Quantitative Verification of System Safety in Event-B

Anton Tarasyuk\textsuperscript{1,2}, Elena Troubitsyna\textsuperscript{2}, and Linas Laibinis\textsuperscript{2}

\textsuperscript{1} Turku Centre for Computer Science
\textsuperscript{2} Åbo Akademi University
Joukahaisenkatu 3-5, 20520 Turku, Finland
\{anton.tarasyuk,elena.troubitsyna,linas.laibinis\}@abo.fi

Abstract. Certification of safety-critical systems requires formal verification of system properties and behaviour as well as quantitative demonstration of safety. Usually, formal modelling frameworks do not include quantitative assessment of safety. This has a negative impact on productivity and predictability of system development. In this paper we present an approach to integrating quantitative safety analysis into formal system modelling and verification in Event-B. The proposed approach is based on an extension of Event-B, which allows us to perform quantitative assessment of safety within proof-based verification of system behaviour. This enables development of systems that are not only correct but also safe by construction. The approach is demonstrated by a case study – an automatic railway crossing system.

1 Introduction

Safety is a property of a system to not endanger human life or environment \cite{4}. To guarantee safety, designers employ various rigorous techniques for formal modelling and verification. Such techniques facilitate formal reasoning about system correctness. In particular, they allow us to guarantee that a safety invariant – a logical representation of safety – is always preserved during system execution. However, real safety-critical systems, i.e., the systems whose components are susceptible to various kinds of faults, are not “absolutely” safe. In other words, certain combinations of failures may lead to an occurrence of a hazard – a potentially dangerous situation breaching safety requirements. While designing and certifying safety-critical systems, we should demonstrate that the probability of a hazard occurrence is acceptably low. In this paper we propose an approach to combining formal system modelling and quantitative safety analysis.

Our approach is based on a probabilistic extension of Event-B \cite{22}. Event-B is a formal modelling framework for developing systems correct-by-construction \cite{3}. It is actively used in the EU project Deploy \cite{6} for modelling and verifying of complex systems from various domains including railways. The Rodin platform \cite{20} provides the designers with an automated tool support that facilitates formal verification and makes Event-B relevant in an industrial setting.
The main development technique of Event-B is refinement – a top-down process of gradual unfolding of the system structure and elaborating on its functionality. In this paper we propose design strategies that allow the developers to structure safety requirements according to the system abstraction layers. Essentially, such an approach can be seen as a process of extracting a fault tree – a logical representation of a hazardous situation in terms of the primitives used at different abstraction layers. Eventually, we arrive at the representation of a hazard in terms of failures of basic system components. Since our model explicitly contains probabilities of component failures, standard calculations allow us to obtain a probabilistic evaluation of a hazard occurrence. As a result, we obtain an algebraic representation of the probability of safety violation. This probability is defined using the probabilities of system component failures. To illustrate our approach, we present a formal development and safety analysis of a radio-based railway crossing. We believe the proposed approach can potentially facilitate development, verification and assessment of safety-critical systems.

The rest of the paper is organised as follows. In Section 2 we describe our formal modelling framework – Event-B, and briefly introduce its probabilistic extension. In Section 3 we discuss a general design strategy for specifying Event-B models amenable for probabilistic analysis of system safety. In Section 4 we demonstrate the presented approach by a case study. Finally, Section 5 presents an overview of the related work and some concluding remarks.

2 Modelling in Event-B

The B Method [2] is an approach for the industrial development of highly dependable software. The method has been successfully used in the development of several complex real-life applications [19,5]. Event-B is a formal framework derived from the B Method to model parallel, distributed and reactive systems. The Rodin platform provides automated tool support for modelling and verification in Event-B. Currently Event-B is used in the EU project Deploy to model several industrial systems from automotive, railway, space and business domains.

**Event-B Language and Semantics.** In Event-B, a system model is defined using the notion of an **abstract state machine** [18]. An abstract state machine encapsulates the model state, represented as a collection of model variables, and defines operations on this state. Therefore, it describes the dynamic part of the modelled system. A machine may also have an accompanying component, called **context**, which contains the static part of the system. In particular, It can include user-defined carrier sets, constants and their properties given as a list of model axioms. A general form of Event-B models is given in Fig. [1]

The machine is uniquely identified by its name $M$. The state variables, $v$, are declared in the **Variables** clause and initialised in the **Init** event. The variables are strongly typed by the constraining predicates $I$ given in the **Invariants** clause. The invariant clause also contains other predicates defining properties that must be preserved during system execution.