Comparative study on threat identification techniques for dependability requirements

Christian Raspotnig
PhD student at University of Bergen
Institute for Energy Technology, OECD Halden Reactor Project
Halden, Norway
christian.raspotnig@hrp.no

***Abstract*—The elicitation of dependability requirements for dependable software systems is traditionally performed with a variety of mostly unrelated analysis techniques from distinct domains. The techniques should offer both a structured way to identify important dependability attributes of the system, and enabling creativity among the participants on identifying as many of these attributes as possible. In this paper we compare two identification techniques from the safety domain and one technique from the security domain. The aim is to look at the advantages and disadvantages of the techniques up against the preliminary identification of threats and means to sustain dependability. Another aspect is to emphasize what the safety and security domains can learn from each other with respect to identifying dependability issues for requirements elicitation.**

I. INTRODUCTION

The dependability concept **Error! Reference source not found.** is used both within the security and safety domains. Safety is one of the attributes of dependability, while security is usually not explicitly mentioned as an own attribute. Security is known to be a combination of Confidentiality, Integrity and Availability (CIA), which again are defined as attributes. Additionally, the attributes includes maintainability and reliability. The concept of dependability is divided into three related parts: Attributes which are exposed to Threats, but sustained by Means. The means are usually expressed as dependability requirements.

In the security domain, the term threat is more used than the term hazard, which is more used in the safety domain. The two domains have much in common when developing computer systems, as they have to ensure that the systems tolerate threats, do not function in an unspecified way or cause any harm to its environment. In this paper we concentrate on three techniques for identification of threats, and how they in an early stage of the development process can help identify means for the threats. This work is part of the PhD in the "ReqSec project" at University of Bergen funded by Norwegian Research Council, and as part of the ongoing research within the OECD Halden Reactor Project.

II. TECHNIQUES FOR THREAT IDENTIFICATION

A frequently used threat identification technique is the Hazard and Operability study (HAZOP) **Error! Reference source not found.**, which originates from the chemical industry in England in the 1960s, and has now been used for a long time by the safety community. The technique has been applied within different industries at a wide range of applications. The Functional Hazard Assessment (FHA) **Error! Reference source not found.** technique stems from the aerospace industry, but is also known and now widely used by the Air Traffic Management (ATM) organizations in Europe as part of the Eurocontrol Safety Assessment Methodology **Error! Reference source not found.**.

Both the HAZOP and FHA techniques are based on the use of guide words to achieve creative thinking among the participants in a threat identification brainstorming meeting, aiming at eliciting the most relevant domain knowledge relating to safety of the system. The main difference between these two techniques is that the FHA strongly promotes to identify threats on the basis of the functionality of the system, while HAZOP looks at the components of the system and the interconnections between these.

The Misuse Case technique **Error! Reference source not found.** distinguishes itself from the HAZOP and FHA technique, as it is a relatively new technique and originates from the requirements engineering community. It is closely related to the Use Case technique, as they use same graphical notation, and are directly integrated into the development process. Misuse Case has mainly been applied in the security domain, but examples of how it can be used for safety exist [4].

III. PROPOSED APPROACH

In this study the proposed approach for assessing the threat identification techniques against elicitation of dependability requirements, is to evaluate the techniques for their capability of identifying threats. In **Error! Reference source not found.** the threats are categorizes as faults, errors and failures. Further it states that a fault, when activated, can lead to an error, which again can cause a failure. A failure is defined as the event when incorrect service is delivered. The means for sustaining the dependability with respect to the threats is in **Error! Reference source not found.** defined as fault prevention, tolerance, removal and forecasting.

The dependability attributes and means in **Error! Reference source not found.** are defined as have not been

used to assess the techniques in this study. One aspect of the study is to explore what the techniques from both safety and security domains can learn of each other.

TABLE I. EVALUATION OF THREE THREAT IDENTIFICATION TECHNIQUES

Threat id. tech. vs. threats	***HAZOP***	***FHA***	***Misuse case***
Faults	Can identify faults, as the combination of guidewords and parameters stimulate this.	Does not directly stimulate the identification of faults.	Does not directly stimulate the identification of faults.
Errors	The identification of errors in components is somewhat feasible, as the guidewords can help being specific about the nature of the error (e.g. late or part of). Use of guidewords can result in leaving threats out.	The identification of errors in components is feasible, as the technique requires a decomposition of the system functions, into sub- and sub-sub-functions (i.e. a component responsible for a certain function). The use of guidewords detected and undetected corruption (commonly used within ATM) can help reveal latent errors.	The graphical notation can especially stimulate the identification of external errors, but also internal errors can be identified.
Failures	The technique with guidewords can be used at system level, identifying failures of the system. Might be limited to focus on single failures, rather than combinations.	The technique can be used on system services or system functions, identifying the failures modes. Might be limited to focus on single failures, rather than combinations.	Graphical notation is feasible for identifying how the services can be altered or fail, e.g. when there are several components which have to fail (redundancy).

Looking at the specific attributes of dependability is believed to limit this. The scope of the study has so far been narrowed into identification techniques, omitting the further analysis of how dependability can be sustain through means. This keeps the focus of the study on the early development phases of computer systems, where the elicitation of non-functional requirements is a crucial part.

For computer system the early identification of threats is critical, as the system might not useful at all if not specified and developed to deal with the threats. If the threats are discovered at later phases of the development process, the expenses of implementing means are likely to challenge the budgets.

IV. PRELIMINARY RESULTS

I shows the results of evaluating the three techniques against the threats to dependability. This evaluation is useful when comparing the three different techniques to each other. For the identification of faults, the HAZOP was the technique evaluated to be feasible for identifying faults. This is based on the combination of guidewords and parameters, which can stimulate the identification of causes for errors. The two other techniques were evaluated as not directly feasible for this.

All three techniques were assessed to be feasible for identifying errors, but the HAZOP and FHA might leave threats out, that are not related to the guidewords or functions used for identification purposes. The FHA has a benefit of stimulating to the identification of latent errors, when using guidewords as detected and undetected corruption of functions. Misuse Case uses a graphical notation, which stimulates the identification of errors in a system, due to external interference.

Both HAZOP and FHA might be limited to focus on single failures, not being able to identify how the combination of failures can affect the system. The Misuse case is feasible for this, as the multiple failures in a system can be visualized through the graphical notation. An improvement of the FHA and HAZOP would be to allow a structured way of relating failures to each other, directly identifying means for these threats.

A specific improvement of the Misuse case is to extend the technique with guidewords and parameters. At the same time the methodology should encourage to not only use guidewords, so that failures not related to any guidewords still can be identified.

V. CONCLUSION AND FUTHER WORK

The comparative study shows how the three threat identification techniques are different. The most significant difference with respect to the identification of threats is that the Misuse Case technique can visualize the multiple failures in a system. The reason for this might be the nature of security assessment, as the identification of threats often has to focus on multiple failures. This can be an input for the threat identification within the safety domain.

The further work with threat identification techniques will include a more in depth study of these and other techniques, taking into account different faults, errors and failures. It will also consider additional dependability attributes.

REFERENCES

[1] A. Avizienis, J-C. Laprie, B. Randell, Fundamental Concepts of Dependability, Report N01145, LAAS-CNRS, 2001

[2] Ericson II, C. A., Hazard Analysis Techniques for System Safety, Fredericksburg, Virginia: Wiley Interscience; 2005

[3] Eurocontrol Safety Assessment Methodology Task Force, Air Navigation System Safety Assessment Methodology, SAF.ET1.STO1.1000-MAN-01-00, Edition 2.0, April 2004

[4] G. Sindre and A. L. Opdahl, Eliciting Security Requirements with Misuse Cases, Requirements Engineering, 10 (1), 34-44, 2005

[5] I. F. Alexander, Misuse Cases Help to Elicit Non-Functional Requirements, Computing and Control Engineering Journal, 2003