they are often designed for being used to assess the way existing systems handle risk in an auditing mode. This view is no longer sustainable in the context of todays ISs that need to constantly adapt to new environments and handle evolution with minimum human intervention. This is an additional argument for the use of more formal languages supporting the reasoning, evolution, monitoring and traceability of risk related information.

In this paper we report on a research related to the design of a suitable modelling language for supporting security risk management (SRM) activities. Central in this research is to first achieve a deep understanding of the SRM domain, then to design an adequate language with suitable constructs and associated semantics for that domain. A central focus of risk management methods is to consider security issues from the very early phases, a.k.a. *requirements engineering* (RE), of ISs development. The associated scientific literature features a number of modelling languages specifically dedicated to security sensitive contexts; however the risk concepts are only partially supported. This advocates for the design of 'yet another' modelling language. However, defining a complete new notation does not appear to us as viable option from a sustainability perspective for the modelling community. As demonstrated for example with UML in software engineering, a consensus over unified and common notations has been proved to be a big push for the adoption of modelling practices in public and private companies. At RE level we plead for a similar approach and rather than the development of a totally new language we improve existing languages, offering an ontological basis sufficiently closed to the risk management domain.

With respect to the above objective, we have identified Secure Tropos [5], which uses the concept of security constraint and methods such as security attack scenarios to analyse security requirements, as a suitable candidate language. The selection of Secure Tropos results from a detailed analysis of the adequacy of its concepts to the *information system security risk management* (ISSRM) reference model [3] [4]. This reference model defines the fundamental concepts of ISSRM as gathered from a quantity of standards and other sources, e.g., [6] [7] [8]. The overall approach is illustrated throughout this paper reusing the example of the electronic Single Assessment Process (eSAP) [9].

The structure of the paper is as follows: in Section 2 we provide theoretical background for our research. In Section 3 we outline our research method and apply Secure Tropos in the running example. In Section 4 we describe how Secure Tropos is aligned with the concepts of the ISSRM reference model. Finally Section 5 discusses the findings and presents conclusions of the study.

2 Theory

In this section we introduce the security risk management domain concepts, then some candidate supporting security modelling languages.

2.1 Security Risk Domain

The ISSRM Reference model [3] [4] presented in Fig. 1 results from a consolidation of existing security standards, e.g., [6], [7], [8]. Like the Tropos Goal-Risk framework [10], the ISSRM reference model addresses risk management at three different levels, combining together asset, risk, and risk treatment views. However the ISSRM reference model focuses on the *IS security* perspective while the Tropos Goal-Risk framework supports risk in general. In this section we summarise some core definitions of ISSRM concepts; for more details see [4].

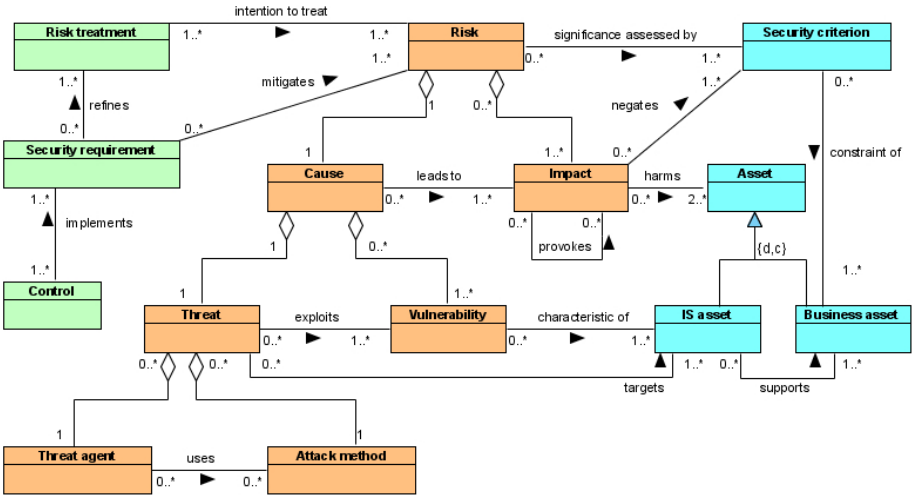

Fig. 1. The ISSRM Reference Model [3] [4]

Asset-related concepts describe what assets are important to protect, and what criteria guarantee asset security. An *asset* is anything that has value to the organisation and is necessary for achieving its objectives. A *business asset* describes information, processes, capabilities and skills inherent to the business and core mission of the organisation, and that has value for it. An *IS asset* is a component of the IS supporting business assets like e.g. a database where is stored medical information of patients. *Security criterion* characterises a property or constraint on business assets describing their security needs. They are often confidentiality, integrity and availability, but sometimes, depending on the context, other specific criteria might be added, like non-repudiation or accountability.

Risk-related concepts present how the risk itself is defined. A *risk* is the combination of a threat with one or more vulnerabilities leading to a negative impact harming the assets. An *impact* describes the potential negative consequence of a risk that may harm assets of a system or an organisation, when a threat (or the cause of a risk) is accomplished. The *cause of the risk* is the combination of a threat and one or more vulnerabilities. A *vulnerability* describes a characteristic of an IS asset or group of IS assets that can constitute a weakness or a flaw in terms of IS security. A *threat* characterises a potential attack or incident, which targets one or more IS assets that may lead to harm the assets. A *threat agent* is an agent that can potentially cause harm to IS assets. An *attack method* is a standard means by which a threat agent carries out a threat.

Risk treatment-related concepts describe what decisions, requirements and controls should be defined and implemented in order to mitigate possible risks. A *risk treatment* is a decision of the intention to treat identified risks. A *security requirement* is the refinement of a treatment decision to mitigate the risk. *Controls* (countermeasures or safeguards) are means designed to improve security, specified by a security requirement, and implemented to comply with it.

Security risk management process. The ISSRM activities follow the general risk management process described in traditional risk management standards, e.g., [6], [7], [8]. It can be summarised into six steps. Here we just briefly recall each step. For more details see [4]. The process begins with a (a) definition of the organisation's *context* and the *identification of its assets*. Next one needs to determine the (b) *security objectives*, such as confidentiality, integrity and availability, based on the level of protection required for the assets. During (c) *risk assessment* one elicits which risks are harming assets and threatening security objectives. Once risk assessment is performed, decisions about (d) *risk treatment* are taken. Decisions might include risk avoidance, risk reduction, risk transfer and risk retention. *Security requirements* (e) on the IS can thus be determined as security solutions to mitigate the risks. Requirements are instantiated into (f) *security controls*, i.e. system specific countermeasures, which are implemented within the organisation. Finally it should be noted that the risk management process is iterative. After determination of the security controls new risks that overcome or are not addressed by these security controls, can emerge.

2.2 Security Modelling Languages

At different IS development phase, security can be addressed using various modelling languages. Abuse frames [11] suggests means to consider security during the early RE. Abuse cases [12], misuse cases [13], and mal-activity diagrams [14] address security concerns through negative scenarios executed by system attacker. SecureUML [15] and UMLsec [16] consider security at system design.

Goal modelling languages have also been adapted to security. Secure i^* [17] addresses security trade-offs during early requirements. KAOS [18] was augmented with so-called *anti-goal models* designed to elicit rationales of attackers. In [19] [20] Tropos has been extended with the notions of *ownership*, *permision*

and *trust*. In this paper we investigate Secure Tropos [5] [21] [22] that models security using *security constraints* and *attack methods*.

All these languages are candidate for supporting largely or partially the SRM activities. For the purpose of this paper we have chosen Secure Tropos that incrementally introduces security concerns through the IS development cycle. The final analysis of the security takes place only during the late development phases – during design [22]. In this paper we address this problem by using the ISSRM reference model to improve the language with SRM aspects.

2.3 Secure Tropos

Secure Tropos enriches a set of Tropos [23] [24] constructs (*actor*, *goal*, *softgoal*, *plan*, *resource*, *threat*, and *belief*) with security constructs such as *security constraint*, and *threat*. An *actor* (see Fig. 3) describes an entity that has strategic goals and intentions within the system or within the organisation settings [23]. A *hardgoal* or simply *goal* hereafter (see Fig. 3), represents an actors' strategic interests. A *softgoal* (see Fig. 5) unlike a *goal*, does not have clear criteria for deciding whether it is satisfied or not and therefore it is subject to interpretation (goals are said to be *satisfied* while softgoals are said to be *satisficed*). A *plan*(see Fig. 4) represents a way of doing things. A *resource* (see Fig. 3) represents an informational or physical entity. A *belief* (see Fig. 7) is the actor's knowledge of the world. All these constructs are present in both Tropos [23] [24] and Secure Tropos [9], [21], [22]. In addition Secure TROPOS introduces *Security constraint* and *Threat*. A *security constraint* represents a restriction related to security that the system must have and actors must respect (see Fig. 3) [5] [21]. A *threat* (see Fig. 6) "represents circumstances that have the potential to cause loss or problems that can put in danger the security features of the system" [5].

Constructs are combined together using relationships: *dependency*, *decomposition*, *means-ends*, *contribution*, *restricts* and *attacks*; so separating between actor and goal models. In the actor model one represents the network of relationships between actors. The relationships are captured using the *dependency* links. *Dependency* between two actors indicates that one actor (the depender) depends for some reason (dependum) on another actor (the dependee) in order to achieve a goal, to execute a plan, or to deliver a resource [23]. *Secure dependency* introduces security constraint(s) that must be respected by actors for the dependency to be satisfied [25]. This means that "the depender expects from the dependee to satisfy the security constraint(s) and also that the dependee will make effort to deliver the dependum by satisfying the security constraint(s)" [21]. The goal model allows a deeper understanding of actors' reasoning about goals to be fulfilled, plans to be performed and available resources [24]. The goal model uses the *means-ends*, *decomposition* and *contribution* relationships. The *means-ends* relationship (see Fig. 4) permits to link a *means* (plan/goal/resource) with a *end* (goal). The *decomposition* relationship (see Fig. 4) permits to define a finer structure of a plan. Only a plan can be decomposed into goals, softgoals, resources and (sub)plans. A *contribution* link (see Fig. 5) describes a positive or negative impact that one element has on another. To facilitate security analysis

Secure Tropos introduces *restricts* and *attacks*. The *restricts* relationship (see Fig. 3) describes how goal achievement is restricted by security constraints. The *attacks* link (see Fig. 7) shows what is the target of an attacker's plan.

3 Research Method

3.1 Method for Aligning Secure Tropos and ISSRM

In order to align Secure TROPOS with the ISSRM reference model, the method shown in Fig. 2 is applied. Our approach is based on the definition of the Secure Tropos language as it is derived from the Secure Tropos meta-model and the description of the language in the literature [9] [5] [21] [22] [25].

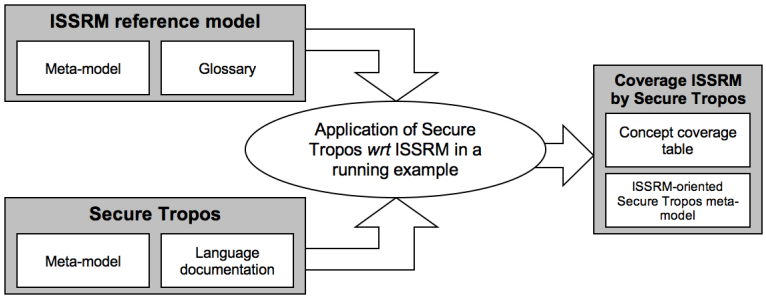

Fig. 2. Research Method

In this paper we use a running example to explain our analysis towards the alignment of the Secure Tropos and ISSRM. The running example is initially used to illustrate the use of the language to address the security risks during early IS development. We then consider the concepts of Secure Tropos *wrt* on how they were used to address ISSRM. The outcome of the comparison is the concept alignment between the language and the ISSRM reference model. We document the final results of our alignment artefacts in Fig. 9. At the same time, an "ISSRM-oriented" Secure TROPOS meta-model is produced. By "ISSRM-oriented", we mean a meta-model[1] aligned on the ISSRM reference model and thus showing only concepts and relationships semantically equivalent to those of the ISSRM reference model.

3.2 Running example

To demonstrate the applicability of our work in a practical and realistic environment we employ the electronic Single Assessment Process (eSAP) [26]. The

[1] In the paper due to the space requirements we do not include Secure Tropos meta-model and ISSRM-oriented Secure Tropos meta-model.

running example is suitable to demonstrate our work for two main reasons: (*i*) security and risk are two important factors in the development and implementation of an electronic system to support the Single Assessment Process; (*ii*) security of the system have been successfully analysed using the Secure Tropos methodology [27]. Therefore, by revisiting the running example, we are able to identify the exact contributions of this paper. eSAP is an IS to support integrated assessment of the health and social care needs of elderly and it is based on the Single Assessment Process, which is part of the National Service Framework (NSF) for Older People Services of the English Department of Health. Due to space limitations, we focus for our running example on one of the most important aspects to make the eSAP running: the Patient personal information.

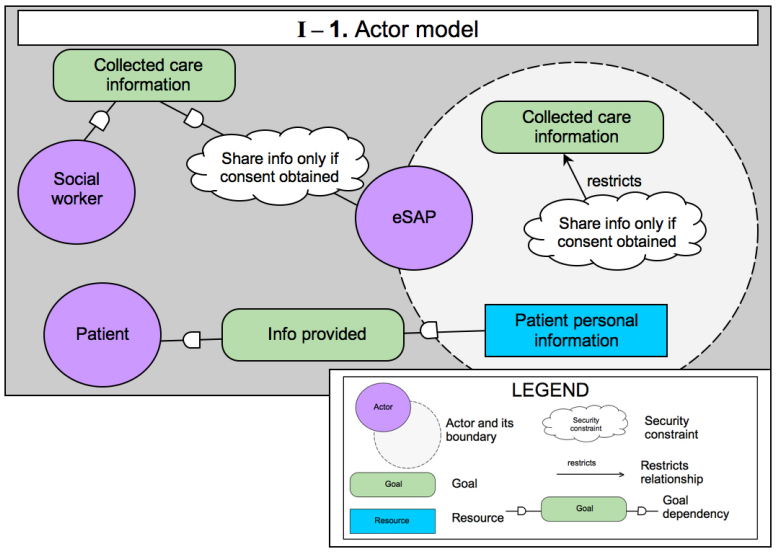

Fig. 3. Actor model

(*a*) ***Context and Asset Identification.*** A Social Worker is in charge of the health care to patients. In order to fulfil her work, she needs the Patient personal information. In Fig. 3 the Social Worker depends on a goal Collected care information held by the eSAP system. As the Patient personal information is a valuable business asset, achievement of the goal Collected care information is restricted by a security constraint Share info only if consent obtained assuring that the consent has to be obtained before the personal information can be sent. The goal Collected care information can be achieved by executing the plan Collect info about treatment, which needs to gather the Patient personal information and to perform the Manage care plan.

(*b*) ***Security objective determination.*** The plan Check data for consent contributes positively to the security constraint Share info only if consent obtained

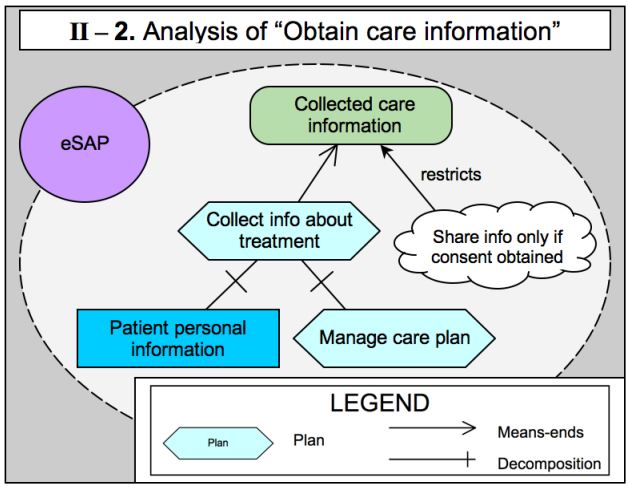

Fig. 4. Analysis of "Obtain care information"

(Fig. 5). This plan is the means to achieve the goal Consent has been obtained. In our example we strive for privacy of the Patient personal information, thus the goal Consent has been obtained takes part in the decomposition of the plan Perform authorisation checks. The latter plan is the means to a goal System privacy ensured and contributes positively to a security constraint Keep system data privacy.

(*c*) *Risk analysis and assessment.* Fig. 6 focuses on a possible cause of the risk to which the eSAP could be exposed. We identify the Authentication attack (which in Secure Tropos is modelled using the threat construct). It describes a situation where a threat agent passes himself off as a trusted actor in order to fake identity and to damage the business assets (e.g., Patient personal information). The Authentication attack has a negative impact on the Privacy softgoal. The constraint Keep system data privacy has a positive impact on the privacy of the system and can make the possible risk difficult to realise. Note that the Authentication attack is not placed as an internal concept. The cause of risk does not depend on the existence of the actor whose assets the risk threatens.

In Fig. 7 we present the view of an Attacker whose aim is to get the Patient personal information. The Attacker has a threat that is characterised by the goal Info about patient received and plan Collect info about breaking the system. Plan is decomposed into two parts: *i*) the attacker has to get information about the consent for the Patient personal information; and *ii*) he needs to find the authentication code to the information. To get the consent, the attacker can Steal data from a social worker or Buy data from the untrusted social worker. In the example the belief Possible to check eSAP access repeatedly corresponds to a vulnerability, known by the attacker, for the eSAP system. The vulnerability contributes positively to the decomposition between two plans Collect info about breaking the system and Check eSAP access repeatedly. The entire Fig. 7 can be seen as the refinement of the cause of the risk identified in Fig. 6.

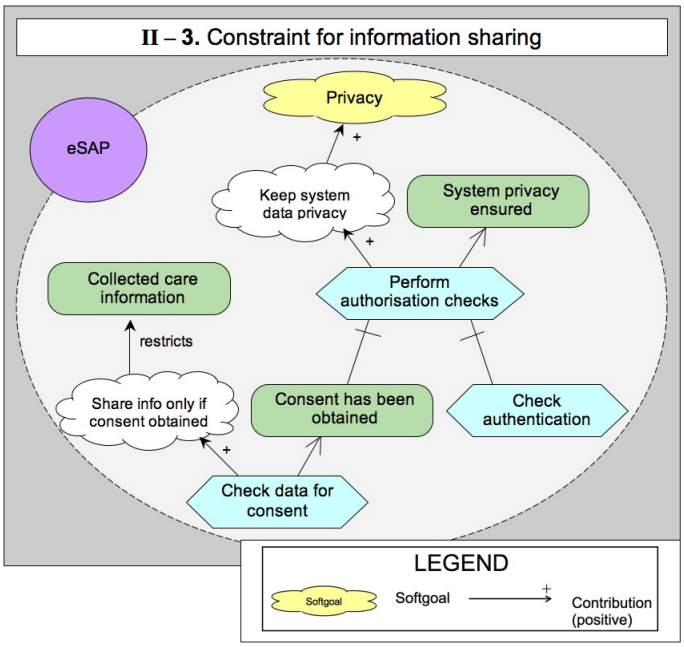

Fig. 5. Constraint for information sharing

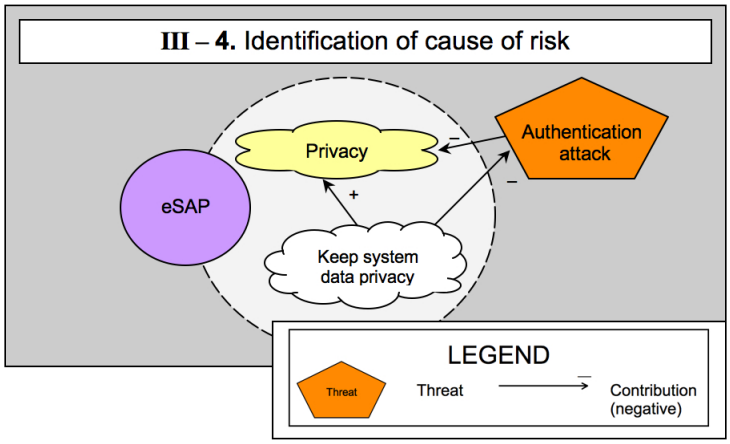

Fig. 6. Identification of an authentication risk

(*d*) *Risk treatment.* Several risk treatment decisions are suggested in [28]. In the example we apply *goal/plan substitution*, meaning that we choose different goals to be fulfilled and plans to be executed to mitigate the risk. This produces different system design but allows avoiding the Authentication attack.

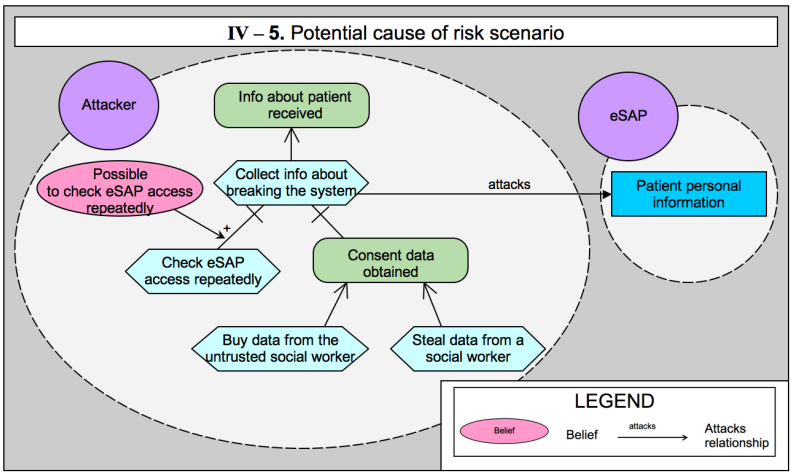

Fig. 7. Potential attack scenario

(*e*) *Security requirements definition.* Next step is the elicitation of the countermeasures that help to mitigate the actual risk. With respect to Fig. 5, we try to find an alternative means to achieve the goal System privacy ensured. Our solution is to Perform cryptographic procedures (Fig. 8). To fulfill the countermeasure, Encrypt data and Decrypt data are performed at the certain time moment. Our countermeasure avoids the Authentication attack because now the eSAP system is designed in a way that it does not require the authentication information. However this might bring other causes of the risk (e.g., Cryptographic attack) which need to be again analysed iteratively.

(*f*) *Control selection and implementation.* Control selection can be reasoned using softgoals distinguishing between different design alternatives. The step takes place after defining controls following the security requirements.

4 Contribution

Our analysis contributes with the semantic alignment between ISSRM and Secure Tropos. In the example we illustrate how we can use the Secure Tropos approach to analyse possible attack scenarios and how from attack scenarios we can withdraw countermeasures. We summarise the discussion on alignment in Fig. 9. First two columns list the concepts of the ISSRM reference model, the third column provides synonyms of the ISSRM concepts found in the Secure Tropos literature [21] [9] [5] [25] [22]. The fourth column list the Secure TROPOS constructs used to address the ISSRM concepts, the last column provides illustration of the Secure TROPOS concept used in the running example in Section 3.2.

Asset-related concepts describe what assets are important to protect, and what criteria guarantee asset security [3]. In Secure TROPOS we identify that

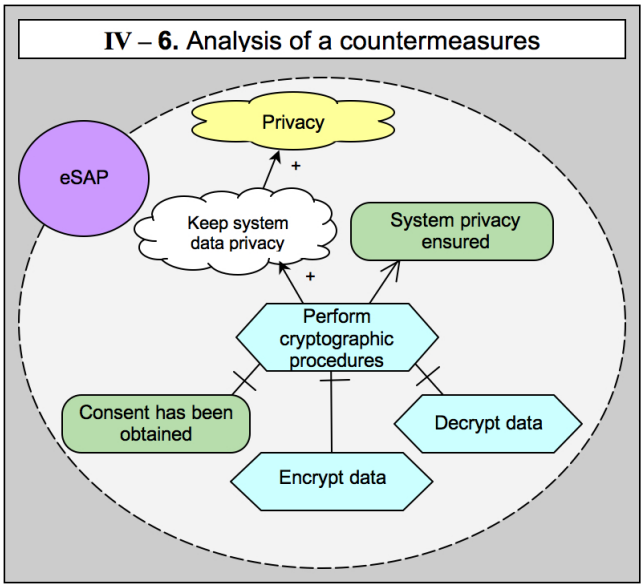

Fig. 8. Analysis of a countermeasure

The ISSRM model		Secure TROPOS		
		Synonym found in literature*	Language constructs	Elements from example
Asset-related concepts	Asset	—	Actor, Goal, Softgoal, Plan, Resource	—
	Business asset	—		**Actor**[Patient]; **Actor**[Social worker]; **Goal**[Obtain care information]; **Goal**[Info provided]; **Resource**[Patient personal information]; **Plan**[Collect info about treatment]; **Plan**[Manage care plan]
	IS asset	—		**Actor**[eSAP]; **Goal**[System privacy ensured]; **Plan**[Perform authorization check]; **Plan**[Check authentication]; **Goal**[Consent has been obtained]; **Plan**[Check data for consent]
	Security criteria	Security feature, Protection property	Security constraint, Softgoal	**Security constraint**[Share info only if consent obtained]; **Security constraint**[Keep system data privacy] **Softgoal**[Privacy]
Risk-related concepts	Risk	—	—	—
	Impact	—	Contribution between the threat and softgoal	**Contribution** between **Threat**[Authentication attack] and **Softgoal**[Privacy]
	Cause of the risk	—	Threat	**Threat**[Authentication attack]
	Threat	—	Goal, Plan	**Plan**[Collect info about breaking the system]; **Goal**[Consent data obtained]
	Vulnerability	—	Belief **	**Belief**[Possible to check eSAP access repeatedly]
	Threat agent	Attacker	Actor	**Actor**[Attacker]
	Attack method	—	Plan, Reatationship attacks	**Plan**[Collect info about breaking the system]; **Plan**[Check eSAP access repeatedly]; **Plan**[Steal data from a social worker]; **Plan**[Buy data from untrusted social worker]; **Plan**[Collect info about breaking the system] **attacks** **Resource**[Patient personal information]
Risk treatment-related concepts	Risk treatment	—	—	—
	Security requirement	Secure goal, secure plan, secure resource, protection objective	Actor, Goal, Softgoal, Plan, Resource Security constraint	**Plan**[Perform cryptographic procedures]; **Plan**[Encrypt data]; **Plan**[Decrypt data]
	Control	—	—	Cryptographic module in the eSAP system

Fig. 9. Alignment between the ISSRM reference model and Secure Tropos. * – literature includes [9] [5] [25] [22]; ** – look for discussion about belief in section 4

actor, *goal*, *resource* and *plan* constructs (and appropriate relationships among them) are used to model both *business* and *IS assets*. For instance, on the one hand *actors* Patient and Social worker (see Fig. 3), *goals* Obtain care information and Info provided and *plans* Collect info about treatment and Manage care plan (see Fig. 4) describe the process necessary for organisation (health care centre) to achieve its objective. On the other hand *resource* Patient personal information characterises the valuable information. All the mentioned examples are identified as *business assets* with respect to the ISSRM reference model [3]. The business processes and information management are supported by the IS, which in our example corresponds to eSAP. In more detail (see Fig. 5) the support for the *business assets* is described by *goals* System privacy ensured and Consent has been obtained and *plans* Perform authorisation check, Check authentication and Check data for consent. The concepts which describe how a component or part of the IS is necessary in supporting *business assets*, are called *IS assets*.

The ISSRM *security criteria* are properties or constraints on business assets characterising their security needs [3]. In Secure Tropos *softgoals* (e.g. Privacy) can help identify higher level security criteria, like privacy, integrity and availability. Depending on the context it might be necessary to specify other *security criteria*, like we do using *security constraints* Share info only if consent obtained and Keep system data privacy (see Fig. 5).

Risk-related concepts present how the risk itself is defined, what are the major principles that should be taken in account when defining the possible risks [3]. Risk is described by the cause of the risk, corresponding to the Authentication attack in Fig. 6. The potentional negative consequence of the risk, identified by a negative contribution link between the Authentication attack and the *security constraint* Privacy is called impact of the risk. Here the impact negates the security criteria and tends to make the *business asset* not private.

In Fig. 7 a combination of the *goal* Info about patient received and the *plan* Collect info about breaking the system corresponds to the *threat* describing the potential attack targeting the *business asset* Patient personal information. The threat is triggered by the *threat agent* Attacker who *knows* about possibility to check the eSAP access repeatedly as identified by *belief* in Fig. 7. To break into the eSAP system the Attacker carries an *attack method* consisting of plans Check eSAP access repeatedly and Steal data from a social worker.

Note that in Fig. 9 *belief* only partially corresponds to ISSRM *vulnerability*. Firstly, the facts that the *actor* (who has role of *attacker*), thinks he knows, might be true – in this case *belief* will correspond to ISSRM *vulnerability*. However, it does not allow lining to a system design solution because this solution might not exist in the early IS development phase. Secondly, facts known by the *attacker* might be wrong; in this case *belief* will not have correspondence in ISSRM. Finally, *belief* does not represent *vulnerabilities* which exist in the system but is not known by the *attacker*.

Risk treatment-related concepts describe what decisions, requirements and controls should be defined and implemented in order to mitigate possible risks [3]. According to [18] [28] in our example we select *goal/plan substitution*

which allows producing a different eSAP design and thus avoiding the identified threat. New *security requirements* (see Fig. 8) that mitigate the risk are identified as *plans* Perform cryptographic procedures, Encrypt data, and Decrypt data. We illustrate the countermeasure only using the Secure Tropos *plan* construct, however we must admit that depending on the selected *risk treatment decision* the combination of *actor*, *goal*, *resource* and *plan* might result in the different security *control* systems.

5 Discussion and Conclusion

In this paper we have analysed how Secure Tropos can be applied to analyse security risks at the early IS development phases. Based on an illustrative example, we showed how a Secure Tropos model can be created following security risk management process. Our purpose was not to develop the complete running example (for instance we do not detail how plan Check data for consent in Fig. 5 has to be performed), but rather to investigate how different language constructs can be used to model security risks. We focus on the early phase (early and late requirements) of the IS development. This means that the analysis of Secure Tropos is not complete *wrt* the late development phases, for instance we do not consider *capabilities* which are the notion used during IS design.

We know that our research method and results could hold a certain degree of subjectivity regarding the selection of the Secure TROPOS language's constructs at the modelling stage, their application and their comparison with ISSRM. To deal with the subjectivity within the team we (*i*) looked at the meta-model of Secure Tropos and make precise unclear use of language constructs; (*ii*) collectively agreed about decisions made when creating the running example; (*iii*) discussed and reasoned about the Secure Tropos and ISSRM alignment.

The alignment suggests a number of improvements for Secure Tropos to use it in the context of security risk management activities:

- Secure Tropos has to provide guidelines as to when and how to use the constructs to avoid misinterpretations of the ISSRM concepts. One of the improvement is inclusion of the tags into the label of the constructs. For example, the *plan* can be used to model *business assets*, *IS assets*, *threats* and *security requirements*. Thus, labels such as [BS] could indicate *business assets*; [IS]– *IS assets*; [Th]– *threat*; and [Sc]– *security requirements*. In our running example we deal with this limitation by decomposing the model into separate diagrams: we use *plan* to represent *business assets* in Fig. 4, *IS assets* in Fig. 5, *theats* in Fig. 7, and *security requirements* in Fig. 8.
- Secure Tropos could be improved with additional constructs to better cover the concepts of ISSRM. Fig. 9 indicates that several concepts such as *risk*, *risk treatment*, and *control* are not reflected in the Secure Tropos approach.
- The semantics of individual modelling constructs should be adapted so that they adequately represent ISSRM concepts. For example, as discussed, *belief* only partially covers *vulnerability*. A possible improvement is recently suggested in [17] by introducing *vulnerable points* in the modelled IS. But some

future research is needed to answer if relationship between *vulnerable points* and *belief* is possible.

Besides Secure Tropos we have also analysed KAOS extension to security [18] and misuse cases [13]. We envision that after analysing a number of security languages it will be possible to facilitate model transformation and languages interoperability. This would allow representing IS using different perspectives, also ensuring IS sustainability.

References

1. Basel Committee on Banking Supervision: International Convergence of Capital Measurement and Capital Standards. Bank for International Settlements (2004)
2. Senate, U.S., of Representatives in Congress (2002), H.: Sarbanes-Oxley Act of 2002. Public Law 107-204 (116 Statute 745) (2002)
3. Mayer, N., Heymans, P., Matulevičius, R.: Design of a Modelling Language for Information System Security Risk Management. In: Proceedings of the 1st International Conference on Research Challenges in Information Science (RCIS 2007). (2007) 121–131
4. Mayer, N., Heymans, P., Matulevičius, R.: Design of a Modelling Language for Information System Security Risk Management. Technical report, CRP Henri Tudor and University of Namur (2006)
5. Mouratidis, H., Giorgini, P., Gordon, M., Philp, I.: A Natural Extension of Tropos Methodology for Modelling Security. In: Proceedings of the Agent Oriented Methodologies Workshop (OOPSLA 2002). (2002)
6. DCSSI: EBIOS–Expression of Needs and Identification of Security Objectives (2004)
7. ENISA: Inventory of Risk Assessment and Risk Management Methods (2004)
8. ISO: Information technology–Security techniques–Information security management systems–Requirements, International Organisation for Standardisation (2005)
9. Mouratidis, H., Giorgini, P., Manson, G.: Using Tropos Methodology to an Model Integrated Health Assessment System. In: Proceedings of the Fourth International Bi-Conference on Agent-oriented Information Systems (AOIS'02). (2002)
10. Asnar, Y., Giorgini, P.: Modelling Risk and Identifying Cuntermeasure in Organizations. In: Proceedings of the 1st Interational Workshop on Critical Information Intrastructures Security, Springer-Verlag Berlin Heidelberg (2006) 55–66
11. Lin, L., Nuseibeh, B., Ince, D., Jackson, M.: Using Abuse Frames to Bound the Scope of Security Problems. In: Proceedings of the 12th IEEE international Conference on Requirements Engineering (RE'04), IEEE Computer Society (2004) 354–355
12. McDermott, J., Fox, C.: Using Abuse Case Models for Security Requirements Analysis. In: Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC'99). (1999) 55
13. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering Journal **10**(1) (2005) 34–44
14. Sindre, G.: Mal-activity Diagrams for Capturing Attacks on Business Processes. In: Proceedings of the Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2007), Springer-Verlag Berlin Heidelberg (2007) 355–366

15. Lodderstedt, T., Basin, D.A., Doser, J.: SecureUML: A UML-based Modeling Language for Model-driven Security. In: Proceedings of the 5th International Conference on the Unified Modelling Language (UML'02), Springer-Verlag (2002) 426–441
16. Jurjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Proceedings of the 5th International Conference on the Unified Modelling Language (UML'02). (2002) 412–425
17. Elahi, G., Yu, E.: A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs. In Parent, C., Schewe, K.D., Storey, V.C., Thalheim, B., eds.: Proceedings of the 26th International Conference on Conceptual Modelling (ER 2007). Volume 4801., Springer-Verlag Berlin Heidelberg (2007) 87–101
18. van Lamsweerde, A.: Elaborating Security Requirements by Construction of Intentional Anti-models. In: Proceedings of the 26th International Conference on Software Engineering (ICSE'04), IEEE Computer Society (2004) 148–157
19. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modeling Security Requirements Through Ownership, Permision and Delegation. In: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE'05), IEEE Computer Society (2005)
20. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modelling social and individual trust in requirements engineering methodologies. In: Proceedings of the 3rd International Conference on Trust Management. LNCS, Springer-Verlag (2005) 161–176
21. Mouratidis, H., Giorgini, P.: Enhancing Secure TROPOS to Effectively Deal with Security Requirements in the Development of Multiagent Systems. In: Proceedings of the 1st International Workshop on Safety and Security in Multiagent Systems (AAMAS 2004). (2004)
22. Mouratidis, H., Jurjens, J., Fox, J.: Towards a Comprehensive Framework for Secure Systems Development. In Dubois, E., Pohl, K., eds.: Proceedings of the 18th International Conference on Advanced Information Systems Engineering (CAiSE'06), Springer-Verlag (2006) 48–62
23. Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Perini, A.: TROPOS: an Agent-oriented Software Development Methodology. Journal of Autonomous Agents and Multi-Agent Systems **8** (2004) 203–236
24. Castro, J., Kolp, M., Mylopoulos, J.: Towards Requirements-Driven Information Systems Engineering: The TROPOS Project. Information Systems **27** (2002) 365–389
25. Mouratidis, H., Giorgini, P., Manson, G.: Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In: Proceedings of the 15th Conference On Advanced Information Systems Engineering (CAiSE'03), Springer-Verlag (2003) 63–78
26. Mouratidis, H., Philp, I., Manson, G.: A novel agent-based system to support the single assessment process of older people. Journal of Health Informatics **9**(3) (2003) 149–162
27. Mouratidis, H.: A Security Oriented Approach in the Development of Multiagent Systems: Applied to the Management of the Health and Social Care Needs of Older People In England. PhD thesis, Department of Computer Science, University of Sheffield, UK (2004)
28. van Lamsweerde, A., Letier, E.: Handling Obstacles in Goal-oriented Requirements Engineering. Transactions on Software Engineering **26**(10) (2000) 978–1005