Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 238 (2009) 71–77
www.elsevier.com/locate/entcs

Model-Based Safety-Cases for Software-Intensive Systems

Peter Braun[1] Jan Philipps[2]

Validas AG
München, Germany

Bernhard Schätz[3] Stefan Wagner[4]

Institut für Informatik
Technische Universität München
Garching b. München, Germany

Abstract

Safety cases become increasingly important for software certification. Models play a crucial role in building and combining information for the safety case. This position paper sketches an ideal model-based safety case with defect hypotheses and failure characterisations. From this, open research issues are derived.

Keywords: Safety case, model-based, structured argument, defect hypothesis, failure characterisation

1 Introduction

The proliferation of software-intensive technical systems has resulted in a growing need for methods to demonstrate their safety and reliability. The goal of such methods is to develop a *safety case* for a system – a line of argument that establishes safety and reliability properties from known properties of the components of the system.

Various approaches exist: Leveson et al. [7] describe an FTA-like approach to examine Ada programs, while Giese et al. [3] show how HAZOP-like safety analyses can be based on component and deployment diagrams of the UML. Within the ISAAC project [2], models of functional, geometrical and human aspects are

[1] Email: peter.braun@validas.de
[2] Email: jan.philipps@validas.de
[3] Email: schaetz@in.tum.de
[4] Email: wagnerst@in.tum.de

doi:10.1016/j.entcs.2009.09.007

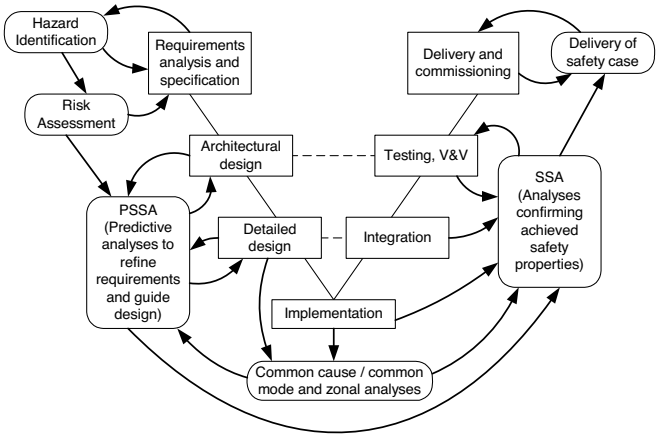

Fig. 1. Safety activities in a development process (Source: [9, p. 29])

integrated for safety analyses. Pumfrey [9] gathers a list of nine factors for success of safety analysis methods and goes on to develop two methods for dealing with mixed hw/sw systems.

In all these approaches the use of models plays a central role in the construction of a safety case. While earlier approaches are based on structured reviews of models, recently formal verification techniques have been applied for model analysis [1,3,5,2]. However, a systematic approach to the definition of those models is still uncommon. It is also an open issue how to justify the appropriateness of the underlying models for the safety case: Is it possible to derive all relevant hazards, system failures and component faults, and is it possible to reason about the causal chains that link them? In other words, we believe that the major open issue is how to reason about the *choice* of models, and not so much how to reason about the *properties* of the models.

In addition to this principal issue of the appropriateness of the models, we believe that there are a number of core success factors for building model-based safety cases:

- *Seamless integration into development processes.* It is not sufficient to merely perform a single safety analysis for certification of the final system – analyses with different focus play their role throughout system development, in order to clarify requirements, designs, and in general to improve both product and process (see Fig. 1).
- *Consideration of system, platform and environment.* It is not sufficient to examine models for the functional behaviour (even if they are augmented with fault models, as in [5]) of a system by verification or tests. Since hazards manifest themselves in the system's environment, the environment must also be modelled and included in the analysis. Since faults often are caused by the underlying computing platform, the platform must also be included; possibly, abstract user models may be needed to reason about operator errors and ways to avoid them or to deal with them. Note that in the development process these different models may well be constructed and analysed at different times: For instance, a prelimi-