Journal of Computational Science xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Journal of Computational Science

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

An integrated toolchain for model based functional safety analysis☆

Lena Rogovchenko-Buffoni[a,*], Andrea Tundis[b], Muhammed Zoheb Hossain[c], Mattias Nyberg[c], Peter Fritzson[a]

[a] Department of Computer and Information Science (IDA), Linköping University, SE-581 83 Linköping, Sweden
[b] Department of Computer Engineering, Modeling, Electronics and Systems Science (DIMES), University of Calabria, Via P. Bucci 41C, 87036 Rende, CS, Italy
[c] Scania AB, SE-151 87 Södertälje, Sweden

ARTICLE INFO

Article history:
Received 7 February 2013
Received in revised form 17 July 2013
Accepted 25 August 2013
Available online xxx

Keywords:
Bayesian networks
Safety analysis
Model-based design
Functional testing

ABSTRACT

The significant increase in the complexity and autonomy of the hardware systems renders the verification of the functional safety of each individual component as well as of the entire system a complex task and underlines the need for integrated, model based tools that would assist this process. In this paper the authors present such a tool, coupled with an approach to functional safety analysis, based on the integration of functional tests into the model itself. The analysis of the resulting model is done through a stochastic Bayesian model. This approach strives to both bypass the necessity for costly hardware testing and integrate the functional safety analysis into an intuitive component development process.

1. Introduction

Functional safety is a key concern in all industry sectors, be it nuclear plants, medical appliance manufactures or the automotive industry [13,14,17,21,25]. The functional correctness of a component is the guarantee that the component behaves the way it should and fulfills all the functional requirements of the system [1,7,24]. In order to ensure the functional correctness of a component, it is necessary to perform a series of rigorous tests on the target device in the appropriate environment context [27]. Skipping this phase and allowing for a component to be tested based on its design specification, without an actual hardware implementation, would make a significant contribution to reducing the skill, labor, time and money required to develop the component [16,29,30].

In this paper we present a novel approach to functional safety verification, where we integrate functional tests as full-fledged components into a model based architecture developed using OpenModelica [9]. This framework can then be used for dynamic requirement verification at simulation time as well as interfaced with different analysis tools, generating for instance a stochastic Bayesian model, which in turn can be used to produce a failure mode effect and analysis (FMEA) table [2].

The rest of the paper is structured as follows: in Section 2 we introduce the framework presented in this paper; Section 3 illustrates a use-case scenario, whereas Section 4 provides more implementation details. Section 5 discusses a generalization of the framework and in Section 6 some related works are presented. A section summarizing the work and mapping out future research directions concludes the paper.

2. A framework for integrated modeling

Our approach aims to combine in a single tool suite the design and verification stages of the development process. With this goal in mind we define the different parts of the framework.

2.1. The system model

The first step is to choose a language for modeling the physical system. Nowadays a large rooster of design and simulation tools is available. Tools such as Matlab, Simulink [18] or SimulationX [26] provide efficient support for the mathematical aspects of component modeling. However, they lack in support for intuitive modular design aspects. The Modelica language is an object-oriented, equation based language aimed at modeling complex, multi domain physical systems in a structured and intuitive fashion. Moreover, this language is supported by a host of free and commercial tools, in particular by OpenModelica [22], an open source compiler and tool suite, complete with a text and graphical modeling editor (OMEdit) for modeling and simulation of physical systems.

☆ A preliminary version of this paper, titled "An Integrated Toolchain for Model Based Functional Safety", appears in the Proceedings of the International Workshop on Applied Modeling and Simulation, Rome, Italy, September 2012.

* Corresponding author. Tel.: +46 13285724.
E-mail address: olena.rogovchenko@liu.se (L. Rogovchenko-Buffoni).

1877-7503/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jocs.2013.08.009

2.2. A formalism for expressing functional requirements

In order to integrate organically the information needed for validation and verification into the modeling process, we define a formalism that can be used from within the modeling environment. In the prototype presented here we have our attention on the expression of the dependency relations between the different components, through the use of special components, labeled services.

Services allow the user to express the requirements of one component for another. These are often extracted from standards or documentation and take the form of equations over the inputs and outputs of the different components of the system. This is convenient, as Modelica is perfectly suited to express behavior in terms of equations, therefore the requirements can be written in the same language as the model, reducing the semantic gap between the desired and actual behaviors of the system.

This relation can be extended to include other information necessary to the analysis, such as failure probabilities for example.

2.3. Analysis during simulation

Expressing the requirement of the model in the language of the model itself, allows the analyst to benefit from all the powerful simulation capabilities the OpenModelica environment offers. Once the combined model is complete, it is possible to simulate the system ensuring that the constraints specified by the services are verified during the whole runtime.

2.4. Interfacing with analysis tools

It is also possible to interface the framework with tools especially targeted at failure analysis. To achieve this, it is necessary to extract the data in a format that is readable by the tool of our choice. This is done through an interface provided by OpenModelica, which allows to retrieve the relationships between the various components. Fig. 1 gives a graphical representation of the whole framework.

2.5. Bayesian networks

Bayesian networks allow for the specification of risk models that represent the key factors and their inter-relationships (a qualitative model) with probability distributions based on expert judgment or on observed data (a quantitative model) [28]. Bayesian networks are already used for the verification of functional validity [3,15], but the existing approaches require the use of dedicated tools. One of the goals of our work is to breach the gap between the tool used to design and program the component and the verification tool.

2.6. Failure mode and effect analysis

Failure modes and effects analysis (FMEA) [2,19] is a step-by-step approach to identifying all the possible failures in a design. With this approach failures are prioritized according to how serious their consequences are, how frequently they occur and how easily they can be detected. FMEA is applicable right from the conception stages of a component and throughout its entire lifespan. This approach is particularly popular with the automotive and aerospace industries [8,20]. The FMEA table produced by analyzing the component model can thus be used to predict possible failures and prevent them right in the design stages.

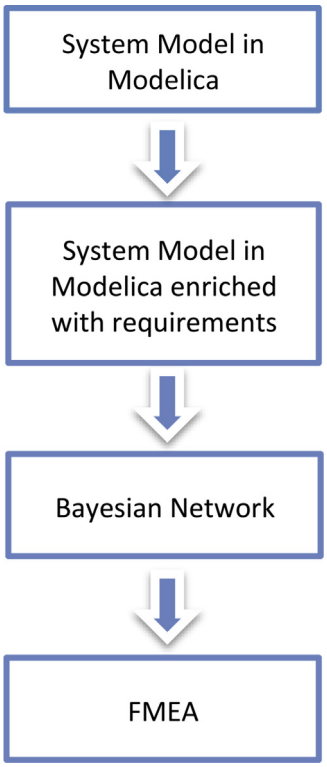

Fig. 1. The modeling framework.

3. Use case scenario

To illustrate our modeling approach we have chosen to model a magnetic valve that is used as a sub-component to start the ignition of an automotive vehicle. Fig. 2 provides a representation of the complete physical model. The rectangular blocks represent the various components and the squares with a circle in the center are the services associated either with an individual component or with the whole environment.

The components and the services are all modeled by classes, in the sense of classical object-oriented programming languages. Services are simply a special kind of components that formalize what is required from one component by another in order to perform the task correctly. These components are interconnected through their interfaces.

Based on the MagneticValve model presented in Fig. 2, a dependency graph is generated by analyzing the relations of the services provided to the different components of the model. We then introduce probabilities at the leaf nodes and perform an inference over the Bayesian Network which allows us to analyze how all the system is affected by a failure. The acronyms AVA, UNA and NF represent available, unavailable and no fault, respectively.

In the first scenario in Fig. 3, we analyze the probability of a component's success and failure, upon fault injection in leaf nodes. Once the Bayesian Network is generated it is automatically opened in GeNie [12], a viewing tool for Bayesian Networks. Here, one can insert probability values into the nodes of the network in order to carry out an inference over the network. Thus in Fig. 3, we have injected a fault into the magnetic valve node, which lies at the bottom right of the figure and after inference we can see that the fault is propagated to all the nodes that are dependent on the magnetic valve node.

In the second scenario in Fig. 4 we investigate the probabilities of the model's success and failure when no fault is injected. For this

Please cite this article in press as: L. Rogovchenko-Buffoni, et al., An integrated toolchain for model based functional safety analysis, J. Comput. Sci. (2013), http://dx.doi.org/10.1016/j.jocs.2013.08.009