Information and Software Technology 66 (2015) 73–96

Contents lists available at ScienceDirect

Information and Software Technology

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

A scientific evaluation of the misuse case diagrams visual syntax

Faisal Saleh [a], Mohamed El-Attar [a,b,*]

[a] Information and Computer Science Department, King Fahd University of Petroleum and Minerals, P.O. Box 5066, Dhahran 31261, Saudi Arabia
[b] Computer Science and Engineering Department, Mississippi State University, Butler Hall, 665 George Perry St., Box 9637, MS 39762, USA

ARTICLE INFO

Article history:
Received 13 September 2014
Received in revised form 14 April 2015
Accepted 15 May 2015
Available online 1 June 2015

Keywords:
Misuse cases diagrams
Visual syntax
Cognitive evaluation

ABSTRACT

Context: Misuse case modeling is a well-known technique in the domain of capturing and specifying functional security requirements. Misuse case modeling provides a mechanism for security analysts to consider and account for security requirements in the early stages of a development process instead of relying on generic defensive mechanisms that are augmented to software systems towards the latter stages of development.
Objective: Many research contributions in the area of misuse case modeling have been devoted to extending the notation to increase its coverage of additional security related semantics. However, there lacks research that evaluates the perception of misuse case models by its readers. A misread or misinterpreted misuse case model can have dire consequences downstream leading to the development of an insecure system.
Method: This paper presents an assessment of the design of the original misuse case modeling notation based on the Physics of Notations framework. A number of improvements to the notation were suggested. A survey and a controlled experiment were carried out to compare the cognitive effectiveness of the new notation in comparison to the original notation.
Results: The survey had 55 participants for have mostly indicated that the new notation is more semantically transparent than the original notation. The results of the experiment show that subjects reading diagrams developed using the new notation performed their tasks an average 6 min quicker, while in general the subjects performed their tasks in approximately 14.5 min. The experimental tasks only required subjects reading diagrams and not creating them.
Conclusion: The main finding of this paper is that the use of colors and icons has improved the readability of misuse case diagrams. Software engineering notations are usually black and white. It is expected that the readability of other software notations will improve if they utilize colors and icons.
© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The complexity of software systems is constantly increasing. Security in such systems is an indispensible key quality attribute. Systems must have the ability to detect potential threats and act accordingly while remaining available to their legitimate users. There is a multitude of threat sources and types of attacks that can occur. Attacks can be orchestrated by humans; whether they are outside (illegitimate) users or insiders with authorized access. Attacks can also be launched by other systems. In either case, systems are expected to keep performing despite of these threats.

Conventional software development methodologies are mainly geared towards completing the development of the business functionality first. Security is usually addressed using generic defensive mechanisms that are "patched-on" downstream in the development process. Such conventional development methodologies are no longer sufficient causing systems to be vulnerable to attacks [13]. The advent of secure software engineering methodologies has allowed developers to overcome the security shortfalls of traditional development methodologies. Secure software engineering entails that security concerns should be accounted for as early as the requirements phase [25]. To this end, many popular requirements and design modeling techniques were extended to specify security aspects. For example, use case models [40] were extended into misuse case models [53], use case maps [7] were extended into misuse case maps [26]; activity diagrams [40] were extended into mal-activity [52] diagrams, etc. The focal security modeling technique considered in this paper is misuse case modeling. The

* Corresponding author at: Information and Computer Science Department, King Fahd University of Petroleum and Minerals, P.O. Box 5066, Dhahran 31261, Saudi Arabia.
E-mail addresses: faisal86@me.com (F. Saleh), melattar@kfupm.edu.sa, melattar@cse.msstate.edu (M. El-Attar).

http://dx.doi.org/10.1016/j.infsof.2015.05.002
0950-5849/© 2015 Elsevier B.V. All rights reserved.

purpose of misuse case modeling is to specify functional security requirements of systems. Ever since the technique was introduced more than a decade ago, it has become one of the more popular security requirements modeling techniques [45]. A misuse case model consists of a diagram and a set of corresponding textual descriptions, similar to use case models. A misuse case diagram visually presents use cases, misuse cases, actors, misusers and various relationships between these entities. While the textual descriptions elaborate expected behavior of a system. The misuse case diagram plays a vital role in communicating the functional security requirements visually to various stakeholders.

1.1. Motivation

The visual communication aspect of modeling is crucial. Modeling is used to convey a mental model from the modeler to a reader. If the reader of a model constructs a different mental model than that intended by the modeler, then the whole modeling exercise has failed. However, this critical aspect is often overlooked in software engineering notations, and misuse case models are no exception. Many software engineering modeling techniques are mainly focused on the range of semantic constructs they can convey, the more the merrier. While empowering the semantic modeling capabilities of notations is desirable, notations also need to be cognitively effective [38]. Cognitive effectiveness in the software engineering notations domain refers to "*the speed, ease and accuracy with which a representation can be processed by the human mind*" [38]. However, notation design has constantly been made subjectively without providing any insight into the design process [36,30,24]. While ostensibly an issue of aesthetics and taste, notation design can greatly affect its cognitive effectiveness. Therefore, it is arguably as equally as important to focus on the cognitive effectiveness of notations as it is to focus on their semantic modeling capabilities. A misread or misinterpreted use case model can lead to the development of a system that does not satisfy its functional requirements. Similarly, a misread or misinterpreted misuse case diagram may lead to the development of an insecure system, likely rendering it useless.

Perhaps this important stream of research has been overlooked due to a lack of a theoretical basis to evaluate and design notations. In 2009, Moody has introduced the "Physics of Notations" (PoN) [36] to help overcome this limitation. The PoN outlines nine principles that can be used to evaluate the cognitive effectiveness of the visual syntax of notations. The PoN principles can also be used as a basis to design and improve cognitively effective notation. PoN principles are derived from theory and evidence from multiple fields, in particular the cognitive science field. These principles focus on the visual aspects of notation design rather than semantic aspects. To this end, the following two research questions have been formulated:

RQ1: What are the sub-optimal design issues that exist?
RQ2: What improvements can be made to current misuse case modeling notation in enhance its cognitive effectiveness.

1.2. Approach

In this paper, we use the PoN principles to evaluate the misuse case modeling notation. This paper also presents a number of improvements to the misuse case modeling notation as a result of the evaluation. The suggested improvements are also in line with the PoN principles. Two empirical user-studies are conducted to validate the suggested improvements. One user study was conducted using software engineering professional and academics, while the other was conducted using software engineering students.

1.3. Potential benefits

Improving the cognitive effectiveness of misuse case diagrams will make them easier and quicker to read, while committing fewer reading mistakes. Another benefit is that the new notation will reduce the learning curve required to read and communicate misuse case diagrams. Ultimately, improving the cognitive effectiveness of misuse case diagrams will improve a neglected dimension of quality of model-driven secure software engineering. Improving the cognitive effectiveness of misuse case diagrams should have a positive effect on all development activities given misuse case diagrams are created and used at the requirements. Improving the cognitive effectiveness of misuse case diagrams should enhance the quality of the end-products that are developed.

The rest of the paper is organized as follows: Section 2 presents a brief background of misuse case modeling and a review of the related literature. Section 3 presents the systematic evaluation results of the misuse case modeling visual syntax. Suggestions to improve the notation are shown in Section 4. Section 5 presents the first of two user studies while Section 6 presents the second user study. Finally, Section 7 concludes and suggests future work.

2. A brief background of misuse case modeling

Use case modeling has been extended to introduce four new security related concepts: *misuse cases, misusers, threaten relationship* and *mitigate relationship*. Sindre and Opdahl formally defined the concepts of a misuse case and misuser in their ensuing work as follows:

- *Misuse Case* – "*a sequence of actions, including variants, that a system or other entity can perform, interacting with misusers of the entity and causing harm to some stakeholder if the sequence is allowed to complete*" [53].
- *Misuser* – "*an actor that initiates misuse cases, either intentionally or inadvertently*" [53].

The semantics of a misuse case is analogous to a use case and a misuser to actor, but inverse. The inverted color scheme is symbolic of these analogous yet inverse relationships between these concepts. The notation legend for misuse case models is show in Fig. 1. The threatens relationship can only be directed from a misuse case to a use case. Conversely, the mitigates relationships can only directed from a use case to a misuse case. The threatens relationship is used to specify a threat to the integrity of a system while a particular use case is being performed. Meanwhile, the

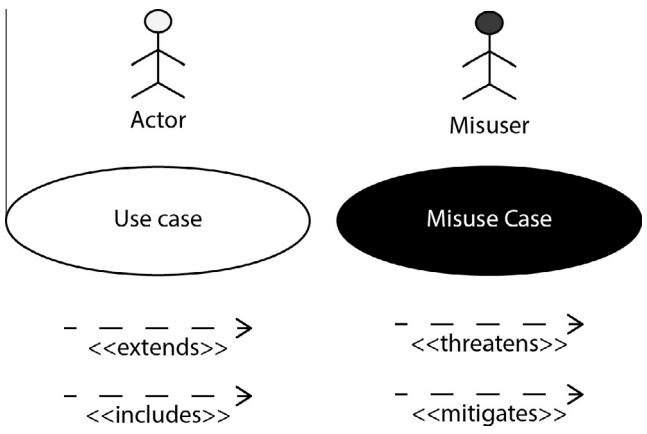

Fig. 1. Misuse case notation legend.

Download English Version:

https://daneshyari.com/en/article/550938

Download Persian Version:

https://daneshyari.com/article/550938

Daneshyari.com