The Journal of Systems and Software 87 (2014) 104–118

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Using SMCD to reduce inconsistencies in misuse case models: A subject-based empirical evaluation

Mohamed El-Attar*

Information and Computer Science Department, King Fahd University of Petroleum and Minerals, P.O. 5066, Dhahran 31261, Saudi Arabia

ARTICLE INFO

Article history:
Received 21 April 2013
Received in revised form
25 September 2013
Accepted 9 October 2013
Available online 18 October 2013

Keywords:
Misuse cases
Subject-based experiment
SMCD

ABSTRACT

Security is a crucial requirement in software systems which need to be addressed as early as the requirements phase. The technique of misuse case modeling has been introduced slightly over a decade ago to elicit and specify functional security requirements. Development efforts downstream will be driven by the functional security requirements specified in misuse case models. Consequently, the quality of a misuse case model influences the effectiveness of downstream development efforts. Inconsistencies are an undesired attribute that can severely reduce the quality of misuse case models. In this paper, a controlled experiment involving students is presented which evaluates the reduction of inconsistencies in misuse case models resulting from utilizing a structure called SMCD (Structured Misuse Case Descriptions). The experiment also examines the impact of using SMCD upon other quality attributes of misuse case models. The results of the experiment indicate that using SMCD improves the consistency levels of the developed misuse case models.

1. Introduction

With the advent of the internet there is an ever increasing need for software systems to be interconnected. It would be inconceivable to trust external entities (people and other systems) with the appropriate use of a system. As soon as a software system joins a larger network of systems, it becomes vulnerable to misuse. Building systems to be secure can no longer be considered a luxury feature, but rather a vital requirement. In traditional software development, end systems are more likely to meet the needs of its customers and users if proper requirements engineering practices were utilized. Similarly, as security becomes an indispensable feature of end systems, end systems are more likely to address their security requirements if functional security requirements were considered at the early stages of development. Considering security requirements during the requirements engineering phase will lead to the required defensive mechanisms being built into the system. This approach was shown to be more effective toward satisfying security requirements in comparison to only patching an end system with defensive mechanisms *after* development is completed (Dubois and Wu, 1996; Jürjens, 2004).

The technique of misuse case modeling (Sindre and Opdahl, 2005) was devised to capitalize on the popularity of use case modeling (Armour and Miller, 2000; Booch et al., 2003; Jacobson et al., 1992, 1995; OMG, 2003). Misuse case modeling extends the use case modeling notation in order to elicit and specify misuse case scenarios in relation to ordinary functional requirements defined in use cases. The literature has reported a number of success stories (Alexander, 2002, 2003; Diallo et al., 2006; Ekremsvik and Tiset, 2004; Mæhre, 2005; Stålhane and Sindre, 2007). For ordinary functional requirements, the likelihood of an end system meeting the specified requirements is heavily dependent on the quality of the functional requirements specifications. Similarly, the likelihood of an end system meeting its functional security requirements heavily depends on the quality of the functional security requirements specifications. In the context of mis/use case modeling, the security quality attribute of an end system is dependent on the quality of the misuse case model that was used to drive the development process. A number of quality attributes for misuse case models have been defined in El-Attar (2012), which indicates that inconsistencies in misuse case models can have a detrimental effect on the overall quality of the misuse case model, and thus, on the development process as a whole. To combat the crucial issue of inconsistencies, the author of El-Attar (2012) proposed a structure named Structured Misuse Case Descriptions (SMCD), originally published in this journal. SMCD aids misuse case modelers during the authoring process and guides them toward developing structurally and syntactically consistent misuse case models. While the functional correctness of SMCD was validated in El-Attar (2012), its usability is yet to be validated. Usability of any technique or tool is undoubtedly an important aspect. A technique or tool maybe the most sophisticated invention, but if it cannot be used, it is rendered useless. SMCD was

* Tel.: +966 38602807; fax: +966 38602174.
E-mail address: melattar@kfupm.edu.sa

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.10.017

intentionally designed so that in can be used to improve consistency without requiring a steep learning curve. However, such is only a hypothesis until empirically proven. Although SMCD was designed to be as simple as possible, is it simple enough? To this end, this paper presents a subject-based empirical evaluation of the usability of SMCD to determine if it can be effectively used to improve consistency in misuse case models. The experiment also investigates the effect of using SMCD on other key quality attributes of misuse case models that were identified in El-Attar (2012).

The remainder of this paper is structured as follows: Section 2 provides a detailed description of misuse case modeling and the SMCD structure. In Section 3, the planning and design of the controlled experiment is presented. The experimental results and their analysis are presented in Section 4. Section 5 provides a review of the related works. Finally, Section 6 concludes and suggests future work.

2. Background

This section provides the relative background that motivated this research and prompted the need to perform the controlled experiment presented in this paper.

2.1. Misuse case modeling

Misuse case modeling extends the already popular use case modeling technique. Misuse case modeling introduces four main concepts: (a) misuse cases, (b) misusers, (c) threatens relationships, and (d) mitigates relationships. A misuse case is a narration of a sequence of actions that are performed (intentionally or inadvertently) by a system while interacting with an external entity that can cause harm to some stakeholder (Sindre and Opdahl, 2005). A misuser is an actor that interacts with a system to perform misuse cases, either intentionally or unintentionally (Sindre and Opdahl, 2005). A threatens relationship is one that is directed from a misuse case toward a use case to indicate that the harmful sequence of actions narrated in a misuse case can be performed while the given use case is being executed (Sindre and Opdahl, 2005). A mitigates relationship is one that is directed from a use case toward a misuse case to indicate that the use case narrates a sequence of actions to eliminate or reduce the risk and harm that is intended by a misuse case (Sindre and Opdahl, 2005).

Graphically, misuse cases and misusers are depicted as use cases and actors, respectively, but with inverted colors. The color inversion is symbolic of that fact that misuse cases and use cases are semantically equivalent, but while being opposites of each other in terms of the behavior they describe. The color inversion also provides a visual distinction between misuse and use cases, and between misusers and actors. The visual distinction allows all constructs to be presented in the same diagram. For an in-depth discussion of the misuse case modeling notation and its semantics and how they can be described, readers are referred to Sindre and Opdahl (2005), Sindre et al. (2002) and Sindre and Opdahl (2001).

2.2. Semantic and structural consistency in misuse case models

This paper focuses only on consistency within a misuse case model and not with other documents such as other requirements or design documents. A misuse case model consists of a diagram as well as a set of descriptions. Within a misuse case model, consistency should be attained at a structural and semantic level.

Semantic inconsistency happens when there are two statements that convey contradicting information. For example, if “Misuse Case-A” states that customer accounts are vulnerable to exploitation for 1 h each day, while “Misuse Case-B” states that customer accounts are vulnerable to exploitation for 3 h each day. This type of inconsistency can only be detected by analyzing and understanding text written in natural language form. The application of natural Language techniques is required in order to improve consistency of this type. Recent pioneering work in the area of semantic inconsistency checking has been presented in Kamalrudin et al. (2011). In Kamalrudin et al. (2011), inconsistency is defined as an occurrence “*when two or more stakeholders have differing, conflicting requirements and/or the captured requirements from stakeholders are internally inconsistent when two or more elements overlap and are not aligned*”. The work presented in Kamalrudin et al. (2011) applies to Essential Use Cases (EUC) (Constantine, 1995) by deriving EUC interaction patterns from the requirements and comparing them to patterns in a EUC Interaction Pattern Library that is developed by the same authors. This juxtaposition process along with an inter-representational traceability approach allows users of the approach to check for requirements quality problems (including inconsistency).

Structural inconsistency happens when a misuse case model presents inconsistent information about its structural components. For example, if a diagram depicts that “Misuse Case-1” *threatens* “Use Case-1” while the textual description of “Misuse Case-1” does not state that it indeed does *threaten* “Use Case-1”. This inconsistency may be a result of the analyst overlooking the behavior by “Misuse Case-1” that *threatens* “Use Case-1”, creating vulnerability to such threat. It is highly beneficial to detect such type of inconsistencies as it would prompt the security analyst to provide this essential information to mitigate threats. Occurrences of structural inconsistencies increases significantly as the size of a misuse case model increases.

2.3. What happens if inconsistent misuse case models are used?

Although misuse cases are the polar opposite of use cases with respect to their narrated intention, their semantics remain the same. As such, the harmful consequences of inconsistencies in use case models are also applicable to misuse case models. The literature has extensively discussed the harmful consequences of inconsistent use case models and thus also misuses case models. For example, Lilly (1999) and Bittner and Spence (2002) outline a number of possible inconsistency instances in use case models. The outlined inconsistencies can lead development teams to implement out-of-scope functionality, causing software development projects to become over-budget and behind-schedule. Scott Amber warns that if too many inconsistencies exist in a use case model then it may be deemed out-of-date (Ambler, 2010). Moreover, inconsistencies in use case models are an indication that relevant information is missing or vague (Ambler, 2010). Inconsistencies in use case models may also be a consequence of an ambiguous domain model (Chandrasekaran, 1997). Overall, inconsistencies in use case models have been found to have a detrimental effect on every aspect of the development process, from the design to the test and maintenance phases (Anda et al., 2001). As use case models are used to specify functional requirements, ultimately inconsistencies in use case models will negatively affect the ability of an end system to deliver its required functionalities. Similarly, as misuse case models are intended to specify functional security requirements, inconsistencies in misuse case models will have an adverse effect on the ability of an end system to perform securely.

Due to the critical importance of consistency in use (and misuse) case models, a large number of research works have contributed toward the goal of improving their consistency. For example, a tool was developed to provide use case authors with a template that will improve the consistency of their requirements specifications (McCoy, 2003). Ben Achour et al. presented a set of styling and content guidelines that assist in improving the consistency of descriptions (Ben Achour et al., 1999). Butler et al. applied the concept of refactoring to use case models (Butler and Xu, 2001). A subset of the refactoring techniques aimed to improve consistency in use case models. A structure named SSUCD (Simple Structured Use Case Descriptions) was presented to improve the structural consistency of use case models (El-Attar and Miller, 2008). It can be deduced that consistency will also be a very crucial quality attribute in misuse case models as well. To this end, SMCD was developed, which extends SSUCD. The extension allows models not only to specify structurally consistent use case models, but also to specify structurally consistent misuse case models.

2.4. SMCD (Structured Misuse Case Descriptions): a brief example

The application SMCD is concerned with utilizing a small set of structural constructs to be used in a template. The application of SMCD does not restrict authors from extending the template to cater to their specific needs. The template utilized for SMCD only contains fields that are common in popular templates already presented in the literature (Cockburn, 1997, 2001; Constantine, 1995; Kruchten, 1999; Kulak and Guiney, 2000; Sindre and Opdahl, 2001). Recall that the SMCD structure extends the SSUCD structure which was proposed to help ensure structural consistency in use case models. Therefore, a detailed explanation of the fields related to describing use cases and actors can be found in El-Attar and Miller (2008). The template fields required to describe misuse cases and misusers are as follows.

2.4.1. Misuse Case Name section

The purpose of this section is to specify the name of the misuse case. This section can also be used to specify if the given misuse case is abstract, implementing an abstract misuse case or whether it is specializing another misuse case.

2.4.2. Associated misusers

This section is used to enumerate the list of misusers that are associated with the given misuse case.

2.4.3. Descriptions section

This section of the template is populated using natural language. The only SMCD keyword that can be used in this section is the “INCLUDE” keyword. The “INCLUDE” keyword is used to specify that the behavior of an inclusion misuse case will be performed. It is important to note that SMCD does not mandate that authors compose the description using bullet points, paragraphs or both. This choice is left up to the author.

2.4.4. Extension Points section and Extended Misuse Cases section

The purpose of the “Extension Points” section is to enumerate any extension points that a misuse case may have. Meanwhile, the “Extended Misuse Cases Section” is populated in extension misuse cases only. The “Extended Misuse Cases Section” is used to describe the exceptional or optional behavior that *extends* a base misuse case at a particular *extension point*.

2.4.5. Threatens section

This section of the template is used to list the set of use cases *threatened* by the behavior described in the given misuse case.

2.4.6. Mitigates section (in use cases)

The “Mitigates” section is part of the extended use case template. The “Mitigates” section is used to specify the list of misuse cases that are mitigated by the behavior described by the given use case.

A brief description of the structural constructs of SMCD and how they are deployed is shown in Table 1. SMCD’s constructs Table 1 describes the diagrammatic notation resulting from each SMCD structural construct. Further details and examples of using SMCD, including its complete E-BNF grammar, can be found in El-Attar (2012).

Fig. 1 presents a misuse case diagram that specifies the functional requirements of a voting system and its inherent threats. The diagram is followed with the textual descriptions of a few entities from the diagram. The complete set of textual descriptions would be very space consuming. However, complete examples of using SMCD to describe misuse cases are presented in the original work (El-Attar, 2012).

Actor:
VIP Voter
SPECIALIZES Voter

Brief Description:
A special type of voter who happens to be an important figure at the University.

Misuser Profile:
VIP Information Thief
SPECIALIZES Information Thief

Brief Description:
A crook who is only interested in gaining information about VIP voters. This voter is not interested in manipulating their votes.

Use Case Name:
Submit Vote

Associated Actors:
Voter, VIP Voter

Description:
. . .before submitting the vote to the central server, the INCLUDE <Encrypt Data> use case is performed to encrypt the vote. . .

Use Case Name:
Encrypt Data

Description:
. . .<*description written in syntax-free natural language using any template*>. . .

Mitigates:
Steal Sensitive Information

Misuse Case Name:
ABSTRACT
Steal Sensitive Information

Description:
. . .in order to gain intercept transmitted information, the INCLUDE <Get Access Privileges> misuse case is performed to obtain access privileges. . ..

Misuse Case Name:
Steal Voter Information
IMPLEMENTS Steal Sensitive Information

Associated Misusers:
Information Thief

Description:
. . .<*description written in syntax-free natural language using any template*>. . .

Threatens:
Register Voter

2.5. Other expected indirect effects of using SMCD to describe misuse case models

SMCD was intentionally designed to have a direct effect on the consistency levels of misuse case models. SMCD is also expected to have an indirect effect on two other quality attributes. Firstly, using SMCD is expected to guide authors to place textual information appropriately within the misuse case descriptions. For example, an *include* relationship should be described in the *including* misuse case, while an *extend* relationship should be described in the *extending* misuse case (El-Attar, 2012). Secondly, SMCD is expected to have an indirect effect on the correctness of the information stated by the authors El-Attar (2012). Correctness in this context indicates the truthfulness of the information stated as well as the analytical view level of the misuse cases. It is certainly undesired to have a

Table 1
SMCD's constructs.

Section	Keyword	Diagrammatic representation
Misuse Case Name	ABSTRACT	Used to declare that the given misuse case is *abstract*. *Abstract* misuse cases appear in italic font in diagrams
	IMPLEMENTS	Used to declare the names of *abstract* misuse cases that the given misuse case *implements*. An arrow depicted as ——▷ depicted from given misuse case to the *abstract* misuse case
	SPECIALIZES	Used to declare the name of a parent misuse case. This relationship is depicted as a generalization relationship arrow (——▷) that is directed from the given misuse case to the *parent* misuse case
	Name (in NL)	Name appears as-is in the misuse case oval
Associated Misusers	Names (in NL)	A solid line (association relationship) links the given misuse case with each of the listed misusers
Description	INCLUDE	An arrow depicted as $\overset{<<include>>}{\dashrightarrow}$ is directed from the *base* use case to the *inclusion* use case
Extended Misuse Cases	Base Misuse Case:	An arrow depicted as $\overset{<<extend>>}{\dashrightarrow}$ is directed from the *extension* use case to the *base* use case
	Extension Point:	Used to specify the name of the *extension point* in the *base* misuse case. The name of the Extension Point is depicted between parenthesis on the corresponding *extend* relationship arrow
	IF	Used to specify a condition the must be true in order for the extension behavior to be performed. The condition is depicted in squared-brackets on the corresponding *extend* relationship arrow
Extension Points	Names (in NL)	Names of the Extension Points are shown within the oval of the belonging misuse case. The names appear under a horizontal dotted line that runs across the oval of the given Misuse Case Name
Threatens	Names (in NL)	An open-head arrow ($\xrightarrow{<<threatens>>}$) is directed from the misuse case to each of the listed use cases it *threatens*
Mitigates	Names (in NL)	An arrow depicted as ($\xrightarrow{<<mitigates>>}$) is directed from the use case to each of the listed misuse cases it *mitigates*

misuse case model become consistent by using SMCD but while worsening the levels of the other quality attributes as a side-effect. Therefore, the experiment presented in this paper also evaluates the indirect effect of using SMCD on the placement and correctness levels of misuse case models. While the literature describes many other quality attributes of mis/use case models (El-Attar, 2012), SMCD has no direct or indirect effect any other quality attributes and hence they are not investigated in our experiment.

3. Research method

In this section, a controlled experiment is described that took place at King Fahd University of Petroleum and Minerals, Saudi Arabia. The experiment follows the well-known experimentation process proposed by Wohlin et al. (2000) and accordingly the following subsections describe the experiments: definition, hypotheses formulation, context, subject selection, design, instrumentation and measurement techniques, and validity evaluation, respectively.

3.1. Research questions

The primary research question addressed by this experiment is whether users of SMCD can develop misuse case models with higher consistency levels in comparison to using the conventional UNL (unstructured natural language). In this paper we refer to textual descriptions that are written using UNL as descriptions that use free-flow text and that do not include any syntax or keywords. UNL does not imply that the descriptions do not use a template. The use of templates improves the quality of use case descriptions and similarly misuse case descriptions as well. As such, the UNL descriptions that will be created as part of this experiment will use a template. The template used will be similar to that proposed by the inventors of misuse cases and which can be found in Sindre and Opdahl (2001), which was based on more popular templates

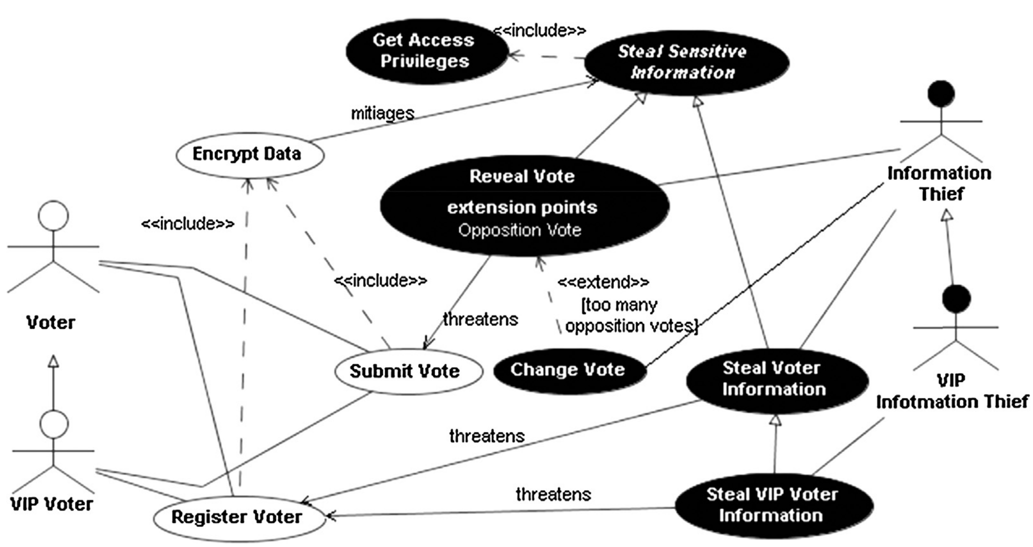

Fig. 1. The misuse case diagram of an online voting system.

for describing use cases such the templates proposed by Kulak and Guiney (2000), Cockburn (2001), or RUP (Kruchten, 1999). The secondary self-imposed research question is whether using SMCD has an effect on other quality attributes of a misuse case model in comparison to using UNL. In this experiment, UNL (along with the template proposed in Sindre and Opdahl (2001)) are used as a control as it is the current state-of-art method to describe mis/use cases as evident in the literature.

Using SMCD is concerned with embedding misuse case descriptions with structural elements. If such structural elements were removed, the resulting artifact would simply be misuse case descriptions in traditional UNL form. The template used to create SMCD descriptions is the same as that will be used to create UNL descriptions. Therefore, the UNL version of the SMCD misuse case descriptions for the online voting system (presented earlier in Fig. 1) will be as shown below (note that fields populated in the same manner as in SMCD misuse case descriptions were omitted for clarity):

Actor:
The VIP voter is a special type of voter who happens to be an important figure at the University.

Misuser:
There is also a VIP information thief whom is a crook who is only interested in gaining information about VIP voters. This voter is not interested in manipulating their votes.

Use Case Name:
Submit Vote

Description:
. . .before submitting the vote to the central server, the vote needs to be encrypted. . .

Use Case Name:
Encrypt Data

Summary:
. . .<*description written in syntax-free natural language using any template*>. . .

Mitigation points:
. . .< *description written in syntax-free natural language using any template* >. . .

Misuse Case Name:
Steal Sensitive Information

Summary:
. . .in order to gain intercept transmitted information, access privileges needs to be obtained. . .

Misuse Case Name:
Steal Voter Information

Description:
. . .<*description written in syntax-free natural language using any template*>. . .

Threats:
. . .<*description written in syntax-free natural language using any template*>. . .

As such, the usage of SMCD is the only independent variable; and hence two treatments exist, SMCD and UNL (non-SMCD). To assess the effect of using SMCD, this experiment has three dependent variables: inconsistency mistakes (I), information misplacement (M) and the content incorrectness (W). Inconsistency mistakes are those that result in structural inconsistency between the misuse case descriptions and the diagram. Information misplacement occurs when correct information is placed at an incorrect section of the descriptions. Content incorrectness occurs when information is described that is factually incorrect outright.

3.2. Hypotheses formulation

To account for the potential effects of using SMCD to misuse case models, three hypotheses were produced (see Table 2). For the *inconsistency* variable (I), the alternative hypothesis indicates that it is expected that using SMCD will result in less instances of inconsistency. Since SMCD was designed to deliberately tackle the issue of inconsistency, therefore inconsistency is the only one-tailed hypothesis. Another expected effect of using SMCD is the correct placement of information within the misuse case descriptions. This effect is represented by the *misplaced* variable (M). Finally, to investigate whether the use of SMCD has an effect on the correctness of the information found in the misuse case descriptions, the *incorrectness* variable (i.e. wrong information) (W) is used to account for this effect. However, since SMCD was not designed to directly improve information placement and correctness, the hypotheses involving the (M) and (W) variables are considered non-directional hypotheses (see Table 2). Al three hypotheses are related to the primary research question.

3.3. Experiment context

This experiment involved software engineering undergraduate students enrolled in a Software Design course. The experiment was conducted three times during the Fall 2011, Winter 2012 and Fall 2012 semesters, in order to gather a sufficient number of subjects as required to perform meaningful statistical analysis. The experiment was conducted as a voluntary laboratory exercise. The laboratory exercise was divided into two main components. The first component was a series of four 1-h lectures to introduce misuse case modeling, its notation and guidelines, and to allow them to practice the technique using a number of examples. The second component of the course consisted of two 3-h lectures in which the subjects would undertake the assigned experimental tasks. The subjects were never informed about the hypotheses under investigation throughout the duration of the experiment.

3.4. Subject selection

All students involved in the experiment were enrolled in a third year undergraduate course and therefore their relative educational experience can be determined. At the time of participating in the experiment, the students have already completed two courses that touch upon the subject of use case modeling. The subject of one of the two courses being Software Requirements Engineering, focused heavily on the topic of use case modeling than the other course which was an introductory course to the broad field of Software Engineering. It is beneficial that the subjects were familiar with use case modeling as this significantly reduced their learning curve since they can readily relate misuse case modeling to the more popular use case modeling technique. Students were not expected to have any experience with misuse case modeling as this topic is not taught in any of the undergraduate courses at the University. Informal interviews with the students later confirmed that none of the students had any experience with misuse case modeling. It is beneficial that the subjects do not have previous experience with misuse case modeling as this helps assure that the subjects do not ignore the prescribed techniques to perform the experimental tasks in favor of other techniques that they are more familiar with from their experience. This lack of knowledge that the subjects have about misuse case modeling does however raise concern with respect to external validity. This issue is later discussed in more detail in Section 4.2.4. Moreover, the fact that all students had similar educational background raises concern with respect to conclusion validity due to the large degree of homogeneity. This issue is later discussed in more detail in Section 4.2.1.

Prior to performing the experimental tasks (Section 3.5), the subjects were given a lecture that revised the use case modeling technique they already knew and introduced the topic of misuse case modeling. In a subsequent lab session, the subjects performed two misuse case modeling exercises that help clarify and strengthen their misuse case modeling skills. The exercises

Table 2
Three dependent variables and their corresponding hypotheses.

Dependent variable	Null hypothesis (Ho)	Alternative hypothesis (Ha)
Inconsistency	(Ho1): I (SMCD) ≤ I (UNL)	(Ha1): I (SMCD) > I (UNL)
Misplacement	(Ho2): M (SMCD) = M (UNL)	(Ha2): M (SMCD) ≠ M (UNL)
Incorrectness	(Ho3): W (SMCD) = W (UNL)	(Ha3): W (SMCD) ≠ W (UNL)

Table 3
Structural and content properties of the RFID and Portfolio systems.

	RFID system	Banking system
Structural elements		
Actors	1	5
Use cases	13	13
Misuse cases	12	6
Misusers	1	2
Include relationships	23	19
Extend relationships	0	2
Threatens relationships	12	2
Mitigates relationships	14	8
Association relationships	2	20
Total	76	72
Information facts	98	112

were concerned with drafting a misuse case diagram based on functional security requirements then creating their corresponding descriptions using UNL. The final preparation sessions introduced the subjects to the SMCD language and they practiced creating misuse case descriptions using SMCD.

3.5. Experimental design tasks

During the experiment, all subjects were required to consider two distinct systems, a RFID-based product authentication system (Lehtonen et al., 2007) (see Fig. 2) and a Swiss portfolio management company system (see Fig.) (Regev et al., 2005). For simplicity, from this point onwards in this paper the RFID-based product authentication system will be referred to as the "RFID system". Meanwhile, the Swiss portfolio management company system will be referred to as the "Banking system". The misuse case diagrams for both systems are provided in their respective sources. Using systems not developed by the author of this paper is important to eliminate any bias, since the author created SMCD. The two misuse case models used in this experiment were also selected as they present real-world systems which reduce the external threat to validity. Moreover, the two misuse case models used in this experiment were also selected as they are relatively large models (in terms of the number of entities present). To the best of our knowledge, the misuse case models used were the largest models that are made publicly available. Most misuse case models present in the literature or online are usually much smaller.

The subjects were evenly distributed into two groups (A and B) of 16 each. The subjects were provided with the Requirements Documents (RDs) and the respective misuse case diagrams of both systems and were asked to develop their respective misuse case descriptions. The RDs were created by two different authors whom have no connection with this experiment in order to eliminate bias toward using SMCD and UNL. A 2 × 2 partial factorial design with repeated measure was used to mitigate the effect of individual and group abilities (see Table 3 for details). As shown in Table 3, the systems used in the experiment were of approximate "difficulty level" in terms of structural and content properties, which is crucial in order to mitigate learning effects. The creators of the RDs used the originating literature sources (Lehtonen et al., 2007; Regev et al., 2005) to guide the development of the RDs. During all sessions of the experiment the subjects were allocated to different rooms based on their belonging group. Prior to each session of the experiment, the subjects were introduced to the business cases behind the misuse case models (see Fig. 3).

3.5.1. Time allocation

Subjects were expected to finish the given exercises in approximately 1.5 h (±15 min). It was believed that the subjects will not face any timing pressures since the sessions were 3 h long. Informal interviews with the subjects later confirmed that they did not feel any timing pressures. Time was measured by the experiment conductor. After conducting the experiment, it was determined that all subjects were able to finish their experimental tasks within the expected time frame with no great time differences witnessed.

3.6. Instrumentation

To eliminate any bias that might be introduced by tool support, the subjects were required not to use any tool support to author the misuse case descriptions whether in SMCD or UNL. The subjects were instructed to use pencil and paper to write their misuse case descriptions.

3.7. Analysis procedure

The quantitative data presented in this paper was considered as discrete count data since there was no empirical evidence that would prevent from assuming that all deficiencies have an equal unit weighting. The data sets were examined for their compliance to the normality assumptions using the Shapiro–Wilk test (Shapiro and Wilk, 1972). This test was preferred over other common "normality" tests as it tends to be more powerful and it does not require that the mean or variance of the hypothesized normal distribution to be specified in advance. This feature of the Shapiro–Wilk test was essential since there was no causal explanation as to the nature of the distribution that the data points are sampled from.

3.8. Scoring and measurement

This section presents examples of misuse case modeling defects (see Table 4). An exhaustive set of defect examples would require a great deal of space to represent. However, Table 4 outlines a large cross section of defect examples that can result from using (or not using) SMCD. Measurement is performed by manually inspecting the misuse case models for the existence of any defects such as those outlined in Table 4. When a defect is detected, it is categorized based on the type of quality attribute it violates. A particular defect can be counted more than once if it violates more than one quality attribute. All misuse case models were evaluated by the author in addition to two other independent researchers. Any disagreements were resolved through discussion. The majority of disagreements were based on the issue of identifying the type of quality attribute violated by a particular defect. Only a small minority of the disagreements were based on the detection of a particular defect in the first place.

4. Analysis and interpretation

We present a descriptive summary for each non-parametric variable in the form of notched box and whiskers plot (see Fig. 4).

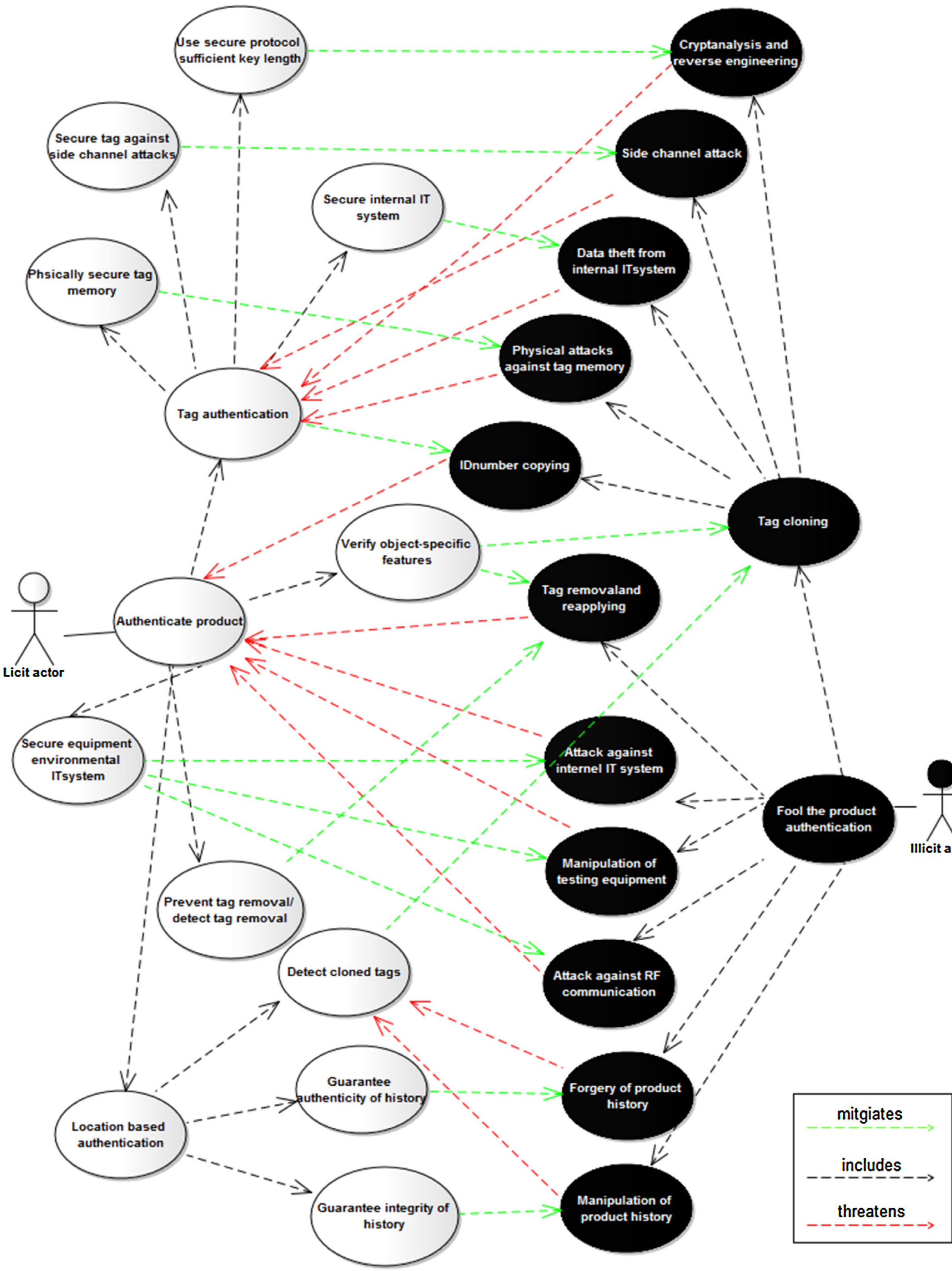

Fig. 2. The misuse case diagram of the RFID system [Lehtonen].

The upper and lower horizontal lines of the notched box represents the lower and upper quartiles. The middle horizontal line shows the median. The tilted lines show the confidence interval around the median. Whiskers (vertical lines) extending from the notched boxes extend to the furthest observations within ±1.5 IQR (interquartile ranges) of the 1st and 3rd quartile. Observations outside 1.5 IQRs are marked as near outliers (+), and those outside 3.0 IQRs are marked as far outliers (*).

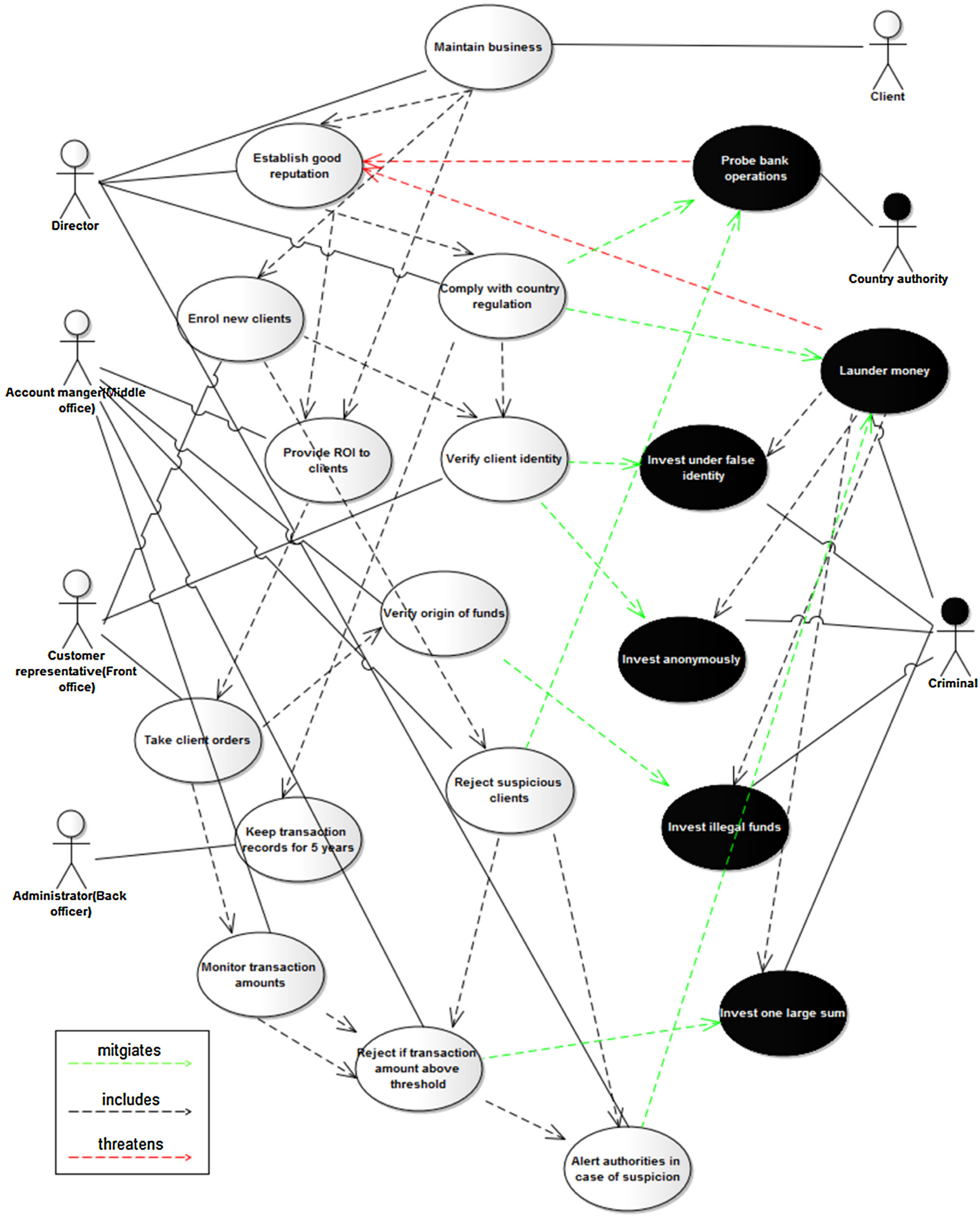

Fig. 3. The misuse case diagram of the Banking system [Regev].

The Mann–Whitney U statistic was used to test for differences between the medians of related samples. The Mann–Whitney U statistic was calculated as prescribed in Siegel and Castellan (1988). It should be noted that the probability provided should be considered as an underestimation due to the presence of a number of ties in the related samples which prevents the use of an exact set. Moreover, the Hodges–Lehman method (Lehmann, 1998) was used to also compute the confidence intervals around the difference between medians. All confidence intervals are given at the standard 95% level.

Table 4
Examples of misuse case modeling defects.

Quality attribute	Examples
Consistency	• A relationship between two entities that is documented but not shown in a diagram • A relationship shown between two entities in a diagram that is not documented • A type relationship that is shown in a diagram but documented as a different type of relationship • Two contradicting statements • A diagram that depicts a relationship between two entities while another diagram depicts a contradicting relationship between the same entities
Placement	• An inclusion relationship described in the inclusion mis/use case • An extension relationship described in the extension mis/use case • A threatens relationship described in a use case • A mitigates relationship described in a misuse case
Correctness	• An incorrect fact or information about an actor or a misuser • An incorrect activity occurring in the workflow described in a mis/use case • An incorrect dependency between any two elements • The introduction of an incorrect mis/use case • The introduction of an incorrect actor or misuser • Any other general incorrect facts or information described in a mis/use case • An information or fact about selecting a particular technology solution • An information or fact that prescribes an algorithm or procedure to satisfy a business requirement • A presumption of the internal behavior of an external entity • An assumption about the GUI interface • An assumption about the deployment of a particular architecture •An ambiguous information or fact stated about an actor or misuser • An ambiguous information or fact stated in a mis/use case • An omitted statement of necessary facts

For tests that indicate statistical significance, an estimate of the size of the difference between the two groups of related datasets is provided by estimating the associated effect size. Given the non-normal nature of the given datasets, Cliff's delta (Cliff, 1993, 1996a,b) is selected as a non-parametric effect size measure. Cliff's delta have been empirically proven to be superior to Cohen's *d* and Hedge's *g* when the data is non-parametric and possesses a variance heterogeneity (Hess et al., 2005; Kromrey et al., 2005; Kromrey and Hogarty, 1998). Cliff's delta is used to examine the probability that individual observations within one dataset are likely to be greater than the observations in the other dataset. A critical feature of using Cliff's delta is that it overcomes the inability to propose directional hypothesis due to the nature of the majority of the dependent

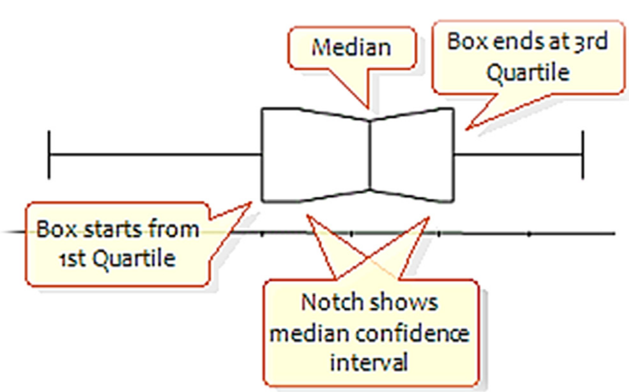

Fig. 4. Notched box and whiskers.

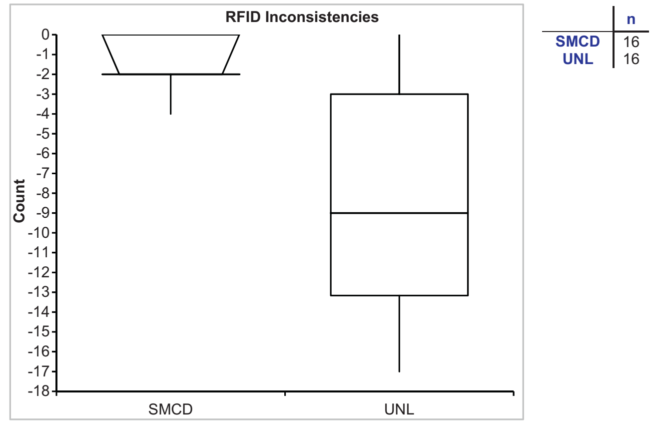

Fig. 5. Inconsistencies – RFID system.

variables determined for this experiment. Two populations are considered equal if zero is included within the confidence interval of Cliff's delta. If the confidence interval only includes positive numbers then SMCD > UNL (favoring SMCD subjects); if it only includes negative numbers then UNL > SMCD (favoring UNL subjects).

4.1. Performed analyses

The analyses performed investigates the effects of the treatment variables and experimental artifacts in isolation: in Section 4.1.1 the effect of using SMCD vs. UNL on each system and with respect to each quality attribute separately; in Section 4.1.2, the results obtained for the RFID and Banking systems with respect to each quality attribute separately; and in Section 4.1.3 the performance of Group A vs. Group B with respect to each quality attribute individually, using both systems.

4.1.1. SMCD vs. UNL

Figs. 5 and 6 and Tables 5 and 7 show the results for the inconsistencies count for the RFID and Banking systems, respectively. Tables 6 and 8 show the results of the Mann–Whitney test for all quality attributes for the RFID and Banking systems, respectively. It can be shown from Tables 6 and 8 that SMCD subjects have statistically significant lower inconsistency values than UNL subjects with respect to both the RFID and Banking systems, respectively. This indicates that the approach based on embedding the structural constructs of SMCD within descriptions explicitly prompts subjects to consider and crosscheck diagrammatic and descriptive elements for consistency. This statistical significance is re-enforced by a positive $\hat{\delta}$ and a confidence interval that only includes positive values

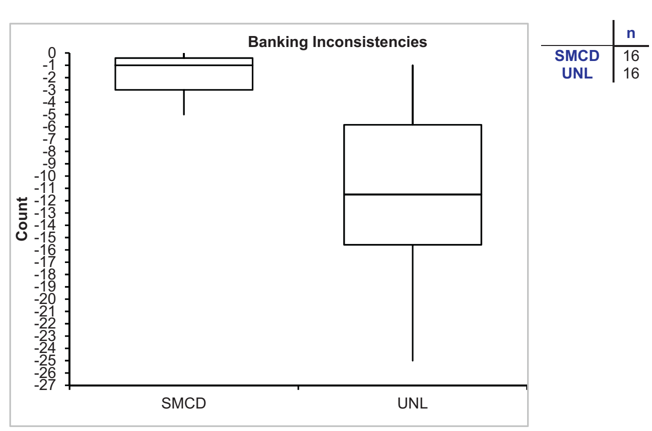

Fig. 6. Inconsistencies banking.

Table 5
Descriptive statistics of the results for SMCD vs. UNL (RFID system).

	System	n	Median	IQR	95% CI of median
Inconsistencies	SMCD	16	−2.0	2.0	−2.0 to 0.0
	UNL	16	−9.0	10.2	−14.0 to −3.0
Misplacements	SMCD	16	0.0	0.0	0.0–0.0
	UNL	16	0.0	0.0	0.0–0.0
Incorrectness	SMCD	16	0.0	0.0	0.0–0.0
	UNL	16	0.0	0.6	−1.0 to 0.0

Table 6
Mann–Whitney test for the RFID system.

Quality attribute	Technique	Rank sum	Mean rank	U	Difference between medians	95% CI	Mann–Whitney U statistic	p
Inconsistencies	SMCD	358.5	22.41	33.5	7.0	3.0 to +∞	33.5	0.0001
	UNL	169.5	10.59	222.5				
Misplacements	SMCD	248.0	15.50	144.0	0.0	0.0–0.0	144.0	0.1508
	UNL	280.0	17.50	112.0				
Incorrectness	SMCD	275.0	17.19	117.0	0.0	0.0–0.0	117.0	0.5652
	UNL	253.0	15.81	139.0				

Table 7
Descriptive statistics of the results for SMCD vs. UNL (Banking system).

	System	n	Median	IQR	95% CI of median
Inconsistencies	SMCD	16	−1.0	2.6	−3.0 to 0.0
	UNL	16	−11.5	9.8	−16.0 to −5.0
Misplacements	SMCD	16	0.0	0.0	0.0–0.0
	UNL	16	0.0	0.0	0.0–0.0
Incorrectness	SMCD	16	0.0	0.0	0.0–0.0
	UNL	16	0.0	0.0	0.0–0.0

Table 8
Mann–Whitney test for the for the Banking system.

Quality attribute	Technique	Rank sum	Mean rank	U	Difference between medians	95.2% CI	Mann–Whitney U statistic	p
Inconsistencies	SMCD	374.0	23.38	18.0	10.0	7.0 to +∞	18.0	<0.0001
	UNL	154.0	9.63	238.0				
Misplacements	SMCD	280.0	17.50	112.0	0.0	0.0–0.0	1.44	0.1508
	UNL	248.0	15.50	144.0				
Incorrectness	SMCD	278.5	17.41	113.5	0.0	0.0–0.0	113.5	0.3416
	UNL	249.5	15.59	142.5				

(Table 9). No statistical significance was observed, in either system, for the *misplaced* and *incorrectness* variables. It should be noted that the majority of the subjects did not commit *misplacement* and *incorrectness* mistakes. Therefore, a notched box and whiskers plot is not provided for these two variables since they do not present any useful information. For the misplacement variable, it is believed that the subjects have carefully learned, from the tutorials just prior to the experiment, the correct placement of the information within the descriptions. The subjects applied this lesson correctly while using both treatments. For the incorrectness variable, it is believed that the subjects have used the requirements documents of the systems as the only source for writing the descriptions. Very few subjects have decided to conjure new (incorrect) information from their own minds.

4.1.2. *RFID system vs. Banking system*

This section investigates the relationship between the two systems. We investigate the relationship between the systems for a possibility of a learning effect. It should be noted that due to the limited partial factorial nature of the experiment, it was not

Table 9
Cliff's delta for the RFID and Banking systems.

Quality attribute	System	Cliff's delta ($\hat{\delta}$)	Variance	Confidence interval around delta ($\hat{\delta}$)	
				Maximum	Minimum
Inconsistencies	RFID	0.739	0.019	0.906	0.370
	Banking	0.859	0.010	0.962	0.536
Misplacements	RFID	−0.125	0.007	0.044	−0.287
	Banking	0.125	0.008	0.298	−0.056
Incorrectness	RFID	0.086	0.025	0.376	−0.279
	Banking	0.113	0.016	0.346	−0.132

Table 10
Descriptive statistics of the results for RFID vs. Banking system.

	System	*n*	Median	IQR	95% CI of median
Inconsistencies	RFID	32	−2.5	8.3	−5.0 to 2.0
	Banking	32	−4.0	10.6	−11.0 to −1.0
Misplacements	RFID	32	0.0	0.0	0.0–0.0
	Banking	32	0.0	0.0	0.0–0.0
Incorrectness	RFID	32	0.0	0.0	0.0–0.0
	Banking	32	0.0	0.0	0.0–0.0

possible to distinguish between these two factors. For simplicity, this section will only refer to the evaluation with respect to the systems. For this analysis, the score for each subject with each system, regardless of the treatment, were added from both groups. This analysis is applied for each quality attribute. Table 10 shows the results of the subjects' performances for both systems with respect to each quality attribute while Fig. 7 shows the results of the subjects' performances with respect to the inconsistencies variable only. As mentioned in Section 4.1.1, the majority of the subjects committed no mistakes with respect to the *misplacement* and *incorrectness* variables and hence a notched box and whiskers plot was not provided for these two variables.

It can be shown from Tables 11 and 12 that no statistical significance was observed for any of the three quality attributes. Therefore, the complexity of the systems and their order is not believed to affect the subjects' performances.

4.1.3. *Group A vs. Group B*

A significant difference between the performances of each group may indicate a bias toward a particular technique. In this section we investigate the difference between the two groups. For this analysis, the score for each subject in each group for each quality attribute, regardless of the treatment used, were added for both systems. Table 13 shows the results of the subjects' performances from both groups with respect to each quality attribute. Fig. 8 shows the results of the subjects' performances with respect to the inconsistencies variable only.

As shown in Tables 14 and 15, no statistical significance was observed with respect to the *inconsistencies* and *incorrectness* variables. However, a statistical significance was observed between the subjects' performances with respect to the *misplacement* variable (in favor of Group A). Further examination of the subjects' performances was conducted to shed more "light" into this situation. It was revealed that two subjects (out of 16) in Group B scored very poorly with respect to the *misplacement* variable. The subjects scored '−15′, '−13′, '−12′ and '−11′. This situation has been avoided by all other subjects (30) simply by applying the instructions given by the tutorial instructor. It most likely that the two subjects missed the instructor's instructions about proper placement of information, perhaps due to a lack of concentration. A significant threat of validity however is not considered as the two concerned subjects actually performed better than the average score of the other subjects with respect to the other two quality attributes.

4.2. *Threats to validity*

This section presents the threats to the validity of the study in accordance with the standard classification as outlined in Wohlin et al. (2000).

4.2.1. *Conclusion validity*

Heterogeneity within subjects is a common validity threat in student-based experiments. There is a serious threat that the observed results are attributed to individual differences rather than the applied treatments if a population exhibits a large degree of heterogeneity. To mitigate against this threat and to increase homogeneity, the experiment was conducted using students enrolled in the same course and whom share a similar educational background. All subjects were familiar with the technique of use case modeling through their previous studies however none were familiar with misuse case modeling. Moreover, all students underwent the same seminars and practice exercises before the experiment.

The fact that the subjects were not familiar with misuse case modeling aided in ensuring that the subjects applied the prescribed treatments as instructed rather that utilizing techniques they have learned previously.

4.2.2. *Internal validity*

As with any subject-based experiment, there is a natural threat of fatigue and maturity. To combat these threats, the subjects were allotted 3 h time slots to perform tasks that are expected to be completed in approximately 1.5 h (±15 min). This expectation was later verified as all subject finished their tasks within the expected time frame with no significant variations. Therefore, the subjects did not

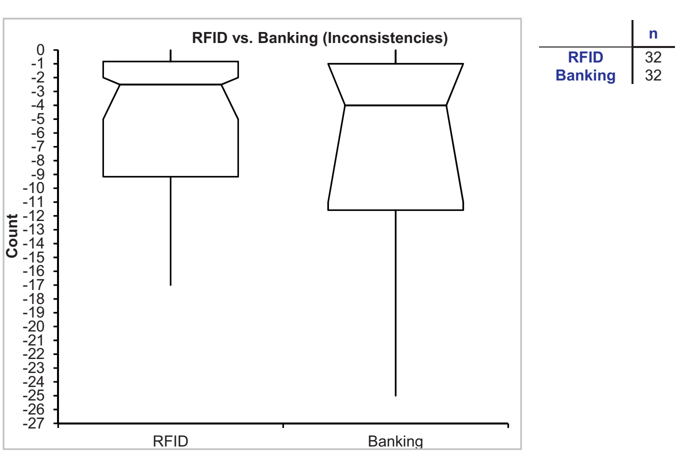

Fig. 7. Inconsistencies – RFID vs. Banking system.

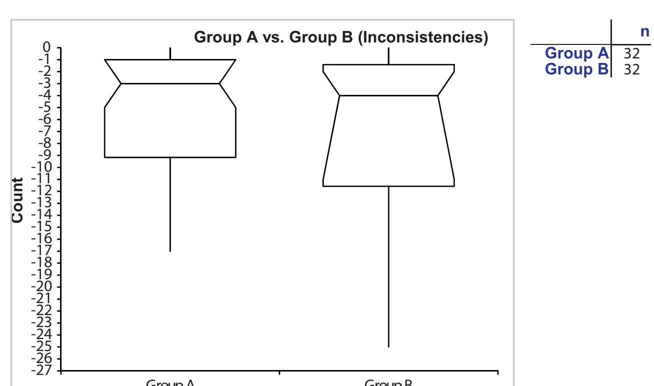

Fig. 8. Inconsistencies – Group A vs. Group B.

Table 11
Mann–Whitney test for RFID vs. Banking system.

Quality attribute	System	Rank sum	Mean rank	U	Difference between medians	95% CI	Mann–Whitney *U* statistic	*p*
Inconsistencies	RFID	1107.5	34.61	444.5	1.0	−1.0 to 3.0	444.5	0.3613
	Banking	972.5	30.39	579.5				
Misplacements	RFID	1040.0	32.50	512.0	0.0	0.0–0.0	512.0	1.0000
	Banking	1040.0	32.50	512.0				
Incorrectness	RFID	1000.5	31.27	551.5	0.0	0.0–0.0	551.5	0.4193
	Banking	1079.5	33.73	472.5				

Table 12
Cliff's delta for all quality attributes.

	Cliff's delta ($\hat{\delta}$)	Variance	Confidence interval around delta ($\hat{\delta}$)	
			Maximum	Minimum
Inconsistencies	0.132	0.021	0.396	−0.153
Misplacements	−0.077	0.009	0.111	−0.260
Incorrectness	−0.053	0.010	0.142	−0.243

Table 13
Descriptive statistics of the results for Group A vs. Group B.

	Group	*n*	Median	IQR	95% CI of median
Inconsistencies	A	32	−3.0	8.2	−5.0 to −1.0
	B	32	−4.0	10.2	−11.0 to −2.0
Misplacements	A	32	0.0	0.0	0.0–0.0
	B	32	0.0	0.0	0.0–0.0
Incorrectness	A	32	0.0	0.0	0.0–0.0
	B	32	0.0	0.0	0.0–0.0

Table 14
Mann–Whitney test for Group A vs. Group B.

Quality attribute	Group	Rank sum	Mean rank	*U*	Difference between medians	95.2% CI	Mann–Whitney *U* statistic	*p*
Inconsistencies	A	1077.0	33.66	475.0	1.0	−1.0 to 3.0	475.0	0.6168
	B	1003.0	31.34	549.0				
Misplacements	A	1104.0	34.50	448.0	0.0	0.0–0.0	448.0	0.0406
	B	976.0	30.50	576.0				
Incorrectness	A	1052.0	32.88	500.0	0.0	0.0–0.0	500.0	0.8062
	B	1028.0	32.13	524.0				

feel any significant time pressure to complete the tasks. Moreover, subjects are accustomed to performing lab exercises that exceed 1.5 h.

The experiment was conducted under the context of a voluntary lab exercise, which raises the issue of self-selection. This issue can only be fully eliminated if the exercises were made to be mandatory. However, the lab exercise was made voluntary in order not to change the learning value the subjects were originally intended to receive by the course. Moreover, mandating the lab exercise will raise morality threats. Subjects with low academic standing may not be performing their experimental tasks to the best of their abilities due to disinterest. In this experiment, morality threats were mitigated on two fronts: (a) Firstly, subjects participated on voluntary basis, which is an indication that the subjects were self-motivated to learn from and participate in this experiment. (b) Secondly, the grading policy at King Fahd University of Petroleum and Minerals states that the final grade that a student receives is based on their performance in comparison to the performance of their classmates. Therefore, subjects who participated in this experiment will be interested to perform to the best of their abilities to gain a competitive advantage over their peers.

4.2.3. *Construct validity*

This experiment was designed to minimize the construct validity of the dependent variables. The subjects were evenly and randomly distributed into two groups. A traditional 2 × 2 fractional factorial design was deployed in order to minimize the effects of individual capabilities, system differences and ordering effects. Bias toward using SMCD or UNL with respect to the systems used was

Table 15
Cliff's delta for all quality attributes.

	Cliff's delta ($\hat{\delta}$)	Variance	Confidence interval around delta ($\hat{\delta}$)	
			Maximum	Minimum
Inconsistencies	0.072	0.022	0.344	−0.210
Misplacements	0.125	0.003	0.239	0.008
Incorrectness	0.023	0.009	0.208	−0.163

eliminated by using RDs of two misuse case models provided by two different authors whom have no connection with this experiment.

In this experiment, all defects were weighted equally. Although it can certainly be argued that different defects have different levels of importance, there lacks empirical evidence that allows us to safely quantify defects differently.

4.2.4. External validity

As usually is the case in experiments that utilize students as subjects, it is unsafe to generalize these results to industry professionals. This conservative view is due to the fact that professionals in general have richer technical experience which improves their ability to learn and apply new methods and technologies in comparison to students. However, in other software engineering related empirical studies, it was found that the difference between students and professional is not always clear cut (Höst et al., 2000; Arisholm and Sjøberg, 2003). In fact, it has been argued that students are better representatives in controlled experiments since professionals have a tendency to stray away from the techniques that they were prescribed and resorting to techniques they have learned from prior experiences in industry (Arisholm et al., 2006).

A common reason for not generalizing the results of software engineering related experiments to industry is the size of the artifacts utilized in the experiments. It is often the case that the size of artifacts considered in an experiment are much simpler than artifacts used in full-scale industrial settings. However, in the case of misuse case modeling, the largest industrial misuse case models available in the literature were used in this experiment. It may be the case that the average size of misuse case models used in industrial settings are much larger than those used in this experiment, however there lacks any evidence of that in the literature.

5. Related work

There have been a number of empirical studies conducted that involve misuse cases. Karpati et al. (2012) presented a controlled experiment to compare the efficiency of using misuse cases and mal-activity diagrams (Sindre, 2007) for modeling social engineering attacks. The results show that both techniques are relatively similar in efficiency. The only significant difference observed is the perceived ease of use of mal-activity diagrams (Karpati et al., 2012).

A subject-based empirical comparison of Secure Tropos and misuse cases was presented in Pan (2012). The results of the experiment indicate that misuse cases and Secure Tropos (Yu, 2001) are complementary techniques. Secure Tropos being a goal based modeling approach, it is used to model and analyze security constraints as the attainment of security goals. Meanwhile, misuse case modeling is a problem-based approach that uses a notational set based on the UML to identify threats and mitigation points.

An experimental comparison of two approaches, one that uses a combination of use case diagrams and FMEA (Failure Mode and Effect Analysis) tables, and another that uses misuse case models, was presented in Stålhane and Sindre (2007). The purpose behind the experiment was to determine which technique found the most hazards. The results of the experiment showed that misuse case modeling performed better than a combination of use cases and FMEA. Another experiment was conducted to compare the use of textual misuse case descriptions with misuse case diagrams (Stålhane and Sindre, 2008) for the same purpose as with the experiment presented in Stålhane and Sindre (2007). The results of the experiment showed that misuse case textual descriptions performed better than misuse case diagrams. An experiment comparing textual misuse case descriptions and sequence diagrams was presented in Stålhane et al. (2010). The results of the experiment indicate that misuse cases were better for identifying hazards resulting from the system operations. On the other hand, sequence diagrams were better for identifying system-internal hazards.

To the best of the author's knowledge there has been no specific empirical study that evaluates the effect of applying a treatment to the overall quality of misuse case models such as is the case with the experiment presented in this paper. However, there have been a number of empirical studies conducted that investigated the impact of applying certain devised technique on the quality of use case models.

A reading technique called TUCCA was presented in Belgamo et al. (2005). The TUCCA technique is concerned with reviewing the requirements document to find defects. A subject-based experiment was conducted to compare the effectiveness of using TUCCA with checklist-based and perspective-based reading techniques for finding defects. The results of the experiment shows that TUCCA has allowed the subjects to detect more defects than when they used a checklist-based or a perspective-based reading technique. Anda and Sjøberg (2002) presented an inspection technique to find defects in use case models. Two case studies were conducted using undergraduate Software Engineering students. The case studies reveal that using the inspection technique allowed the students to detect more defects in use case models than using an ad-hoc technique (Anda et al., 2001).

An experiment was conducted to evaluate the effect of two sets of guidelines with respect to authoring use case descriptions; styling and content guidelines (Ben Achour et al., 1999). The styling guidelines are used to improve the structural aspect of a use case model, while content guidelines are used to ensure that all the necessary contents are stated in the descriptions. The results indicate that the overall quality of use case models is improved when using these guidelines.

Cox and Phalp (2000) conducted an experiment to evaluate the effectiveness of applying the CREWS guidelines to improve the quality of use case models. It was concluded from that experiment that the CREWS guidelines were undoubtedly effective, however, they were not always easy to understand and use.

6. Conclusion

This paper presents a subject-based controlled experiment that mainly explores the research question of whether SMCD can be effectively used to produce consistent misuse case models. Other research questions were posed by this experiment to evaluate the effect of using SMCD on the overall quality of the developed misuse case models.

The experiment presented in this paper was conducted in the context of a voluntary laboratory exercise but for bonus marks toward the final grade of undergraduate software engineering students. The subjects applied two treatments; SMCD and UNL, to two distinct systems. The results indicate a statistically significant improvement with respect to the consistency levels of the developed misuse case models favoring the use of SMCD. There were no other statistically significant results observed to support any possible conjecture. The experiment yields an important conclusion with respect to using SMCD given that consistency, solely, is a highly sought after quality attribute in misuse case models.

When the experiment was concluded, it was found that all subjects finished within the expected time frame, indicating that the subjects did not face any time pressures given that they finished their experimental tasks in roughly half the allowed time. Moreover, there was no significant difference between the times subjects required to complete their tasks while using SMCD and UNL. This eliminates the possibility that using SMCD may be much more time consuming than using UNL. Informal post-interview revealed that subjects found SMCD easy to learn and apply, and that its deployment did not pose a burden to their authoring efforts. Questions by the subjects during the experiment were mainly regarding the business cases of the two systems. A minority of the questions were regarding the application of SMCD or UNL. In general, there were no obvious problems observed during the experiment.

Misuse case modeling is one of the most popular techniques to elicit and model functional security requirements (Crook et al., 2002). The validity of the approach and some success stories from the industry has been reported [51, den Barber, 29, Breivik, 1]. Consequently, misuse case modeling has become the focus of many research works where the extensions to the modeling notation has been proposed (Dwaikat and Parisi-Presicce, 2004; Herrmann and Paech, 2005; Pauli and Xu, 2005; Røstad, 2006). Although not as mainstream as use case modeling, the work presented in El-Attar and Miller (2008) takes a proactive approach toward improving quality in misuse case models as the technique is expected to become more popular. It is therefore important to empirically evaluate any promising techniques that can help improve quality in misuse case models. It is also important to empirically evaluate the effect of high quality misuse case models upon other aspects of the development process. For example, how does a higher quality misuse case model improve the coverage of tests cases generated? And how does it improve the ease of maintenance efforts? Such questions and many others are worth investigating empirically. Hopefully the experiment presented in this paper provides practical guidelines for future research to evaluate other alternatives.

Acknowledgments

We would like to thank all the students who participated in this experiment. We would also like to thank the independent researchers who volunteered to evaluate the misuse case descriptions. We would also like to thank the authors of Lehtonen et al. (2007) and Regev et al. (2005) for allowing us to use their misuse case models in this paper. We would also like to thank the IEEE Systems Journal and the Business Process Management Journal for allowing us to use the misuse case diagrams presented in this paper. The author would like to acknowledge the support provided by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum and Minerals (KFUPM) for funding this work through project No. IN111028.

References

Alexander, I.F., 2003. Misuse cases, use cases with hostile intent. IEEE Software 20 (1), 58–66.

Alexander, I.F., 2002. Initial industrial experience of misuse cases in trade-off analysis. In: Proceedings of the 10th Anniversary IEEE International Requirements Engineering Conference (RE'02), Essen, Germany.

Ambler, S., 2010. Agile Modeling Website, Available at: http://www.agilemodeling.com/essays/whenIsAModelAgile.htm (last accessed October).

Anda, B., Sjøberg, D., Jørgensen, M., 2001. Quality and understandability in use case models. In: Lindskov Knudsen, J (Ed.), 15th European Conference Object-Oriented Programming (ECOOP). Springer, Budapest, pp. 402–428.

Anda, B., Sjøberg, D.I.K., 2002. Towards an inspection technique for use case models. In: Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, pp. 127–134.

Armour, F., Miller, G., 2000. Advanced Use Case Modeling. Addison-Wesley.

Arisholm, E., Briand, L., Hove, S., Labiche, Y., 2006. "The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation". IEEE Transaction on Software Engineering 32, 365–381.

Arisholm, E., Sjøberg, D., Technical Report 2003-6 2003. A Controlled Experiment with Professionals to Evaluate the Effect of a Delegated versus Centralized Control Style on the Maintainability of Object-Oriented Software. Simula Research Laboratory http://www.simula.no/

Belgamo, A., Fabbri, S., Maldonado, J.C., 2005. TUCCA: improving the effectiveness of use case construction and requirement analysis. In: International Symposium on Empirical Software Engineering, pp. 266–275.

Ben Achour, C., Rolland, C., Maiden, N.A.M., Souveyet, C.,1999. Guiding use case authoring: results of an empirical study. In: Proceedings IEEE Symposium on Requirements Engineering. IEEE Comput. Soc., Los Alamitos.

Bittner, K., Spence, I., 2002. Use Case Modeling. Addison-Wesley, MA.

Booch, G., Rumbaugh, J., Jacobson, I., 2003. The Unified Modeling Language User Guide, 2nd ed. Addison-Wesley.

Butler, G., Xu, L., 2001. Cascaded refactoring for framework evolution. In: Proceedings of 2001 Symposium on Software Reusability, ACM Press, pp. 51–57.

Chandrasekaran, P., 1997. How use case modeling policies have affected the success of various projects (or how to improve use case modeling). In: Addendum to the ACMSIGPLAN Conference on Object-oriented Programming, Systems, Languages, and Applications.

Cliff, N., 1993. Dominance statistics: ordinal analyses to answer ordinal questions. Psychological Bulletin 114 (3), 494–509.

Cliff, N., 1996a. Answering ordinal questions with ordinal data using ordinal statistics. Multivariate Behavioral Research 31 (3), 331–350.

Cliff, N., 1996b. Ordinal Methods for Behavioral Data Analysis. Lawrence Erlbaum Associates.

Cockburn, A., 2001. Writing Effective Use Cases. Addison-Wesley, Boston.

Cockburn, A., 1997. Goals and use cases. Journal of Object-Oriented Programming 10 (5), 35–40.

Constantine, L.L., 1995. Essential modeling: use cases for user interfaces. ACM Interactions 2 (2), 34–46.

Cox, K., Phalp, K., 2000. Replicating the CREWS use case authoring guidelines experiment. Journal of Empirical Software Engineering 5 (3), 245–267.

Crook, R., Ince, D., Lin, L., Nuseibeh, B., 2002. Security requirements engineering: when anti-requirements hit the fan. In: IEEE Joint International Conference on Requirements Engineering, pp. 8–13.

Diallo, M.H., Romero-Mariona, J., et al., 2006. A comparative evaluation of three approaches to specifying security requirement. In: REFSQ'06, Luxembourg.

Dubois, E., Wu, S., 1996. A framework for dealing with and specifying security requirements in information systems. In: Information Systems Security. Chapman & Hall, London, UK, pp. 88–99.

Dwaikat, Z., Parisi-Presicce, F., 2004. From misuse cases to collaboration diagrams in UML. In: 3rd International Workshop on Critical Systems Development with UML.

Ekremsvik, S., Tiset, E.M., 2004. Misbrukstilfeller – utprøving i praksis og evaluering av teknikk. Department of Computer and Information Science, Trondheim, Norway, NTNU, Master.

El-Attar, M., Miller, J., 2008. Producing robust use case diagrams via reverse engineering of use case descriptions. Journal of Software and Systems Modeling 7 (1), 67–83.

El-Attar, M., 2012. Towards developing consistent misuse case models. Journal of Systems and Software 85 (2), 323–339.

Herrmann, A., Paech, B., 2005. Quality misuse. In: International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 180–186.

Hess, M.R., Kromrey, J.D., Ferron, J.M., Hogarty, K.Y., Hines, C.V., 2005. Robust inference in meta-analysis: an empirical comparison of point and interval estimates using the standardized mean difference and cliff's delta. In: Annual meeting of the American Educational Research Association, p. 36, Available at: www.coedu.usf.edu/main/departments/me/documents/RobustMeta-AnalysisAERA2005.pdf (last accessed July 2013).

Höst, M., Regnell, B., Wohlin, C., 2000. "Using Students as Subjects – A Comparative Study of Students and Professionals in Lead-Time Impact Assessment". Empirical Software Engineering 5 (Nov), 210–214.

Jacobson, I., Ericsson, M., Jacobson, A., 1995. The Object Advantage. ACM Press.

Jacobson, I., et al., 1992. Object-Oriented Software Engineering. A Use Case Driven Approach. Addison-Wesley.

Jürjens, J., 2004. Secure Systems Development with UML. Springer, Berlin.

Kamalrudin, M., Hosking, J., Grundy, J., 2011. Improving requirements quality using essential use case interaction patterns. In: Proceedings of the 33rd International Conference on Software Engineering, pp. 531–540.

Karpati, P., Sindre, G., Matulevicius, R., 2012. Comparing misuse case and mal-activity diagrams for modelling social engineering attacks. International Journal of Secure Software Engineering 3 (2), 54–73.

Kromrey, J., Hogarty, K., Ferron, J., Hines, C., Hess, M., 2005. Robustness in meta-analysis: an empirical comparison of point and interval estimates of standardized mean differences and cliff's delta. In: American Statistical Association 2005 Joint Statistical Meetings, p. 7, Available at: http://luna.cas.usf.edu/~mbrannic/files/meta/Robust%20Estimates.pdf (last accessed July 2013).

Kromrey, J., Hogarty, K., 1998. Analysis options for testing group differences on ordered categorical variables: an empirical investigation of Type 1 error control and statistical power. Multiple Linear Regression Viewpoints 25, 70–82.

Kruchten, P., 1999. The Rational Unified Process: An Introduction, 2nd ed. Addison Wesley Longman Inc.

Kulak, D., Guiney, E., 2000. Use Cases: Requirements in Context. Addison-Wesley.

Lilly, S.,1999. Use case pitfalls: top 10 problems from real projects using use cases. In: Proceedings of TOOLS USA'99. IEEE Computer Society.

Lehmann, E.L., 1998. Non-Parametrics: Statistical Methods Based On Ranks. Revised. Pearson.

Lehtonen, M., Michahelles, F., Fleisch, E., 2007. Trust security in RFID-based product authentication systems. IEEE Systems Journal 1 (2), 129–144.

Mæhre, M., 2005. Industrial Experiences with Misuse Cases. Department of Computer and Information Science, Trondheim, Norway, NTNU, Master.

McCoy, J., 2003. Requirements use case tool (RUT). In: Companion of the 18th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, pp. 104–105.

OMG, 2003. Unified Modeling Language, version 1.5. Object Management Group, Inc. http://www.uml.org (cited 21.11.03).
Pan, Y., 2012. Comparison of i*-based and use case-based security modelling initiatives for software requirements engineering: an empirical comparison of secure tropos and misuse cases. Norwegian University of Science and Technology, Master of Information Science Thesis.
Pauli, J.J., Xu, D., 2005. Misuse case-based design and analysis of secure software architecture. In: International Conference on Information Technology: Coding and Computing, pp. 398–403.
Regev, G., Alexander, I.F., Wegmann, A., 2005. Modelling the regulative role of business processes with use and misuse cases. Business Process Management Journal 11 (6), 695–708.
Røstad, L., 2006. An extended misuse case notation, including vulnerabilities and the insider threat. In: Proceedings of the 12th International Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 33–43.
Shapiro, S.S., Wilk, M.B., 1972. An analysis of variance test for the exponential distribution. TechnoMetrics 14, 355–370.
Siegel, S., Castellan Jr., N.J., 1988. Non-Parametric Statistics for the Behavioral Sciences, 2nd ed. McGraw-Hill.
Sindre, G., Opdahl, A., 2005. Eliciting security requirements with misuse cases. Requirements Engineering Journal 10, 34–44.
Sindre, G., Opdahl, A.L., Breivik, G.F., 2002. Generalization/specialization as a structuring mechanism for misuse cases. In: Proceedings of the 2nd Symposium on Requirements Engineering for Information Security (SREIS'02), pp. 1–16.
Sindre, G., Opdahl, A.L., 2001. Templates for misuse case description. In: Proceedings of the 7th International Workshop on Requirements Engineering: Foundation.
Sindre, G., 2007. Mal-activity diagrams for capturing attacks on business processes. In: Proceedings of the 13th international Working Conference on Requirements Engineering: Foundation for Software Quality, pp. 355–366.
Stålhane, T., Sindre, G., 2007. A comparison of two approaches to safety analysis based on use cases. In: Proceedings of ER'07, Auckland, 6–9 November, 2007, Springer LNCS 4801.
Stålhane, T., Sindre, G.,2008. Safety hazard identification by misuse cases: experimental comparison of text and diagrams. In: Model Driven Engineering Languages and Systems, Lecture Notes in Computer Science 5301. Springer, Heidelberg, pp. 721–735.
Stålhane, T., Sindre, G., du Bousquet, L., 2010. Comparing safety analysis based on sequence diagrams and textual use cases. Advanced Information Systems Engineering Lecture Notes in Computer Science 6051, 165–179.
Wohlin, C., Runeson, P., Host, M., Ohlsson, M.C., Regnell, B., Wesslen, A., 2000. Experimentation in Software Engineering – An Introduction. Kluwer.
Yu, E., 2001. Agent-oriented modelling: software versus the world. In: Proceedings of the Agent-Oriented Software Engineering AOSE-2001Workshop, pp. 206–225.

Mohamed El-Attar received his B.Eng. degree from Carleton University, Canada, and his Ph.D. degree from the University of Alberta, Canada. In 2009, he joined the department of Information and Computer Science at King Fahd University of Petroleum and Minerals, Saudi Arabia, as an assistant professor. His research interests include Requirements Engineering, in particular with UML and use cases, objectoriented analysis and design, model transformation and empirical studies. For information about his research see: http://faculty.kfupm.edu.sa/ICS/melattar/index.html.