A Graphical Approach to Risk Identification, Motivated by Empirical Investigations

Ida Hogganvik and Ketil Stølen

SINTEF ICT and Department of Informatics, University of Oslo
{iho, kst}@sintef.no

Abstract. We propose a graphical approach to identify, explain and document security threats and risk scenarios. Security risk analysis can be time consuming and expensive, hence, it is of great importance that involved parties quickly understand the risk picture. Risk analysis methods often make use of brainstorming sessions to identify risks, threats and vulnerabilities. These sessions involve system users, developers and decision makers. They typically often have completely different backgrounds and view the system from different perspectives. To facilitate communication and understanding among them, we have developed a graphical approach to document and explain the overall security risk picture. The development of the language and the guidelines for its use have been based on a combination of empirical investigations and experiences gathered from utilizing the approach in large scale industrial field trials. The investigations involved both professionals and students, and each field trial was in the order of 250 person hours.

1 Introduction

We have developed a graphical approach supporting the identification, communication and documentation of security threats and risk scenarios. The approach has been applied in several large industrial field trials and the major decisions regarding its underlying foundation, notation and guidelines are supported by empirical investigations. Our modeling approach originates from a UML [17] profile [15, 16], developed as a part of the EU funded research project CORAS (IST-2000-25031) [24] (http://coras.sourceforge.net). As a result of our work to satisfy the modeling needs in a security risk analysis, the language and its guidelines have evolved into a more specialized and refined approach. The language is meant to be used by the analyst during the security risk analysis, and has different purposes in each phase of the analysis. A normal risk analysis process often includes five phases: (1) context establishment, (2) risk identification, (3) risk estimation, (4) risk evaluation and (5) treatment identification [2]. In the following we refer to security risk analysis as security analysis. In the context establishment our language is used to specify the stakeholder(s) of the security analysis and their assets in *asset diagrams*. The purpose is to obtain a precise definition of what the valuable aspects of the target of analysis are, and which ones that are more important than others. From empirical investigations [5] and field trials we know that asset identification and valuation is very difficult, and that mistakes or inaccuracies made there may jeopardize the value of the whole security analysis.

O. Nierstrasz et al. (Eds.): MoDELS 2006, LNCS 4199, pp. 574 – 588, 2006.

During risk identification we use *threat diagrams* to identify and document how vulnerabilities make it possible for threats to initiate unwanted incidents and which assets they affect. The threat diagrams give a clear and easily understandable overview of the risk picture and make it easier to see who or what the threat is, how the threat works (threat scenarios) and which vulnerabilities and assets that are involved.

The threat diagrams are used as input to the risk estimation phase, where unwanted incidents are assigned likelihood estimates and possible consequences. The likelihood estimation is often a difficult task, but illustrating the unwanted incidents in the correct context has proved very helpful in practice.

After the risk estimation the magnitude of each risk can be calculated on the basis of its likelihood and consequence, and modeled in *risk diagrams.* The risk diagrams specify which treats that initiate the different risks and exactly which assets they may harm. This risk representation is then compared to predefined risk tolerance levels to decide which ones that need treatments.

In the treatment identification, the threat diagrams that contain the non-tolerated risks are used as basis for *treatment diagrams*. In this phase the appropriate treatments are identified and modeled in treatment diagrams, where they point to the particular place where they should be implemented (e.g. pointing to a vulnerability). The resulting treatment diagrams can be seen as a plan for how to deal with the identified risks.

The contribution of our work is a revised, specialized, graphical language supporting risk identification based on structured brainstorming, and a comprehensive guideline for its use. Moreover, and in particular, we provide empirical support for the language's underlying, conceptual foundation and the main design decisions.

This paper is structured as follows: Sect. 2 introduces structured brainstorming in risk analysis. Sect. 3 explains the language's underlying conceptual foundation and Sect. 4 gives an example-driven introduction to our approach, including guidelines for modeling. Sect. 5 describes the major design decisions made during the development and the empirical investigations supporting these. Sect. 6 discusses the threats to validity for the empirical results, Sect. 7 presents related work and summarizes the main conclusions.

2 Structured Brainstorming for Risk Identification

A frequently used technique in security analysis, and in particular in risk identification, is so-called structured brainstorming (HazOp-analysis [18] is a kind of structured brainstorming). It may be understood as a structured "walk-through" of the target of analysis. The technique is not limited to a specific type of target, but can be used to assess anything from simple IT systems to large computerized process control systems or manual maintenance procedures in e.g. nuclear power plants. This does not mean that exactly the same technique can be applied directly to all kinds of targets, it has to be adapted to fit the target domain. The main idea of structured brainstorming is that a group of people with different competences and backgrounds will view the target from different perspectives and therefore identify more, and possibly other, risks than individuals or a more heterogeneous group. The input to a brainstorming session is various kinds of target models (e.g. UML models). The models are assessed in a stepwise and structured manner under the guidance of the security analysis leader. The identified risks are documented by an analysis secretary.

Deciding which roles should be present is part of tailoring the brainstorming technique towards the target in question. In some cases an expert is only participating in the part of the analysis where his or her expertise is needed. This makes it essential that the information is simple to communicate and comprehend.

3 The Conceptual Foundation

The first step in developing the language was constructing a conceptual model based on standardized security risk analysis terminology. We aimed to use the most intuitive and common interpretations. The conceptual model can be seen as a kind of abstract syntax for the language, and is shown in Fig. 1 using UML class diagram notation.

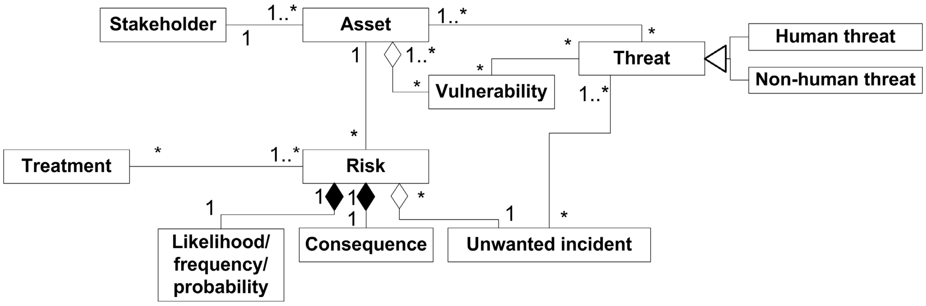

Fig. 1. The conceptual model

The conceptual model may be explained as follows: **stakeholders** are those people and organizations who may affect, be affected by, or perceive themselves to be affected by, a decision or activity regarding the target of analysis [2]. An **asset** is something to which a stakeholder directly assigns value, and hence for which the stakeholder requires protection [3]. Assets are subject to **vulnerabilities,** which are weaknesses which can be exploited by one or more threats [9]. A **threat** is a potential cause of an unwanted incident [9]. An **unwanted incident** [9] is an event that may harm or reduce the value of assets and is something we want to prevent. A **risk** is the chance of something happening that will have an impact upon objectives (assets) [2]. Our model captures this interpretation by defining a risk to consist of an unwanted incident, a likelihood measure and a consequence. The abstract concept "risk", the more concrete "unwanted incident", and their respective relationships to "asset" require some explanation. In our definition, an unwanted incident that harms more than one asset gives rise to one unique risk for each asset it harms. This enables us to keep the consequences for different stakeholders separate, since an asset is always defined with respect to a single stakeholder. The level of risk is measured by a **risk value** [2] (e.g. low, medium, high or other scales) which is based upon the estimated **likelihood** (a general description of frequency or probability [2]) for the unwanted incident to happen and its **consequence** in terms of damage to an asset. A **treatment** is the selection and implementation of appropriate options for dealing with risk [2].

To validate our model and investigate the understanding of security analysis terminology we conducted an empirical study [5]. The 57 subjects included both students who had little or no knowledge of security analysis and more experienced professionals. We found that many of the terms as used in our conceptual model are well understood, even by people without training in security analysis.

The study showed that *human beings* are most commonly viewed as threats, followed by *events* (even if they are in fact initiated by a human). To increase the awareness of non-human threats, we decided to specify threat as either *human threat* and *non-human threat.* Our original specialization of frequency into probability and likelihood was not satisfactory. Both frequency and probability are measures that can be covered by likelihood. Likelihood was on the other hand found to be one of the least understood terms and we therefore decided to include all three concepts. For simplicity we only use "frequency" throughout the rest of the paper. On the question of what it is most common to treat, the subjects gave priority to vulnerability and thereafter risk. To make our model suitable for all treatment strategies we associate treatment with risk. This means that treatments can be directed towards vulnerabilities, threats, unwanted incidents or combinations of these.

4 Graphical Risk Modeling – An Example Driven Introduction

This section provides an example of a security analysis with guidelines for how and where the diagrams are made during the process. Fig. 2 gives the syntactical representation of the concepts from Sect.0, plus *initiate* (arrow), *treatment direction* (dashed arrow), *relationship* (line), *logical gates* (and & or) and *region* (used to structure the models). The symbols are a revised version of the ones used in Sect. 5.

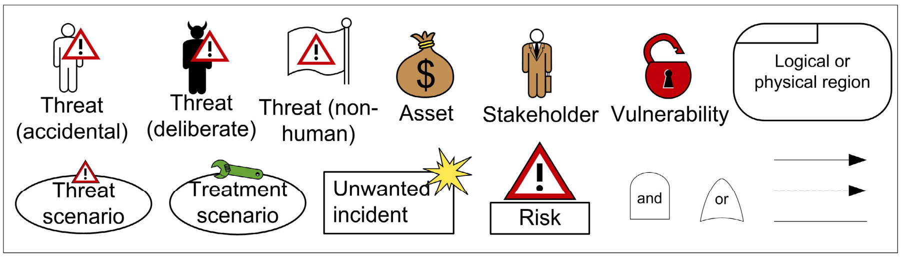

Fig. 2. The graphical representation of the main concepts

The target of analysis in our example is a web-based application which communicates confidential information between an insurance company and its customers. The development project is expensive and prestigious to the company, but the governmental data inspectorate is concerned about the level of privacy of the data provided by the service.

1 - Context establishment: The purpose of the context establishment is to characterize the target of the analysis and its environment. The web application is represented as a logical region (inspired by [20]) with two independent stakeholders (Fig. 3). The stakeholders value assets differently: the governmental data inspectorate has "GDI1.data privacy" as its main asset, while the company management has identified four assets, ordered according to value as follows: "CM1.company brand & reputation", "CM2.data privacy", "CM3.application availability" and finally "CM4.application interface usability".

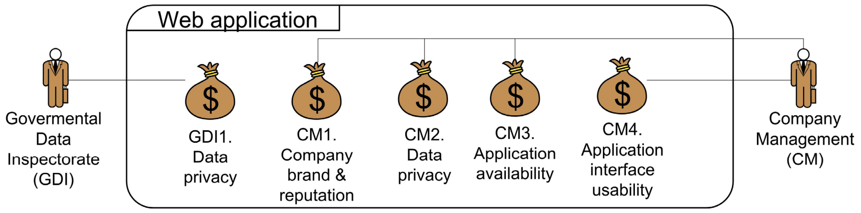
Web application
Govermental Data Inspectorate (GDI)
GDI1. Data privacy
CM1. Company brand & reputation
CM2. Data privacy
CM3. Application availability
CM4. Application interface usability
Company Management (CM)

Fig. 3. Asset diagram

Modeling guideline for asset diagrams:

1. Draw a region that logically or physically represents the target of analysis.
2. Place the assets within the region, numbered according to its importance to the stakeholder, and with a reference to its stakeholder.
3. Associate the stakeholders with their assets.

2 - Risk identification: In the risk identification phase the security analysis leader and the brainstorming participants must find answers to questions like: *what are you most concerned about with respect to your assets?* (threat scenarios and unwanted incidents), *who/what initiates these*? (threats), *what makes this possible*? (vulnerabilities). This information is modeled in *threat diagrams*. Consider the threat diagram in Fig. 4. This diagram focuses on network related threat scenarios. The asset "CM4.application interface usability" is not harmed by this kind of incidents and therefore left out. The stakeholders are concerned about the unwanted incidents: "disclosure of data", "corruption of data" and "unavailability of application". Both threats are considered to cause incidents accidentally and are therefore modeled in the same diagram. We now explain one chain of events from the initiation caused by a threat to the left, to the impact on an asset to the right[1]: "IT-infrastructure" may first use "hardware failure" to make the server crash, and second use the "poor backup solution" to corrupt data and make the application unavailable.

Modeling guideline for threat diagrams:

1. Use the region from the asset diagram and add more regions if useful.
2. Model different kinds of threats in separate diagrams. E.g. deliberate sabotage in one diagram, mistakes in an other, environmental in a third etc. (classification from [9]). This makes it easier to generalize over the risks, e.g. "these risks are caused by deliberate intruders" or "these risks are caused by human errors".
3. Threats are placed to the left in the diagram.

[1] Disregard the frequency and consequence information, this is added later in risk estimation.

4. Assets are listed to the right, outside the region.
5. Unwanted incidents are placed within the region with relations to assets they impact.
6. Assets that are not harmed by any incidents are removed from the diagram.
7. Add threat scenarios between the threats and the unwanted incidents in the same order as they occur in real time (i.e. in a logical sequence).
8. Insert the vulnerabilities before the threat scenario or unwanted incident they lead to. E.g.: "poor backup solution" is placed before the threat scenario "application database fails to switch to backup solution".

3-Risk estimation: The threat diagrams are input to the risk estimation where threat scenarios and unwanted incidents are assigned frequencies and consequences. In Fig. 4 the final frequency for "corruption of data" is based on the frequencies of the two threat scenarios "application servers malfunctioning" and "application database fails to switch to backup solution". We here use the consequence scale: large (L), medium (M) and small (S). In a full security analysis the scale would be mapped to what the client considers to be large, medium and small reductions of asset value (e.g. 10% customer loss, 2 hours unavailability, 10.000$ etc.).

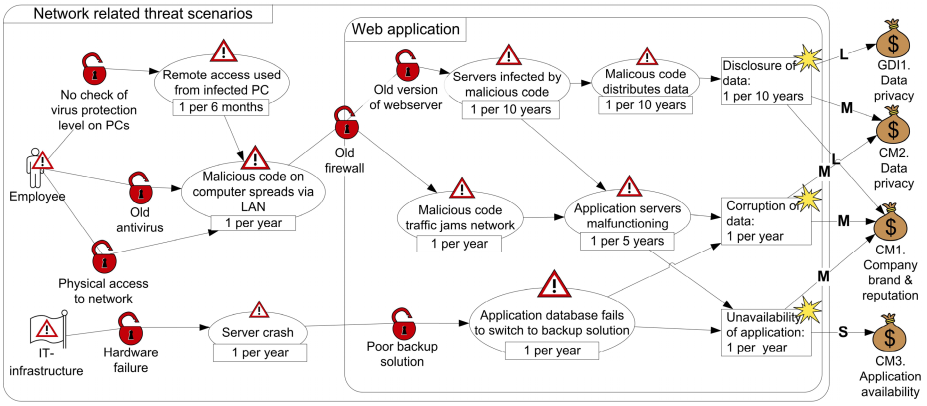
Network related threat scenarios
Web application
No check of virus protection level on PCs
Remote access used from infected PC
1 per 6 months
Employee
Old antivirus
Malicious code on computer spreads via LAN
1 per year
Physical access to network
Old firewall
Old version of webserver
Servers infected by malicious code
1 per 10 years
Malicous code distributes data
1 per 10 years
Disclosure of data:
1 per 10 years
Malicious code traffic jams network
1 per year
Application servers malfunctioning
1 per 5 years
Corruption of data:
1 per year
IT-infrastructure
Hardware failure
Server crash
1 per year
Poor backup solution
Application database fails to switch to backup solution
1 per year
Unavailability of application:
1 per year
L
M
L
M
M
M
S
GDI1. Data privacy
CM2. Data privacy
CM1. Company brand & reputation
CM3. Application availability

Fig. 4. Threat diagram (after risk estimation)

Modeling guideline for risk estimation:

1. Add frequency estimates to the threat scenarios.
2. Add frequency estimates to the unwanted incidents, based on the threat scenarios.
3. Annotate the unwanted incident-asset relations with consequences.

4 - Risk evaluation: On the basis of the risk estimation we model the resulting risks with their associated risk values in a risk diagram. Risk diagrams help the stakeholders to get an overview and evaluate which risks that need treatments. The example in Fig. 5 presents the threats and the risks they represent against the two most important assets in our example ("GDI1.data privacy" and "CM1.company brand & reputation"). We use a short hand notation where "Disclosure of data" in reality represents two risks with value = "major" (shown with associations to two assets).

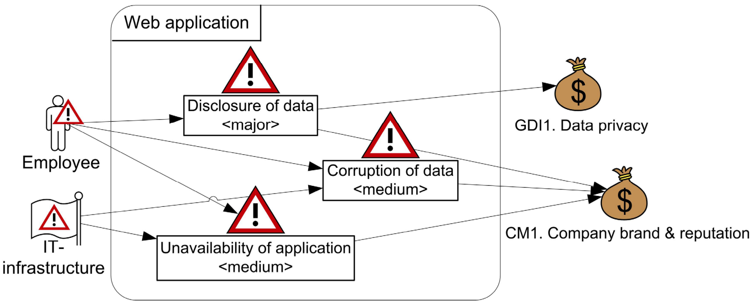

Fig. 5. Risk diagram (for the two most important assets)

Modeling guideline for risk diagrams:

1. Use the threat diagram and annotate all relations between unwanted incidents and assets with the risk symbol, showing a short risk description and the risk value (similar risks may be grouped as in the short hand notation shown in Fig. 5).
2. Split the risk diagrams into several diagrams according to risk value or asset importance (show all intolerable risks, all risks for specific assets etc.).
3. Remove threat scenarios and vulnerabilities, but keep the relations between the threats and the unwanted incidents.

5 - Treatment identification: The threat diagrams are also used as basis for treatment diagrams and extended with treatment options. The treatments must then be assessed to see whether they bring the risk value to an acceptable level. This makes it possible to optimize the treatment strategy according to a cost/benefit analysis. Fig. 6 shows the risks for the two most important assets. The proposed treatments in our example, “upgrade server”, “upgrade backup solution” and “limit access to the network”, are directed toward three vulnerabilities. Since treatments often address vulnerabilities,

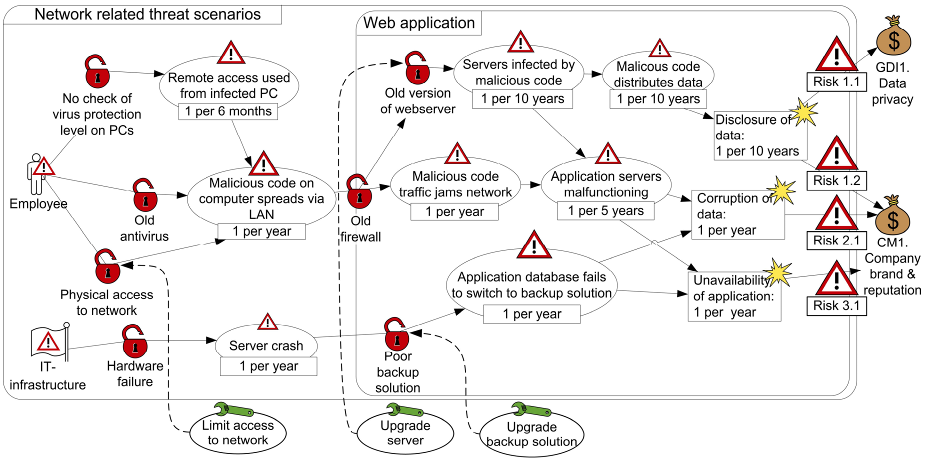

Fig. 6. Treatment diagram for: GCI1.data privacy and CM1.company brand & reputation

one may use the analogy "to close the padlock", - meaning closing the specific pathway through the graph.

Modeling guideline for treatment diagrams:
1. Use the threat diagrams as a basis and annotate all arrows from unwanted incidents to assets with risk icons.
2. If the diagram becomes complex, split it like the risk diagrams.
3. Annotate the diagram with treatments, pointing to where they will be applied.

5 Design Decisions

Designing a language and its guidelines requires many design decisions. In the following we describe our design choices based on experiences from four major field trials and the results from two empirical investigations.

5.1 Field Trial Experiences

The field trials were carried out within the setting of the research project SECURIS (152839/220). Each security analysis required about 250 person hours from the analysis team and 50-100 hours from the stakeholder. The clients and scopes for the analyses were:

- Vessel classification company: A web based information sharing service between customer and service provider.
- Telecom company: Mobile access for employees to e-mail, calendar and contacts.
- Energy company: A control and supervisory system for power grid lines.
- Metal production company: A web based control and supervisory system for metal production.

The participants (the analysis team as well as the representatives of the stakeholders) were requested to evaluate the use of graphical risk modeling after each session and their feedback was:

- the use of the graphical models during the analysis made it easier to actively involve the participants and helped ensure an effective communication between the analysis team and the participants.
- the participants found the notation itself easy to understand and remember. It was considered to be a good way of visualizing threat scenarios and very suitable for presentations. According to one of the participants this type of visualization emphasizes the "message" or the purpose of the analysis.
- one of the main benefits was how the language helped specifying the relations between threats and the chain of events they may cause, the various states of the target, and potential incidents. The modeling method made the participants more conscious of the target of analysis and its risks by representing threats and vulnerabilities more explicitly than "just talking" about them.
- the language provides an opportunity to document cause-consequence relations in a precise and detailed manner.

The participants emphasized that they need a proper introduction to the notation and sufficient time to understand the information that is presented to them. One should limit the amount of information in one diagram, and rather split it according to threat type, scenario type or asset type. It is also important to strive towards correctness from the very beginning. This means that the participants must be involved early, enabling them to adjust the diagram as they find appropriate.

The diagrams capture information gathered during brainstorming sessions and one of the main concerns of the participants was whether or not the diagrams would be complete. To help overcome this we recommend utilizing check lists to ensure that all important aspects are covered. Any changes to the diagrams suggested during the brainstorming session represent important information. Their rationale should be captured by the analysis secretary, and the diagrams should be updated real time.

5.2 The Graphical Icons Experiment

Graphical icons are often seen as mere "decoration" just to make diagrams look nicer, but a major hypothesis in our work is that people unfamiliar with system modeling, may considerably benefit from carefully designed icons. In particular they may help the participants in a structured brainstorming to arrive at a common understanding of the security risk scenarios without wasting too much time. The original UML profile [15, 16] is characterized by its use of special graphical icons. To validate the effect of this we conducted an experiment which compared the usefulness of the UML profile icons compared to standard UML use case icons (Fig. 7) [5].

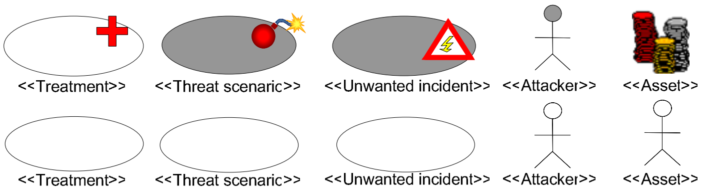

Fig. 7. The notation with and without special icons

In most of the tasks the group receiving normal UML icons had similar mean score to the group using profile icons, but when the time pressure increased the subjects with profile icons managed to complete more tasks than the other group. The positive effects of using graphical icons in models is also supported by [13]. The icons did on the other hand not significantly affect the correctness of interpretation of risk scenarios. Nevertheless, we decided to use special graphical icons to help the participants in a structured brainstorming to quickly get an overview of the risk scenarios.

5.3 Modeling Preferences Experiment

In the already mentioned empirical study of the conceptual foundation we found that some concepts were difficult to understand. Our hypothesis was that an explicit representation of these concepts in the models would mitigate this problem. To select the best representation we conducted an experiment involving 33 professionals. We used

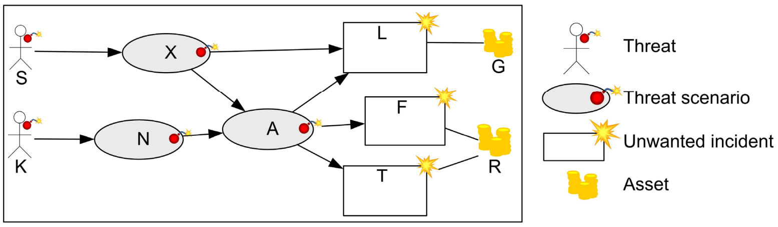

Fig. 8. The basic diagram (using the symbols from the previous version of the language)

variations of Fig. 8, which is a simplified version of a real threat diagram, to illustrate the various options. For a full description of the study we refer to [4].

Representing frequency: To help visualizing which paths that are most likely to be chosen by a threat, we investigated the effect of using different line types (**Fig. 9**). Line type (thick, thin, dashed etc.) has been suggested by [23] to represent aspects of associations between elements in a graph notation.

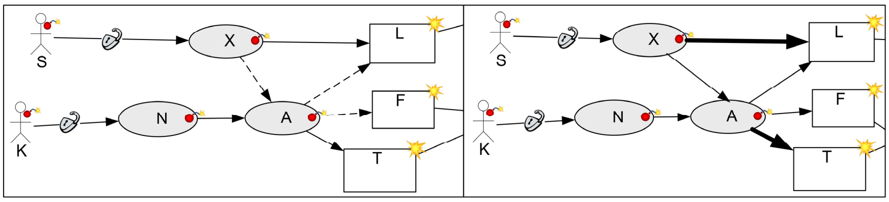

Fig. 9. Paths more likely than others in the graph

The result showed that neither of the line types is preferred for this purpose, possibly because a thick or dashed line does not convey a unique interpretation. During field trials we have found it more helpful to annotate threat scenarios with frequency estimates. The unwanted incident frequency is then estimated on the basis of the frequency estimates of the threat scenarios that cause the incident. This has reduced the need for assigning frequency information to the graph pathways.

Representing risk: The concept "risk" often leads to some confusion due to its abstract nature. We tried to make the risk notion less abstract by decorating the arrows between the unwanted incidents and the assets they harm with risk representations.

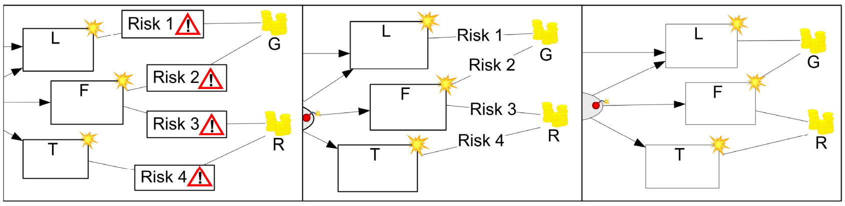

Fig. 10. "F": one incident, but two risks

The alternatives we tested included "text label and icon in box", "text label only" and "association only" (Fig. 10).

The investigation showed that neither of the suggested representations is significantly preferred. Based on this we decided to specify risk in separate risk diagrams. In treatment diagrams, where we need to represent risks in their context, we use a text label and a risk icon. This alternative received the highest score, although the difference was not sufficiently large to be statistically significant from the two others.

In risk evaluation and treatment identification it may be useful to specify the magnitude of risks or unwanted incidents. We tested color, size and text label (Fig. 11), where the two first, according to [23], can be used to visualize magnitude. The result was clear, the participants preferred the text label version over the two other alternatives. A possible explanation is that a text label has unique interpretation, while size or color may have many interpretations.

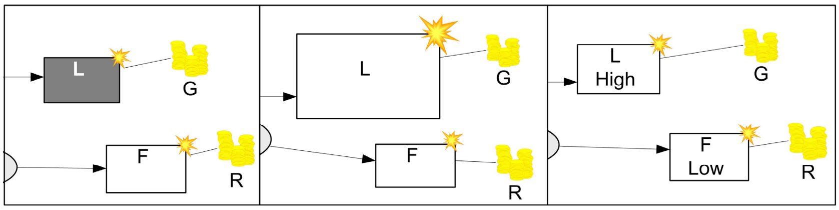

Fig. 11. The magnitude of an unwanted incident

Representing vulnerability: In the original UML profile [15, 16], vulnerabilities were only represented as attributes of the assets they were associated with. During field trials we experienced a definite need for representing vulnerabilities explicitly in the diagrams. We tested the UML profile representation against one that uses a vulnerability symbol (Fig. 12). The result showed that the alternative using chain locks as a vulnerability symbols was significantly preferred.

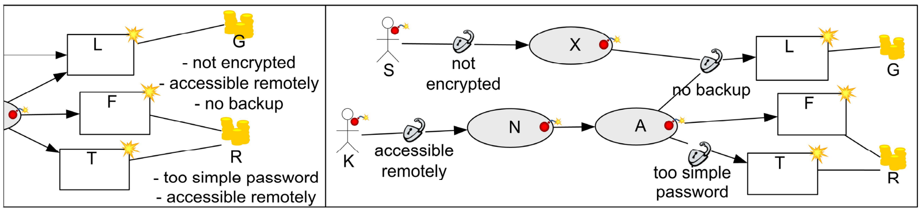

Fig. 12. Modeling vulnerabilities

The vulnerability symbols are helpful in describing threat scenarios and an excellent support in treatment identification. One of the participants in a field trial actually pointed to a vulnerability and claimed "*we've fixed this one now, so you should close the chain lock*".

6 The Validity of the Empirical Results

The main threats to validity of the empirical results on which our approach builds, are summarized in this section. For full details we refer to [4-6].

The investigation of the underlying conceptual foundation was organized as a survey and the subjects included both master students in informatics and professionals. Despite their heterogeneous backgrounds, we believe that the results give a pretty good picture of the opinions of people working with system development in a general sense. The survey was formulated in natural language which always gives room for misinterpretations; some questions were discarded from statistical analysis due to this.

The four major field trials have given us a genuine opportunity to make the modeling guidelines appropriate for actual and realistic modeling situations. One of the main challenges we faced was how to optimize the structure of the diagrams. In some cases it was feasible to structure according to the kind of threat or incident, while in others we structured according to work processes in the target of analysis. This problem must be addressed early in the risk analysis process when the client specifies the acceptable risk level. The other main issue was how to capture changes and comments that arise during the brainstorming sessions. The solution seems to be modeling "on-the-fly". Updating and modifying diagrams real time can be challenging, but with the appropriate tool-support it can be overcome.

The subjects that participated in the icon experiment were master students in informatics. They had been introduced to the UML profile icons in a lesson focusing on the principles of risk modeling. This gave the subjects that received profile icons an advantage, but we believe it was too small to make a real difference.

In the modeling preferences investigation, the various alternatives were tested on a population with mainly technical background. This kind of background is similar to the background of most of the participants in our field trials and therefore considered to be a representative population.

The diagrams we used were a compromise between realistic diagrams and naive notation-only tests. This is a weakness that is difficult to avoid, and we will therefore validate our findings by testing them in real threat diagrams in our next field trial.

In general, humans who get involved in a security analysis are rarely familiar with the concepts or the process used. Concepts that students find intuitive to understand and model are probably intuitive also to professionals. We therefore believe that our use of students in two of the three investigations had little effect on the validity of the results.

7 Conclusions and Related Work

Misuse cases [1, 21, 22] was an important source of inspiration in the development of the above mentioned UML profile. A misuse case is a kind of UML use case [10] which characterizes functionality that the system should *not* allow. The use case notation is often employed for high level specification of systems and considered to be one of the easiest understandable notations in UML. A misuse case can be defined as "a completed sequence of actions which results in loss for the organization or some specific stakeholder" [22]. To obtain more detailed specifications of the system, the misuse cases can be further specialized into e.g. sequence- or activity diagrams. There are a number of security oriented extensions of UML, e.g. UMLSec [11] and SecureUML [14]. These and related approaches have however all been designed to capture security properties and security aspects at a more detailed level than our language. Moreover, their focus is not on brainstorming sessions as in our case.

Fault tree is a tree-notation used in fault tree analysis (FTA) [8]. The top node represents an unwanted incident, or failure, and the different events that may lead to the top event are modeled as branches of nodes, with the leaf node as the causing event. The probability of the top node can be calculated on the basis of the probabilities of the leaf nodes and the logical gates "and" and "or". Our threat diagrams often look a bit like fault trees, but may have more than one top node. Computing the top node probability of a fault tree requires precise quantitative input, which rarely is available for incidents caused by human errors or common cause failures. Our approach can model fault trees, but we also allow qualitative likelihood values as input.

Event tree analysis (ETA) [7] focuses on illustrating the consequences of an event and the probabilities of these. It is often used as an extension of fault trees to create "cause-consequence" diagrams. Event trees can to a large extent also be simulated in our notation.

Attack trees [19] aim to provide a formal and methodical way of describing the security of a system based on the attacks it may be exposed to. The notation uses a tree structure similar to fault trees, with the attack goal as the top node and different ways of achieving the goal as leaf nodes. Our approach supports this way of modeling, but facilitates in addition the specification of the attack initiators (threats) and the harm caused by the attack (damage to assets).

The Riskit method [12] includes a risk modeling technique that makes it possible to specify factors that may influence a software development project. It has similarities to our approach, but targets project risks and therefore utilizes different concepts and definitions and has not the special focus on security risks.

In security risk analysis various types of brainstorming techniques are widely used to identify risks. The participants use their competence to discover relevant threats, vulnerabilities and unwanted incidents that the analysis object may be subject to. The overall idea of brainstorming is that the brainstorming group will identify more risks together than the same individuals working separately. The participants have extensive knowledge of the target of analysis, but often within different areas. The challenge is to make them understand and discuss the overall risk picture without being hampered by misunderstandings and communication problems. Since a brainstorming session may be exhausting, we need techniques and guidelines that make the session as efficient and effective as possible. The findings must also be documented in a precise and comprehensive manner.

We have developed a graphical approach to risk identification to meet these requirements. It captures the complete risk picture, including assets, stakeholders and how vulnerabilities make security threats able to harm the assets. The graphical notation is both simple and expressive, and does not require extensive knowledge of modeling. The language originates from a UML profile for risk assessment [15, 16]. Through application in field trials and empirical studies it has evolved towards a more refined and specialized language. To our knowledge our approach is the only one that combines system modeling techniques with risk modeling techniques for use in structured brainstorming within the security domain.

A major presumption for an intuitive and understandable language is the underlying conceptual foundation, which defines the various concepts in the language and how they relate to each other. We have selected our definitions from well recognized standards within security and risk analysis [2, 3, 9]. To ensure that the best definitions of concepts and relations were chosen, we investigated the alternatives empirically. We found that risk analysis terminology in general is quite well understood. In general, those concepts that caused most uncertainty were the concepts that are least used in daily language. As an example, frequency is less used in the daily language than consequence and was one of the terms many of the respondents had trouble understanding. The subjects tend to care less about the frequency than the consequence of a risk, even though a high frequency risk with low consequence over time may cost the same as a large consequence risk.

The language has been utilized in four large industrial field trials. We have experienced that the diagrams facilitate active involvement of the participants in the brainstorming sessions, and they are very helpful in visualizing the risk picture. According to the participants, the diagrams explicitly illustrate the threats and vulnerabilities in a way that makes it easy to see the relations and precisely define the risk consequences.

The notational design decisions are supported by two empirical studies. The first one investigated the usefulness of special graphical icons contra traditional UML use case icons. The result showed that the group that received diagrams with special icons was able to conclude faster. This and similar supporting studies convinced us to use graphical icons in our language. The second study tested different modeling approaches on a group of professionals. From this we conclude amongst others that text labels are preferable over visualization mechanisms like size, color or line type. For our language this means that we mark a major risk with the text label "major", rather than representing it with a large risk icon.

The final language description will include a formal semantics and a more detailed modeling guideline. The language is currently being implemented in the CORAS' risk analysis tool (v3.0) which is planned for release autumn 2006.

Acknowledgments. The research on which this paper reports has partly been funded by the Research Council of Norway project SECURIS (152839/220). The authors especially thank Mass Soldal Lund for his valuable input. We also thank the SECURIS analysis team: Fredrik Seehusen, Bjørnar Solhaug, Iselin Engan, Gyrd Brændeland and Fredrik Vraalsen.

References

1. Alexander, I., *Misuse cases: Use cases with hostile intent.* IEEE Software, 2003. **20**(1), pp. 58-66.
2. AS/NZS4360, *Australian/New Zealand Standard for Risk Management*. 2004, Standards Australia/Standards New Zealand.
3. HB231, *Information security risk management guidelines*. 2004, Standards Australia/Standards New Zealand.

4. Hogganvik, I. and K. Stølen, *Investigating Preferences in Graphical Risk Modeling* (Tech. report SINTEF A57). SINTEF ICT, 2006. http://heim.ifi.uio.no/~ketils/securis/the-securis-dissemination.htm.
5. Hogganvik, I. and K. Stølen. *On the Comprehension of Security Risk Scenarios. In Proc. of 13th Int. Workshop on Program Comprehension (IWPC'05).* 2005, pp. 115-124.
6. Hogganvik, I. and K. Stølen. *Risk Analysis Terminology for IT-systems: does it match intuition? In Proc. of Int. Symposium on Empirical Software Engineering (ISESE'05).* 2005, pp. 13-23.
7. IEC60300-3-9, *Event Tree Analysis in Dependability management - Part 3: Application guide - Section 9: Risk analysis of technological systems.* 1995.
8. IEC61025, *Fault Tree Analysis (FTA).* 1990.
9. ISO/IEC13335, *Information technology - Guidelines for management of IT Security.* 1996-2000.
10. Jacobson, I., et al., *Object-Oriented Software Engineering: A Use Case Driven Approach.* 1992: Addison-Wesley.
11. Jürjens, J., *Secure Systems Development with UML.* 2005: Springer.
12. Kontio, J., *Software Engineering Risk Management: A Method, Improvement Framework, and Empirical Evaluation.* PhD thesis, Dept. of Computer Science and Engineering, Helsinki University of Technology, 2001.
13. Kuzniarz, L., M. Staron, and C. Wohlin. *An Empirical Study on Using Stereotypes to Improve Understanding of UML Models. In Proc. of 12th Int. Workshop on Program Comprehension (IWPC'04).* 2004, pp. 14-23.
14. Lodderstedt, T., D. Basin, and J. Doser. *SecureUML: A UML-Based Modeling Language for Model-Driven Security. In Proc. of UML'02.* 2002, pp. 426-441.
15. Lund, M.S., et al., *UML profile for security assessment* (Tech. report STF40 A03066). SINTEF ICT, 2003.
16. OMG, *UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms.* 2005, Object Management Group.
17. OMG, *The Unified Modeling Language (UML) 2.0.* 2004.
18. Redmill, F., M. Chudleigh, and J. Catmur, *HAZOP and Software HAZOP.* 1999: Wiley.
19. Schneier, B., *Attack trees: Modeling security threats.* Dr. Dobb's Journal, 1999. **24**(12), pp. 21-29.
20. Seehusen, F. and K. Stølen. *Graphical specification of dynamic network structure. In Proc. of 7th Int. Conference on Enterprise Information Systems (ICEIS'05).* 2005, pp. p.203-209.
21. Sindre, G. and A.L. Opdahl. *Eliciting Security Requirements by Misuse Cases. In Proc. of TOOLS-PACIFIC.* 2000, pp. 120-131.
22. Sindre, G. and A.L. Opdahl. *Templates for Misuse Case Description. In Proc. of Workshop of Requirements Engineering: Foundation of Software Quality (REFSQ'01).* 2001, pp. 125-136.
23. Ware, C., *Information Visualization: Perception for Design.* 2 ed. 2004: Elsevier.
24. Aagedal, J.Ø., et al. *Model-based risk assessment to improve enterprise security. In Proc. of Enterprise Distributed Object Communication (EDOC'02).* 2002, pp. 51-64.