Information and Software Technology 75 (2016) 71–89

Contents lists available at ScienceDirect

Information and Software Technology

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

Requirements engineering for safety-critical systems: A systematic literature review

Luiz Eduardo G. Martins[a,*], Tony Gorschek[b]

[a] Department of Science and Technology, Federal University of São Paulo, São José dos Campos, Brazil
[b] Software Engineering Research Lab, Blekinge Institute of Technology, Karlskrona, Sweden

ARTICLE INFO

Article history:
Received 15 October 2015
Revised 23 March 2016
Accepted 3 April 2016
Available online 12 April 2016

Keywords:
Safety requirements
Safety-critical systems
Hazard
Accident
Systematic literature review
Requirements engineering

ABSTRACT

Context: Safety-Critical Systems (SCS) are becoming increasingly present in our society. A considerable amount of research effort has been invested into improving the SCS requirements engineering process as it is critical to the successful development of SCS and, in particular, the engineering of safety aspects.
Objective: This article aims to investigate which approaches have been proposed to elicit, model, specify and validate safety requirements in the context of SCS, as well as to what extent such approaches have been validated in industrial settings. The paper will also investigate how the usability and usefulness of the reported approaches have been explored, and to what extent they enable requirements communication among the development project/team actors in the development of SCS.
Method: We conducted a systematic literature review by selecting 151 papers published between 1983 and 2014. The research methodology to conduct the SLR was based on the guidelines proposed by Kitchenham and Biolchini.
Results: The results of this systematic review should encourage further research into the design of studies to improve the requirements engineering for SCS, particularly to enable the communication of the safety requirements among the project team actors, and the adoption of other models for hazard and accident models. The presented results point to the need for more industry-oriented studies, particularly with more participation of practitioners in the validation of new approaches.
Conclusion: The most relevant findings from this review and their implications for further research are as follows: integration between requirements engineering and safety engineering areas; dominance of the traditional approaches; early mortality of new approaches; need for industry validation; lack of evidence for the usefulness and usability of most approaches; and the lack of studies that investigate how to improve the communication process throughout the lifecycle. Based on the findings, we suggest a research agenda to the community of researchers and advices to SCS practitioners.

1. Introduction

Computing systems are becoming ubiquitous and a basic part of human life. They help us in so many activities that it is difficult to imagine modern society without their support. Nevertheless, this ubiquitous presence carries a high level of dependency, which inevitably demands the need for systems that are increasingly available, reliable, safe, and secure [18,119] (for SLR references please see Appendix A). In many situations we rely on computing systems to help us control highly critical activities [81,115], such as in medical procedures [59,79,85,104,117,130,179], human transportation [48a,53,87,88,96,110,164], aerospace and defence systems [23,71,128,131,132,183], and high energy handling [142,173,182,189]. Failures during the control of these systems might cause serious damage to the environment, property and people [8,17,29], impacting companies, the marketplace, as well as the quality of life of people and society in general.

Systems with these characteristics are generally called safety-critical systems (SCS) [113,124,148]. Over the last 40 years, a considerable amount of research effort has been invested into improving the engineering of SCS. One of the most significant challenges for companies that develop SCS is to create and establish a complete, correct, unambiguous, testable, and yet understandable requirements specification and/or understanding shared among stakeholders [15,64,65,92,106,126]. This is crucial for the IT mainstream [20], but it is even more important for companies developing SCS, when considering product/system safety certification process compliance [100,180]. Compliance and ultimate

* Corresponding author. Tel.: +551997628279.
E-mail addresses: legmartins@unifesp.br (L.E.G. Martins), tony.gorschek@bth.se (T. Gorschek).

http://dx.doi.org/10.1016/j.infsof.2016.04.002
0950-5849/

certification and associated processes are required for companies developing SCS [91,98].

The literature on SCS has reported on many cases where systems have failed due to a lack in requirements specifications, or misunderstandings traced to problems in requirements engineering, contributing to accidents that cause damage to the environment, injury to people and even the loss of lives [8,13,177]. Of course, when accidents happen, they have a strong negative impact for the companies responsible for the associated SCS. As discussed by Leveson in [16], the causes of accidents involving complex technological systems usually are multifactorial. The hierarchical model of accidents causes proposed by Lewycky [39] points out that it is the constraints, or lack of them, on technical and physical conditions, social and human interactions [60,82,93,125,146], the management system and organizational culture, as well as governmental or socioeconomic policies [16,24] that are the root causes of accidents. Such factors are closely related and have influenced the way of approaching the safety requirements of SCS [61,74].

With the increasing complexity of SCS [12,191], the rules and standards for safety certification and associated processes defined by governments and international agencies are becoming more difficult and expansive [25]. The requirements specifications, and related processes for requirements engineering, play a very important role during the safety certification process [26,157,159,160], both in relation to process-based compliance and safety-assurance standards [15,19,145,150,174]. In addition, with the system functionalities increasingly moving from hardware to software [70], the safety certification process becomes even more complex.

Considering the importance of the requirements engineering process for improving the safety of SCS [78,165,171,178,185], we conducted a systematic literature review (SLR) to investigate what approaches have been proposed to elicit, model, specify or validate safety requirements in the context of SCS, as well as to what extent such approaches have been validated in industrial settings. Furthermore, we investigate the relationship between safety analysis and requirements engineering practices in order to analyse how integrated these areas are and what communication issues emerged from them. In this paper we analyse and discuss the SLR results considering four perspectives: (i) the requirements engineering approaches to treat safety requirements; (ii) how the safety requirements approaches have been validated by their proponents; (iii) how the usefulness and usability of the approaches have been measured by their proponents; and (iv) to what extent the safety requirements approaches support communication among the actors throughout the SCS lifecycle. To the best of our knowledge, this is the first SLR on the topic of requirements engineering for SCS.

This paper is organized as follows. Section 2 presents background and related work. The research methodology adopted to conduct the SLR is presented in Section 3. The results and the analysis related to our research questions are presented in Section 4. Our conclusions are presented in Section 5.

2. Background and related work

2.1. Definitions

In order to set the scope and make clear the adopted terms used in the SLR, and to ensure consistency throughout this paper, we present the following definitions, organized in alphabetical order:

Accident. An undesirable (negative) event involving damage, loss, suffering or death [7,120].

Approach. In the context of this SLR, we are interested in the following types of approaches: technique, model, framework, method, process, methodology or tool to elicit, model, specify or validate safety requirements for safety-critical systems.

Functional safety requirement. The requirement to prevent or mitigate the effects of failures identified in safety analysis [6].

Hazard. A system state that might, under certain environmental conditions, lead to a mishap. Hence, a hazard is a potentially dangerous situation that may lead to an accident [1,7].

Safety. Firesmith defines safety as “the degree to which accidental harm is properly addressed (e.g. prevented, identified, reacted to, and adapted to)” [2]. According to Leveson “safety must be defined in terms of hazards or states of the system that when combined with certain environmental conditions could lead to a mishap.” [7,133].

Safety requirement. A requirement that describes the constraints or actions to support and improve a system’s safety. Firesmith defines the safety requirement as “any quality requirement that specifies a minimum, mandatory amount of safety in terms of a system-specific quality criterion and a minimum level of an associated metric.” [2].

Safety-critical. According to Medikonda and Panchumarthy “those software or system operations that, if not performed, performed out of sequence, or performed incorrectly could result in improper control functions, or lack of control functions required for proper system operation, that could directly or indirectly cause or allow a hazardous condition to exist” [1].

Usability. How easy an approach is to be used by practitioners.

Usefulness. The fact of being useful and bringing value for practitioners.

Validated approach. It is one that was tested, piloted or performed in some way into the industry setting.

2.2. Related work

Nair et al. [62] conducted a SLR on evidence for safety. The study considered 171 peer-reviewed publications with the intention of answering four questions: “What constitutes the evidence for safety?”, “What techniques are used for structuring safety evidence?”, “What techniques are used for assessing safety evidence?” and “What challenges and needs have been the target of the investigation in relation to safety evidence?” The authors argue that they intentionally conducted a SLR with a broad scope, not restricting themselves to a particular standard or domain. The stated reason for such a decision is that the breadth of scope enabled them to provide a more general and thorough analysis of the state of the art on evidence for safety. Additionally, they classified the various notions of evidence gleaned from the literature into a hierarchical taxonomy, which includes 49 evidence types.

Mellado et al. [3] conducted a systematic review of the literature concerning security requirements engineering in order to summarize evidence regarding security requirements approaches and to provide a framework to appropriately support new research activities. The research question that they tried to answer was “Which approaches have been carried out to develop secure Information Systems by means of Security Requirements Engineering?” They found 22 studies that completely fit their previously defined inclusion criteria.

Rodríguez-Dapena [19] discusses software safety certification as a multi-domain challenge. This work highlights the problem of new systems that are built from subsystems, which come from different application domains, because there is no certification scheme for inter-domain systems yet. The author comments on the