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).

Safety and Security Methods in the ATM Domain

The ATM domain faces extensive worldwide changes through ongoing research and development programs, such as Single European Sky ATM Research (SESAR)[1] and Next Generation (NextGen)[2] in the United States. Their main goals are to create additional capacity in the airspace and to create more efficient ATM systems, of course with safety as a central concern.

The current safety process suggested for assessing ATM systems in Europe is the *Eurocontrol Safety Assessment Methodology (SAM)* (Eurocontrol, 2006), which has used the ARP 4761 standard as a basis (SAE, 1996). We will review it briefly here, because it will appear in the case studies later. SAM considers three main elements — humans, equipment and procedures — in three steps: (1) functional hazard assessment (FHA), which identifies hazards and establishes safety objectives; (2) preliminary system safety assessment (PSSA), which elicits safety requirements and apportions them to the elements of the ATM system for further analysis; and (3) system safety assessment (SSA) that collects evidence during the development, implementation and operation of the ATM system, in order to ensure and demonstrate safety (Eurocontrol, 2006). These activities require several workdays of meetings and safety analysis involving many different stakeholders.

Whereas mature safety processes have become common in the ATM domain, security processes are not equally well established. Security is the focus of recent and current research and development programs, such as work package 16.2 in the SESAR project[3], and recognised and reflected in its Security Risk Assessment Method (SecRAM) and Security Risk Management Toolkit (SRMT). Eurocontrol's ATM Security Team also investigated potential relationships between ATM security and safety. The CORAS method, which provides an extensive framework for model-driven risk analysis (Lund et al. 2011a), has also been used on an ATM case (Lund et al. 2011b) and has similarities with SecRAM.

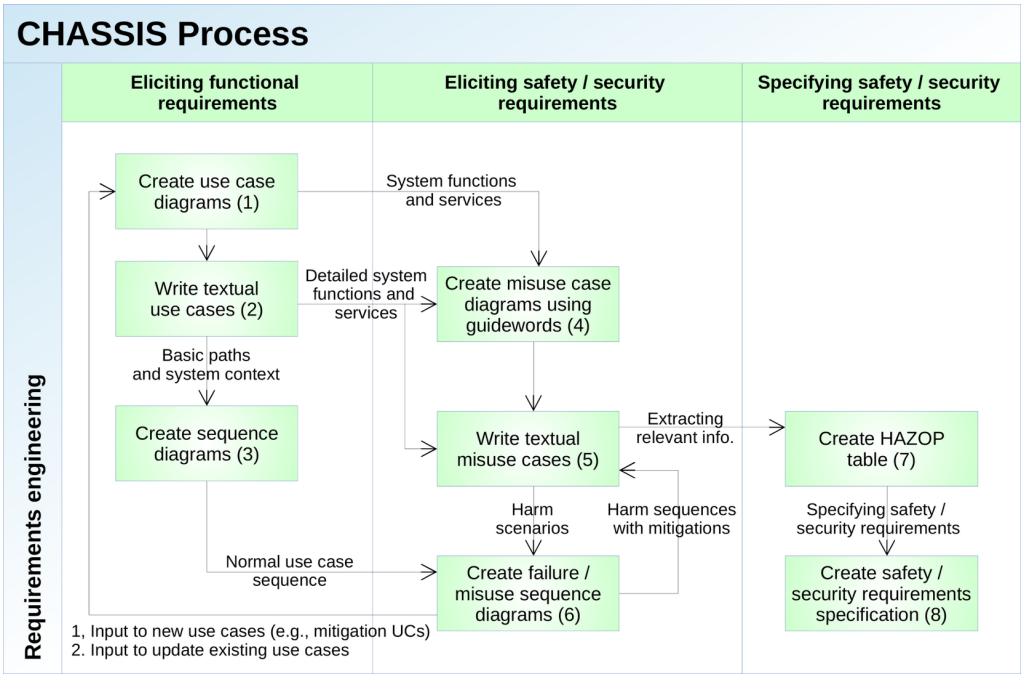
CHASSIS

The purpose of CHASSIS (Combined Harm Assessment of Safety and Security for Information Systems) is to integrate safety and security considerations with early requirements determination activities (Raspotnig et al., 2012a; 2013a). The method has been developed as part of a national industry network of small-to-medium sized ATM suppliers in response to their calls for security assessment methods for safety-critical systems[4].

CHASSIS takes functional requirements as its starting point and offers a process and a set of extended UML techniques along with HAZOP for collaboratively capturing and documenting safety and security requirements. The CHASSIS method is particularly based on diagrammatic and textual MUCs that were originally developed to elicit security requirements (Sindre & Opdahl, 2000; 2005), but which were later adapted to safety (Sindre, 2007). MUSD extends SD with a notation for representing attackers' steps against components of a system by exploiting vulnerabilities (Katta et al., 2010). FSD has instead extended SD for safety, providing a notation for representing failures of system components and how they can propagate (Raspotnig & Opdahl, 2012b). Whereas MUC uses inverted icons to distinguish between what the system should and should not do, MUSD and FSD both use colour (i.e., green versus red). MUSD and FSD were discussed for security and safety modelling in (Raspotnig & Opdahl, 2012c).

Figure 1 shows the overall CHASSIS process (Raspotnig et al., 2012a). The functional lane (Eliciting functional requirements) comprises three activities that are part of regular requirements determination: First, activity (1) creates Use Case Diagrams (D-UC), which are detailed into Textual Use Cases (T-UC) in activity (2). The basic paths and systems context from the T-UC are then used by activity (3) to create Sequence Diagrams (SD).

Figure 1. Overview of the CHASSIS process



The safety/security lane (Eliciting safety/security requirements) also has three activities, which build on the three functional ones: First, activity (4) creates Misuse Case Diagrams (D-MUC) by systematically using HAZOP guidewords on the system functions and services described in the D-UC and T-UC, in order to systematically assess each function or service against a range of deviating conditions, each described by a guideword from HAZOP, inspired by earlier work on combining them with regular UCs for security requirements analysis (Srivatanakul et al., 2004). Using HAZOP guidewords also supports creativity. The results are input to activity (5) where Textual Misuse Cases (T-MUC) are written based on the textual use cases (T-UC) and diagrammatic misuse cases (D-MUC). The resulting harm scenarios are used further in activity (6), where they are used to extend the sequence diagrams (SD) to create Failure Sequence Diagrams (FSD) for safety hazards and Misuse Sequence Diagrams (MUSD) for security threats. The FSDs and MUSDs are analysed further to identify possible mitigations to the hazards and threats, which can be fed directly back to the textual misuse cases (T-MUC) and to the overall use case diagram (D-UC) from activity (1).

The specification lane (Specifying safety/security requirements) has two activities: activity (7) extracts relevant info from the textual misuse cases (T-MUC) to create HAZOP tables extended to account for security as well as safety; activity (8) then creates a requirements specification for safety and security.

The resulting overall process is highly iterative, with the results of safety and security assessment repeatedly being fed back into functional requirements determination, incrementally making the HAZOP table in the specification lane more precise and complete. Whenever they require specialised competences, parts of the safety and security assessment can be done in separate passes through the safety/security lane. In the case studies, we will perform separate safety passes before the security passes, focussing on security-for-safety rather than on (the equally important) security-for-its-own-sake.

RESEARCH METHOD

Research Questions

The purpose of this work is to contribute towards software systems that are both safe and secure by evaluating the use of CHASSIS in industrial settings. Our main research questions are: (1) Can the same basic concepts be used to deal with both safety and security aspects? (2) Is the CHASSIS method easy to use? and (3) Is the CHASSIS method useful? We ask the first question to assess how well CHASSIS succeeds in integrating safety and security assessment with other systems development tasks. We ask the two latter questions because perceived ease of use and usefulness are central determinants of future intention to use (Davis 1989). We have detailed each main question further into more specific sub-questions as shown in Table 1. These questions have played a central role to drive and focus our study: when creating our questionnaires; when observing and collecting data about the cases; when analysing the observations, questionnaires and other data; and when discussing our findings.

Research Approach

Because CHASSIS was developed in an industrial network, we saw opportunities for evaluating the method with network partners. The safety and security areas are both highly sensitive and there is a dearth of studies that evaluate safety and security techniques in practice. To recruit partners, we offered to conduct gap analyses to benchmark their safety and security processes against standards and best-practice guidelines.[5] As part of the analysis, we used CHASSIS in collaboration with the partners, getting a rare change to evaluate our safety and security method in real industrial settings.

We chose to conduct the evaluation as a case study for the following reasons (Yin 2008): our research questions were qualitative, exploring contemporary events in a work situation, and the study was set in the participants' usual environment, over which we had limited control. The evaluation would be a multiple-case study because several network partners were involved. It would be a participative case study because we would need to involve ourselves as facilitators of the gap analyses and as experts on CHASSIS.

Table 1. Research questions and sub-questions used in the evaluation

1. Can the same basic concepts be used to deal with both safety and security aspects? a) How are the safety and security techniques similar and how do they differ? b) Can the same system models be used as a starting point for modelling both safety and security aspects? c) How are the resulting safety and security modelling processes similar and how do they differ? d) How are the resulting safety and security models similar and how do they differ?
2. Is the CHASSIS method easy to use? a) Do the participants perceive the method and its techniques as easy to use? b) Do the participants understand the method and its techniques? c) Are the participants able to distinguish safety from security aspects?
3. Is the CHASSIS method useful? a) Do the participants perceive the method and its techniques as useful? b) Does the method and its techniques encourage systematic consideration of both safety and security aspects? c) Does the method and its techniques encourage creative thinking about safety and security aspects? d) Does the method and its techniques facilitate a common understanding of the system and its safety and security aspects?

Participants

Two commercial companies from the industry network participated in our study. Both were small-to-medium suppliers of specialised computer-based systems to the ATM sector, and both were looking for better ways to systematically address security in safety-critical software development. Hence, they were motivated to take part in our study to learn about model-based safety and security assessment and about CHASSIS. We will call them the RadioSystems and AirportLights suppliers in the rest of the paper.

The RadioSystems supplier provides radio systems to ANSPs (air navigation service providers) and other ATM suppliers, depending on the complexity of the voice-communication system required by their customers. We have studied and modelled a radio system that is used by an air-traffic control officer (ATCO) to send and receive digital-voice messages to and from aircraft. The system is safety-relevant, but also has security aspects. Two employees took part with, respectively, 25 and 15 years of working experience from the IT industry. They were both experienced in system modelling and safety assessments, but had no practical experience with the hazard and operability study (HAZOP) method. They had both tried modelling with UML and conducting security assessments, but did not consider themselves experienced.

The AirportLights supplier develops computer-based and safety-relevant control and monitoring systems for airports. We have studied and modelled a system used by ATCOs to control airport lights, e.g., on runways, taxiways and the apron. It is also used to monitor the status of other systems, infrastructure and weather, such as navigation aids, electrical power supplies and wind. Three participants took part (a fourth person was also present as an observer for parts of the CHASSIS sessions), with 22, 20, and 10 years of working experience from the IT industry, respectively. They were all experienced in safety assessments, but had less practical experience with security assessments. One was experienced in HAZOP, another had tried it, and the third had only heard about it. The two former had also tried modelling with SD, whereas the third was experienced. None of the participants were experienced in modelling with UCs, although the third informant had some experience with UML.

Existing Processes

For safety, like most European ATM companies, the two suppliers were already following Eurocontrol's (2006) SAM as outlined in the Background section. They sometimes conducted its first stage, FHA, in order to establish the prerequisite for deriving the safety requirements for their products. However, the PSSA and SSA stages were considered more important, and both companies used FMEA as their main technique during these stages.

For security, also like most European ATM companies, the two suppliers were not following a formalised security process specific to the ATM domain. But they both had measures in place to avoid and handle some security threats.

Research Design

The participants used CHASSIS to assess the safety and security of one of their own software products and working in their own offices. Because both our companies were from safety-critical domains, they focused on security aspects that were related to safety, i.e., security-for-safety (or security-as-part-of-safety). In other words, our evaluation did not consider (the equally important) security for business and other purposes (security-for-its-own-sake). Because we planned to cover parts of the FHA and PSSA stages with CHASSIS, we encouraged the participants to also take the ANSP's point of view when needed for the FHA. In some situations, the first author also acted as a domain expert accounting for the ANSP's point of view.

Although the participants from the two companies worked on different systems, they otherwise proceeded in similar ways. Because the participants already knew the functional aspects of their systems well, and we wanted to focus on the safety-and-security specific parts of CHASSIS, we prepared

UCs and SDs in advance, using CHASSIS' functional lane (Eliciting Functional Requirements) to present, review, and revise them. This is not untypical of common practice, where safety and security assessments often involve models created by others. Each group of participants then performed two iterations of CHASSIS' safety/security lane (Eliciting Safety/Security Requirements) – first for safety and then for security – carrying out activities from the specification lane (Specifying Safety/Security Requirements) in parallel. We regulated how much time was spent on each activity and how the techniques were used, but also allowed flexibility to exploit the case-study setting, for example when the participants suggested to use techniques in a way we had not intended. As a result, some of the CHASSIS activities received less attention than others.

Data Collection and Analysis

Guided by our research questions, we collected data about the feasibility, ease of use, and usefulness of the CHASSIS method, combining qualitative and some quantitative methods. In order to increase repeatability and ensure that our observations and data collection were systematic and similar in both companies, we created a detailed study protocol, shown in Table 2, which we followed strictly.

Before the CHASSIS sessions, we administered a pre-session questionnaire to the participants to collect data about their prior experiences with UML and with safety and security techniques related to CHASSIS. The participants were also given an introduction to CHASSIS and to the purpose of the study.

During the sessions, the first and second authors acted as participant observers in the roles of facilitator and secretary. Each was responsible for specific clearly-defined tasks as shown in Table 2 (participants sometimes took over the facilitator role during the sessions, a realistic behaviour we encouraged). We wrote down observations during the sessions and audio-recorded all verbal communication. We also collected all the changes that were made to the UCs and SDs we had prepared in advance, and we took photos of all the drawings made during the sessions (although in the RadioSystem case, we completed the FSD and MUSD diagrams ourselves and later sent them to the participants by email for comments).

After the sessions, we conducted a post mortem with the participants immediately after the CHASSIS session to reflect upon and discuss our observations. Finally, we administered a post-session questionnaire by email to collect the participants' experiences with using CHASSIS. Their feedback will be summarised in the Results section and elaborated in an electronic addendum available at http://hdl.handle.net/1956/16161.

Data Analysis

We used the research questions in Table 1 to drive data analysis. We attempted to answer each question by iteratively going through the different types and sources of data we had collected: both observation notes, produced artefacts and audio recordings. Towards the end, we involved a fourth researcher, a colleague of the first two authors and an expert in safety and security assessments, who reviewed the transcribed notes, compared them to the collected data, and commented on the resulting answers to the research questions, i.e., researcher triangulation. This fourth researcher colleague had not taken part in the sessions, but was familiar with the CHASSIS method. Based on his comments, we extracted more data from the collection that was overlooked, solved some contradicting statements in the extracted data material and reflected more balanced on differences of safety and security D-MUCs.

RESULTS

We first present the results from the participating organisations, broadly following the CHASSIS process in Figure 1. We discuss understanding and learning specifically, before we review the results from the post-session questionnaire (results from the pre-session questionnaire about participant

Table 2. The protocol for the case studies

1. Language – ask participants to use English during the study
2. Pre-study questionnaire – collect information about the participants from each company before the gap analysis
3. Introduce the CHASSIS method and the purpose of the study
   a) Clear confidentiality aspects with all participants
   b) Explain how safety and security assessments will be the focus of the study
   c) Present CHASSIS by scope, process, methods, and techniques included
4. Collection of data during analysis sessions
   a) Acting as participating observers with the following roles:
      i. Facilitator
         A) For the safety and security assessments: lead the group through the analysis
         B) For the case studies: investigate how they see the CHASSIS method and encourage them to use each technique and complete the gap analysis of one process
      ii. Secretary
         A) For the safety and security assessments: record the rationale behind the created artefacts
         B) For the case studies: observe the participants and record how they use the method
      iii. Domain and method experts
         A) Provide domain knowledge (of ATM) when needed
         B) Provide suggestions on how to use the techniques
         C) Answer questions about the techniques
   b) Record data
      i. Audio-record all verbal communication between participants during the sessions
      ii. Written and drawn information
         A) Record changes to advance-prepared UCs and SDs
         B) Take photos of drawings made during meeting
         C) Record T-UCs and T-MUCs made
      iii. Write down observations during sessions
5. Perform post mortems to reflect upon and discuss our observations right after the sessions
6. Post-study questionnaire
   a) Email post-study questionnaire and recorded data to all participants right after sessions
   b) Collect information about the participants' experiences with the CHASSIS method

backgrounds were summarised in the previous section). A fuller account of the sessions and questionnaire is available in the electronic addendum to this paper at http://hdl.handle.net/1956/16161.

Case Studies

Preparation

Because the participants already knew the functional aspects of their systems well, and we wanted to focus on the safety-and-security specific parts of CHASSIS, and because safety and security assessments often involve models created by others, we had prepared D-UCs, T-UCs and SDs beforehand. Table 3 shows the general T-UC we created for the RadioSystems supplier.

Functional Lane

At the beginning of each session, we presented our advance-prepared D-UCs, T-UCs and SDs to the participants, asking them to review and possibly improve them, corresponding to the left lane of Figure 1. In the RadioSystems case, the D-UC and T-UC were discussed first. The participants had no specific comments to the D-UC, but suggested changes to the terms used in the more detailed T-UC. They also pointed to a missing actor (a radio mast) and the need for digital/analogue conversion at some point in the scenario. The participants did not need much facilitation in order to extend the

Table 3. T-UC describing the transmit radio message from the RadioSystems supplier

Name	Transmit radio message
Iteration	1
Summary	An ATCO is transmitting a radio message to an aircraft
Basic path	• bp1. Pushed transmit button activates radio client modulation • bp2. Radio client records information • bp2.1. Radio client transforms voice to digital signal (packets) • bp2.2. Radio client sets frequency to transmit on • bp3. Radio client sends packets to radio server • bp4. Radio server identifies frequency to send on • bp5. Radio server sends packets to correct radio • bp6. Radio converts to AM (amplitude modulated) signal and sends to the antenna
Alternative paths	• ap1. Replaces bp3,4,5: Radio client sends directly sends to radio • ap2. Replaces bp3,4,5: Has analogue interface to the channel (would be done with other boxes)
Exception paths	• ep1. Affects all bps. Failure in network, will not have any communication. Dual network.
Extension points	
Triggers	• tr1. Transmitter button pushed
Preconditions	• preC1. Setting the correct frequency • preC2. Radio channel is free for communication
Postconditions	• postC1. Radio message was sent to antenna.
Related business rules	
Authors	• Informant-1-RadioSystemsSupplier • Informant-2-RadioSystemsSupplier

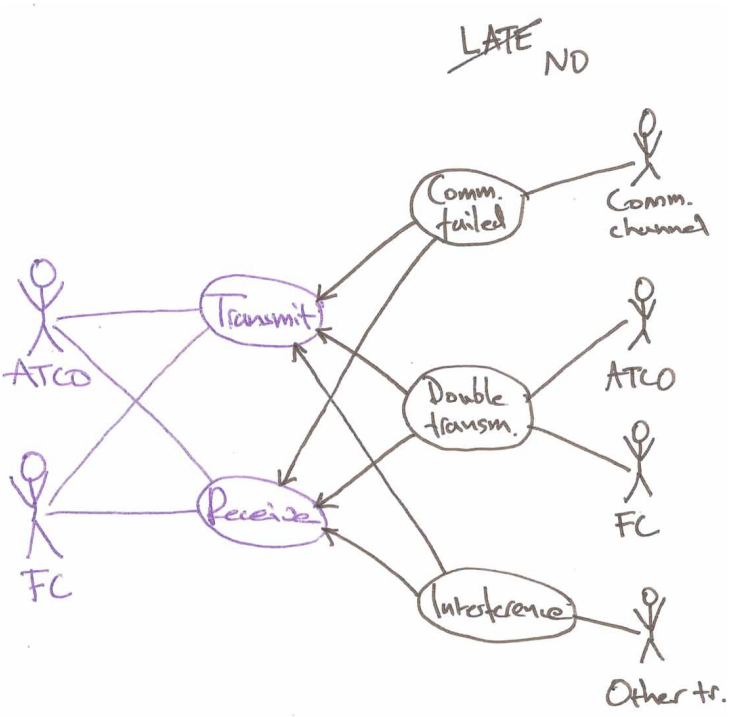
T-UC with alternative and exception paths, assumptions, and pre- and post-conditions. The comments and extensions to the T-UC gave enough information to also update the three SDs we had prepared in advance. Already at this stage, the participants began to discuss failures of system components, having some difficulty with the distinction between alternative and exception paths, perceiving the latter as places to document failures.

In the AirportLights case, the SD was instead discussed first. The participants found the diagram clear and useful and started discussing their understandings of the system. In this case too, the participants quickly began discussing possible failures. Moving from the SD to the T-UC did not flow naturally, and we had to guide the participants about how to use the alternative path. The T-UC was used to summarise the results of the SD session, without many new contributions from the participants.

Safety and Specification Lanes

Having validated and improved the premade models, we moved on to safety analysis, corresponding to the middle and right lanes of Figure 1. In the AirportLights case, the D-MUC discussion lead to many causes being suggested for the misuse cases. The participants entered concrete discussions of operational air-traffic control situations, narrowing the discussion down specific phases of operation and introducing environmental effects, such as weather and time of day, as factors. In the RadioSystems case, the participants quickly delved into technical details, but then realised that it was better to use D-MUCs (Figure 2) on a higher abstraction level this early in safety assessment. The guidewords helped the participants to brainstorm for hazards and three misuse cases were quickly identified. For example, the misuse case "communication failure" was identified by applying the guideword "late" to the use case "transmit", and the ensuing discussion lead to the new guideword "no" (as in "no transmission").

Figure 2. The created safety D-MUC by use of the guideword "no" (redrawn from photo image)



Although mitigation ideas started to emerge during the D-MUC discussions in the RadioSystems case — and became more concrete while the FSD was being drawn — they were not captured in any of the diagrams, perhaps because the participant who facilitated the discussion did not know the notations well, was not used to drawing time in the downwards direction, and did not naturally think of using colour to distinguish between what the system should and should not do. However, the FSD (Figure 3) encouraged discussing concurrency in more detail, which was recognised as a limitation of the D-MUC. The idea of breaking the FSD itself into more detailed FSDs also emerged.

In the AirportLights case, the participants used FSD to describe a detailed scenario for one of the safety D-MUCs, building on the common understanding that had been established and leveraging domain knowledge about environmental and operating conditions. They created the FSD in the following steps:

1. Drawing and discussing the actors, system components and their interactions;
2. Identifying the failure in the system for the scenario;
3. Drawing the failure and analysing the failure's effect in the system and on the actors; and
4. Identifying mitigations for the failure and discussing both failure and mitigations for effectiveness.

Only one cause of failure was described in the resulting FSD, whereas many causes were identified in the safety D-MUC. However, the causes identified with the D-MUC were on different levels of abstraction. If more FSDs had been created, we think more causes would have been identified and described.

In both sessions, the T-MUC was mostly written after (and partly in parallel with) the FSD. Table 4 shows the T-MUC for the RadioSystems case. The T-MUC was well suited to record the ideas from the discussions taking part during the D-MUC and, in particular, FSD sessions. The

Figure 3. The FSD representing the double transmission from the RadioSystems supplier (redrawn from photo image)

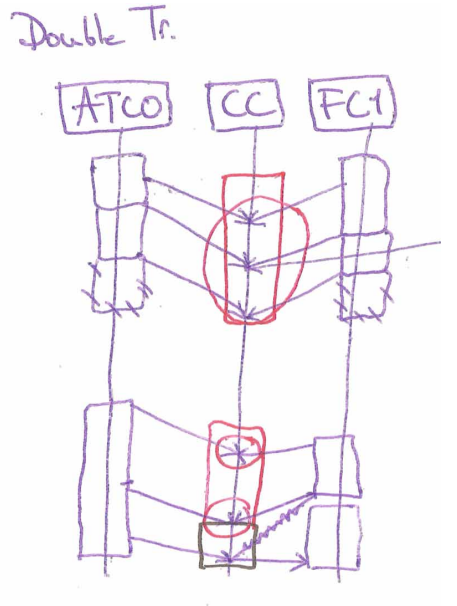


Table 4. The safety T-MUC for double transmission from the RadioSystems supplier

Name	Double Transmission
...	...
Basic path	• bp1. flight crew and ATCO initiate the transmitting at the same time • bp2. Other flight crew might receive the double transmission • bp3. flight crew and ATCO releases the transmitting button at the same time
Mitigation points	• mp1. If another flight crew hears double transmission and makes ATCO/flight crew aware of the double transmission • mp2. If double transmission happens less than specific timeframe, receiver recognizes it and informs the voice-communication system. • mp3. Procedure for re-transmit after a certain amount of time
Assumptions	• as1. In bp1: Same length of transmission • as2. In bp1: Communication channel is simplex • as3. In mp2: double transmission does not happen at same time (resolution of some miliseconds)
Preconditions	• preC1. flight crew and ATCO ready to transmit on the same time
Postconditions	• postC1. The ATCO and flight crew not aware of the double transmission
Misuser profile	ATCO and flight crew (communication channel)
Authors	• Informant-1-RadioSystemsSupplier • Informant-2-RadioSystemsSupplier

T-MUC was also used to record assumptions made during the FSD and MUC sessions. Many of the discussions during the D-MUC and FSD sessions had brought up issues that could have been directly recorded in the T-MUC fields, such as pre-conditions about weather, time of the day and flight phase, assumptions about ATCO and flight crew operations, stakeholders and the risks involved. Using T-MUC in parallel with D-MUC and FSD might therefore be a good way to record information that

might otherwise be forgotten. Although a few such issues were recorded in the D-MUC and FSD, they were less suitable than the T-MUC for this purpose. We also think information in a T-MUC is more likely to be reused in later development steps, both while still using CHASSIS and after moving on to later development methods.

In the RadioSystems case, the main results of safety analysis were condensed further into a HAZOP table (Table 5). Although the T-MUC had already collected and structured the most central information from the FSD and (in part) the D-MUC, the HAZOP table was useful for eliciting more general information, such as the hazard of a message double transmission and its consequences. The mitigation points in the T-MUC could directly be referenced from the HAZOP table to the recommended action for avoiding or treating the hazard.

In both sessions, the flow of information and ideas between the techniques was working well throughout safety assessment. Although time was limited, hazards were identified and analysed from different viewpoints and abstraction levels in both sessions. The overall time spent for the safety assessment was a little over an hour in both cases. Of the four modelling activities (D-MUC, FSD, T-MUC, and HAZOP), most time was spent on the T-MUC in the RadioSystems case and on FSD in the AirportLights case.

Security and Specification Lanes

Having conducted the safety assessment, we reiterated the middle and right lanes of Figure 1 to assess security. In the RadioSystems case, the participants quickly took the attackers point of view when working with the D-MUC (Figure 4), discussing a scenario where an attacker is simulating the ATCO and blocking normal communication between the ATCO and the flight crew. The HAZOP guidewords worked well for security, and the CIA triad generated more detailed ideas about how the attacker might proceed. After a while, the participants agreed the scenario was becoming too complex, creating a natural transition to continuing with MUSD.

In the AirportLights case, the D-MUC session remained on a high abstraction level, as the participants were discussing the capabilities of a potential attacker and where he or she might attack. The generation of misuse cases depended on how the originating UC was phrased and how the guideword was applied. We see the potential for guidephrase generation, for example using a list of the different combinations of guidewords and UC phrases. This could be done automatically in combination with ontologies, potentially making the security D-MUC more complete.

In the transition from D-MUC to MUSD, the focus moved to identifying different ways an attacker might access the airfield's lighting system. The discussion in the MUSD session was thus on a lower abstraction level but, at the same time, some of the new threats to the aircraft were on a higher one. In the RadioSystems case, the MUSD produced a detailed discussion of how an attacker could pose as ATCO, thereby continuing the scenario from the D-MUC. The participants still took the attacker's point of view, collaborated to build on each other's ideas, and brought in elements from the D-MUC for further elaboration. At this stage, they also identified possible mitigations. This session only produced a sketch of an MUSD, which we completed into a full MUSD (Figure 5) afterwards and distributed by email for validation.

Table 5. The HAZOP table created in the RadioSystems supplier

#	Item	Parameter	GW	Consequence	Cause	Hazard	Recomm.
1	Voice comm. system	Safety-related messages from ATCO and FC	No	ATCO and FC transmits at the same time; ATCO not aware of FC not receiving safety; FC does not follow safety instructions of ATCO in time	Start of transmission at the same time on simplex channel	Two aircraft get on collision course; aircraft flying towards an obstacle	Mp. 1, 2 and 3

Figure 4. The security D-MUC from the RadioSystems case (redrawn from photo image)

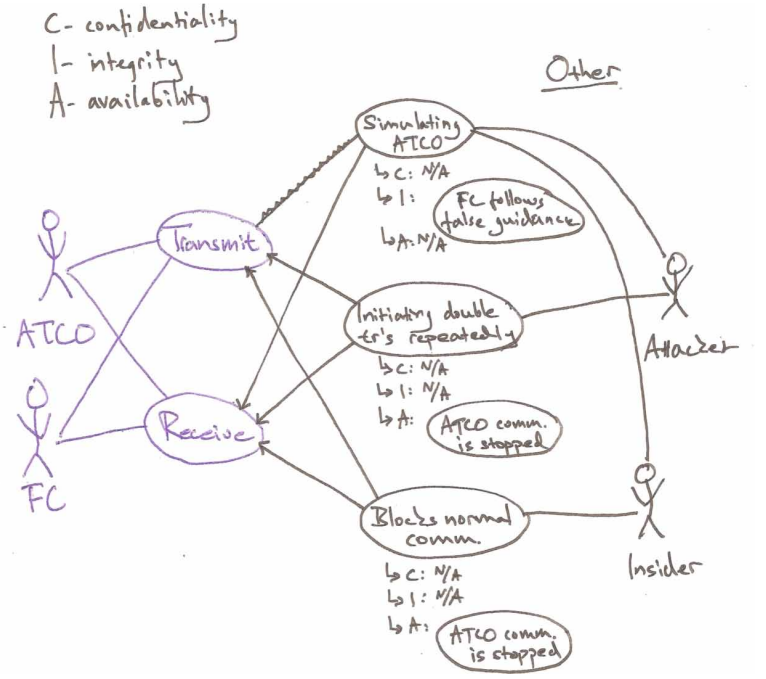
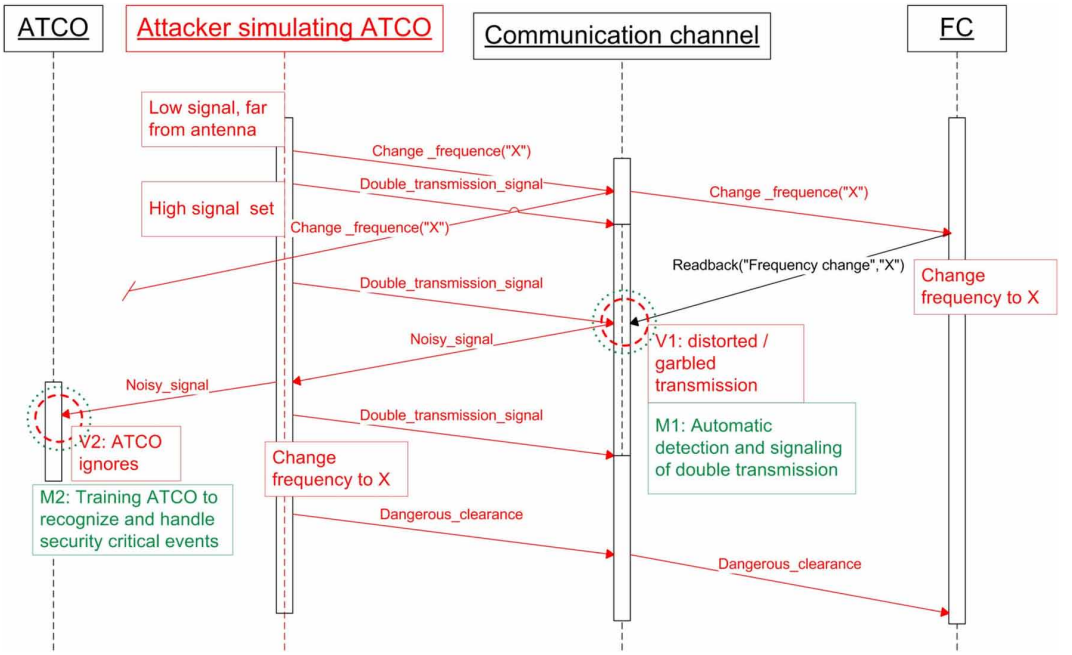


Figure 5. The MUSD where an outsider is simulating the ATCO from the RadioSystems case



In both sessions, less time (around 35 minutes in each case) was spent on security than on safety assessment. With the RadioSystems supplier, most time was spent on the security D-MUC, and the T-MUC the was skipped for lack of time, although the D-MUC and MUSD sessions provided much information that would have fit in the T-MUC fields. In the AirportLights case, the MUSD session was again the longest, underlining the participants' preference in this case for SD-based over UC-based notations.

Understanding

For the safety D-MUCs, some time was initially spent explaining the guidewords. But once explained, most participants were immediately able to relate guidewords to use cases. The participants also came up with suggestions of new causes of misuse cases. In the AirportLights case where the guideword "late" was used, there were suggestions to change it to "slow" or "delayed", showing that the participants understood guidewords as flexible. When proceeding with the security D-MUC, the participants were able to associate the guideword and use cases directly without our support.

In the AirportLights case, the participants also discussed the transition from D-MUC to either T-MUC or FSD. One participant wanted to start drawing something, another thought it would be easier to write down the steps in a T-MUC. He commented that FSD would be better in a group setting and T-MUC better when working alone. The third participant argued that a T-MUC gives more information than a FSD. We think all of them may be useful situational ways of adapting CHASSIS to the problem at hand.

The uses of FSD and MUSD in the AirportLights case were straightforward. Although the participant-facilitator needed some support in getting started with both FSD and MUSD, he was then able to draw illustrative diagrams and facilitate useful discussions. The resulting FSD showed both failures and mitigations, whereas there was not enough time to draw a complete and realistic MUSD. In the RadioSystems case, the discussions during the creation of MUSD and FSD showed that the participants understood how to detail the identified misuse cases. One participant did, however, have difficulties drawing a new FSD (although he could understand an already drawn FSD). A possible reason is that none of the participants in this company had previous experience with SD.

Learning

The participants reused many of the ideas from the safety assessment in the security assessment. For example, knowledge of the system and possible safety failures were used to identify vulnerabilities during security assessment, such as the double transmission from the AirportLights safety assessment.

The scenarios created by the participants, in particular during the FSD session, also created a common understanding that was used in both the safety and the security assessments. In particular, the participants reused knowledge about systems, domains, functionality, components, environments, and assumptions. Since they already had a common understanding of the system, the discussions during the security assessment became more focused on security issues. The double transmission detection, which was discussed as a mitigation in the safety assessment, was reused in the security assessment as one way to detect the attack scenarios in the security D-MUC and MUSD.

In both sessions, the participants were able to distinguish between safety and security aspects, although one security issue was mentioned in passing during safety analysis. When the participants reused knowledge from the safety assessment in the security assessment, they also translated the knowledge into a security setting.

The participants' understanding of the techniques was better during the security than during the safety assessments, most likely due to learning. In particular for the MUSD, the participants' modellings skills had improved over the FSD. They also seemed to align more naturally to the scope of assessment and limitations and to what to discuss and not during the security assessment. Although we did not use the same guidewords for safety and security D-MUCs, the participants understood the security guidewords more readily than the safety guidewords.

Questionnaire

After the CHASSIS sessions, we administered a post-session questionnaire to the participants. The questionnaire had two parts:

- Statements that we asked the participants to rank on a Likert-like scale ranging from 1—strongly disagree to 5—strongly agree; and
- Open-Ended questions that allow the participants to describe their experiences with using CHASSIS.

The four participants that answered agreed most strongly that CHASSIS facilitated discussions and common understanding among participants. They also agreed that it was easy to familiarise with CHASSIS, and that they would consider using CHASSIS again in the future. The detailed responses are summarised in the electronic addendum to this paper available at http://hdl.handle.net/1956/16161.

DISCUSSION

We proceed to answer our three research questions from Table 1, before we discuss CHASSIS from a broader perspective. We will also review our research approach.

Research Questions

1. Can the Same Basic Concepts be Used to Deal with Both Safety and Security Aspects?

Our results suggest that the same basic concepts can indeed be used for both safety and security aspects, with minor differences in interpretation of certain concepts, such as stakeholders and risk.

Turning to the sub-questions:

- The safety and security techniques of CHASSIS are mostly similar, although certain concepts need to be clarified and better explained, such as stakeholders and risks for the safety T-MUC and the alternative and exception paths for FSD. The learning effect from safety to security assessment also suggests that the techniques are indeed similar.
- The same system models can be used as a starting point for modelling both safety and security aspects, as was demonstrated in both cases.
- The resulting safety and security modelling processes are mostly similar. The order of creating MUCs and FSD/MUSDs differed, but the difference was a difference between the two cases as much as a difference between safety and security. For MUSD and FSD there were some differences in the starting point, where the misactor was the starting point for the MUSD, whereas for FSD it was the actor.
- The resulting safety and security models are also similar, although the misactors of the safety D-MUCs were somewhat different from the misactors in security D-MUCs. Also, the FSD and MUSD diagrams had different levels of detail, but this may be mostly due to limited time.

In both the cases, we only used the security techniques to address threats related to safety (security-for-safety). We did not consider other security aspects, such as information or business security (security-for-its-own-sake).

2. Is the CHASSIS Method Easy to Use?

Our results suggest that CHASSIS is easy to use for safety experts. The participants understood the concepts after some time and naturally brought experience from the safety assessments with them into the security work. They distinguished safety from security aspects without much effort.

However, when participants attempted to facilitate use of the FSD and MUSD techniques themselves, they sometimes needed help to get started drawing diagrams, whereas facilitating discussions around the diagrams came naturally. Answering the sub-questions:

- The participants mostly perceived the method and its techniques as easy to use, as evidenced by the questionnaire results. During the sessions, some time was spent explaining the participants how to proceed, in particular during safety assessment, but they quickly returned to discussing the cases, and each technique eventually produced useful information. Using CHASSIS was more challenging when the facilitator did not know the underlying techniques well beforehand, but both participant-facilitators were successful in the end.
- Given the limited time the participants were exposed to CHASSIS, they understood the method and its techniques well.
- The participants were able to distinguish safety from security aspects without guidance. Except for one comment about an ATCO being subject to a malicious attack during the safety D-MUC, we did not observe any confusion or attempt to discuss security during the safety sessions or vice versa. The participants reused ideas and information from the safety sessions in the security sessions, but they were able to adapt this knowledge to the security setting, and they always distinguished clearly between safety and security.

3. Is the CHASSIS Method Useful?

Our results also suggest that CHASSIS was perceived as useful by the participants. In the sessions, the method seemed to encourage creativity and foster common understanding. When answering the questionnaire, the participants reported interest in incorporating CHASSIS into their existing systems development processes. Returning to our sub-questions again:

- The participants perceived the method and its techniques as useful. Participants from both companies indicated they would be interested in integrating CHASSIS into their development processes, and they pointed out that the visualisation aspect is a strength, as it makes it easier to understand, discuss and distribute knowledge.
- The method and its techniques seem to encourage systematic consideration of both safety and security aspects. The method appeared structured at the method level. The T-MUC and HAZOP tables are a way to systematically structure much of the information that was elicited by the other techniques. It is possible to integrate HAZOP into T-MUCs by providing the appropriate fields.
- The method and its techniques encourage creative thinking about safety and security aspects. In both sessions, the participants were actively looking for and suggesting reasons for hazards. They also identified operational phases and environmental conditions that would have to be present for the hazard to become true.
- The method and its techniques facilitate a common understanding of the analysed processes and their safety and security aspects. While using D-MUC, FSD and MUSD, the participants exchanged and built on each other's understandings and explanations of how the system worked, of how it could fail or be attacked, and of how mitigations could prevent failures and attackers. We also observed how the participants corrected one another while referring to elements in the diagrams and how they brought knowledge established in the FSD sessions over into the MUSD sessions. The participants also rated common understanding high in their questionnaire responses.

Implications for Practice

We proceed to discuss implications of our findings for industrial practice in this section, before turning to research implications in the next.

Scope of CHASSIS

CHASSIS seems well fit for small-to-medium sized companies, such as the two suppliers of safe and secure ATM equipment that participated in our study. Such companies are in need of security processes, but cannot afford to implement extensive security processes that conflict with their established, mission-critical safety processes. According to the participants, CHASSIS can help visualising and documenting the safety and security aspects and assessments of their systems. Such documentation could be helpful in meetings with their customers, whether they are service providers (ASPNs) or equipment suppliers, to specify the correct level of safety and security for their components in the overall ATM system. We noted that some of the participants recognised the potential for using CHASSIS results to discuss and communicate safety and security aspect with other groups of stakeholders (e.g., ATCOs and flight crew).

Need for Guidelines

The study has made it clear that CHASSIS needs better guidelines in order to be used by the industry. Although CHASSIS appeared intuitive to the participants when the authors facilitated, it was not always easy for the participants to facilitate themselves. Although the participants stated that they would like to use CHASSIS as part of their development processes, more detailed guidelines are needed to ensure that it is applied in a systematic manner. Such guidelines should explain how to use the techniques together and detail what input that is expected and what output that should be produced by each technique. For example, T-MUC should be used with both D-MUC and FSD/MUSD, in order to record important information about safety and security aspects.

Abstraction Levels

The guidelines should clarify the appropriate abstraction level for each technique, so that it is clear whether to make new T-MUCs for each D-MUC, FSD and MUSD diagram, or whether one should refine one common T-MUC for all diagrams. Different abstraction levels should also be considered for the remaining techniques. For FSD and MUSD we have suggested decomposition, but this is from a system level point of view. One should also consider a scenario point of view, as an abstraction of which actors and system components to include for each scenario is important to keep the diagrams from becoming too complex. For D-MUC, the include and extend features of regular UCs should be considered.

Visualising Complex Safety and Security Issues

When answering our open-ended questions, participants emphasised the visualisations provided by CHASSIS, whose diagrams were used to consider safety and security aspects. However, some of them were also worried the diagrams might become too complex and the overview might be lost. The modelling techniques included in CHASSIS represent both the system (functions, components and their interactions), the negative aspects (hazards, threats, failures and vulnerabilities), and the potential remedies to those negative aspects. CHASSIS thus encourages adding a lot of information to models, potentially resulting in too complex diagrams. Our limited evaluations did not produce such complex diagrams, but the discussions of safety D-MUCs became so detailed that we switched to the FSD technique in one case. Another issue was that important information was not always recorded during the modelling sessions with D-MUC, FSD and MUSD. Two possible ways to deal with such problems are recording important information with T-MUC and decomposition of FSD and MUSD.

Recording Important Information With T-MUC

In the RadioSystems case, the secretary used T-MUC to record important information during the FSD session. Afterwards the information was reviewed by the participants, and some of it was changed and some more was added. This worked well in the CHASSIS sessions, but might not be the optimal way of using T-MUC with D-MUC and FSD. A previous study has suggested that the best way to use FSD and FMEA, also a table-based technique, was to use them iteratively (Raspotnig & Opdahl 2012b): FMEA was good for structuring the failure analysis, whereas FSD was suitable for facilitating discussions of failure propagations and system reactions. We might have included FMEA too in CHASSIS, but so far we have relied on T-MUCs only.

Recording information iteratively with T-MUCs during the use of D-MUC, FSD and MUSD would avoid adding information as annotations to the diagrams (e.g., "(low visibility)" and "< 10 s from touch down" in the MUSD from the AirportLights case shown in the addendum to this paper). For more complex diagrams this could reduce the complexity and increase the readability, in particular when the diagram is instead annotated with references to the T-MUC.

Decomposition of FSD and MUSD

One feature of FSD and MUSD that was not explored during the sessions is decomposition (Raspotnig & Opdahl 2012c). The advantage of decomposition is that it allows detailing both the system and its failures. In the RadioSystems case, Informant-2-RadioSystemsSupplier recognised that his discussions became too complex and detailed during the FSD session. He suggested having one higher level FSD for the consequences and one lower level FSD that would be more detailed about the causes. This is an example of FSD decomposition that would have been a possible continuation of the evaluation if more time had been available. We ended up adding the details to the T-MUC, although a decomposed FSD would have been more appropriate.

Decomposition is not the only way to reduce complexity in FSD and MUSD. SD also supports interaction use, which allows long sequences in an FSD or MUSD to be replaced by a simple label (Raspotnig & Opdahl 2012c).

Implications for Research

Visualising Complex Safety and Security Issues

Along the lines discussed in the previous section, further research such investigate how to avoid creating diagrams that are too complex when using CHASSIS, both by recording important information with T-MUC and though decomposition of FSD and MUSD.

Integration With Other Safety Methods

Although CHASSIS is a generic method that can be used in any domain where safety and security is important, it fits particularly well with current needs in the ATM domain. The main reason is the nature of the safety assessment method for ATM, the Eurocontrol Safety Assessment Methodology (SAM). As outlined in the Background section, SAM applies FHA as its first step, where system functions are assessed for identifying hazards. This step corresponds well to the D-MUC, which applies guidewords to use cases that represent system functions or services in order to identify hazards. The Preliminary System Safety Assessment (PSSA) is the next step in SAM, which is concerned with analysing the system architecture, both for regular system interactions and potential failures that can lead to the identified hazards. This is indeed close in nature to FSD. However, SAM only considers safety, whereas CHASSIS considers both safety and security. CHASSIS could thereby supplement SAM with a visual approach to safety assessment, and offer a corresponding security assessment process. This will be investigated in future work.

Integration With Risk-Oriented Methods

Several existing methods complement CHASSIS by explicitly assessing risk, which should somehow be covered by future versions of CHASSIS, either by extension or integration. The CORAS method provides an extensive framework for model-driven risk analysis (Lund et al. 2011a). It focuses on security risk analysis, whereas CHASSIS focusses on safety and security requirements elicitation. Also, CORAS is based on its own UML profile for addressing security, whereas CHASSIS extends existing UML diagrams with notations for representing both safety and security aspects. Like CHASSIS, work on CORAS has explored, among others, HAZOP and failure mode and effect analysis (FMEA) together with Unified Modeling Language (UML) diagrams to assess security aspects (Gran 2003). CHASSIS does, however, only use HAZOP guidewords together with use cases to elicit misuse cases.

The SafSec approach, developed for the avionics sector, also combines safety and security (Dobbing & Lautieri 2007). Unlike CHASSIS, it does not promote a model-based approach to safety and security assessment, but outlines a dependability process based on risk assessment. It also does not suggest concrete techniques, but focuses on the integrated risk process and how it relates to safety and security standards.

In contrast to the qualitative focus of CHASSIS, BDMP (Piètre-Cambacédès & Bouissou 2010) supports powerful quantitative modelling and analysis. And whereas BDMP can model hybrid scenarios that include both malicious and accidental sequences, CHASSIS aims at uniform, but separated sequences generated by FSD and MUSD.

Covering Other Dependability Aspects

An informant from the AirportLights supplier stated that "The exercise should perhaps be based on the requirements list, and then all the requirements with any relation to safety/security should be considered. The risks should also be classified, so that all are listed, but only those that will have any impact are analyzed." These are good suggestions, but currently out of the scope for the CHASSIS method. For risk considerations, CHASSIS has been compared to the Boolean-Logic Driven Markov processes (BDMP) technique (Kriaa 2013). This technique was considered as complementary to some of the CHASSIS techniques; it allows quantifying sequences of malicious and accidental scenarios that can complement FSD- and MUSD-generated sequences in order to classify risk. Instead of developing features in CHASSIS to classify risk, we will investigate how CHASSIS can be used together with other techniques that are specialised for this purpose.

For other dependability aspects, CHASSIS could consider reliability and availability aspects for FSD during decomposition, in particular if FMEA is included as part of CHASSIS. Furthermore, dependability aspects that relate to both safety and security should be considered for CHASSIS. The ideas of fault, error, and failure related to safety and security by Avizienis (2004), should be investigated for CHASSIS. One possibility could be to use the suggested categories of faults with some of the techniques included in CHASSIS, either as guidewords or as a checklist during the brainstorming sessions. Furthermore, the outlined fault and failure handling in (Avizienis 2004) could be used for specifying various mitigation strategies by CHASSIS techniques.

Validity and Reliability

We have taken care throughout this study to identify and ameliorate threats to validity and reliability (Yin 2008).

Construct Validity

Construct validity is threatened when the collected data are not really about the concepts studied, which could happen if we did not communicate well with the participants or if they did not properly understand CHASSIS. We have tried to avoid this in several ways. We have established good communication by working with the participants' companies in the industrial network over a longer

period, and the first author has extensive experience with the ATM domain as a practitioner. In the first part of the gap analysis, we carefully studied their existing safety and security practices: both in general and using SAM, SecRAM and SRMT. Participating in the CHASSIS session ourselves gave us the chance to quickly discover and rectify any misunderstandings about CHASSIS and its techniques.

We have also tried to mitigate threats to construct validity by using several different types and sources of data about the concepts we have studied, i.e., data triangulation. For example, when investigating ease of use, we used both responses to the post-session questionnaire and transcribed notes from the sessions. The session transcripts covered both the participants' comments about ease of use and problems we observed while they used CHASSIS. We kept pictures of the models drawn as further evidence of usability problems.

A threat that remains is the short time we had available, which only allowed a single iteration of CHASSIS and forced us to advance-prepare (and in one case also post-produce) certain diagrams that would normally have been created by the participants themselves. Our intimate knowledge of both CHASSIS, the ATM domain and the participants' companies has limited the impact of this threat.

Internal Validity

Internal validity is threatened when conclusions are not sufficiently grounded in data, which is a challenge when trying to conclude from a study that involves few companies and participants. We have tried to limit the problem in several ways. We have chosen research questions that encourage exploration and discussion over firm acceptance or rejection of hypotheses. The gist of our study has been to draw as much useful knowledge as possible out of a rare opportunity to study a safety and security technique as used by practitioners in a real industrial setting. We have taken care not to draw firmer conclusion than we can ground in our data.

We have used the research questions in Table 1 to streamline the research process, so that all later research steps – data collection, data analysis, and discussion – come out of the questions. We have established and followed detailed protocols for the evaluation sessions and data collection (Table 2). We have used different types and sources of data to back up our exploration and discussion (data triangulation). We have analysed the resulting combination of observation notes, produced artefacts and audio recordings through several iterations. As already explained in the Research Method section, we also involved a fourth researcher to review our analysis and results, i.e., researcher triangulation.

External Validity (Generalization)

External validity is threatened when findings cannot be generalised to a larger population: because our study has involved small-to-medium sized suppliers to the ATM sector, the findings may not apply to larger companies or to more complex problems. The results may not generalise to other domains than ATM either, but this may be less critical, because adopting safety and security techniques across industry sectors is often successful.

The small numbers of organisations and participants in our study also limits external validity, not least because the two companies had similar safety practices. Yet, in a qualitative and intensive study such as ours, the quality of the cases, data and analysis is more important than their numbers and sizes. The low number of organisations and participants is to some extent out-weighed by the richness of the data that has emerged from our close collaboration with industrial practitioners in realistic settings – in an area where empirical studies are rare.

Also, our study has focused only on security aspects that were related to safety, i.e., security-as-part-of-safety. We did not consider security-for-its-own-sake, e.g., general security of information and business processes. Our study also only involved experts on safety and software development, no security experts. Our study is a useful starting point, but further research is needed to establish more general knowledge about industrial use of CHASSIS.

Reliability

Reliability is threatened when repeating the study would have generated different data. To ensure reliability we have let a detailed set of research questions (Table 1) drive our data collection and we have made our detailed study protocol (Table 2) and questionnaires available in the electronic addendum to this paper at http://hdl.handle.net/1956/16161.

CONCLUSION

This paper has presented the CHASSIS method for combined safety and security assessment and evaluated it in two industrial case studies involving small-to-medium sized suppliers to the ATM sector. The evaluation has shown that CHASSIS successfully ties the safety and security areas together and is perceived as both useful and easy to use. The previous section has discussed several implications for both industrial practice and further research.

We originally planned to integrate safety and security assessments more closely in our evaluation. One idea was to follow an iterative approach, by first applying D-MUC for safety and security, then applying FSD and MUSD to identify safety and security mitigations, before returning to the D-MUCs for safety and security to analyse the mitigations found. Unfortunately, this was not possible due to time limitations and had to be left for further work. In general, further industrial case studies are called for with more companies in more sectors, preferably involving more companies and participants that use CHASSIS on larger systems for a longer period. Decomposition of safety and security models should also be explored as part of CHASSIS. The present study has spent more time on safety than on security assessment. Further studies should provide balance and investigate security-for-its-own-sake in addition to security-as-part-of-safety. Ideas from CHASSIS can even be adapted to other dependability areas, for example combining security with privacy assessment.

Our study has shown that for CHASSIS to be taken up by industry, better usage guidelines are needed and the method must be better aligned with current safety processes. In further work, we therefore want to use the experiences and knowledge obtained from this evaluation to develop more detailed process guidelines. The guidelines should cover using both CHASSIS and its techniques alone and using CHASSIS as part of broader safety, security and software development processes. Our study has made us realise that the organisations that need CHASSIS are likely to have mature safety assessment processes in place already, but less clear security assessment process. The guidelines and perhaps the method itself should be tailored to reflect this imbalance.

CHASSIS can also be complemented with other techniques. One possibility we have mentioned is introducing FMEA tables to supplement HAZOP. Another path for further work is to investigate how CHASSIS can be augmented by methods such as CORAS, SafSec and Boolean-Driven Markov Processes (BDMP) in order to explicitly assess risk. We also hope to compare and integrate CHASSIS more specifically with existing methods for safety and security assessments in the ATM domain, in particular SAM/SecRAM/SRMT. Yet another path is to integrate CHASSIS with argumentation systems to better support development of safety and security cases.

ACKNOWLEDGMENT

We would like to acknowledge Vikash Katta for his great effort of going through the complete material thoroughly and providing us with precise comments on required improvements, in particular to the analysis.

REFERENCES

Amoroso, E. G. (1994). *Fundamentals of Computer Security Technology*. Upper Saddle River, NJ: Prentice-Hall.

Antón, A.I., and Earp, J.B. (2000). Strategies for developing policies and requirements for secure electronic commerce systems. *E-commerce security and privacy, 2*.

Avizienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Trans. Dep. Sec. Comp.*, *1*(1), 11–33. doi:10.1109/TDSC.2004.2

Rodríguez, A., Fernández-Medina, E., & Piattini, M. (2006). Capturing Security Requirements in Business Processes through a UML 2. 0 Activity Diagrams Profile. In Proc. Advances in Conceptual Modeling – Theory and Practice (ER 2006 Workshops). Springer. doi:10.1007/11908883_6

Dardenne, A., van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of Computer Programming*, *20*(1-2), 3–50. doi:10.1016/0167-6423(93)90021-G

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Systems Quarterly*, *13*(3), 319–340. doi:10.2307/249008

Dobbing, B., & Lautieri, S. (2007). Dependability-by-Contract. In F. Redmill & T. Anderson (Eds.), *The Safety of Systems, Proc. Fifteenth Safety-Critical Systems Symposium*, Bristol, UK, February 13–15 (pp. 35–51). Springer.

Elahi, G. (2012). Making Trade-offs among Security and Other Requirements during System Design [PhD Thesis]. Univ. of Toronto.

Ericson, C. A. (1999). Fault Tree Analysis – A History. In *Proc. 17th International System Safety Conference*.

Ericson, C. A. II. (2005). *Hazard analysis techniques for system safety*. Wiley-Interscience. doi:10.1002/0471739421

Eurocontrol Safety Assessment Methodology Task Force. (2004). Functional Hazard Assessment – Guidance Material B1, edition 2.0.

Eurocontrol Safety Assessment Methodology Task Force. (2006). *Air navigation safety assessment methodology* (ed. 2.1).

Gran, B. A. (2003). The CORAS methodology for model-based risk assessment. Information Society Technology.

Herrmann, A., Morali, A., Etalle, S. & Wieringa, R. (2011). RiskREP: Risk-based Security Requirements Elicitation and Prioritization.

International Electrotechnical Commission. (2006). *IEC 60812 Analysis techniques for system reliability – Procedure for failure mode and effects analysis*.

Jürjens, J. (2002). UMLsec: Extending UML for Secure Systems Development. The Unified Modeling Language. In *Proceedings of the 5th International Conference (UML 2002)*, Dresden, Germany. Springer.

Katta, V., Karpati, P., Opdahl, A. L., Raspotnig, C., & Sindre, G. (2010). Comparing two techniques for intrusion visualization. In Bommel et al. (Eds.), *The Practice of Enterprise Modeling, LNBIP* (Vol. 68, pp. 1–15). Springer Berlin Heidelberg. doi:10.1007/978-3-642-16782-9_1

Kriaa, S., Raspotnig, C., Bouissou, M., Pièrre-Cambacédès, L., Karpati, P., Halgand, Y., & Katta, V. (2013). Comparing two approaches to safety and security modelling: BDMP technique and CHASSIS method. In *Proc. Enlarged Halden Group Meeting*, Storefjell, Norway. OECD Halden Reactor Project.

Leveson, N. G. (2011). *Engineering a Safer World: Systems Thinking Applied to Safety*. Engineering Systems.

Lin, L., Nuseibeh, B., Ince, D., Jackson, M., & Moffett, J. (2003). Introducing Abuse Frames for Analysing Security Requirements. In *Proc. 11th IEEE International Requirements Engineering Conference (RE'03)*, Monterey Bay, CA. IEEE. doi:10.1109/ICRE.2003.1232791

Lin, L., Nuseibeh, B., Ince, D., & Jackson, M. (2004). Using Abuse Frames to Bound the Scope of Security Problems. In *Proc. 12th IEEE International Requirements Engineering Conference (RE'04)*, Kyoto, Japan. IEEE.

Liu, L., Yu, E., & Mylopoulos, J. (2003). Security and Privacy Requirements Analysis within a Social Setting. In *Proc. 11th International Requirements Engineering Conference (RE'03)*, Monterey Bay, CA. IEEE. doi:10.1109/ICRE.2003.1232746

Lodderstedt, T., Basin, D., & Doser, J. (2002). SecureUML: A UML-Based Modeling Language for Model-Driven Security. The Unified Modeling Language. In *Proc. 5th Int'l Conf. (UML 2002)*, Dresden, Germany. Springer.

Lund, M.S., Solhaug, B., & Stølen, K. (2011a). *Model-Driven Risk Analysis – The CORAS Approach.* Springer.

Lund, M. S., Solhaug, B., & Stølen, K. (2011b). Risk analysis of changing and evolving systems using CORAS. In A. Aldini & R. Gorrieri (Eds.), *Foundations of Security Analysis and Design VI, LNCS 6858* (pp. 231–274). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-23082-0_9

Massacci, F., & Zannone, N. (2006). *Detecting Conflicts between Functional and Security Requirements with Secure Tropos: John Rusnak and the Allied Irish Bank. Social Modeling for Requirements Engineering. P. Giorgini, N. A. M. Maiden, J. Mylopoulos and E. Yu*. Cambridge, MA: MIT Press.

Matulevicius, R., Mayer, N., Mouratidis, H., Dubois, E., Heymans, P., & Genon, N. (2008). Adapting Secure Tropos for security risk management in the early phases of information systems development. In *Advanced Information Systems Engineering* (pp. 541–555). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-69534-9_40

McDermott, J., & Fox, C. (1999). Using Abuse Case Models for Security Requirements Analysis. In *Proc. 15th Annual Computer Security Applications Conference (ACSAC'99)*. IEEE CS Press. doi:10.1109/CSAC.1999.816013

Mouratidis, H., & Giorgini, P. (2007). Secure Tropos: A Security-Oriented Extension of the Tropos Methodology. *International Journal of Software Engineering and Knowledge Engineering*, *17*(02), 285–309. doi:10.1142/S0218194007003240

Mouratidis, H., Giorgini, P., & Manson, G. (2005). When security meets software engineering: A case of modelling secure information systems. *Information Systems*, *30*(8), 609–629. doi:10.1016/j.is.2004.06.002

Perrow, C. (1999). *Normal Accidents: Living with High-Risk Technologies*. Princeton University Press.

Piètre-Cambacédès, L., & Bouissou, M. *(*2010*).* Beyond attack trees: Dynamic security modeling with boolean logic driven markov processes (BDMP). In *Proc. European Dependable Computing Conference (EDCC)* (pp. 199–208). doi:10.1109/EDCC.2010.32

Raspotnig, C., Karpati, P., & Katta, V. (2012a). A combined process for elicitation and analysis of safety and security requirements. In I. Bider et al. (Eds.), *Enterprise, Business-Process and Information Systems Modeling, LNCS 113* (pp. 347–361). Berlin, Heidelberg: Springer. doi:10.1007/978-3-642-31072-0_24

Raspotnig, C., Katta, V., Karpati, P., & Opdahl, A. L. (2013a). Enhancing CHASSIS: A Method for Combining Safety and Security. In *Proc. Eighth International Conference on Availability, Reliability and Security (ARES)* (pp. 766–773). IEEE. doi:10.1109/ARES.2013.102

Raspotnig, C., & Opdahl, A. L. (2012b). Improving security and safety modelling with failure sequence diagrams. *International Journal of Secure Software Engineering*, *1*(3), 20–36. doi:10.4018/jsse.2012010102

Raspotnig, C., & Opdahl, A. L. (2012c). Supporting failure mode and effect analysis: A case study with failure sequence diagrams. In B. Regnell & D. Damian (Eds.), *Requirements Engineering: Foundation for Software Quality, LNCS 7195* (pp. 117–131). Springer Berlin Heidelberg. doi:10.1007/978-3-642-28714-5_10

Raspotnig, C., & Opdahl, A. L. (2013b). Comparing risk identification techniques for safety and security requirements. *Journal of Systems and Software*, *86*(4), 1124–1151. doi:10.1016/j.jss.2012.12.002

Schneier, B. (1999). Attack Trees. Dr. Dobb's Journal, December.

Schneier, B. (2000). *Secrets and Lies: Digital Security in a Networked World*. Indianapolis: Wiley.

Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., & Sommerlad, P. (Eds.). (2005). Security Patterns: Integrating Security and Systems Engineering. Wiley.

Sindre, G. (2007). A look at misuse cases for safety concerns. In J. Ralyté, S. Brinkkemper, & B. Henderson-Sellers (Eds.), *Situational Method Engineering: Fundamentals and Experiences, IFIP 244* (pp. 252–266). Boston: Springer. doi:10.1007/978-0-387-73947-2_20

Sindre, G., & Opdahl, A. L. (2000). Eliciting Security Requirements by Misuse Cases. In *Proc. TOOLS Pacific 2000*, Sydney. IEEE CS Press.

Sindre, G., & Opdahl, A. L. (2005). Eliciting security requirements with misuse cases. *Requir. Eng.*, *10*(1), 34–44. doi:10.1007/s00766-004-0194-4

Society of Automotive Engineers (SAE). (1996). Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment, December.

Srivatanakul, T., Clark, J., & Polack, F. (2004). Effective security requirements analysis: Hazop and use cases. In K. Zhang & Y. Zheng (Eds.), *Information Security, LNCS* (Vol. *3225,* pp. 416–427). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-30144-8_35

Stallings, W., & Brown, L. V. (2008). *Computer security*. Prentice-Hall.

Stamatis, D. H. (1995). *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. American Society for Quality. ASQ Press.

van Lamsweerde, A. (2004). Elaborating Security Requirements by Construction of Intentional Anti-Models. In *Proc. International Conference on Software Engineering*, Los Alamitos, CA. doi:10.1109/ICSE.2004.1317437

van Lamsweerde, A., & Letier, E. (2000). Handling Obstacles in Goal-Oriented Requirements Engineering. *IEEE Transactions on Software Engineering*, *26*(10), 978–1005. doi:10.1109/32.879820

Winther, R., Johnsen, O.-A., & Gran, B. A. (2001). Security Assessments of Safety Critical Systems Using HAZOPs. In *Computer Safety* (pp. 14–24). Reliability and Security. doi:10.1007/3-540-45416-0_2

Yin, R. K. (2008). Applied Social Research Methods Series. In *Case Study Research: Design and Methods* (4th ed.). SAGE Publications.

ENDNOTES

1 See Single European Sky ATM Research (SESAR): *SESAR Joint Undertaking (JU)*, http://www.sesarju.eu/, accessed 2017-04-11.

2 See Federal Aviation Administration (FAA): *Next Generation Air Transportation System (NextGen)*, https://www.faa.gov/nextgen/, accessed 2017-04-11.

3 SESAR Joint Undertaking (SESAR JU): WP16 – R&D Transversal Areas, Description of Work Version 5.0, June 2009.

4 ATM Bedriftsnettverk: ATM Safety & Security Bedriftsnettverk, Sub-activity DA-1.1 – Analysis of Current Standards and Regulations. Technical Report L-1.1.1, September 2011.

5 ATM Bedriftsnettverk: ATM Safety & Security Bedriftsnettverk, Sub-activity DA-1.2 – GAP Analysis of Safety and Security Processes. Technical Report L-1.2.2, April 2012.