Chapter 4

The ForMoSA Approach to Qualitative and Quantitative Model-Based Safety Analysis

Axel Habermaier
Universität Augsburg, Institut für Informatik, Germany

Matthias Güdemann
Otto-von-Guericke University of Magdeburg, Germany

Frank Ortmeier
Otto-von-Guericke University of Magdeburg, Germany

Wolfgang Reif
Universität Augsburg, Institut für Informatik, Germany

Gerhard Schellhorn
Universität Augsburg, Institut für Informatik, Germany

ABSTRACT

This chapter presents ForMoSA (FORmal M Odels and Safety Analysis), an integrated approach for the safety assessment of safety-critical embedded systems. The approach brings together the best of engineering practice, formal methods, and mathematics: traditional safety analysis, temporal logics and verification, as well as statistics and optimization. These three orthogonal techniques cover three different aspects of safety: fault tolerance, functional correctness, and quantitative analysis. The ForMoSA approach combines these techniques to assess system safety in a structured and formal way. Furthermore, the tight combination of methods from different analysis domains results in mutual benefits. The combined approach yields results which cannot be produced by any single technique on its own. The methodology was applied to several case studies from different industrial domains. One of them is an autonomous control of level crossings using radio-based communication, which is used in this chapter to describe the individual steps of the ForMoSA methodology.

DOI: 10.4018/978-1-4666-1643-1.ch004

Copyright © 2012, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
MOTIVATION

Safety-critical systems are expected to operate safely under regular circumstances as well as in many degraded situations. In the latter, these systems have to cope with one or more components that are not working as specified, while at the same time they have to guarantee that no harm is done to people or the environment. Fault Tree Analysis (Vesley, Dugan, Fragola, Minarick, & Railsback, 2002), Failure Modes and Effects Analysis (Reifer, 1979), Hazard and Operability Study (Fenelon, McDermid, Nicholson, & Pumfrey, 1995), and many other traditional safety analysis techniques help safety engineers in systematically analyzing the safety of a system: They dissect the system to determine possible (combinations of) component failures that might result in an occurrence of a hazard and therefore in a violation of the intended behavior. However, the functionality provided by safety-critical systems is becoming increasingly complex, therefore requiring the development of more sophisticated safety analysis techniques to analyze the system behavior under both regular and degraded situations. Additionally, software is becoming a more important factor for the innovation of safety-critical systems; whether it is the automotive sector, avionics, or the railway industry, more and more safety-critical hardware is replaced by software. Among the typical reasons for this change in system design is the increased flexibility offered by implementing certain functionality in software, resulting in a larger amount of features supported by the system as well as more full-fledged implementations of the individual features. On the other hand, software development is complex and error-prone and is thus likely to introduce systematic errors that have the potential of violating safety requirements. In the field of software engineering, formal methods have been developed to deal with software design and implementation errors. These mathematical techniques allow sound reasoning about software designs and implementations with mathematical rigor.

This chapter applies formal methods to the field of safety assessment of safety-critical systems, also pointing out the advantages and disadvantages of the approach over traditional analysis techniques such as the aforementioned Fault Tree Analysis (FTA). The formal techniques presented in the remainder of this chapter focus on discovering cause-consequence relationships under the assumption that all relevant safety requirements, hazards, and component failures are already identified. The techniques also support risk assessment by deriving occurrence probabilities for the hazards from the cause-consequence relationships obtained from the analysis. Costs and other consequences related to the occurrence of a hazard are outside the scope of these techniques, although we briefly outline how the results of the analysis can be used to find an optimal trade-off in the case of antagonistic goals. Furthermore, the techniques presented hereafter can be applied during the entire lifecycle of safety-critical systems, that is, from the beginning of the development of a system until its decommissioning and disposal. During the initial development phase, formal methods are highly recommended by some industry standards such as IEC 61508 (IEC, 2010) and EN 50128 (CENELEC, 2011) in order to reach the highest software safety level. In particular, EN 50128 highly recommends the use of formal methods for the software requirements specification, software design, and verification. Compliance with these standards is often required either by law or contractually. During the maintenance and evolution phase of a system, formal methods can be used to prove that revised safety measures or other changes to the system solve newly identified safety issues or at least do not negatively affect overall system safety. When a system is decommissioned, the techniques can ascertain that a system shutdown does not lead to any safety violations or other unexpected consequences. The techniques also serve as a formal documentation of the safety assessments performed during the entire lifecycle of the system as required by IEC 61508 and EN 50128.