Science of Computer Programming 128 (2016) 2–21

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Validation process for railway interlocking systems

A. Bonacchi [a], A. Fantechi [a,*], S. Bacherini [b], M. Tempestini [b]

[a] DINFO – University of Florence, Via S. Marta 3, Firenze, Italy
[b] General Electric Transportation Systems, Firenze, Italy

ARTICLE INFO

Article history:
Received 16 May 2015
Received in revised form 19 April 2016
Accepted 19 April 2016
Available online 2 May 2016

Keywords:
Railway interlocking systems
System validation
Model-based testing
Formal methods
Model checking

ABSTRACT

An interlocking system monitors the status of the objects in a railway yard, allowing or denying the movement of trains, in accordance with safety rules. The high number of complex interlocking rules that guarantee the safe movements of independent trains in a large station makes the verification of such systems a complex task, which needs to be addressed in conformance with EN50128 safety guidelines.

In this paper we show how the problem has been addressed by a manufacturer at the final validation stage of production interlocking systems, by means of a model extraction procedure that creates a model of the internal behaviour, to be exercised with the planned test suites, in order to reduce the high costs of direct validation of the target system.

The same extracted model is then subject to formal verification experiments, employing an iterative verification process implementing slicing and CEGAR-like techniques, defined to address the typical complexity of this application domain.

1. Introduction

Among railway signalling systems, the *interlocking* systems are responsible of allowing or denying, according to established safety and operational regulations, the routing of the trains through stations or railway yards. Safety and operational regulations are generic for the region or country where the interlocking is located, and their instantiation on a particular station or yard topology is usually captured by a so called *control table*. Control tables of modern computer-based interlocking systems are implemented by means of iteratively executed software tests over the status of the yard objects.

One of the most common ways to describe the interlocking rules given by control tables is through boolean equations or, equivalently, ladder diagrams which are interpreted either by a Programmable Logic Controller (PLC) or by a proper evaluation engine over a standard processor.

In particular, in [7], we have introduced an effort made in a cooperation between the University of Florence and the Safety and Validation (S&V) team of General Electric Transportation Systems (GETS) in Florence, with the final aim of reducing the costs of verifying the safety requirements of the produced interlocking systems, in the system validation phase.

The S&V team performs the final independent validation, according to CENELEC EN50128 standard [9], of the produced railway signalling systems, and hence it acts as an independent verifier of the interlocking systems produced by other branches of the company, with little insight of the followed development process, and focusing on the final product.

* Corresponding author.
E-mail addresses: a.bonacchi@unifi.it (A. Bonacchi), alessandro.fantechi@unifi.it (A. Fantechi), stefano.bacherini@ge.com (S. Bacherini), matteo.tempestini@ge.com (M. Tempestini).

http://dx.doi.org/10.1016/j.scico.2016.04.004
0167-6423/

According to the separation between design teams and validation teams, the information accompanying a system that undergoes validation is constituted solely by the controlled station layout, and by a set of test suites defined by the signalling engineers for that particular system.

To gain more insight over the control tables encoded within the system under evaluation, it is possible to extract those control tables by means of proprietary procedures (called in the following *legacy libraries*) that provide the control tables in a proprietary format. For confidentiality reasons, the format cannot be disclosed, nor any fragment of extracted control tables.

Hence, the scope of our work is the modelling of the control tables extracted from the binary files, by means of a *reverse engineering* process, in order for the tests suites to be simulated on the model; the expected advantages are expected to be given by the early validation of the implemented control tables, before a full physical test bench for the specific equipment is built.

The choice of the modelling tool was taken according to specific constraints posed by the S&V team: in order to smoothly adopt this verification technique inside the internal production process, a commercial development/verification tool, already known within the company, was a requirement. This constraint has favoured the choice of Matlab and *Simulink*, using Simulink logic gates to encode boolean functions extracted from the legacy control tables.

Moreover, in [6] we set up a verification framework based on model checking on the extracted model, employing Matlab *Design Verifier* [33], both because it works on Simulink models and to exploit at best its SAT-solving capabilities on the native boolean coding of the control tables. The verification framework exploits environment abstraction, slicing and CEGAR-like techniques, driven by the detailed knowledge of the interlocking product under verification.

The paper extends results from [5–7], which separately described the preliminary application of the proposed model-based testing and formal verification concepts. The two concepts are now jointly presented in an improved way, with more up-to-date details, reporting also how the former is already consolidated into the industrial validation procedure, while the latter is still at an experimental stage.

The paper is organized as follows: Section 2 presents the current activity of system testing in the S&V Laboratory of GETS, placing it in the context of the overall development cycle of safety critical systems in accordance with EN50128. In Section 3 we introduce Ladder Logic, which is used to implement control tables, and we introduce the model extraction process that allows the control tables to be translated into boolean functions implemented in a Simulink model. Section 4 introduces the testing made on the extracted models, showing the results obtained when three example interlocking systems are simulated with given simulation scenarios. In Section 5 we address formal verification discussing in particular the proposed slicing and abstraction techniques. Section 6 concludes the paper.

2. The role of validation in EN50128

CENELEC EN50128 is the standard that specifies the procedures and the technical requirements for the development of programmable electronic devices to be used in railway control and signalling protection [9]. This standard is part of a family, and it refers only to the software components and to their interaction with the whole system. The basic concept of the standard is the *SIL* (Safety Integrity Level). Integrity levels characterize software modules and functions according to their criticality, and range is defined from 0 to 4, where 0 is the lowest level, which refers to software functions for which a failure has no safety effects and 4 is the maximum level, for which a software failure can have severe effects on the safety of system, resulting in possible loss of human life.

2.1. Model-based design

The standard encourages the usage of models and formal methods in every phase of the development cycle for software of the highest SILs, starting from the design to the verification. The rationale is that models are more related to abstract concepts than the technologies used for their implementation into code, and are therefore closer to the domain of the problem. On the other hand, sufficiently detailed models can be used for automatic generation of code: automatic code generation, according to EN50128, requires that the code generator itself is somehow trusted.

In the past years GETS has committed to model-based design, by developing an internal software production process compliant with EN50128, employing the most advanced software production and verification techniques, such as formal methods, model-based design and testing, automatic code generation [15], static analysis based on abstract interpretation [16]. Several insights into this process can be retrieved in [2]; such a process includes a thorough verification activity on software units [16].

2.2. Independent validation

The EN50128 standard proposes three kinds of preferred organizational structures, according to the SIL, for the software development.

The eight roles (Project Manager, Requirement Manager, Designer, Implementer, Integrator, Tester, Verifier and Validator), defined by the standard for SIL3 and SIL4 software (to which we are interested in this context), should be fulfilled by at least