STF40 A03067 – Unrestricted

REPORT

A UML profile for the identification and analysis of security risks during structured brainstorming

Mass Soldal Lund, Folker den Braber, Ketil Stølen, Fredrik Vraalsen

SINTEF ICT

Cooperative and Trusted Systems

May 2004

SINTEF

SINTEF ICT

Address: NO-7465 Trondheim
NORWAY
Location Trondheim:
S.P. Andersens v 15
Location Oslo:
Forskningsveien 1
Telephone: +47 73 59 30 00
Fax: +47 73 59 43 02

Enterprise No.: NO 948 007 029 MVA

SINTEF REPORT

TITLE

A UML profile for the identification and analysis of security risks during structured brainstorming

AUTHOR(S)

Mass Soldal Lund, Folker den Braber, Ketil Stølen, Fredrik Vraalsen

CLIENT(S)

Research council of Norway

REPORT NO.	CLASSIFICATION	CLIENTS REF.	
STF40 A03067	Unrestricted	152839/220	
CLASS. THIS PAGE	ISBN	PROJECT NO.	NO. OF PAGES/APPENDICES
Unrestricted	82-14-03110-9	40332800	17/0
ELECTRONIC FILE CODE		PROJECT MANAGER (NAME, SIGN.)	CHECKED BY (NAME, SIGN.)
040504.uml-sa-report2.doc		Ketil Stølen	Ida Solheim
FILE CODE	DATE	APPROVED BY (NAME, POSITION, SIGN.)	
	2004-05-04	Bjørn Skjellaug, Research director	

ABSTRACT

Methods for identification and analysis of security risks make use of structured brainstorming sessions. The effectiveness of such sessions depends on the extent to which the stakeholders and analysts involved understand and are understood by each other. Since such sessions involve people with different backgrounds and competencies, like users, system-developers, decision makers and system managers, communication among them may be difficult. This report proposes a carefully designed specification language defined as a UML profile aiming to improve communication and understanding during such sessions. We claim that the profile (1) allows the target of evaluation to be described in a uniform manner at a suitable level of abstraction, (2) improves understanding and communication during structured brainstorming sessions concerned with security, (3) facilitates the documentation of results from such brainstorming sessions, and security assessments in general.

KEYWORDS	ENGLISH	NORWEGIAN
GROUP 1	ICT	IKT
GROUP 2	Risk management	Risikoadministrasjon
SELECTED BY AUTHOR	Modelling	Modellering
	UML	UML
	Security assessment	Sikkerhetsanalyse

A UML Profile for the Identification and Analysis of Security Risks during Structured Brainstorming

Mass Soldal Lund[1,2], Folker den Braber[1], Ketil Stølen[1,2], Fredrik Vraalsen[1]

[1] SINTEF Information and Communication Technology, Oslo, Norway
{msl, fbr, kst, fvr}@sintef.no
[2] Department of Informatics, University of Oslo, Norway

Abstract. Methods for identification and analysis of security risks make use of structured brainstorming sessions. The effectiveness of such sessions depends on the extent to which the stakeholders and analysts involved understand and are understood by each other. Since such sessions involve people with different backgrounds and competencies, like users, system-developers, decision makers and system managers, communication among them may be difficult. This report proposes a carefully designed specification language defined as a UML profile aiming to improve communication and understanding during such sessions. We claim that the profile (1) allows the target of evaluation to be described in a uniform manner at a suitable level of abstraction, (2) improves understanding and communication during structured brainstorming sessions concerned with security, (3) facilitates the documentation of results from such brainstorming sessions, and security assessments in general.

1 Introduction

In this report we present a UML profile intending to support model-based security assessment of IT systems. The profile introduces a meta-model that defines an abstract language supporting model-based security assessment. Furthermore, the profile provides a mapping of classes in the meta-model to UML modelling elements by defining so-called stereotypes, and introduces special symbols (icons) for representing the stereotypes in UML diagrams. The motivation for the profile is to facilitate the practical use of UML to support security management in general, and security assessment in particular.

The background of the profile is the EU-project CORAS (IST-2000-25031).[1] CORAS has developed a framework for model-based security assessment. This framework includes:

- A methodology for model-based security assessment integrating aspects from partly complementary risk analysis methods and state-of-the-art modelling methodology.

[1] For more information on the CORAS project we refer to the project homepage: http://coras.sourceforge.net/

- A UML-based specification language targeting security assessment, namely the profile presented in this report.
- A library of reusable security assessment experiences.
- A computerised platform providing two repositories; one for assessment documentation and one for reusable experiences.
- An XML mark-up for exchange of security assessment data.
- A component for computerised vulnerability management.

In the model-based security assessment methodology of CORAS, UML models are used for three purposes:

1. To describe the target of evaluation at the right level of abstraction.
2. To facilitate communication and interaction between different groups of stakeholders involved in a security assessment.
3. To document security assessment results and the assumptions on which these results depend to support reuse and maintenance.

The UML profile supports all these objectives, but has a special emphasis on communication and documentation. Documentation is supported because the meta-model of the profile is consistent with a data structure of security assessment documentation developed as part of the CORAS project. Communication is supported by the definition of easy-to-understand icons (symbols) associated with the modelling elements of the profile, and because these specialised modelling elements are consistent with the ontology of security assessment.

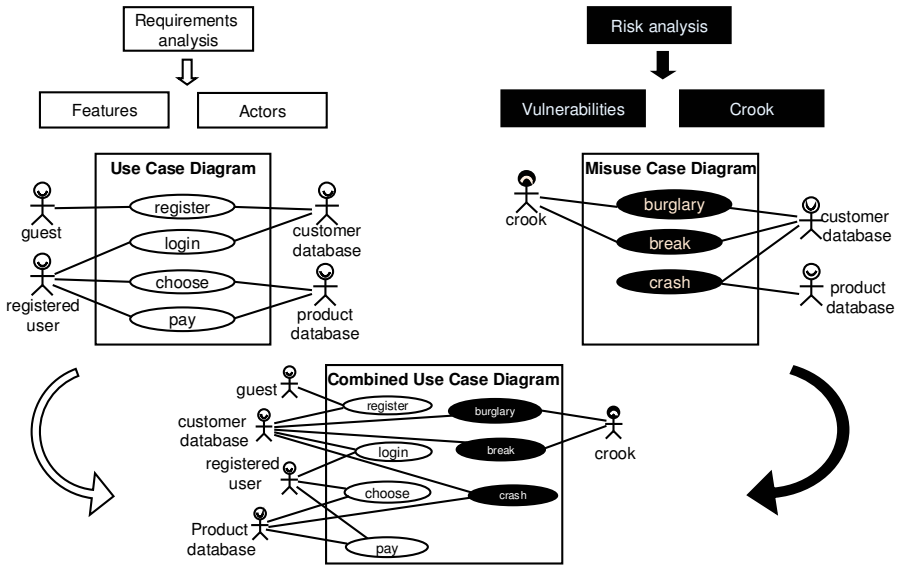

Fig. 1. Requirement capture versus threat identification

In system development, particularly in requirements analysis, the focus is on capturing the *desired* functionality or features. In risk analysis, on the other hand, the

focus is on discovering the *undesired* functionality or vulnerabilities. One of the main ideas of the profile is based on the observation that the same modelling techniques can be used for risk analysis as for system analysis or development.

Fig. 1 highlights how threat identification mirrors requirements capture. Requirements capture resembles risk identification in that both activities require different groups of stakeholders, e.g., users, decision makers, system developers, system managers, etc., to communicate and reach consensus. Graphical specification techniques like use-case and sequence diagrams have proved well suited to support this interaction in the case of requirements capture. The same kind of notations may as well be applied to facilitate threat identification (e.g., misuse-case diagrams [1,14,15]).

The combination of these two views provides us with the "good" as well as the "bad" sides of the system under design or assessment and constitute what we may characterise as a "complete overview" of a security critical system.

The remainder of this report is structured as follows: Section 2 provides the requirements identified for the profile, and Section 3 an overview of the profile. In Section 4 an example using the profile is given, based on a telemedicine trial carried out in the CORAS project, and Section 5 provides conclusions.

2 Requirements to the language

In development of the CORAS methodology, it early became evident that there was a need for a modelling language specialised towards documenting and communicating the results from the security assessments. Methods for identification and analysis of security risks make use of structured brainstorming sessions. The effectiveness of such sessions depends on the extent to which the stakeholders and analysts involved understand and are understood by each other. Since such sessions involve people with different backgrounds and competencies – like users, system-developers, decision makers and system managers – common understanding among the stakeholders is often not the case. In order to facilitate such a common understanding, a common language is needed. For this language the following requirements have been identified:

1. *The language should be graphical.*
   The main benefit of graphical languages is that most people seem to find them easy to comprehend, even when they have a formally defined syntax.
2. *The language should be defined as a UML 2.0 profile.*
   UML is the most widely used specification language in the software industry today, and therefore a natural choice of basis for a language targeting security assessment of IT systems; the use of UML builds a bridge between security assessment and system development. The well-definedness of UML is also of great value, since it supports structured and uniform documentation, as well as support for automated consistency checking and analysis. With UML 2.0 [12] as the new standard, the older version of UML now in use (versions 1.3, 1.4 and 1.5) will be outdated, and therefore not suited as a basis for the language.

3. *The language should reflect standard notation and terminology for security assessment.*
   In a generic framework for evaluating language quality presented in [7], this is related to what is called *domain appropriateness*; the conceptual basis should be able to express all that is in the domain, and should not be able to express anything outside of the domain.
4. *The language should allow the target of evaluation to be described in a uniform manner at a suitable level of abstraction.*
   Security assessments may be carried out at any level of abstraction, from high level assessments at enterprise level to low level assessments of technical implementations. For an assessment to be efficient it is important that the participants are able to stay focused on the desired abstraction level, and models describing the target of evaluation should be guiding the participants in this respect.
5. *The language should facilitate understanding and communication during brainstorming sessions concerned with security risks.*
   Standard risk analysis techniques, like for example Hazard and Operability analysis (HazOp) [13], are based on structured brainstorming, and depend on the participants being able to communicate. For a modelling language to be useful in such settings, it needs to support (and certainly not obscure) the communication between participants. Referring to [7], this is related to *comprehensibility appropriateness*; symbols representing different concepts should be distinguishable, a symbol should represent the same concept in all contexts, etc.
6. *The language should facilitate the documentation of security assessments and maintenance of security assessment documentation.*
   To support documentation, the language must be able to express the output of security assessments. Moreover, security assessments are costly and time consuming and cannot be carried out from scratch each time a system is updated or modified, so support for maintaining security assessment documentation is an important requirement. As explained in [8] security assessment documentation has strong internal dependencies that must be accounted for when doing maintenance, and these dependencies must be reflected in the language.

3 Overview of the profile

The UML profile for security assessment introduces a meta-model that defines an abstract language for supporting model-based security assessment. Further, the profile provides a mapping of classes in the meta-model to UML modelling elements by defining so-called stereotypes, and introduces special symbols (icons) for representing the stereotypes in UML diagrams.

The basis of the CORAS model-based security assessment methodology is the generic risk management process of [2]. This process is divided into five sub-processes (Fig. 2):

- *Sub-process 1: Identify context*. This first sub-process is concerned with identifying and specifying the system under assessment, the target of evaluation,

the stakeholders and assets of the system and the assumptions on which the assessment will be based.

- *Sub-process 2: Identify risks*. The target of this sub-process is to identify the threats to and vulnerabilities of the identified assets, and to identify unwanted incidents that may result from these threats and vulnerabilities.
- *Sub-process 3: Analyse risks*. In this sub-process the risks are analysed. A risk is an unwanted incident that has been quantified with a consequence value representing a loss of asset value and a frequency.
- *Sub-process 4: Evaluate risks.* Based on its frequency and consequence a risk is assigned a risk value. In this sub-process, the values of risks are compared against risk evaluation criteria identified as a part of the assessment context in order to decide which risks are acceptable and which risks need to be treated.
- *Sub-process 5: Treat risks.* The last of the sub-processes is concerned with identifying and evaluating treatments to the risks that in the previous sub-process were judged to be unacceptable.

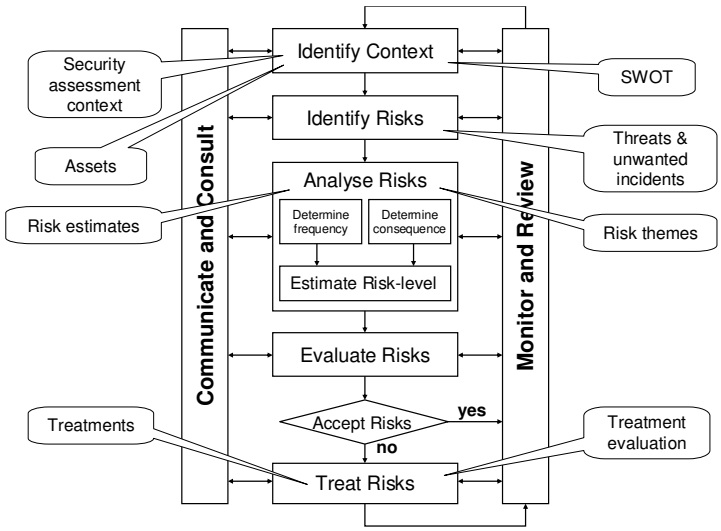

Fig. 2. The profile's relation to the risk management process

In the CORAS methodology, this general risk management process has been specialised towards targeting security critical IT-systems, providing detailed guidelines for the assessments. For a detailed description of this specialised risk management process we refer to [3].

The meta-model is divided into six sub-models that support different stages of the risk management process: the context sub-model (sub-process 1), the SWOT sub-model (sub-process 1), the unwanted incident model (sub-process 2), the threat agent model (sub-process 2), the risk model (sub-process 3), and the treatment model (sub-process 5).

A security assessment always starts with identifying the context of the assessment. A strengths, weaknesses, opportunities and threats (SWOT) analysis may be part of this. After the context has been established, the rest of a security assessment can be divided into the identification and documentation of unwanted incidents, risks and treatments.

The unwanted incident model is concerned with organising and documenting the threats and vulnerabilities opening for incidents that may harm the system. The risk model quantifies unwanted incidents with respect to their potential reductions of asset value. The treatment model supports documenting ways of treating the system and quantifying the effect of treatments with respect to reducing the potential harm of risks. In addition there is a threat agent sub-model associated with the unwanted incident model. This model defines a number of well-known threat agents from the security literature.). The full meta-model and definition of the profile is given in [10].

As illustrated in Fig. 2, the profile provides support for the risk management process by providing modelling support for:

- security assessment context,
- strengths, weaknesses, opportunities and threats (SWOT) analyses,
- assets,
- threats and unwanted incidents,
- risk estimates,
- risk themes,
- treatments, and
- treatment evaluation.

Even though the UML profile for security assessment is developed within the CORAS project, it is based on the concepts and terminology and the general process of [2], and should be sufficient general itself to be applied together with other security assessment methodologies based on the same generic process.

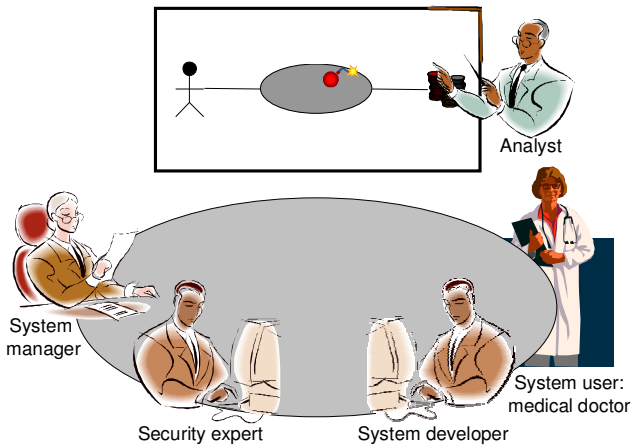

Fig. 3. Typical security assessment setting

4 Example

In the following we present an example of how the profile supports modelling in a security assessment. The example is based on a real field trial conducted within the CORAS project. The trial consisted of a full assessment of a telemedicine system for cardiological examinations. The system uses a dedicated network for transmitting medical data, allowing a general practitioner and a cardiology expert to examine a patient together even though they are physically located at different sites.

The examples focus on two groups of stakeholders of the system; the patients and the primary health care centres (PHCCs) that use the system to get medical advices from medical experts.

Fig. 3 shows the typical use of UML in a security assessment. The analyst is leading a structured brain-storming session, and is presenting diagrams for the rest of the participants. The participants need not be familiar with UML, to them the UML diagrams may as well be presented as merely illustrations of what they are discussing. The analyst, however, will after the session systemise the outcome of the brainstorming, and for him/her the well-definedness of UML supports the task of achieving consistency of the assessment documentation.

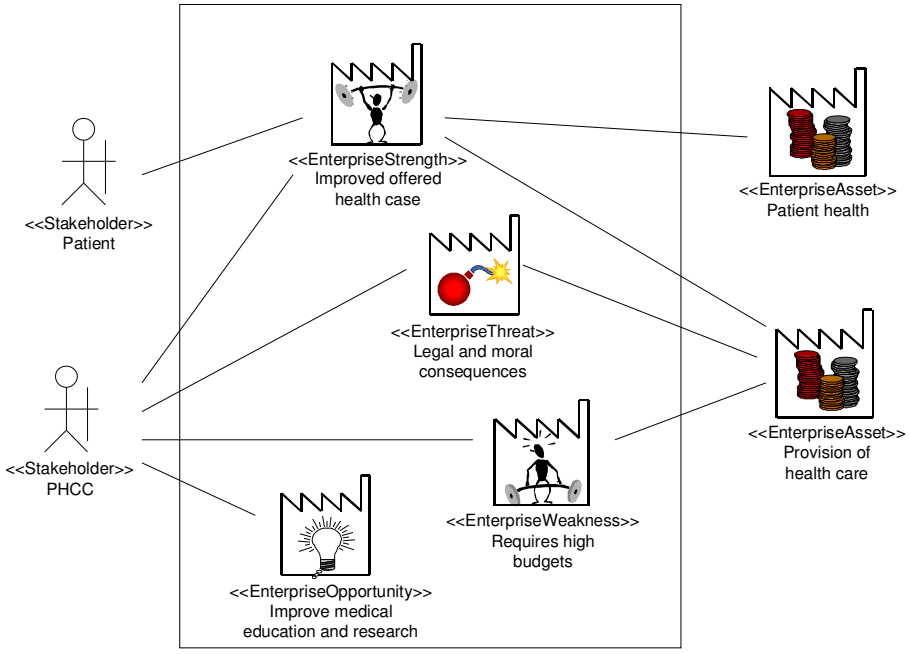

Fig. 4. SWOT diagram

4.1 SWOT (sub-process 1)

A strengths, weaknesses, opportunities and threats (SWOT) analysis is carried out as a part of establishing the context of a security assessment, as they are concerned with identifying the strategic context of the organizations carrying out the security assessment. Thus, SWOT is used for pointing out the general direction of the assessment, and its results are only indirectly used in the further assessment.

Fig. 4 shows how the results of the SWOT analysis of the telecardiology service modelled using the profile. It is worth noticing that a SWOT is usually carried out as part of the context identification, and before the asset identification.

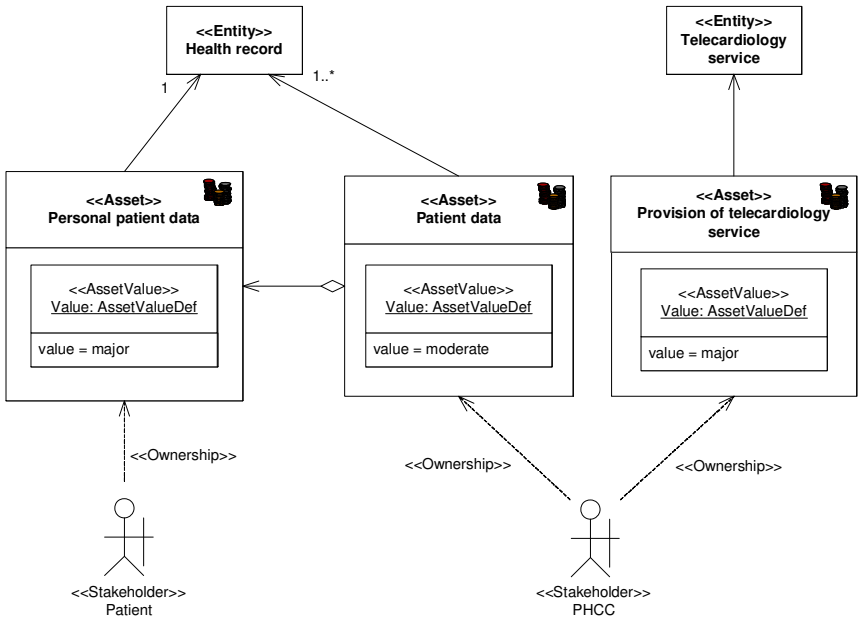

Fig. 5. Modelling of assets

4.2 Security assessment context (sub-process 2)

The most important part of the security assessment context consists of the stakeholders and assets of the system under assessment, on which all further assessment is based. A stakeholder is a person or organisation that has interests in the assessed system, and an asset is part or feature of the system that has value for one of the stakeholders. Security assessments are *asset-driven*, which means that assessments are carried out relative to the identified assets.

Each asset may only be related to one stakeholder and should have an unambiguous value assigned by one stakeholder. If two stakeholders view the same entity as an asset, this should be documented as two different assets related to the

same entity. Two assets are per definition different if valued by different stakeholders; both the values and the reasons for the valuing may be different.

Fig. 5 shows the result of the asset identification. We see how stakeholders are related to assets by the <<Ownership>> stereotype, and how assets are related to assets.

4.3 Identification of threats (sub-process 2)

Threat identification is concerned with exploring the threats to the system under assessment. In Fig. 6, modelling of threats is exemplified. A threat is represented by a threat agent and a threat scenario. A threat agent is a potential cause of an unwanted incident, which may result in harm to a system or an organisation and its assets. Threat agents can be external, (e.g., hackers or viruses) or internal (e.g., system failures or disloyal employees). And a threat scenario is a behavioural description of a threat.

In the example, an eavesdropper with illegal access to data over the network constitutes on of the threats to the asset Personal patient data. The stereotypes <<Eavesdropper>>, <<Insider>>, <<Intruder>> and <<SystemThreat>> in this example are specified in the threat agent submodel as specialisations of <<ThreatAgent>>.

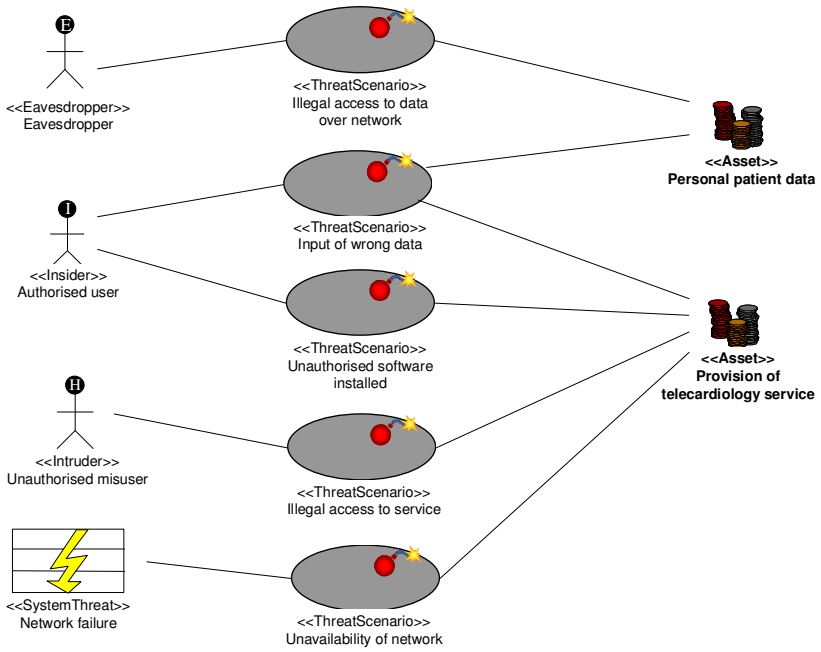

Fig. 6. Modelling of threats

4.4 Identification of vulnerabilities and unwanted incidents (sub-process 2)

In Fig. 7, a part of the model of Fig. 6 is extended with vulnerabilities and unwanted incidents. A vulnerability is a weakness with respect to an asset or group of assets that can be exploited by one or more threats, and an unwanted incident an undesired event that may reduce the value of an asset.

As we see in the example, vulnerabilities are expressed as (undesired) features of assets. The relation between threats and unwanted incidents are expressed by the predefined UML stereotype <<include>>. An unwanted incident may in other words include one or more threat scenarios that model the behavioural aspects of threats. There is also the possibility of an unwanted incident leading to other unwanted incidents. As in the example, this is shown by use of the stereotype <<Initiate>>.

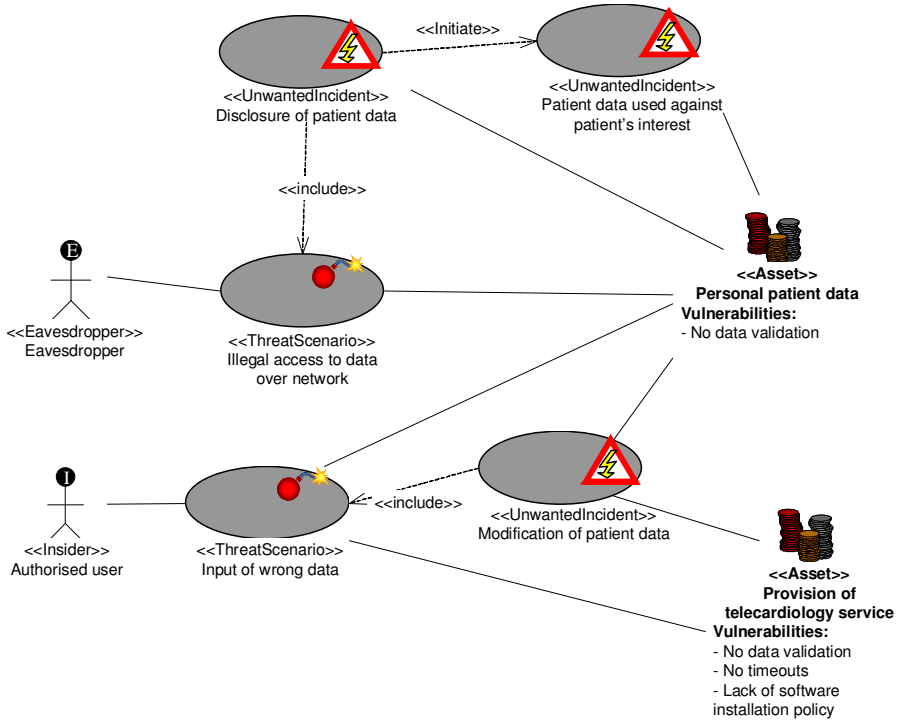

Fig. 7. Modelling of vulnerabilities and unwanted incidents

4.5 Documenting and evaluating risks (sub-process 3)

A risk is an assignment of consequence value and frequency to an unwanted incident on a particular asset. These values are used for calculating a risk value, which

represents loss of value of the related asset. Risks that in some way are related or similar may be categorised into risk themes. A risk theme is itself assigned a risk value based on the risks it contains and is treated like a singular risk with respect to evaluation and treatment.

Fig. 8 illustrates how risks are modelled with the profile. The values are instances of the corresponding value definitions. A risk is related to an unwanted incident by the use of the stereotype <<RiskEvaluation>>. The diagram also shows which asset each risk is related to, and illustrates how an unwanted incident may be evaluated as different risks when related to different assets.

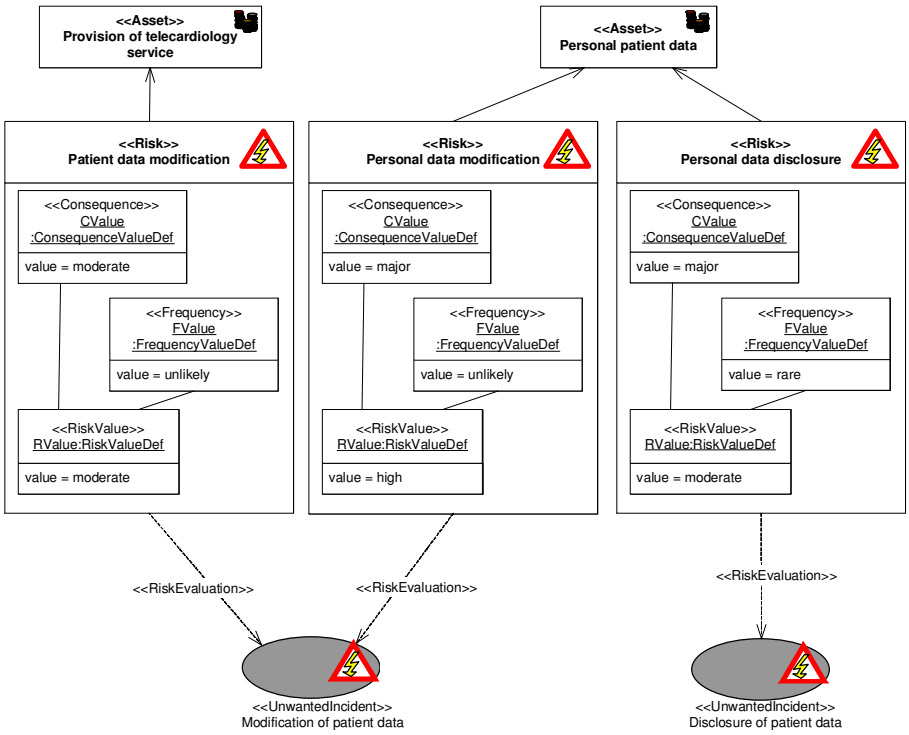

Fig. 8. Risk estimates

In sub-process 4, risks are evaluated by risk evaluation criteria defined in the context of the security assessment. A risk evaluation criterion states which risk values are acceptable, and which are not – implying the need for treatment.

4.6 Identification and evaluation of treatments (sub-process 5)

Sub-process 5 is concerned with identifying and evaluating ways of providing treatments to the system under assessment in order to reduce the value of risks. A treatment may apply to several unwanted incidents. However, when a treatment's capability to reduce risk value is assessed, this is with respect to a single risk or risk theme.

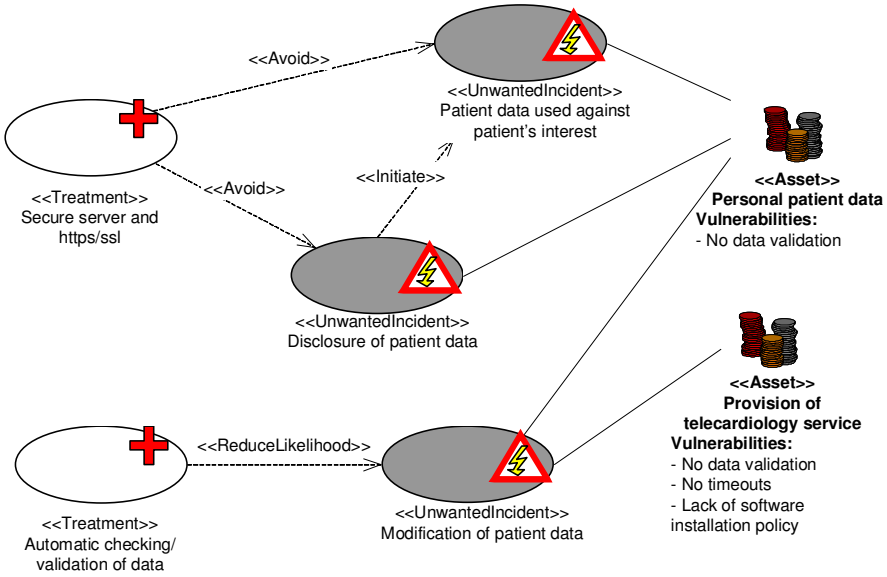

Fig. 9. Modelling of treatments

In Fig. 9 treatments to the unwanted incidents of Fig. 7 is included in the diagram Treatments are related to unwanted incidents by one of the stereotypes <<Avoid>>, <<Transfer>>, <<ReduceLikelihood>> or <<ReduceConsequence>>, representing the main treatment options or main classes of approaches of applying treatment.

In the same manner as a risk is an evaluation of the impact of an unwanted incident on an asset, a treatment effect is the evaluation of the impact of a treatment on a risk. As shown in Fig. 10, treatment effects are modelled in a similar fashion as risks. The treatment effect is bound to a treatment by the means of the stereotype <<TreatmentEvaluation>> and assigned a risk reduction. Risk reduction is the "value" of a treatment with respect to a risk, in this example showing that implementing the treatment will reduce the value of the risk from "moderate risk" to "no risk".

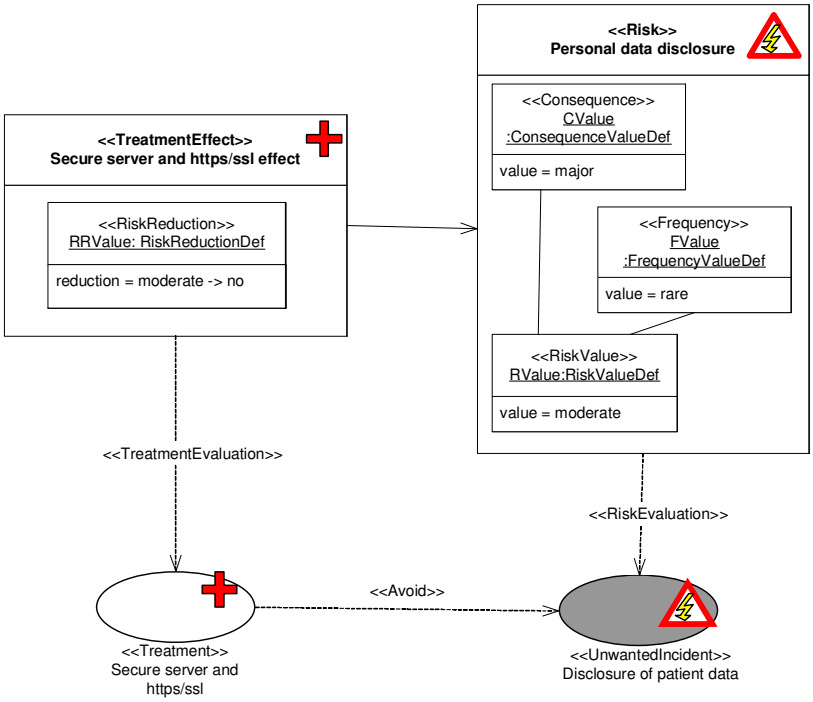

Fig. 10. Evaluation of treatments

5 Conclusions

In this report we have presented a graphical modelling language, implemented as a UML profile supporting security assessments with a special emphasis on improving communication between the stakeholders and other participants of assessments. At the same time the profile is founded in a carefully designed meta-model which facilitates automated analysis of the models expressed in the language.

There exist other UML profiles targeting security [6,16], but our profile is, to our knowledge, the only UML profile with specialised support for security risk analysis, assessment and management.

The language meets the requirements specified by Section 2 in the following sense: (1) The language is based on the same graphical style as UML. Furthermore, carefully designed icons have been incorporated to make specifications easy to comprehend for non-technical people involved in security analysis. (2) The language is defined as a UML 2.0 profile and contained in a recommended OMG standard [11]. (3) The language is founded on a meta-model for security risk analysis developed in the CORAS-project [9,10]. The model builds on international standards for risk

management [2] and security [4,5]. (4) The language provides the same kind of abstraction mechanisms as standard UML. The misuse cases may be detailed and refined in the same way as ordinary use cases using sequence diagrams, activity diagrams and state machines. (5) The language has been designed with the aim to be easily understandable for all groups of stakeholders involved in a security assessment. Icons distinguish symbols representing different concepts; for example, "piles of coins" tags assets, while an "ignited bomb" illustrates treats. The icons are also used to bind together concepts that are related; for example, the icons for "enterprise threat" and "threat scenario" are different, but related since both contain the "ignited bomb". (6) Many dependencies between different diagrams can be deduced from the underlying meta-model and checked automatically. This means that when one part of the security assessment documentation is updated, a tool may automatically detect other parts of the documentation that also must be updated. In this sense the language supports reuse of maintenance of security assessment documentation.

The profile was developed over an extended period of time with continual feedback from several sources: the development of the model-based security assessment methodology of CORAS, from use of the profile in large scale trials where the methodology was applied to real systems, as well as use of the profile in teaching of Master students. The profile is also a part of the proposal for "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms" [11] that was adopted as an OMG standard in November 2003. The standardisation process itself provided much useful input to the development of the profile.

Acknowledgements

The research presented in this report has partly been funded by the Research Council of Norway projects SECURIS (152839/220), SARDAS (152952/431) and QuA (152950/431), and partly by the 5th Framework EU project CORAS (IST-2000-25031). The CORAS consortium consists of eleven partners from four countries: CTI (Greece), FORTH (Greece), IFE (Norway), Intracom (Greece), NCT (Norway), NR (Norway), QMUL (UK), RAL (UK), SINTEF (Norway), Solinet (Germany) and Telenor (Norway). Telenor and SINTEF were responsible for the administrative and scientific coordination, respectively. The results reported in this report have benefited from the joint efforts of the CORAS consortium.

Thanks to Jon Oldevik, Ida Solheim and Jan Øyvind Aagedal for valuable comments on earlier drafts of this report.

References

1. Alexander, I.: Misuse cases: Use cases with hostile intent. IEEE Software **20** (2003) 58-66
2. AS/NZS 4360:1999 Australian standard: Risk Management. Standards Australia (1999)
3. den Braber, F., Dimitrakos, T., Gran, B.A., Lund, M.S., Stølen, K., Aagedal, J.Ø.: The CORAS methodology: Model-based risk assessment using UML and UP. In Favre, L., ed.: UML and the Unified Process. IRM Press (2003) 332-357
4. ISO/IEC TR 13335-1:2001 Information technology – Guidelines for the management of IT security – Part 1: Concepts and models for IT security. International Organization or Standardization (2001)
5. ISO/IEC 17799:2000 Information technology – Code of practice for information security management. International Organization for Standardization (2000)
6. Jürjens, J.: UMLsec: Extending UML for secure systems development. In UML 2002 - The Unified Modeling Language: 5th International Conference, Dresden, Germany, volume 2460 of Lecture Notes in Computer Science. Springer-Verlag (2002) 412-425
7. Krogstie, J.: Evaluating UML using a generic quality framework. In Favre, L., ed.: UML and the Unified Process. IRM Press (2003) 1-22
8. Lund, M.S., den Braber, F., Stølen, K.: Maintaining results from security assessments. In: Proc. 7th European Conference on Software Maintenance and Reengineering (CSMR'03), IEEE Computer Society (2003) 341-350
9. Lund, M.S., Hogganvik, I., Seehusen, F., Stølen, K. (editor): The CORAS framework, the CORAS UML profile for security assessment, and the CORAS library of reusable elements. CORAS public deliverable D3.7 (2003)
10. Lund, M.S, Hogganvik, I., Seehusen, F., Stølen, K.: UML profile for security assessment. Technical report STF40 A03066. SINTEF Telecom and Informatics (2003)
11. OMG: UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, Revised submission to OMG RFP ad/2002-01-07. OMG document: realtime/2003-08-06. Object Management Group (2003)
12. OMG: UML 2.0 superstructure specification. OMG adopted specification ptc/2003-08-02. Object Management Group (2003)
13. Redmill, F., Chudleigh, M., Catmur, J.: Hazop and software Hazop. Wiley (1999)
14. Sindre, G., and Opdahl, A.L.: Eliciting Security Requirements by Misuse Cases. In: Proc. TOOLS-PACIFIC 2000, IEEE Computer Society (2000) 120-131
15. Sindre, G., and Opdahl, A.L.: Templates for Misuse Case Description. In: Proc. Workshop of Requirements Engineering: Foundation of Software Quality, (REFSQ'01). (2001)
16. Lodderstedt, T., Basin, D., and Doser, J.: SecureUML: A UML-based modeling language for model-driven security. In UML 2002 - The Unified Modeling Language: 5th International Conference, Dresden, Germany, volume 2460 of Lecture Notes in Computer Science. Springer-Verlag (2002) 426-441