Journal of Cases on Information Technology
Volume 20 • Issue 1 • January-March 2018

Combined Assessment of Software Safety and Security Requirements:
An Industrial Evaluation of the CHASSIS Method

Christian Raspotnig, ATM System Development, Avinor Air Navigation Services, Gardermoen
Peter Karpati, Institute for Energy Technology, Halden, Norway
Andreas L Opdahl, Department of Information Science and Media Studies, University of Bergen, Bergen, Norway

ABSTRACT

Safety is a fundamental concern in modern society, and security is a precondition for safety. Ensuring safety and security of complex integrated systems requires a coordinated approach that involve different stakeholder groups going beyond safety and security experts and system developers. The authors have therefore proposed CHASSIS (Combined Harm Assessment of Safety and Security for Information Systems), a method for collaborative determination of requirements for safe and secure systems. In this article, the authors evaluate CHASSIS through industrial case studies of two small-to-medium sized suppliers to the air-traffic management (ATM) sector. The results suggest that CHASSIS is easy to use, and that handling safety and security together provides benefits because techniques, information, and knowledge can be reused. The authors conclude that further exploration and development of CHASSIS is worthwhile, but that better documentation is needed—including more detailed process guidelines—to support elicitation of security and safety requirements and to systematically relate them to functional requirements.

KEYWORDS

Air Traffic Management (ATM), Case Study, Industrial Evaluation, Requirements Elicitation, Requirements Engineering, Safety, Security, Stakeholder Involvement

INTRODUCTION

Safety can be defined as resilience to unintended hazards. The goal of system safety is the protection of life, systems, equipment and the environment (Ericson, 2005). Safety is a fundamental concern in modern society, whose infrastructures for, e.g., health, welfare, energy, transport, communication and the environment have become critically dependent on the complex and tightly coupled ICT systems that support them (Perrow, 1999; Leveson, 2011). Software safety is therefore a central research problem with great industrial and societal importance. Security (Stallings & Brown, 2008) can be defined as resilience to intended threats. Security is a prerequisite for safety. Whereas safety-critical systems of the past ran in isolation on specialised software and hardware, modern systems are internetworked and based on standard technologies. In recent years, safety-critical systems in areas such as Air-Traffic

DOI: 10.4018/JCIT.2018010104

Management (ATM) have thus become increasingly exposed to security threats. Software safety and security have become central research areas of great industrial and societal importance.

New methods are therefore needed that integrate assessment of safety and security when developing software and other systems. Such new methods must take into account that modern safety- and security-critical system are complex, typically spanning both organisational boundaries and domains of expertise. Ensuring the safety and security of such systems thus requires collaboration between different stakeholder groups beyond safety and security experts and system developers. The new methods can exploit that safety and security are — to an extent — similar because they are both concerned with what a new system should not do, whereas existing methods focus on what the system should do (Raspotnig & Opdahl, 2013b).

Combined Harm Assessment of Safety and Security for Information Systems (CHASSIS) (Raspotnig et al., 2012a; 2013a) is a method for determining requirements for safe and secure systems, in particular software and information systems. The method comprises a requirements analysis process and a set of extended UML techniques, specifically *Misuse Cases (MUC)* (Sindre & Opdahl 2000, 2005; Sindre 2007), *Misuse Sequence Diagrams (MUSD)* (Katta et al., 2010), and *Failure Sequence Diagrams (FSD)* (Raspotnig & Opdahl, 2012b; 2012c). It also uses guidewords from the *hazard and operability study (HAZOP)* (Winther, 2001; Ericson, 2005) technique to identify hazards and threats, and a HAZOP table is used to collect and summarize important information about potential harm.

The purpose of this paper is to contribute towards software systems that are both safe and secure. We have therefore for the first time evaluated the feasibility, ease of use, and usefulness of CHASSIS through industrial case studies of two small-to-medium sized suppliers of software in the Air-Traffic Management (ATM) sector. We have asked whether the same basic concepts can be used to deal with both safety and security aspects; whether the CHASSIS method is easy to use; and whether the method is useful. The paper is structured as follows: Section 2 presents the background and the CHASSIS method, before Section 3 describes our research method. Section 4 presents the two case studies along with the survey data we collected. Section 5 summarises and discusses our results, before Section 6 concludes the papers and presents ideas for further work.

BACKGROUND

This section reviews existing safety and security practices along with earlier work on the CHASSIS method.

Safety and Security Methods

A broad range of methods and techniques already exist for both safety and security analysis. Examples of safety techniques are *Functional Hazard Assessment (FHA)* (Eurocontrol, 2004), *Preliminary Hazard Analysis (PHA)* (Ericson, 2005), *HAZard and OPerability (HAZOP) (*Ericson, 2005, Winther el al., 2001), *Failure Mode and Effect Analysis (FMEA)* (Stamatis, 1995), *Fault Tree Analysis (FTA)* (Ericson, 2005), *Event Tree Analysis (ETA)* (Ericson, 1999; 2005), and *Boolean-logic Driven Markov Processes (BDMP)* that models malicious and accidental scenarios in a tree structure (Pietre-Cambacedes & Bouissou, 2010). Examples of security techniques are attack trees (Schneier, 1999; 2000), threat trees (Amoroso, 1994), various adaptations of UML (Rodriguez et al., 2006, Jurjens, 2002, Lodderstedt et al., 2002), abuse cases (McDermott & Fox, 1999), misuse cases (Sindre & Opdahl, 2000; 2005), security policies (Anton & Earp, 2000), KAOS with anti-goals (Dardenne et al., 1993; van Lamsweerde, 2000; van Lamsweerde & Letier, 2004), extensions of *i** (Liu et al. 2003; Elahi, 2012), Secure Tropos (Mouratidis et al., 2005; 2007; Massacci & Zannone, 2006), abuse frames (Lin et al., 2003; 2004), security patterns (Schumacher et al., 2005) and risk-based elicitation of security requirements (Matulevicius et al., 2008; Herrmann et al., 2011).