Using Misuse cases with non-quality requirements

Antti Holopainen

Abstract—
Misuse cases are like use cases, but with bad intent. They can be used when defining non-quality attributes. This paper describes the main concepts concerning misuse cases and shows ways to present them. A literature study is performed to find answers to research questions.

Index Terms—

1. INTRODUCTION

USE cases are a descriptions of how end-users will use a software code. They are mostly used in gathering requirements quite early in the software development process. They define how software should work in the user perspective. (Cockburn 2001)

Misuse cases describe a sequence when user does something that he is normally not supposed to do, either on purpose trying to break the system or by accident when making an error. Some research has been made about the topic but misuse cases are not yet in wide use in software development.

2. RESEARCH OBJECTIVE

The objective of this research is to study the benefits of using misuse cases in software development. This research also tries to find areas and quality attributes where using misuse cases are most useful and to find experiences when using misuse cases in the real industry and in software development.

3. RESEARCH QUESTIONS

RQ1 what are misuse cases?
RQ2 for what can misuse cases be used in?
RQ3 Are there results of using misuse cases in the software industry?
RQ4 what are the areas or quality attributes that misuse cases can be used with?

4. RESEARCH SCOPE

The scope of this study is within the research done and reported by various authors and researchers about misuse cases and usage of misuse cases in software development. If good examples are found, other fields than software can be used as a reference too.

5. RESEARCH METHODS AND RESULTS

The research questions are answered by performing a literature survey and analysis. References were selected with keyword queries from article databases and more references added from the reference lists of found documents. Best documents were selected and studied more careful.

As a result this study process gathers results from different articles and studies about the research questions and answers them.

6. WHAT ARE MISUSE CASES

6.1 Use Cases

A use case is an organized branching structure of steps (activities taken by named roles) in time-sequence, together with supporting information such as preconditions, defining both normal and exceptional sequences of events, to achieve a named goal. (Cockburn 2001)

6.2 Misuse cases

Sindre and Opdahl (Sindre, Opdal, 2005) propose a way to modify the use case diagrams by adding negative use cases – misuse cases – that specify not-wanted behavior. The authors define misuse case and misusers in line with the UML definitions use case and actor. They also show two new relationships between use cases and misuse cases. New definitions are presented below.
6.2.1 Misuse case

“Misuse case is 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.”

6.2.2 Misuser

“Misuser is an actor that initiates misuse cases, either intentionally or inadvertently.”

6.3 Relationships

Two new relationships are also introduced; “mitigate” and “threaten”. A use case can mitigate a misuse case and a misuse case can threaten a use case. Standard UML relationships like extends and includes are also used with misuse cases.

6.3.1 Threaten-relationship

When a use case is exploited or hindered by a misuse case, the misuse case is said to threaten the use case. “Misuse case A threatens use case B if achieving the goal of A reduces the system’s ability to achieve the goal of B” (Alexander 2003).

6.3.2 Mitigate-relationship

When a use case mitigates a misuse case, the use case is used as a countermeasure against a misuse case. Use case A mitigates misuse case B if it reduces B’s effect on the use cases that it threatens.” (Alexander 2003) Mitigate is defined as “lessen or to try to lessen the seriousness or extent of” (Cognitive Science Laboratory of Princeton University, WordNet). Mitigating use cases are sometimes called security use cases and they often extend a use case which is being threatened.

6.3.3 Examples

For example a misuse case “Steal the car” threatens the use case “Drive the car” since if the misuse case succeeds and the car is stolen, the good actor cannot drive the car.

The misuse case “steal the car” can be mitigated with a use case “Lock the car”, when that use case succeeds and car is locked, it is harder to steal the car and achieve the goal of the misuse case. (See fig.1)

One misuse case can threaten many use cases and a use case can mitigate many misuse cases. If a misuse case is “Cause denial of service of a system”, achieving the goal of that misuse case threatens basically all use cases the system has been designed to implement. A mitigating use case “Verify input data” can mitigate misuse cases like “Use SQL injection to break integrity of data” or “cause a buffer overflow”. See figure 3 for a larger network of use and misuse cases.

Microsoft introduces a similar concept like misuse cases, threat modeling (Meier, Mackman, Wastell). They also try to find threats with a goal-based approach considering the goals of the attacker. Threats are also identified along use cases by thinking how a use case can be maliciously or unintentionally being abused. At the end, mitigating use cases and functions must be planned to avoid the threats or at least lessen their probability to occur.

<table>
<thead>
<tr>
<th>Source Case</th>
<th>Target Case</th>
<th>Mitigates</th>
<th>Includes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case type</td>
<td>Use</td>
<td>Misuse</td>
<td></td>
</tr>
<tr>
<td>Use</td>
<td>Includes</td>
<td>Threatens</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Creation of relationships between use and misuse cases (Alexander 2003)

7. HOW TO PRESENT MISUSE CASES

Sindre and Opdahl define the notation for misuse cases in (Sindre, Opdahl 2002)

7.1 UML-related graphical presentation

Misuse cases are not part of the UML standard, but because they resemble use cases, symbols with inverted colors are used to represent them. Mitigate and threaten-relationships are added compared to traditional graphs. Threatening misuse cases and mitigating use cases are shown in single graph and their dependencies form a use case network model. Use case networks are often used and connecting use cases and misuse cases are presented in same graph. Graphs use standard UML relationships and extended relationships described earlier in this paper. Figure 3 from (Sindre, Opdahl 2002) presents a metamodel of the basic concepts and their relation to the UML metamodel.
Sindre and Opdahl provide templates for textual representation of misuse cases in (Sindre, Opdahl 2002).

7.2.1 Lightweight misuse case description
Kulak and Guiney in (Kulak, Guiney 2000) and Cockburn in (Cockburn 2001) proposed templates for use cases. Sindre and Opdahl extend them with a field called threats. Threats can also be represented as an additional column instead of additional field. In security-intensive systems the threat-field works as a warning. The designers can notice if security is being thought in the particular use case. If threat-field is missing, it is possible that the security issues have not been considered enough.

7.2.2 Extensive misuse case description
Sindre and Opdahl propose a template to present misuse cases extensively with terms like with regular use cases. With misuse cases most of the same fields are needed, but some adaptations and new fields are introduced. Fields like name, summary and author remain the same, but sequence descriptions like basic and alternative path describe the actions that the misuser(s) go trough to cause harm. They introduce many new fields like mitigation points, mitigation guarantees and related business rules. Full list can be found from in (Sindre, Opdahl 2002).

8. USING MISUSE CASES IN TRADE-OFF ANALYSIS
Alexander in (Alexander 2003) extends the relationships of misuse cases more and defines two new types for use in trade-off analysis.

Use case or misuse case A aggravates misuse case B if it increases either the probability of success or the seriousness of the damage that B threatens. Use case A conflicts with use case B if achieving A’s goal makes achieving B’s goal more difficult (or impossible), and vice versa for B’s effect on A. Figure 3 presents few examples of aggravating and conflicting use and misuse cases. The main conflict is the trade-off between strict and loose control of services and users. If control is very strict, threats from malicious misusers may be better mitigated but it also creates new misuse cases when legitimate users frustrate on the control or don’t anymore know how to use the system with all the extra functions from security use cases. So strict control aggravates misuse case “Frustrated by controls” whereas loose control aggravates misuse case “Intrude into system”.

According to Alexander, the extended set now enables the analyst to describe threats, to discover both functional and non-functional requirements, and to show the areas where costs and benefits can be expected. Analyzing the use cases needed to mitigate a misuse case and studying the conflicting use cases more information can be used for trade-off analysis.
Figure 4 (Alexander 2003)

9. AREAS WHERE MISUSE CASES CAN BE USED

Originally Sindre and Opdahl (Sindre, Opdahl 2002) used misuse cases to elicit security-related requirements. However, Alexander writes in (Alexander 2003) that also other non-functional requirements can also be handled in a similar way with use/misuse case elicitation and analysis. The idea of misuse cases is focused much on security requirements and others are forgotten. Misuse cases can aid for example in reliability, maintainability and portability requirements. Usability is also a good area to use misuse cases, using users that give wrong input by accident as misusers. Andrea Herrmann and Barbara Paech demonstrate the usage of other non-functional requirements in (Herrmann, Paech 2005). They show misuse cases for attributes like efficiency, usability (see table 2), maintainability and portability.

Paulu and Xu in (Paulu, Xu 2005) have studied a case where misuse cases were used in architectural design. The case study is about a hospital information system (HIS) and they investigate how security requirements in the form of misuse cases can be used to guide architectural design and analysis. They first designed use and misuse cases and determined the actors involved and then made sure each actor and their cases are in the architectural plan. With misuse cases they could ensure that they included also the mitigating use cases.

Table 2 Usability misuse case (Herrmann, Paech 2005)

10. EXPERIENCES OF USING MISUSE CASES

Not many references of actual use of misuse cases in software industry were found, but a university study and an example from train industry is presented. Requirement analysis has common points even in train seats and software industry. Damodaran studied misuse cases with his students with a class experiment in university of Houston-Victoria (Damodaran, 2006) about secure software development. A task was given to the students to study a presentation of an example of requirements gathering in a case concerning eBay shopping model from the customer view. They then had to develop a set of misuse cases and the mitigating security use cases related to the example case. The professor was pleased with the results and those students who finished the task and learned the technique came up with good requirements.

On a different area from software requirements, Alexander used the misuse case technique with defining security and safety requirements for a train seat. Alexander held a one-day workshop meeting of about 20 stakeholders, including project management, design engineers and quality assurance representatives. They had problems with coming up with good solutions for trade-offs between seat comfort and mitigation from hostile threats like vandals breaking the seats or ‘pests’ who seated themselves next to women, especially in seats with lacked dividing armrests. The technique and concepts used were same as when discussing software. Alexander noted that the stakeholders came up with many requirements and solutions to problems with misuse cases that they did not come up using traditional methods. They found trade-offs and used the misuse cases to analyze them. Alexander stated that “The apparent success of the workshop suggests that the techniques may be worth more detailed exploration, and application in other domains.”

11. CONCLUSIONS

Misuse cases are use cases with negative intent. They can be presented graphically like use case networks or textually with small modifications to use case presentation or more extensively. Misuse cases can be useful with many non-functional requirements, but the literature is mostly concerned about security. Misuse cases can also be used in architectural design and trade-off analysis. No major experiences from software industry have yet been found, but results from other fields are promising.

REFERENCES


Alexander, I: Modelling the Interplay of Conflicting Goals with Use and Misuse Cases Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'02)

Guttorm Sindre and Andreas L. Opdahl: Eliciting security requirements with misuse cases, Journal Requirements Engineering ISSN 0947-3602 Volume 10, Number 1 January, 2005 Pages 34-44


G Sindre, AL Opdahl - Proc. Seventh International Workshop on Requirements: Templates for Misuse Case Description

Cognitive Science Laboratory of Princeton University, WordNet http://wordnet.princeton.edu/

Meledath Damodaran: Secure software development using use cases and misuse cases Issues in Information Systems Volume VII. No. 1, 2006


Cockburn A (2001) Writing effective use cases. Addison-Wesley, Boston

Andrea Herrmann and Barbara Paech: Quality misuse Workshop (CAISE-Conference) 2005