Experimental Threat Model Reuse with Misuse Case Diagrams

Jostein Jensen, Inger Anne Tøndel, and Per Håkon Meland

SINTEF ICT, SP Andersens vei 15 B, N-7465 Trondheim, Norway
{jostein.jensen,inger.a.tondel,per.h.meland}@sintef.no
http://www.sintef.com/

Abstract. This paper presents an experiment on the reusability of threat models, specifically misuse case diagrams. The objective was to investigate the produced and perceived differences when modelling with or without the aid of existing models. 30 participants worked with two case studies using a Latin-squares experimental design. Results show that reuse is the preferred alternative. However, the existing models must be of high quality, otherwise a security risk would arise due to false confidence. Also, reuse of misuse case diagrams is perceived to improve the quality of the new models as well as improve productivity compared to modelling from scratch.

1 Introduction

There is a general agreement that software needs to become more secure, but secure software development methodologies and techniques are seldom applied. A recent survey article by Geer [5] shows that only 10% of the most technically sophisticated companies tend to apply secure development techniques, and only for the most critical 10% of their applications. A quote by Chris Wysopal in the same article suggests that *"part of the solution is to make software-security technology and processes require less time and less security-specific expertise."*

Though there is a wide selection of secure software methodologies available, most of them include some kind of threat modelling in order to understand the dangers and determine the security needs of a system, preferably at an early stage of the development. A threat model is a suitable medium for sharing knowledge about relevant threats, core functionality and assets between security experts and developers, thereby bridging the information gap that tends to exist between these camps.

This paper presents an experiment on reuse of threat models in order to reduce the need of security expertise and time spent creating a model. Many systems face similar threats [4], especially the ones with similar functionality and/or within same application domains. We have worked with a flavor of threat models named misuse cases [11], and investigated how helpful a catalogue of existing models for various systems seems when creating a new model for a new system, compared to not having access to this catalogue. We have also looked at if such a catalogue would impact the creativity, e.g. the number, type and range

M. Soriano, S. Qing, and J. López (Eds.): ICICS 2010, LNCS 6476, pp. 355–366, 2010.
© Springer-Verlag Berlin Heidelberg 2010

of threats, when creating a model. An example of a misuse case diagram can be seen in Figure 1.

Techniques for using UML use case based models to identify threats and to support development of security requirements is not new. The roots can be traced back to 1999 when McDermott and Fox [7] published their paper on using abuse cases to analyse security requirements. In 2000 Sindre and Opdahl first introduced the concept of misuse cases [12], a concept which they refined in 2005 [11]. The misuse case is an extension to the UML use case notation, where inverted symbols for actors and use cases are used to represent malicious actors and misuse cases/threats. Industrial experiences from using the misuse case modelling technique is presented by Alexander [1]. Among his concluding remarks he states that misuse case modelling is a promising approach for eliciting various non-functional requirements, such as security, and to identify threats to system operation. Yet, it is recognised after ten years of existence that industrial uptake of the modelling technique has been low [9]. As a first attempt to provide an empirical ground to select modelling techniqes for early elicitation of security threats Opdahl and Sindre performed an experiment [9] to compare misuse cases and attack trees [10]. They found that the use of attack trees in general resulted in more identified threats than when using misuse case modelling from scratch. However, when the experiment participants were given pre-made use case models which they should extend with misuse cases, the advantage of attack trees with respect to number of identified threats was evened out. Meland et al. [8] carried out a small controlled experiment with developers from the industry to find out more about the potential for reuse of misuse cases. This experiment investigated what form a reusable misuse case should have, comparing full misuse case diagrams to categorised stubs of threats and mitigations. For both approaches the participants generally found that they were able to identify threats and security use cases they would not have identified otherwise, and that both were easy to learn and easy to use.

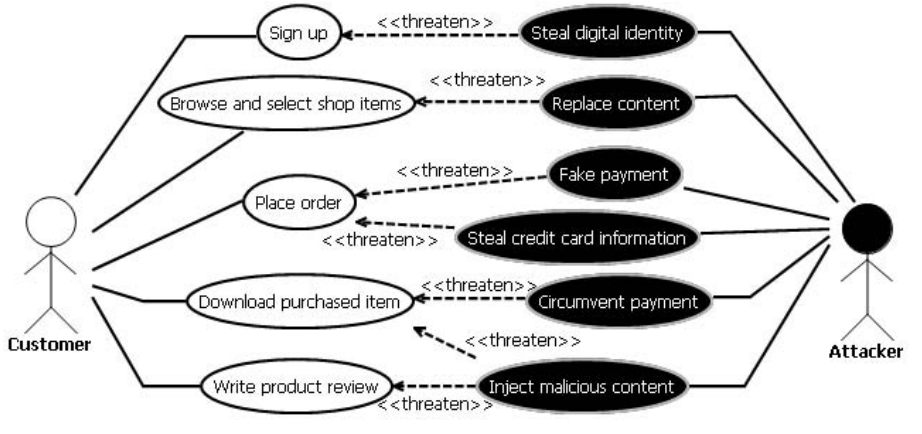

Fig. 1. A simple misuse case diagram showing core functionality and associated threats (see case study A)

2 Method

We have used a controlled experiment to address the following research questions:

RQ1: *For someone who is not a security expert, how is reuse of existing misuse case diagrams (M1) perceived compared to modelling misuse cases from scratch (M2)?*
From this question we derived the following null-hypothesis: **$H1_0$:** *There will be no significant differences in the participants opinion with respect to the two different misuse case modelling approaches.*

RQ2: *Will there be significant differences in the resulting models/threats from these two methods?* This question also includes: *To what extent are the diagrams being reused?*

The experiment involved a group of 30 students with at least three years of university-level computer science and software education, all of them taking a course in software security at the Norwegian University of Science and Technology (NTNU). This course had already taught them threat modelling with misuse cases, so the notation was familiar to them. All participation was voluntary and anonymous.

A Latin-Squares experimental design was chosen so that all the participants could construct misuse case models using both M1 and M2, while we could control for the order in which the different approaches were used. The participants were divided into two groups; Group I started working with case study A using M1, while Group II started out with case study A using M2. After the first run through both groups got a new case study (B) and now performed the modelling with the other approach (group I used M2 while group II used M1).

With M1, the participants were to create their model by going through a "misuse case diagram catalogue" and look for similar applications and diagram elements, and then import what they thought was relevant into their own model. The catalogue contained eight full misuse case diagrams for various applications, but did deliberately not contain any perfect matches for any of the case studies. With M2, the participants were short of this catalogue, but had access to their curriculum on software security and threat modelling.

For comparison reasons, we used the same catalogue and case studies as in the experiment by Meland et al. [8]. These case studies are from two somewhat different application areas, and while each had a 1-2 pages long description, they can be summarised as:

Case study A: Online store for mobile phones. This case study was loosely based on the *mobile Web shop* previously used by Opdahl et al. [9]. The context is an online store with digital content like music, videos, movies, ring tones, software and other products for mobile equipment such as 3G mobile phones. It is mainly to be accessed by customers directly through their mobile phones, but otherwise resembles any other online store.

Case study B: Electronic Patient Record (EPR) system. This case study has a different environment, stakeholders and assets compared to case study A. An electronic patient record system (EPR) is used by clinicians to register and share information within and between hospitals. Much of the patient information should be regarded as strictly private/sensitive, and one fundamental assumption is that the system is only available on a closed health network for hospitals (not accessible from the Internet or terminals outside of the hospitals).

For each case study, the participants had about 25 minutes to create their model. Questionnaires on background information were completed before the experiment began, and post-method questionnaires were filled in after each case (approximately 5 minutes for each questionnaire). The post-method questionnaires consisted of a set of statements inspired by the *Technology Acceptance Model* [3] to measure the participant's perception of the methods:

Q1: This method helped me find many threats I would never have identified otherwise.
Q2: This method helped me find many security use cases I would never have identified otherwise.
Q3: This method was easy to learn and use.
Q4: This method made me spend little time creating the misuse cases.
Q5: I am confident in the results I have created.
Q6: It would be easy for me to become more skilful using this method.
Q7: Using this method is a good idea.
Q8: Using this method would enhance the quality when creating misuse cases.
Q9: I feel more productive when using this method.
Q10: I would like to use this method in the future.
Q11: If I am going to do threat modelling by means of misuse cases in the future, I would prefer to have existing misuse case diagrams available (including a free text area).

3 Results

3.1 Analysis of Questionnaires

The statements were answered according to a 5-point Likert-scale with values from 1 to 5, where 1 represents *Strongly agree* and 5 represents *Strongly disagree*. To check if there were significant differences between the groups' answers for the two approaches, we used an ANOVA single factor analysis with $\alpha = 0{,}05$. The p-values calculated based on this analysis is shown in Table 1.

Looking at the calculated p-values for all questions but the last (Q11), there is evidence of significant differences between the groups after they applied each approach. This means that we can use the statistical results shown in Table 2 and Table 3 to look for perceived differences between M1 and M2, respectively. We will come back to the conclusion on validity with respect to sample size in the discussion. For now we refer to our calculations of measured effect size and calculated Type II error rate, shown in Table 1.

Table 1. Calculated p-values, effect sizes and Type II error rates between groups using M1 and M2

Question	p-value	Measured effect size	Calculated Type II error rate
Q1	0,00001	1,2898	0,075
Q2	0,00045	0,9613	0,283
Q3	0,01343	0,6585	0,589
Q4	0,00007	1,1039	0,171
Q5	0,01690	0,6352	0,612
Q6	0,00228	0,8243	0,416
Q7	0,0044	0,7705	0,472
Q8	0,00027	1,0084	0,165
Q9	0,00010	1,1136	0,178
Q10	0,01199	0,6747	0,572
Q11	0,45820	0,1940	0,926

Q1 and Q2 are related to perceived usefulness of the modelling approaches, and the participants indicated if they could identify more threats and countermeasure using their respective methods than they otherwise would have done. While the participants modelling from scratch (M2) give a somewhat negative indication of this, the participants using the misuse case catalogue (M1) believe they identified more threats and countermeasures than they would otherwise.

Q3 shows us that there are only small differences on how easy the participants felt it was to apply the two methods. Both groups think it was easy, although M1 is perceived a bit easier. When we look at the imagined learning curve (Q6) we see that the experiment participants think it will be easier for them to become more skillful using existing security models as modelling basis than to start modelling from scratch. From Q4 we see that M1 is considered quite more effective in terms of time spent on the modelling effort compared to M2.

In general, the results show that misuse case modelling seems like a good idea (Q7). However, the M1 group is more satisfied with their approach. The participants also feels more confident that M1 will enhance the quality of the models (Q8). Participants using M2 are neutral to the question on whether they feel more productive when using the approach (Q9), while it is a clear indication that M1 is perceived to increase the productivity. Looking into the future, the experiment shows that there is a higher interest in using M1 in future development projects than M2 (Q10).

In Q5 we asked the participants whether they were confident in the results they had produced. The results show that neither of the methods led to confidence. M1 participants were in general neutral to the question, while M2 results show that the experiment group is somewhat negative to the statement.

The last question we asked (Q11) was whether it would be preferable to model misuse case diagrams by support from existing models in the future. For this question the statistical analysis shows that there are no significant differences among the groups. However, the results from both groups are clear that such help would be preferable. In the questionnaire, this last question also included

Table 2. Descriptive statistics for M1

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Q11**
Mean	2,366667	2,733333	2,166667	2,433333	3,033333	1,966667	2	2,066667	2,275862	2,133333	1,866667
Standard Error	0,182469	0,178971	0,118257	0,163885	0,155241	0,13116	0,126854	0,126249	0,130333	0,149584	0,141692
Median	2	2	2	2	3	2	2	2	2	2	2
Mode	2	2	2	2	3	2	2	2	2	2	2
Standard Deviation	0,999425	0,980265	0,647719	0,897634	0,850287	0,718395	0,694808	0,691492	0,701862	0,819307	0,776079
Sample Variance	0,998851	0,96092	0,41954	0,805747	0,722989	0,516092	0,482759	0,478161	0,492611	0,671264	0,602299
Kurtosis	0,41524	-0,09342	1,425783	1,208036	-0,72367	-0,95372	-0,78912	-0,76989	0,179577	1,028134	0,51662
Skewness	0,727527	1,053602	0,649577	0,826619	0,294619	0,049603	-3,3E-17	-0,0874	0,219258	0,950041	0,715927
Range	4	3	3	4	3	2	2	2	3	3	3
Minimum	1	2	1	1	2	1	1	1	1	1	1
Maximum	5	5	4	5	5	3	3	3	4	4	4
Sum	71	82	65	73	91	59	60	62	66	64	56
Count	30	30	30	30	30	30	30	30	29	30	30
Confidence Level(95,0%)	0,373191	0,366037	0,241862	0,335182	0,317502	0,268253	0,259446	0,258207	0,266974	0,305934	0,289793

Table 3. Descriptive statistics for M2

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Q11**
Mean	3,633333	3,633333	2,633333	3,466667	3,6	2,633333	2,62069	2,862069	3,142857	2,758621	2,034483
Standard Error	0,176057	0,162476	0,139649	0,177682	0,170193	0,162476	0,167669	0,162524	0,16031	0,189991	0,175345
Median	4	4	2,5	3	4	3	3	3	3	3	2
Mode	4	4	2	3	3	3	3	3	3	2	2
Standard Deviation	0,964305	0,889918	0,76489	0,973204	0,932183	0,889918	0,902924	0,87522	0,848279	1,023131	0,944259
Sample Variance	0,929885	0,791954	0,585057	0,947126	0,868966	0,791954	0,815271	0,76601	0,719577	1,046798	0,891626
Kurtosis	-0,83309	-0,59003	1,741215	-0,87717	0,780243	-0,59003	-0,65299	0,639441	-0,80012	-0,72533	1,867974
Skewness	-0,15895	-0,11836	1,250135	0,100117	-0,4488	-0,11836	-0,0751	0,969899	0,104017	0,309188	1,021325
Range	3	3	3	3	4	3	3	3	3	4	4
Minimum	2	2	2	2	1	1	1	2	2	1	1
Maximum	5	5	5	5	5	4	4	5	5	5	5
Sum	109	109	79	104	108	79	76	83	88	80	59
Count	30	30	30	30	30	30	29	29	28	29	29
Confidence Level(95,0%)	0,360078	0,332301	0,285615	0,3634	0,348083	0,332301	0,343454	0,332916	0,328928	0,389178	0,359177

a comment field where the participants were encouraged to answer why, or why not. It is interesting to look at some of the answers:

Askeladden: *[Using a misuse case catalogue is]Good if you do not have particular knowledge about the domain, and limited experience with modelling. The more experience - the less need to look at existing cases.*
Neo: *(Plus) [Using a misuse case catalogue makes it]Easier to start. Lots of ideas to get inspiration from. Effective: do not have to repeat work if misuse cases from similar situations are available. (Minus) Do not use your own creativity in a satisfying way.*
Roland: *The [misuse case catalogue] method did not make me feel creative - I'm not 100% certain that I have modeled all important threats, as I have not been using my brain too much. But if the catalogue is good - that would mean it's a very useful tool. If not I guess it might already be a security problem.*
Bart: *I would use the existing diagrams only after I did by my own [misuse case modelling] just to check whether I haven't omitted anything important and well-known.*
Ender: *It's good to have some templates which you know identify many known risks.*
Leela: *It [the misuse case catalogue] shows things you maybe wouldn't have thought of. It gives some extra perspective. Although it also can make it easy to forget to think for yourself.*
Bilbo: *[With a misuse case catalogue, it is] Easier to spot things you might have missed or hadn't though of at all. Very nice to have as reference material if you're new to security modelling.*
Padme: *Risks and vulnerabilities are often known and should be re-used.*

From the statistical analysis we get an indication that misuse case modelling with support from existing misuse cases is perceived as a favorable approach. However, taking a look at the free text answers provided by the experiment participants we see that there are both strengths and weaknesses of the approach to consider. To find answers related to our second research question (RQ2) it is interesting to take a look at the misuse case diagrams handed in after the case study.

3.2 Analysis of Threat Models

For the analysis of the produced misuse case diagrams we registered all threats and grouped the similar ones in categories.

Table 4 shows threats identified for case study A, and table 5 shows threats identified for the case study B. Threats that are only present in one diagram are not listed. For case study A, this was the case for a total of 13 threats (M1: 1, M2: 12). For case study B, 18 threats were only identified once (M1: 3, M2: 15). An *X* in the table means that the threat is included in closely related misuse case diagram in the catalogue. One participant ended up modelling attack trees instead of misuse cases, these models were disregarded.

Table 4. Threats identified for case study A

Threats		M1	M2
Fake payment, e.g. pay with stolen/fake credit card	X	11/16	1/13
Register fake credit card/payment info	X	12/16	-
Fake user ID or phone, or impersonate another user		5/16	6/13
Steal session		2/16	4/13
Manipulate properties like price and availability	X	9/16	-
Manipulate shopping cart, order history or similar	X	2/16	2/13
Manipulate search results/recommendations		3/16	-
Edit another user's personal information/profile	X	11/16	2/13
Add malware		2/16	2/13
Manipulate emails/ confirmation email	X	5/16	-
Modify data in transit		2/16	1/13
Eavesdrop	X	10/16	6/13
Access sensitive info, e.g. credit card info or personal info	X	13/16	10/13
Collect email addresses/phone numbers	X	9/16	-
Get access to unpaid content, content paid for by another user, or download content for additional phones		5/16	6/13
Obtain legitimate user's access credentials		3/16	2/13
Get access to server, exploit vulnerabilities (e.g. in input handling)	X	13/16	4/13
Get access to administrative operations		1/16	1/13

Table 5. Threats identified for case study B

Threats		M1	M2
Spoof user identity (includes authentication as another user, but also to claim to be another ward/hospital)		1/13	3/16
Register false info, or modify legitimate content		5/13	11/16
Fake confirmation (includes confirm own changes)		2/13	4/16
Delete information		3/13	2/16
Import corrupted/false/malicious data		2/13	3/16
Collect (transmitted) user information for profiling purposes	X	3/13	-
Access private/secret data, e.g. patient information	X	10/13	14/16
Steal ID/username and or password		2/13	6/16
Eavesdrop	X	6/13	4/16
Gain access to logs		-	4/16
Utilise import/export of files in order to get access to or spread sensitive information		3/13	3/16
Make service unavailable		1/13	2/16
Get administrator access	X	4/13	1/16
Override/bypass access control		1/13	5/16
Gain control over/steal mobile terminal		3/13	2/16
Access bedside terminal		2/13	1/16
Abuse emergency access control override mechanism		2/13	1/16
Exploit vulnerabilities in the system, e.g. to access server		1/13	4/16
Fake an emergency		-	2/16
Give or sell information to outsiders or non-relevant users		-	2/16

From the table we see that threats marked with an X are more commonly seen in the models created by M1 than M2. This indicates that models are being reused and that the misuse case catalogue influenced the new models to a great extent. We can also see from the analysis that modelling from scratch seems to identify more unique threats that are difficult to place inside one of the pre-made categories.

4 Discussion

4.1 Experiment Validity

Conclusion validity *is the degree to which conclusions we reach about relationships in our data are reasonable.*[1] There are four factors that are essential when looking at conclusion validity: sample size, effect size, Type I error rate and Type II error rate. The change of one causes effects on the other. In our experiment the sample size was dependent on the number of students showing up for class, in this case 30 students. Based on their questionnaire answers we measured the effect size as shown in table 1. This again allowed us to calculate the Type II error rate shown in the same table. Since we wanted to minimise the risk of rejecting the $H1_0$ when it was true, we kept the α level low (0,05). Consequently, our ability to come to right conclusions is given by the Type II error rate (β). A power level (1-β) of 0.80 is considered good when you want to find relationships in the collected data. Four out of the eleven questions, Q1, Q4, Q8 and Q9 satisfy this rule, and a fifth question (Q2) is within a 70% power level. The relationships we have found for the other questions are under the influence of a higher degree of uncertainty.

Consequently we see that there are problems with conclusion validity related to some of the questions, however, we claim to have sufficient evidence to reject $H1_0$. In our opinion there are significant differences between the modelling approaches tested within this experiment.

Construct validity is concerned with whether the study observations can be used to make inference on the theoretical concepts the study was built on. Our questionnaires on participants' background indicate that the experience level on misuse case modelling is low among the participants, and the time given to work on each case study was limited. These factors would have influenced on the construct validity if we were looking at the quality of the produced misuse case models. The conditions for both groups were similar, and consequently will influence all results equally. In our study we focus on perceived differences between two approaches, and look at differences in (not quality of) the modelling results. As such, these factors should not influence our conclusions.

There are several risks in student experiments, e.g. the participants tries to guess the hypothesis and provide answers according to what they think the most correct answer is. A reason for this could be that they believe they will

[1] Definition from Research Method Knowledge Base:
`http://www.socialresearchmethods.net/kb/concval.php`

be evaluated based on their answers. However, in this experiment this risk was mitigated by the fact that the experiment leader was external to the course, the participants were anonymous, and it was explicitly made clear prior to the experiment that it was not a test of skills.

Internal validity is concerned with the ability to conclude whether there is a cause-effect relationship between the treatment and the outcome of the experiment, e.g. whether the change of method made a difference. A normal threat to internal validity for studies involving multiple groups is that the groups are not comparable before the study. In this experiment, however, the questionnaires on participant background show that the software security knowledge and experience with misuse case modelling is similar for both groups. Additionally, our Latin-squares experiment design allows both groups to test both methods so that differences in background would have been evened out. On the other hand, a Latin-squares design, as it is used in this experiment can introduce some degree of uncertainty in the results due to the learning effect between the two runs. Since we used case studies from two different domains we believe the learning effect to be minimal, and if present it can influence the model analysis (RQ2) somewhat, more than the perceived differences between the modelling approaches (RQ1).

The limited time spent on each modelling task can also be a factor influencing the results. When time is short it is convenient to lean to the alternative taking less time to perform, and it is inviting to look at existing solutions. Our results show that modelling with support from existing diagrams is a preferred alternative. The effect of the time limitation related to this result can only be investigated by performing new experiments with more time for the modelling task. For now we note this threat to internal validity, but at the same time we claim that time is a limited resource in most development projects and that the time constraint brings some realism into the experiment.

External validity is concerned with the ability to generalise the results from the experiment participants to the target population. In our setting this means generalising the experiment results to software developers with little to moderate security knowledge. It is common to criticise the use of students as participants in empirical studies as this may cause a threat to external validity. In many situations they are not representative for the target population of the study. Carver et al. [2], on the other hand, provide a good discussion of which situation students are suitable experiment subjects. One of their claims is that students very well can be used for *Studying issues related to a technology's learning curve or the behavior of novices.* They also claim that one requirement for obtaining valuable results from empirical studies with students is that the studies must be *properly integrated with the course.* Since the topic of the experiment was within the course curriculum we caught the opportunity to perform an experiment in line with this requirement. Further, a study performed by Höst et al. [6] concludes that: *Only minor differences between students and professionals can be shown concerning their ability to perform relatively small tasks of judgment.* Their study compared the validity of software engineering studies where students were used instead of professionals.

From this we deduce that the use of students within this experiment is a good match for answering our research questions, especially when considering that our target group is software developers who are not security experts.

Reliability Our post-task questionnaire was originally designed to be aligned with TAM [3] in order to measure the tree factors leading to technology acceptance: perceived usefulness, ease of use and intention to use. However, a factor analysis using VARIMAX rotation indicates that our questions do not correspond to each of these factors the way we intended. Consequently, we have chosen to analyse each question independently.

4.2 Model Analysis

All threat models were analysed to find if there were differences in identified threats between M1 and M2, and if the diagrams from the catalogue were actually reused. The analysis indicated that diagrams modelled using M1 on the same case study are more similar compared to M2, and also that misuse case diagrams are being reused.

A weakness in our analysis is that we created threat categories after the experiment and placed similar threats within the best matching category. Due to this qualitative evaluation, other researchers might have ended up with slightly different results.

The quality of the catalogue also needs to be taken into consideration. For case study B, the most similar misuse case diagram in the catalogue was less detailed and with fewer threats than the most similar diagram for case study A. These differences might affect the final results. The fact that our Latin-squares design allowed one group to look at the misuse catalogue in the first run of the experiment might also colour the results. This group will be more confident with respect to how a misuse case model should look, and create similar models to the first run. However, since we changed the domain for the case studies we believe this effect is limited.

5 Conclusion

We have performed an experiment in order to measure the produced and perceived differences between creating misuse case models from scratch and creating models with support from existing models. The results presented in this paper indicate that software developers who are not security experts, prefer an approach where they can reuse existing models in their threat modelling activity. Analysis also confirms that reuse of models has an influence on the perceived end results. However, caution must be taken before making models available and before utilising existing diagrams; reusable diagrams must be of high quality, otherwise they might themselves introduce new security problems, as one of the experiment participants pointed out.

Acknowledgment

We would like to thank Adjunct Associate Professor Lillian Røstad and her students at the Norwegian University of Science and Technology participating in the experiment, and Martin G. Jaatun for valuable comments. The research leading to these results received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreements no 215995 (SHIELDS) and 257930 (Aniketos).

References

1. Alexander, I.: Initial industrial experience of misuse cases in trade-off analysis. In: Proceedings of IEEE Joint International Conference on Requirements Engineering, pp. 61–68 (2002)
2. Carver, J.C., Jaccheri, L., Morasca, S., Shull, F.: A checklist for integrating student empirical studies with research and teaching goals. Empirical Softw. Engg. 15(1), 35–59 (2010)
3. Davis, F.: Perceived usefulness, perceived ease of use, and user acceptance of information technologies. MIS Quarterly 13(3), 319–340 (1989)
4. Firesmith, D.: Specifying reusable security requirements. Journal of Object Technology 3, 61–75 (2004)
5. Geer, D.: Are companies actually using secure development life cycles? Computer 43, 12–16 (2010)
6. Höst, M., Regnell, B., Wohlin, C.: Using students as subjects - a comparative study of students and professionals in lead-time impact assessment. Empirical Softw. Engg. 5(3), 201–214 (2000)
7. McDermott, J., Fox, C.: Using abuse case models for security requirements analysis. In: Proceedings of 15th Annual Computer Security Applications Conference, ACSAC 1999, pp. 55–64 (1999)
8. Meland, P.H., Tøndel, I.A., Jensen, J.: Idea: Reusability of threat models - two approaches with an experimental evaluation. In: Massacci, F., Wallach, D., Zannone, N. (eds.) ESSoS 2010. LNCS, vol. 5965, pp. 114–122. Springer, Heidelberg (2010)
9. Opdahl, A.L., Sindre, G.: Experimental comparison of attack trees and misuse cases for security threat identification. Information and Software Technology 51(5), 916–932 (2009)
10. Schneier, B.: Attack trees. Dr. Dobb's Journal (1999)
11. Sindre, G., Opdahl, A.L.: Eliciting security requirements with misuse cases. Requirements Engineering 10(1), 33–44 (2005)
12. Sindre, G., Opdahl, A.: Eliciting security requirements by misuse cases. In: Proceedings of 37th International Conference on Technology of Object-Oriented Languages and Systems, TOOLS-Pacific 2000, pp. 120–131 (2000)