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

Raimundas Matulevičius[1], Nicolas Mayer[1,2], Haralambos Mouratidis[3], Eric Dubois[2], Patrick Heymans[1], and Nicolas Genon[1]

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

Abstract. Security is a major target for today's information systems (IS) designers. Security modelling languages exist to reason 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. We analyse it and suggest improvements in the light of an existing reference model for IS security risk management. This allows for checking 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.

Keywords: Risk management, information system, security, Secure Tropos.

1 Introduction

Information systems (ISs) undoubtedly play an important role in today's society are more and more 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, such as Basel II [1] or SOX [2]. Risk management is considered as central by IS professionals. The risk management does not only support security officers in the handling of security vulnerabilities but it also provides a framework that allows evaluation of the return on investment of the security solutions against the economic and business consequences of not implementing them. There are more than 200 risk management methods making it a

Z. Bellahsène and M. Léonard (Eds.): CAiSE 2008, LNCS 5074, pp. 541–555, 2008.

challenge to select the most adequate one. In a previous analysis [3] we 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, they are often designed for assessing 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 to 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 new and complete notation does not appear to us as a 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 proven to be a big push for the adoption of modelling practices in public and private companies. At the RE level we plead for a similar approach and rather than to develop a totally new language we improve existing languages, offering an ontological basis sufficiently close to the risk management domain.

With respect to the above objective, we have identified Secure Tropos [4], 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]. This reference model defines the fundamental concepts of ISSRM as gathered from a quantity of standards and other sources, e.g., [5] [6] [7]. The overall approach is illustrated throughout this paper reusing the example of the electronic Single Assessment Process (eSAP) [8].

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

2.1 Security Risk Domain

The ISSRM Reference model [3] presented in Fig. 1 results from a consolidation of existing security standards, e.g., [5], [6], [7]. In this section we summarise some core definitions of ISSRM concepts.

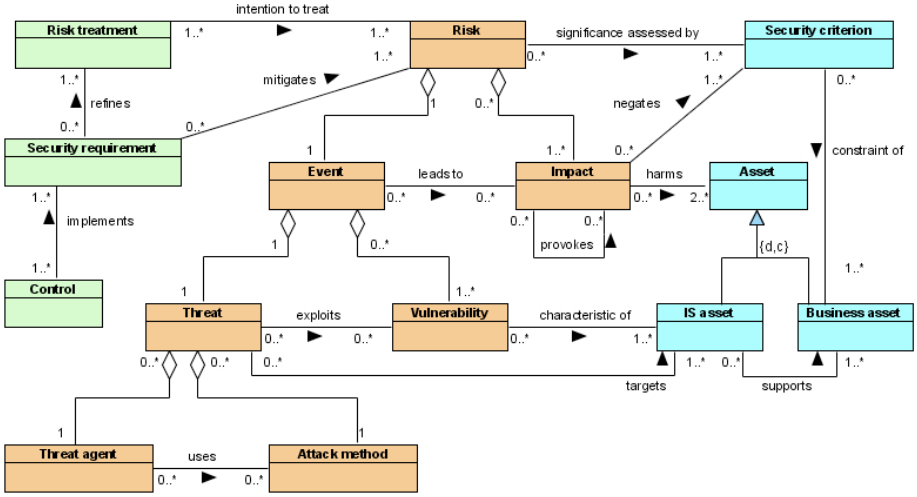

Fig. 1. The ISSRM Reference Model [3] [9]

Asset-related concepts describe assets and the criteria which guarantee asset security [3] [9]. 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, having value for it. An *IS asset* is a component of the IS supporting business assets like a database where information is stored. A *security criterion* characterises a property or constraint on business assets describing their security needs, usually for confidentiality, integrity and availability.

Risk-related concepts present how the risk itself is defined [3] [9]. 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. An *event* 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 and may lead to the assets being harmed. 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 [3] [9]. A *risk treatment* is an intentional decision to treat identified risks. A *security requirement* is the refinement of a treatment decision to mitigate the risk. *Controls* (countermeasures or safeguards) are designed to improve security, specified by a security requirement, and implemented to comply with it.

Like the Tropos Goal-risk framework [10], the ISSRM reference model addresses risk management at three different levels, combining asset, risk, and risk treatment views. However the ISSRM reference model focuses on the *IS* security while the Tropos Goal-risk framework supports risk in general.

Security risk management process. The ISSRM activities follow the general risk management process [3] [9]. This process originates from the risk management standards (e.g., [5], [6], [7]) and consists of six steps. It 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* (confidentiality, integrity and/or 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. The risk management process is *iterative*. Each step can be repeated to obtain an outcome of higher quality. Furthermore, 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 phases, security can be addressed using various modelling languages. Abuse frames [11] suggest 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 the attacker. SecureUML [15] and UMLsec [16] consider security during system design.

Goal modelling languages have also been adapted to security. Secure *i** [17] addresses security trade-offs. KAOS [18] was augmented with *anti-goal models* designed to elicit attackers' rationales. In [19] [20] Tropos has been extended with the notions of *ownership*, *permision* and *trust*. Here we investigate Secure Tropos [4] that models security using *security constraints* and *attack methods*.

All these languages are candidates for supporting largely or partially the SRM activities. In this paper we specifically target security risk management in the *early* IS development. Thus, we have chosen Secure Tropos, which incrementally introduces security concerns from the requirements phases. However, the final analysis of the security concerns takes place only during the design phases [21]. Therefore by aligning Secure Tropos with the ISSRM reference model, we suggest improvements needed for the SRM in the early (requirements) IS phases.

2.3 Secure Tropos

Secure Tropos enriches a set of Tropos [22] [23] constructs (*actor*, *goal*, *softgoal*, *plan*, *resource*, 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 organisational setting [22]. A *hardgoal* or simply *goal* hereafter (see Fig. 3), represents an actor's 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 [22] [23] and Secure Tropos [8] [21] [24]. In addition Secure TROPOS introduces *security constraints* and *threats*. A *security constraint* represents a restriction related to security that the system must have and actors must respect (see Fig. 3) [4] [24]. 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" [4].

Constructs are combined using relationships: *dependency*, *decomposition*, *means-ends*, *contribution*, *restricts* and *attacks*. 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 [22]. *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)" [24]. The *goal model* allows a deeper understanding of how the actors reason about goals to be fulfilled, plans to be performed and available resources [23]. 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 an *end* (goal). The *decomposition* relationship (see Fig. 4) permits to define a finer structure of a plan. 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

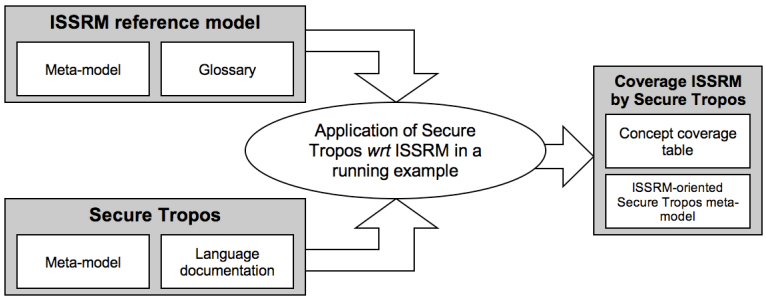

Fig. 2. Research Method

Tropos language as it is derived from the Secure Tropos meta-model and the description of the language in the literature [4] [8] [21] [24] [25].

In this paper we use a running example to explain our analysis of the alignment of Secure Tropos and the ISSRM. The running example is initially used to illustrate the use of the language. We then consider the concepts of Secure Tropos *wrt* how they were used to address ISSRM. The outcome of this comparison is the concept alignment between Secure Tropos 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 [26][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 use it to analyse the electronic Single Assessment Process (eSAP) [27]. The eSAP is an IS to support integrated assessment of the health and social care needs of elderly. It is based on the Single Assessment Process, which is part of the National Service Framework for Older People Services of the English Department of Health. The eSAP is suitable to demonstrate our work for two main reasons: (*i*) security and risk are two important factors in its development and implementation; (*ii*) the security of the system have been successfully analysed using the Secure Tropos methodology [28]. Therefore, by revisiting the running example, we are able to identify the exact contributions of this paper. Due to space limitations, we focus on one of the most important aspects to make the eSAP: the Patient personal information.

(a) Context and asset identification. A Social Worker is in charge of the health care to patients. In order to fulfill her work, she needs the Patient personal information. In Fig. 3 the Social Worker depends on a goal Collected care

[1] Due to space requirements we did not include the Secure Tropos meta-model nor the ISSRM-oriented Secure Tropos meta-model.

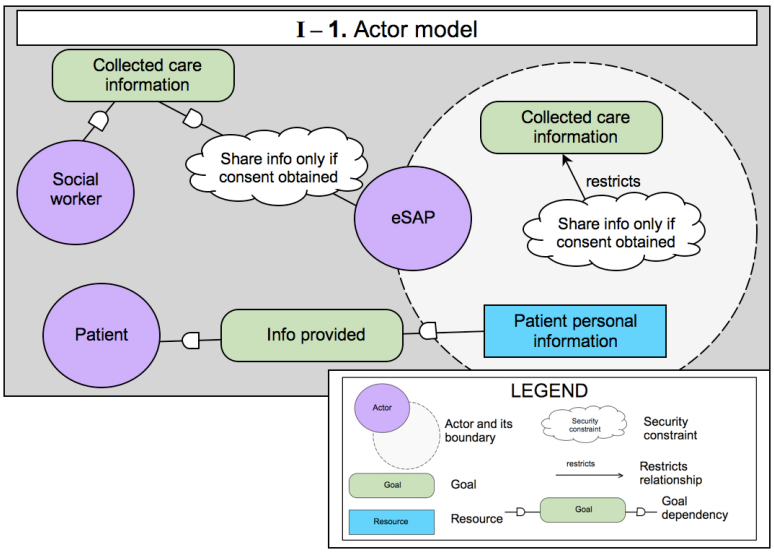

Fig. 3. Actor model

information held by the eSAP system. As the information is a valuable business asset, achievement of the goal Collected care information is restricted by a security constraint 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, see Fig. 4.

(*b*) *Security objective determination.* The plan Check data for consent contributes positively to the security constraint Share info only if consent obtained (Fig. 5). This plan also realises 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 fulfill the goal System privacy ensured and contributes positively to the security constraint Keep system data privacy.

(*c*) *Risk analysis and assessment.* Fig. 6 focuses on a possible risk event. We identify an Authentication attack (modelled using the *threat* construct). It describes a situation where a threat agent fakes his identity to pass himself off as a trusted actor in order to damage the business assets (e.g., Patient personal information). The Authentication attack has a negative impact on Privacy. On the other hand the constraint Keep system data privacy mitigates the possible risk difficult to realise. Note that the Authentication attack does not depend on the existence of an actor whose assets are threatened.

In Fig. 7 we present the view of an Attacker whose aim is to get the Patient personal information. The Attacker poses a threat (the goal Info about patient received and plan Collect info about breaking the system in Fig. 7). The plan is decomposed into two parts: (*i*) the attacker has to get the consent for the

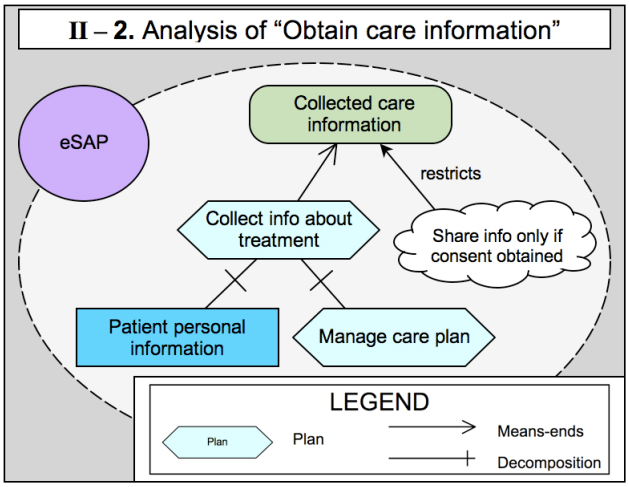

Fig. 4. Analysis of "Obtain care information"

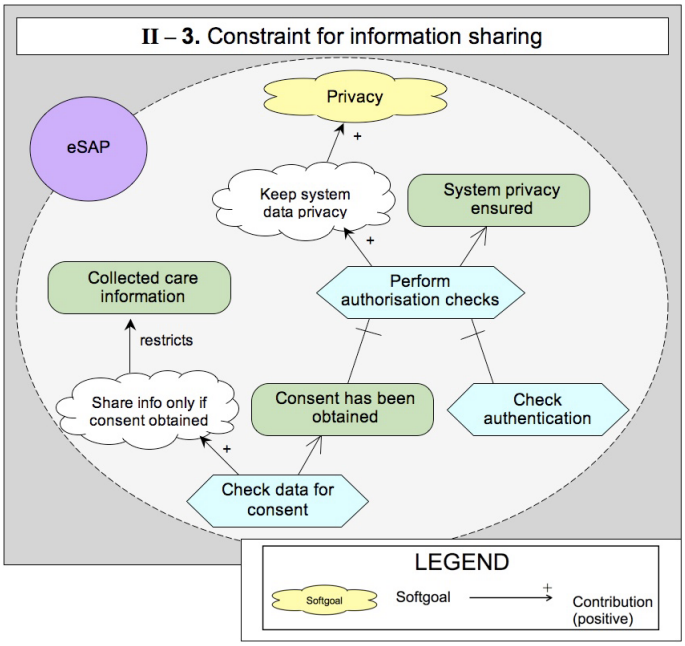

Fig. 5. Constraint for information sharing

Patient personal information; and (*ii*) he needs to find the authentication code for the system. To get the consent, the attacker can Steal data from a social worker or Buy data from the untrusted social worker. Here, belief Possible to check eSAP access repeatedly corresponds to a vulnerability, known by the attacker.

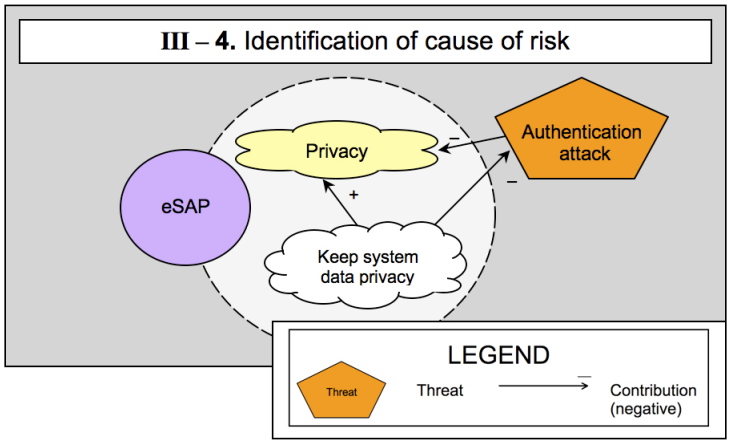

Fig. 6. Identification of an authentication risk

The vulnerability contributes positively to the decomposition between two plans Collect info about breaking the system and Check eSAP access repeatedly. Fig. 7 can be seen as the refinement of the cause of the risk identified in Fig. 6.

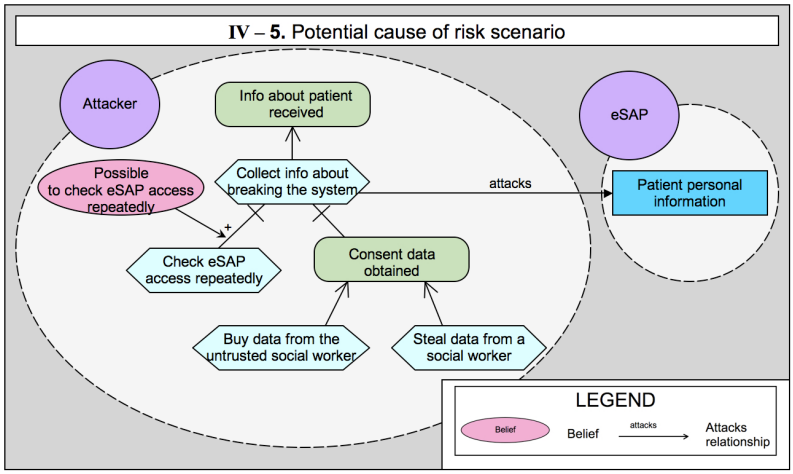

Fig. 7. Potential attack scenario

(*d*) *Risk treatment.* Several risk treatment decisions are suggested in [29]. 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 a different system design but allows avoiding the Authentication attack.

(*e*) *Security requirements definition.* The 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 realise the countermeasure, Encrypt data and Decrypt data are performed at a certain time. Our countermeasure avoids the Authentication attack because now the eSAP system is designed so that it does not require the authentication information. However this might result in other events of the risk (e.g., Cryptographic attack) which need to be analysed as well.

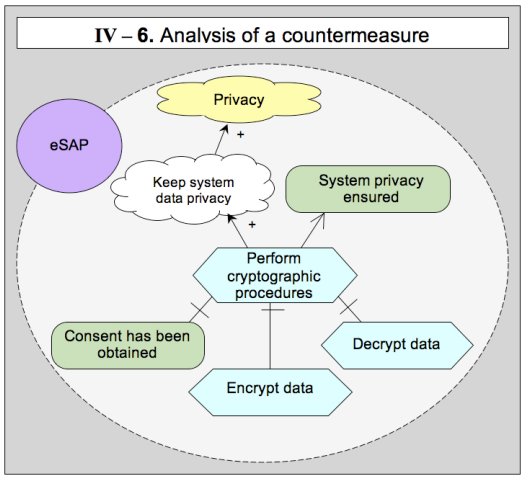

Fig. 8. Analysis of a countermeasure

(f) Control selection and implementation. Softgoals can be used to reason on the differences between control alternatives. This step takes place after controls are defined, that usually happens during the design phase.

4 Contribution

The contribution of the analysis is with the semantic alignment between ISSRM and Secure Tropos. We illustrate how we can use the Secure Tropos approach to analyse possible attack scenarios and how derive countermeasures from attack scenarios. 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 [4] [8] [21] [24] [25]. The fourth column lists the Secure TROPOS constructs used to address the ISSRM concepts. The last column illustrates the Secure TROPOS concepts 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 the *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

The ISSRM model		Secure TROPOS		
		Synonyms 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]
	Event	—	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, Relationship 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**
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	—	New model which implements security requirements	Cryptographic module in the eSAP system

Fig. 9. Alignment between the ISSRM reference model and Secure Tropos. * – literature includes [8] [4] [21] [25]; ** – look for discussion about belief in Section 4.

hand the *actors* Patient and Social worker (see Fig. 3), the *goals* Obtain care information and Info provided and the *plans* Collect info about treatment and Manage care plan (see Fig. 4) describe the process necessary for the organisation (health care centre) to achieve its objectives. On the other hand the *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 is the eSAP. In more details (see Fig. 5) the support for the *business assets* is described by the *goals* System privacy ensured and Consent has been obtained and the *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 by using the *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, and what major principles should be taken into account when defining the possible risks [3]. Risk is described by the event 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 compromises the *business asset* 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 the possibility to check the eSAP access repeatedly as identified by the *belief* in Fig. 7. To break into the eSAP system the Attacker carries an *attack method* consisting of the 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 fact that the *actor* (who has the role of the *attacker*) thinks he knows, might be true. In this case the *belief* will correspond to *vulnerability* in the sense of the ISSRM. 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 there is no corresponding concept in the 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] [29] in our example we use *goal/plan substitution* which leads to a different eSAP design 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 note that, depending on the selected *risk treatment decision*, the combination of *actor*, *goal*, *resource* and *plan* might result in 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. Based on an illustrative example, we showed how a Secure Tropos model can be created following the security risk management process. Our purpose was not to develop the example in detail (for instance we do not detail how the 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 IS development. This means that the analysis of Secure Tropos is not complete *wrt* the late development, 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 Secure Tropos meta-model, clarified unclear use of language constructs; (*ii*) collectively agreed on 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 in the context of security risk management activities:

- Secure Tropos has to provide guidelines as to when and how to use each constructs in order to avoid misinterpretations of the ISSRM concepts. One improvement could be inclusion of tags in the label of a construct. For example, the *plan* construct 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 [SR]– *security requirements*. In our running example we deal with this limitation by decomposing the model into separate diagrams: we use the *plan* construct to represent *business assets* in Fig. 4, *IS assets* in Fig. 5, *threats* 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 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, the *belief* construct 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 a relationship between *vulnerable points* and *belief* is possible.

Note that the *research method* used for alignment between language constructs and the ISSRM reference model can be used to evaluate of any security modelling language. In addition to Secure Tropos we also investigated *KAOS extended to security* [26] and *misuse cases* [9]. We envision that after analysing a number of security languages it will be possible to facilitate model transformation and language interoperability. This would allow representing ISs using different perspectives, also ensuring IS sustainability.

Acknowledgment. This work is partially funded by the Interuniversity Attraction Poles Programme, Belgian State, Belgian Science Policy. We also thank A. Classen for proofread of the paper.

References

1. Basel Committee on Banking Supervision: International Convergence of Capital Measurement and Capital Standards. Bank for International Settlements (2004)
2. United States Senate and House of Representatives in Congress: 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), pp. 121–131 (2007)
4. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering (IJSEKE) 17(2), 285–309 (2007)
5. DCSSL: EBIOS–Expression of Needs and Identification of Security Objectives (2004)
6. ENISA: Inventory of Risk Assessment and Risk Management Methods (2004)
7. ISO: Information Technology–Security Techniques–Information Security Management Systems–Requirements, International Organisation for Standardisation (2005)
8. 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 2002) (2002)
9. Matulevičius, R., Mayer, N., Heymans, P.: Alignment of Misuse Cases with Security Risk Management. In: Proceedings of the ARES 2008 Symposium on Requirements Engineering for Information Security (SREIS 2008), pp. 1397–1404. IEEE Computer Society, Los Alamitos (2008)
10. Asnar, Y., Giorgini, P.: Modelling Risk and Identifying Cuntermeasure in Organizations. In: Proceedings of the 1st Interational Workshop on Critical Information Intrastructures Security, pp. 55–66. Springer, Heidelberg (2006)
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 2004), pp. 354–355. IEEE Computer Society, Los Alamitos (2004)
12. McDermott, J., Fox, C.: Using Abuse Case Models for Security Requirements Analysis. In: Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC 1999), p. 55 (1999)
13. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirements Engineering Journal 10(1), 34–44 (2005)
14. Sindre, G.: Mal-activity Diagrams for Capturing Attacks on Business Processes. In: Sawyer, P., Paech, B., Heymans, P. (eds.) REFSQ 2007. LNCS, vol. 4542, pp. 355–366. Springer, Heidelberg (2007)
15. Lodderstedt, T., Basin, D.A., Doser, J.: SecureUML: A UML-based Modeling Language for Model-driven Security. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 426–441. Springer, Heidelberg (2002)
16. Jurjens, J.: UMLsec: Extending UML for Secure Systems Development. In: Jézéquel, J.-M., Hussmann, H., Cook, S. (eds.) UML 2002. LNCS, vol. 2460, pp. 412–425. Springer, Heidelberg (2002)
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.) ER 2007. LNCS, vol. 4801, pp. 87–101. Springer, Heidelberg (2007)

18. van Lamsweerde, A.: Elaborating Security Requirements by Construction of Intentional Anti-models. In: Proceedings of the 26th International Conference on Software Engineering (ICSE 2004), pp. 148–157. IEEE Computer Society, Los Alamitos (2004)
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 2005). IEEE Computer Society, Los Alamitos (2005)
20. Giorgini, P., Massacci, F., Mylopoulos, J., Zannone, N.: Modelling social and individual trust in requirements engineering methodologies. In: Proceedings of the 3nd International Conference on Trust Management. LNCS, pp. 161–176. Springer, Heidelberg (2005)
21. Mouratidis, H., Jurjens, J., Fox, J.: Towards a Comprehensive Framework for Secure Systems Development. In: Dubois, E., Pohl, K. (eds.) CAiSE 2006. LNCS, vol. 4001, pp. 48–62. Springer, Heidelberg (2006)
22. 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, 203–236 (2004)
23. Castro, J., Kolp, M., Mylopoulos, J.: Towards Requirements-Driven Information Systems Engineering: The TROPOS Project. Information Systems 27, 365–389 (2002)
24. Mouratidis, H., Giorgini, P., Manson, G.A.: When Security Meets Software Engineering: a Case of Modelling Secure Information Systems. Information Systems 30(8), 609–629 (2005)
25. Mouratidis, H., Giorgini, P., Manson, G.: Integrating Security and Systems Engineering: Towards the Modelling of Secure Information Systems. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, pp. 63–78. Springer, Heidelberg (2003)
26. Genon, N.: Modelling Security during Early Requirements: Contributions to and Usage of a Domain Model for Information System Security Risk Management. Master thesis, University of Namur (2007)
27. 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), 149–162 (2003)
28. 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)
29. van Lamsweerde, A., Letier, E.: Handling Obstacles in Goal-oriented Requirements Engineering. Transactions on Software Engineering 26(10), 978–1005 (2000)