Moving from Event-B to Probabilistic Event-B

Mohamed Amine Aouadhi, Benoit Delahaye, Arnaud Lanoix

Moving from Event-B to Probabilistic Event-B*

Mohamed Amine Aouadhi
University of Nantes
LINA UMR CNRS 6241
mohamed-amine.aouadhi@univ-nantes.fr

Benoît Delahaye
University of Nantes
LINA UMR CNRS 6241
benoit.delahaye@univ-nantes.fr

Arnaud Lanoix
University of Nantes
LINA UMR CNRS 6241
arnaud.lanoix@univ-nantes.fr

ABSTRACT

We propose a fully probabilistic extension of Event-B where all the non-deterministic choices are replaced with probabilities. We present the syntax and the semantics of this extension and introduce novel and adapted proof obligations for proving the correctness of probabilistic Event-B models. As a preliminary step towards handling refinement of probabilistic Event-B models, we propose sufficient conditions for the almost-certain convergence of a set of events and express them in terms of proof obligations. We illustrate our work by presenting a case study specified in both standard and probabilistic Event-B.

CCS Concepts

•Mathematics of computing → Markov processes; •Software and its engineering → Formal software verification; •Theory of computation → *Proof theory;*

Keywords

Event-B; Probabilistic systems; Markov Chains

1. INTRODUCTION

As systems become more and more complex, with randomised algorithms [19], probabilistic protocols [3] or failing components, it is necessary to add new modelling features in order to take into account complex system properties such as reliability [24], responsiveness [8, 23], continuous evolution, energy consumption etc.

In this way, several research works have focused on the extension of Event-B to allow the expression of probabilistic information in Event-B models. Event-B [1] is a formal method used for discrete systems modelling. It is equipped with *Rodin* [2], an open toolset for modelling and proving systems. The development process in Event-B is based on refinement: systems are typically developed progressively using an ordered sequence of models, where each model contains more details than its predecessor.

In [17], Morgan *et al.* have summarised the difficulties of embedding probabilities into Event-B. This seminal paper suggests that probabilities need to be introduced as a refinement of *non-determinism*. In Event-B, non-determinism occurs in several places such as the choice between enabled events in a given state, the choice of the parameter values in a given event, and the choice of the value given to a variable through some non-deterministic assignments. The ideal probabilistic extension of Event-B should therefore allow using probabilities in all these places. To the best of our knowledge, the existing works on extending Event-B with probabilities have mostly focused on refining non-deterministic assignments into probabilistic assignments. In [9], Hallerstede and Hoang propose to focus on a qualitative aspect of probability. They propose to refine non-deterministic assignments into *qualitative* probabilistic assignments where the actual probability values are not specified, and adapt the Event-B semantics and proof obligations to this new setting. In [25], Yilmaz study the refinement of qualitative probabilistic Event-B models and propose a tool support inside Rodin. Other works [20, 22, 21] have extended this approach by refining non-deterministic assignments into *quantitative* probabilistic assignments where, unlike in [9], the actual probability values are specified. This new proposition is then exploited in order to assess several system properties such as reliability and responsiveness.

Unfortunately, other sources of non-determinism than assignments have been left untouched in these works. In [17], the authors argue that probabilistic choice between events or parameter values can be achieved by transformations of the models that embed these choices inside probabilistic assignments. While this is unarguably true, such transformations are not trivial and greatly impede the understanding of Event-B models. Moreover, these transformations would need to be included in the refinement chain when designers need it, which would certainly be counter-intuitive to engineers.

Instead, we propose a different approach in which probabilistic choices can be introduced as a refinement of any potential non-deterministic choice, be it between enabled events, parameter values or assignments. Our long-term goal is to produce a probabilistic extension of Event-B where probabilistic events/parameters/assignments can be introduced natively either as standalone modelling artifacts or as a refinement of their non-deterministic counterparts. This long-term goal is clearly ambitious and will require several years of study to be achieved.

As a first step towards this long-term objective, we consider a slight-

*This work is partially supported by the ANR national research program PACS (ANR-14-CE28-0002).

SAC 2017,April 03-07, 2017, Marrakech, Morocco

http://dx.doi.org/10.1145/3019612.3019823

ly simplified modelling process where the engineer introduces probabilities in the last refinement step of a model, when the system is already sufficiently detailed. For now, we also restrict ourselves to purely probabilistic systems: when probabilities are introduced in the model, they replace *all* non-deterministic choices. We therefore propose a *fully probabilistic* extension of Event-B where *all* non-deterministic choices are replaced with probabilistic ones. As for standard Event-B models, the consistency of probabilistic Event-B models is expressed in terms of proof obligations. We therefore introduce new proof obligations dedicated to the consistency of probabilistic Event-B models and explain how standard Event-B proof obligations can be adapted to the probabilistic setting. In order to prove the correctness of our approach, we show that the semantics of a probabilistic Event-B model is a (potentially infinite-state) discrete time Markov chain.

As explained in [17], ensuring the refinement of Event-B models where probabilistic choice is not reduced to assignments is a difficult problem. While we do not solve this problem in its entirety, we take a preliminary step towards this goal by providing sufficient conditions, expressed in terms of proof obligations, for the almost-certain convergence of a set of events. Convergence is a required property in standard Event-B for proving refinement steps as soon as new events are introduced in the model. Almost-certain convergence has already been studied in [9] and [12], in the context of non-deterministic models with probabilistic assignments, but we show that the proof obligations developed in this context are not sufficient for models where probabilistic choice also appears in the choice of events and parameter valuations.

Finally, we illustrate our work on a classical case study: the emergency brake system. In particular, we show that some of the requirements provided in this case study cannot be taken into account using standard Event-B while their specification using probabilistic Event-B is intuitive, in particular when probabilities can be taken into account for the choice between enabled events.

All the results we present in this paper are being implemented in a prototype plugin for Rodin, which we briefly present at the end of the paper.

Related work. As said above, [17] is a seminal paper that identifies the challenges when considering a probablistic extension for Event-B: introducing probabilities in the three places where non-determinism appears in standard Event-B (between enabled events, on the parameter values choice and on non-deterministic assignments) is the major dificulty to preserve the practicality of Event-B, and in particular the refinement development framework.

Existing works such as the book of Morgan and McIver [18] partially answer these challenges. In particular, [18] introduces a probabilistic refinement calculus where the refinement of probabilistic guarded commands ($\equiv$ assignements) is worked out. The PhD work of Hoang [16, 14, 11] adapts these results to the classical B-Method and the underlying guarded substitution language. Following these works, Hallerstede and Hoang [9] have proposed a first probabilistic extension of Event-B, where probabilities are introduced as a refinement of non-determinism in non-deterministic assignments. In this first extension, Hallerstede and Hoang focus on a *qualitative* aspect of probabilities and adapt the Event-B semantics and proof obligations to this new setting. *Quantitative* probabilistic assignments are then introduced in [20, 22, 21]. Almost-certain convergence of a set of events is studied in [9] and [12] in this context of non-deterministic models with probabilistic assignments. Event-B refinement of such models is studied in [25]. We show in Section 6 that the proof obligations dedicated to almost-certain convergence for non-deterministic systems with probabilistic assignments cannot be adapted to our setting and that additionnal proof obligations are required. The inherent complexity of introducing probabilistic choice between events and for parameter valuations is such that the introduction of a functionnal refinement procedure in our setting is still out of our reach, although we are making progress as Section 6 shows.

Outline. The paper is structured as follows. Section 2 presents an overview of the Event-B method and of our running case study. In Section 3, we introduce the syntax of fully probabilistic Event-B and illustrate our approach on the running case study. Section 4 presents new and modified proof obligations for the consistency of probabilistic Event-B models. The semantics of a fully probabilistic Event-B model is described in Section 5 and Section 6 treats the almost-certain convergence of fully probabilistic Event-B models. Finally, Section 7 concludes and presents hints for future work. For space reasons, the full proofs of our results as well as additional material and examples are presented in a separate appendix, to be consulted at the discretion of the reviewers. cd

2. EVENT-B

We first present the basic elements of the Event-B method and then introduce our running case study.

2.1 Preliminaries

Event-B [1] is a formal method used for the development of complex systems. Systems are described in Event-B by means of models. For the sake of simplicity, we assume in the rest of the paper that an Event-B model is expressed by a tuple $M = (\bar{v}, I(\bar{v}), V(\bar{v}), \mathsf{Evts}, \mathsf{Init})$ where $\bar{v} = \{\mathsf{v}_1 \ldots \mathsf{v}_n\}$ is a set of variables, $I(\bar{v})$ is an invariant, $V(\bar{v})$ is an (optional) variant used for proving the convergence of the model, Evts is a set of events and $\mathsf{Init} \in \mathsf{Evts}$ is an initialisation event. The invariant $I(\bar{v})$ is a conjunction of predicates over the variables of the system specifying properties that must always hold.

Events. An event has the following form:

$$\textbf{event } \mathsf{e}_i \mathrel{\widehat{=}} \textbf{any } \bar{t} \textbf{ where } \mathsf{G}_i(\bar{t},\bar{v}) \textbf{ then } \mathsf{S}_i(\bar{t},\bar{v}) \textbf{ end}$$

where e_i is the name of the event, $\bar{t} = \{\mathsf{t}_1 \ldots \mathsf{t}_n\}$ represents the set of parameters of the event, $\mathsf{G}_i(\bar{t},\bar{v})$ is the guard of the event and $\mathsf{S}_i(\bar{t},\bar{v})$ is the action of the event. An event is *enabled* in a given valuation of the variables (also called a configuration) if and only if there exists a parameter valuation such that its guard $\mathsf{G}_i(\bar{t},\bar{v})$ is satisfied in this context. Parameters and guards are optional. The action $\mathsf{S}_i(\bar{t},\bar{v})$ of an event may contain several assignments that are executed in parallel. Assignments can be written in several forms in standard Event-B, but they can always be reduced to what we call *Predicate assignments* in the following. In order to simplify the writing of (Probabilistic) Event-B programs, we only distinguish the following three forms of assignments:

- *Deterministic assignment*: $\mathsf{x} := \mathsf{E}(\bar{t}, \bar{v})$ means that the expression $\mathsf{E}(\bar{t},\bar{v})$ is assigned to the variable x.
- *Predicate assignment*: $\mathsf{x} :| \mathsf{Q}_x(\bar{t},\bar{v},\mathsf{x}')$ means that the variable x is assigned a new value x' such that the predicate $\mathsf{Q}_x(\bar{t},\bar{v},\mathsf{x}')$ is satisfied.

- *Enumerated assignment*: $x :\in \{E_1(\bar{t},\bar{v}) \ldots E_n(\bar{t},\bar{v})\}$ means that the variable x is assigned a new value taken from the set $\{E_1(\bar{t},\bar{v}) \ldots E_n(\bar{t},\bar{v})\}$.

Before-after predicate. The formal semantics of an assignment is described by means of a before-after predicate (BAP) $Q_x(\bar{t},\bar{v},x')$, which describes the relationship between the values of the variable before (x) and after (x') the execution of an assignment.

- The BAP of a deterministic assignment is $x' = E(\bar{t},\bar{v})$.
- The BAP of a predicate assignment is $Q_x(\bar{t},\bar{v},x')$.
- The BAP of an enumerated assignment is $x' \in \{E_1(\bar{t},\bar{v}) \ldots E_n(\bar{t},\bar{v})\}$.

Recall that the action $S_i(\bar{t},\bar{v})$ of a given event may contain several assignments that are executed in parallel. Assume that $v_1 \ldots v_i$ are the variables assigned in $S_i(\bar{t},\bar{v})$ – variables $v_{i+1} \ldots v_n$ are thus not modified – and let $Q_{v_1}(\bar{t},\bar{v},v'_1) \ldots Q_{v_i}(\bar{t},\bar{v},v'_i)$ be their corresponding BAP. Then the BAP $S_i(\bar{t},\bar{v},\bar{v}')$ of the event action $S_i(\bar{t},\bar{v})$ is:

$$S_i(\bar{t},\bar{v},\bar{v}') \;\widehat{=}\; Q_{v_1}(\bar{t},\bar{v},v'_1) \wedge \ldots \wedge Q_{v_i}(\bar{t},\bar{v},v'_i) \wedge (v'_{i+1}=v_{i+1}) \wedge \ldots \wedge (v'_n=v_n)$$

Proof obligations. The consistency of a standard Event-B model is characterised by *proof obligations* (POs) which must be discharged. These POs allow to prove that the model is sound with respect to some behavioural semantics. Formal definitions of all the standard Event-B POs are given in [1]. In the following, we only recall the most important of them: (event/INV) for *invariant preservation*, which states that the invariant still holds after the execution of each event in the Event-B model M. Given an event e_i with guard $G_i(\bar{t},\bar{v})$ and action $S_i(\bar{t},\bar{v})$, this PO is expressed as follows:

$$I(\bar{v}) \wedge G_i(\bar{t},\bar{v}) \wedge S_i(\bar{t},\bar{v},\bar{v}') \vdash I(\bar{v}') \qquad \text{(event/INV)}$$

2.2 The Emergency Brake System

We now introduce our running example, based on a simplified scenario of the emergency brake system in charge of manoeuvring the brake of a vehicle.

Specification. To command the brake, a pedal is provided to the driver: when the pedal is switched to "down", the brake must be applied; when the pedal is switched to "up", the brake must be released. Some requirements constrain the model:

R1. Pedal failure: when the driver tries to switch "down" the pedal, it may stay in the same position;

R2. Risk of pedal failure: the risk of pedal failure is set to 10%;

R3. Brake failure: the brake may not be applied although the pedal has been switched down;

R4. Maximum brake wear: the brake cannot be applied more than a fixed number of times;

R5. Brake wear: due to brake wear, the risk of brake failure increases each time the brake is applied.

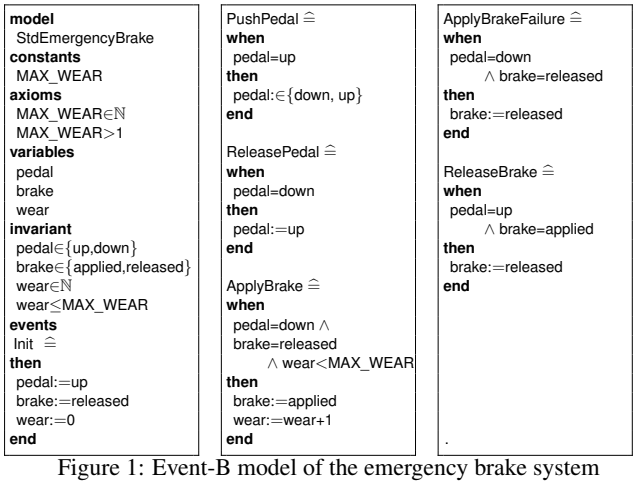

Figure 1: Event-B model of the emergency brake system

Event-B model. The model StdEmergencyBrake given in Fig. 1 presents an Event-B specification of the emergency brake system. The state of the system is described by means of three variables: pedal models the driver command, brake represents the state of the emergency brake (applied or released) and wear counts the number of times the brake is applied. The constant MAX_WEAR represents the maximum number of times the brake can be applied.

The event PushPedal models the driver command, i.e, switching the pedal to down. For taking into account the possible pedal failure mentioned in R1, we use an enumerated non-deterministic assignment pedal :∈ {down,up} to express that the pedal is switched to down (the attempted behaviour) or remains in the up position (failure). Using standard Event-B, we cannot take into account the quantitative risk of failure expressed in R2. The event ApplyBrake models the brake application, *i.e.* the variable brake is assigned the value applied (and the variable wear is increased). The event ApplyBrakeFailure models failure during the brake application: the value of variable brake remains released. When wear<MAX_WEAR, the events ApplyBrake and ApplyBrakeFailure are enabled at the same time (when pedal=down ∧ brake=released), the subsequent non-determinism between these two events reflects requirement R3. On the other hand, when wear=MAX_WEAR, ApplyBrakeFailure is enabled while the guard of ApplyBrake is not satisfied. Therefore, the brake necessarilly fails as soon as wear=MAX_WEAR, which means that the brake event cannot be triggered more than MAX_WEAR times (the maximum brake wear) as expressed by R4. Requirement R5 cannot be modelled in standard Event-B.

3. PROBABILISTIC EVENT-B

The typical way of defining a probabilistic Event-B model from a classical Event-B model *M* is to go through *M* and replace all occurrences of non-deterministic choices with probabilistic choices. In Event-B, non-determinism can appear in three places: the choice of the enabled event to be executed, the choice of the parameter value to be taken and the choice of the value to be assigned to a given variable in a non-deterministic assignment. In the following, we go through these three sources of non-determinism and explain how to turn them into probabilistic choices.

3.1 Introducing probabilistic choices

In standard Event-B, when several events are enabled in a given

configuration, the event to be executed is chosen non-deterministically. In order to resolve this non-deterministic choice, we propose to equip each probabilistic event with a *weight*. In configurations where several probabilistic events are enabled, the probability of choosing one of them will therefore be computed as the ratio of its weight against the total value of the weights of all enabled events in this state. Using weights instead of actual probability values is convenient as the set of enabled events evolves with the configuration of the system. Moreover, for the sake of expressivity, we propose to express the weight $W_i(\bar{v})$ of a probabilistic event e_i as an expression over the variables $\bar{v}$ of the probabilistic Event-B model. The probability of executing a given event can therefore evolve as the system progresses. A probabilistic event is therefore allowed to be executed only if *i)* its guards is fulfilled and *ii)* its weight is strictly positive.

In standard Event-B, events can be equipped with parameters. In each configuration where this is possible, a valuation of the parameters is chosen such that the guard $G_i(\bar{t},\bar{v})$ of the event is satisfied. When there are several such parameter valuations, one of them is selected non-deterministically. We therefore propose to replace this non-deterministic choice by a uniform choice over all parameter valuations ensuring that the guard of the event is satisfied. The uniform distribution is a default choice but our results can be extended to any other discrete distribution.

Recall that non-deterministic assignments in Event-B are expressed in two forms: predicate non-deterministic assignments and enumerated non-deterministic assignments. We propose to replace predicate non-deterministic assignments by *predicate probabilistic assignments* written

$$x :\oplus \mathsf{Q}_x(\bar{t},\bar{v},x')$$

Instead of choosing non-deterministically among the values of x' such that the predicate $\mathsf{Q}_x(\bar{t},\bar{v},x')$ is true as in standard predicate non-deterministic assignments, we propose to choose this new value using an uniform distribution. For simplicity reasons, we enforce that this uniform distribution must be discrete, and therefore that the set of values x' such that $\mathsf{Q}_x(\bar{t},\bar{v},x')$ is true must always be finite. As above, the uniform distribution we propose by default could be replaced by any other discrete distribution.

We propose to replace enumerated non-deterministic assignments by *enumerated probabilistic assignments* written

$$x := E_1(\bar{t},\bar{v})@_{p_1} \oplus \ldots \oplus E_m(\bar{t},\bar{v})@_{p_m}$$

In this structure, the variable x is assigned the expression E_i with probability p_i. In order to define a correct probability distribution, each p_i must be strictly positive and smaller or equal to 1, and they must sum up to 1. Although rational numbers are not natively handled in Event-B, we assume that an adequate context is present. That can be done by defining a "Rational" theory in Rodin using the theory plug-in providing capabilities to define and use mathematical extensions to the Event-B language and the proving infrastructure [7].

Remark that standard deterministic assignments are conserved, but can also be considered as enumerated probabilistic assignments where $m = 1$.

3.2 Syntax

Turning all non-deterministic choices into probabilistic choices has side effects on the syntax of events and models. In probabilistic Event-B, we therefore propose to use the following syntax for a probabilistic event e_i :

$$e_i \;\widehat{=}\; \textbf{weight}\ W_i(\bar{v})\ \textbf{any}\ \bar{t}\ \textbf{where}\ G_i(\bar{t},\bar{v})\ \textbf{then}\ S_i(\bar{t},\bar{v})\ \textbf{end}$$

where $W_i(\bar{v})$ is the weight of the event, $G_i(\bar{t},\bar{v})$ is the guard of the event and $S_i(\bar{t},\bar{v})$ is a *probabilistic action*, *i.e.* an action consisting only of deterministic and *probabilistic* assignments which are executed in parallel.

For simplicity reasons we impose, as in standard Event-B, that the initialisation event must be deterministic. The results we present in the rest of the paper can nevertheless easily be extended to probabilistic initialisation events.

Definition 1 (Probabilistic Event-B Model). A probabilistic Event-B model is a tuple $M = (\bar{v}, I(\bar{v}), \mathsf{PEvts}, \mathsf{Init})$ where $\bar{v}$ is a set of variables, $I(\bar{v})$ is the invariant, PEvts is a set of *probabilistic* events and Init is the initialisation event.

3.3 Running Example

A probabilistic version of the emergency brake system from Section 2.2 is given in Fig. 2. This model has the same variables pedal, brake and wear, the same invariants and the same events as the Event-B model StdEmergencyBrake from Fig. 1. Remark that, unlike in standard Event-B, requirements R2 and R5 can be taken into account in this probabilistic version. R2 is specified in the probabilistic event PushPedal by using an enumerated probabilistic assignment instead of a non-deterministic assignment: the variable pedal is assigned the value down with a probability $9/10$ (attempted behaviour) and the value up with a probability $1/10$ (failure), hence resulting in a risk of pedal failure of 10%. Requirement R5 is taken into account by annotating probabilistic event ApplyBrake with a weight MAX_WEAR−wear and probabilistic event ApplyBrakeFailure with a weight wear. As the probabilistic event ApplyBrake increases the variable wear when it is executed, the weight of the probabilistic event ApplyBrake decreases each time it is executed whereas the weight of the probabilistic event ApplyBrakeFailure in-

model
ProbaEmergencyBrake

constants
MAX_WEAR

axioms
MAX_WEAR∈ℕ
MAX_WEAR>1

variables
pedal
brake
wear

invariant
pedal∈{up,down}
brake∈{applied,released}
wear∈ℕ
wear ≤ MAXWEAR

events

Init ≙
then
pedal:=up
brake:=released
wear:=0
end

PushPedal ≙
weight
MAX_WEAR
when
pedal=up
then
pedal:= down @9/10
⊕ up @1/10
end

ReleasePedal ≙
weight
MAX_WEAR
when
pedal=down
then
pedal:=up
end

ApplyBrake ≙
weight
MAX_WEAR − wear
when
pedal=down ∧
brake=released
∧ wear < MAXWEAR
then
brake:=applied
wear:=wear+1
end

ApplyBrakeFailure ≙
weight
wear
when
pedal=down
∧ brake=released
then
brake:=released
end

ReleaseBrake ≙
weight
MAX_WEAR − wear
when
pedal=up
∧ brake=applied
then
brake:=released
end

.

Figure 2: Probabilistic model of the emergency brake system

creases. The failure of the brake is modelled by means of a probabilistic choice between ApplyBrake and ApplyBrakeFailure instead of a non-deterministic choice as in the standard version, which implies that the more ApplyBrake is executed, the higher the probability that ApplyBrakeFailure occurs instead. In this version, all requirements are therefore taken into account.

4. CONSISTENCY

As in standard Event-B, the consistency of a probabilistic Event-B model is defined by means of proof obligations (POs). In this section, we therefore introduce new POs specific to probabilistic Event-B and explain how we adapt standard Event-B POs in order to prove the consistency of probabilistic Event-B models.

4.1 Specific POs for Probabilistic Event-B

Numeric weight. For simplicity reasons, we impose that the expression $\mathsf{W}_i(\bar{v})$ representing the weight of a given probabilistic event must evaluate to natural numbers.

$$\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \vdash \mathsf{W}_i(\bar{v}) \in \mathsf{NAT} \qquad \text{(event/WGHT/NAT)}$$

Parameter values finiteness. In order to be able to use a discrete uniform distribution over the set of parameter valuations ensuring that the guard of a probabilistic event is satisfied, we impose that this set must be finite.

$$\mathsf{I}(\bar{v}) \vdash \text{finite}(\{\bar{t} \mid \mathsf{G}_i(\bar{t},\bar{v})\}) \qquad \text{(event/param/pWD)}$$

Enumerated probabilistic assignments well-definedness and feasibility. In all enumerated probabilistic assignments, it is necessary to ensure that the discrete probability values $\mathsf{p}_1 \ldots \mathsf{p}_n$ define a correct probability distribution. Formally, this leads to two POs:

1. Probability values p_i in enumerated probabilistic assignments are strictly positive and smaller or equal to 1.

   $$\vdash 0 < \mathsf{p}_i \leq 1 \qquad \text{(event/assign/pWD1)}$$

2. The sum of the probability values $\mathsf{p}_1 \ldots \mathsf{p}_n$ in enumerated probabilistic assignments must be equal to 1.

   $$\vdash \mathsf{p}_1 + \ldots + \mathsf{p}_n = 1 \qquad \text{(event/assign/pWD2)}$$

Feasability of enumerated probabilistic assignments is trivial: as soon as at least one expression $\mathsf{E}_i(\bar{t},\bar{v})$ is present and well-defined, it always returns a value.

Predicate probabilistic assignment well-definedness and feasibility. To define a discrete uniform distribution over the set of values of a variable x making the predicate $\mathsf{Q}_x(\bar{t},\bar{v},\mathsf{x}')$ of the corresponding assignment satisfied, this set must be finite.

$$\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \wedge \mathsf{W}_i(\bar{v})>0 \vdash \text{finite}(\{\mathsf{x}' \mid \mathsf{Q}_x(\bar{t},\bar{v},\mathsf{x}')\}) \qquad \text{(event/assign/pWD3)}$$

Feasibility of predicate probabilistic assignments is ensured by the standard feasibility PO [1] inherited from Event-B. It ensures that the set $\{\mathsf{x}' \mid \mathsf{Q}_x(\bar{t},\bar{v},\mathsf{x}')\}$ is not empty.

4.2 Modifications to Standard POs

Where standard Event-B POs are concerned, the main difference in probabilistic Event-B is the condition for a probabilistic event to be enabled. Indeed, while it suffices to show that the guard of an event is satisfied for this event to be enabled in standard Event-B, we also have to show in probabilistic Event-B that its weight is strictly positive. We therefore modify standard Event-B POs as follows.

Invariant preservation. The invariant must be preserved by all enabled probabilistic events.

$$\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \wedge \mathbf{W}_i(\bar{v}) > \mathbf{0} \wedge \mathsf{S}_i(\bar{t},\bar{v},\bar{v}') \vdash \mathsf{I}(\bar{v}') \qquad \text{(event/pINV)}$$

Deadlock freedom. In all acceptable configurations, there must exist at least one enabled probabilistic event.

$$\mathsf{I}(\bar{v}) \vdash (\mathsf{G}_1(\bar{t},\bar{v}) \wedge \mathbf{W}_1(\bar{v}) > \mathbf{0}) \vee \ldots \vee (\mathsf{G}_n(\bar{t},\bar{v}) \wedge \mathbf{W}_n(\bar{v}) > \mathbf{0}) \qquad \text{(model/pDLF)}$$

For the sake of understanding, we hereby insist on the separation between the guard of an event, which reflects the classical notion of enabledness, and the fact that its weight must be strictly positive. Obviously, one could also automatically re-write the guard of all probabilistic events in order to include the condition on its weight. This solution would allow conserving most of the standard Event-B consistency POs without modifications in the probabilistic setting.

5. SEMANTICS

Semantics of standard Event-B models can be expressed in terms of Labelled Transition Systems [6]. Informally, given an Event-B model $M = (\bar{v}, I(\bar{v}), \mathsf{Evts}, \mathsf{Init})$, its semantics is the LTS $\mathcal{M} = (S, s_0, AP, L, \mathsf{Acts}, T)$ where S is a set of states, Acts is the set of actions (event names), $s_0 \in S$ is the initial state obtained by executing the Init event, AP is the set of valuations of the variables in $\bar{v}$ that satisfy the invariant $I(\bar{v})$, $L : S \to AP$ is a labelling function that provides the valuations of the variables in a given state, and $T \subseteq S \times \mathsf{Acts} \times S$ is the transition relation corresponding to the actions of the events. In the following, we extend this work by presenting the semantics of probabilistic Event-B models in terms of Discrete Time Markov Chains (DTMC).

5.1 Notations

Let $M = (\bar{v}, I(\bar{v}), \mathsf{PEvts}, \mathsf{Init})$ be a probabilistic Event-B model and σ be a valuation its variables. Given a variable $x \in \bar{v}$, we write $[\sigma]x$ for the value of x in σ. Given an expression $E(\bar{v})$ over variables in $\bar{v}$, we write $[\sigma]E(\bar{v})$ (or $[\sigma]E$ when clear from the context) for the evaluation of $E(\bar{v})$ in the context of σ. Given a probabilistic event e_i with a set of parameters $\bar{t}$ and a valuation σ of the variables, we write $T_\sigma^{e_i}$ for the set of parameter valuations θ such that the guard of e_i evaluates to true in the context of σ and θ. Formally, $T_\sigma^{e_i} = \{\theta \mid [\sigma,\theta]G_i(\bar{t},\bar{v}) = true\}$. Recall that parameter valuations are chosen uniformly on this set. We write $P_{T_\sigma^{e_i}}$ for the uniform distribution on the set $T_\sigma^{e_i}$. Given a valuation σ of the variables and a probabilistic event e_i, we say that e_i is *enabled* in σ iff (a) the weight of e_i evaluates to a strictly positive value in σ and (b) either e_i has no parameter and its guard evaluates to true in σ or there exists at least one parameter valuation θ such that the guard of e_i evaluates to true in the context of σ and θ, i.e. $T_\sigma^{e_i} \neq \emptyset$. Given a probabilistic event e_i, we write $Var(e_i)$ for the set of variables in $\bar{v}$ that are modified by the action of e_i, i.e. the variables that appear on the left side of an assignment in $S_i(\bar{t},\bar{v})$. Recall that a variable $x \in Var(e_i)$ must be on the left side of either a predicate probabilistic assignment or a enumerated probabilistic assignment. In both cases, given an original valuation σ of the variables, a valuation θ of the parameters of e_i and a target valuation σ' of the variables, we

write $P^{e_i}_{\sigma,\theta}(x,\sigma')$ for the probability that x is assigned the new value $[\sigma']x$ when executing e_i from the valuation σ and with parameter valuation θ. If e_i is not equipped with parameters, this is written $P^{e_i}_{\sigma}(x,\sigma')$. In the following, we use the more general notation and assume that it is replaced with the specific one when necessary. The formal definition of $P^{e_i}_{\sigma,\theta}(x,\sigma')$ is given in [5].

5.2 DTMC semantics

Informally, the semantics of a probabilistic Event-B model $M = (\bar{v}, I(\bar{v}), \mathsf{PEvts}, \mathsf{Init})$ is a Probabilistic LTS $[\![M]\!] = (S, s_0, AP, L, \mathsf{Acts}, P)$ where the states, labels, actions, atomic propositions and initial state are similarly obtained as for the standard LTS semantics of Event-B. The only difference with the standard LTS semantics is that the transitions are equipped with probabilities, which we explain below. In the following, we identify the states with the valuations of the variables defined in their labels.

Intuitively, the transition probabilities are obtained as follows: Let $e_i \in \mathsf{PEvts}$ be a probabilistic event, $x \in \bar{v}$ be a variable and s, s' be two states of $[\![M]\!]$ such that (s, e_i, s') is a transition in the standard LTS semantics, i.e. where e_i is enabled in s and there exists a parameter valuation $\theta \in T_s^{e_i}$, if any, such that the action of e_i may take the system from s to s'. The probability assigned to transition (s, e_i, s') is then equal to the product of ***(1)*** the probability that the event e_i is chosen from the set of enabled events in state s, ***(2)*** the probability of choosing each parameter valuation θ, and ***(3)*** the overall probability that each modified variable is assigned the value given in s' under parameter valuation θ.

Definition 2 (Probabilistic Event-B Semantics). The semantics of a probabilistic Event-B model $M = (\bar{v}, I(\bar{v}), \mathsf{PEvts}, \mathsf{Init})$ is a *PLTS* $[\![M]\!] = (S, s_0, AP, L, \mathsf{Acts}, P)$ where S is a set of states where each state is uniquely identified by its label, $s_0 \in S$ is the initial state obtained after the execution of the Init event, AP represents the valuations of all variables that satisfy the invariant of the model: $AP = \{\sigma \mid [\sigma]I(\bar{v}) = \mathsf{true}\}$, $L : S \to AP$ is the labelling function that assigns to each state the corresponding valuation of the variables, Acts is the alphabet of actions (event names), and $P : S \times \mathsf{Acts} \times S \to [0,1]$ is the transition probability function such that for a given state s, for all $e_i, s' \in \mathsf{Acts} \times S$, we have $P(s, e_i, s') = 0$ if $e_i \notin \mathsf{Acts}(s)$ or $\exists x \in X \backslash \{Var(e_i)\}$ st $[s]x \neq [s']x$ and otherwise

$$P(s,e_i,s') = \underbrace{\frac{[s]W_i(\bar{v})}{\sum_{e_j \in \mathsf{Acts}(s)} [s]W_j(\bar{v})}}_{(1)} \times \sum_{\theta \in T_s^{e_i}} \Big(\underbrace{P_{T_s^{e_i}}(\theta)}_{(2)} \times \underbrace{\prod_{x \in Var(e_i)} P^{e_i}_{s,\theta}(x,s')}_{(3)} \Big)$$

In the following proposition, we show that the semantics of a probabilistic Event-B model as defined above is indeed a DTMC. For space reasons, the proof of this proposition is given in [5].

Proposition 1. The semantics of a probabilistic Event-B model M satisfying the POs given in Section 4.1 is a *DTMC*.

For space reasons, the DTMC of the probabilistic emergency brake system is given in [5].

6. CONVERGENCE

The development process in Event-B is inherently based on refinement. As said earlier, systems are typically developed progressively using an ordered sequence of models, where each model contains more details than its predecessor. One key aspect of refinement is the addition, in one refinement step, of new variables and new events that characterize the evolution of those variables. In order to preserve certain properties, it is then necessary to show that the introduction of these new events in a refined model cannot prevent the system from behaving as specified in the abstract model. In particular, it is necessary to show that such new events are "convergent", in the sense that they cannot keep control indefinitely: at some point the system has to stop executing new events in order to follow the behaviour specified in its abstract model.

Although this paper does not adress refinement in probabilistic Event-B, we propose a solution in order to prove that a given set of events almost-certainly converges in a probabilistic Event-B model, which is a necessary step for adressing refinement in the future. We therefore start this section with a brief recall of how events can be proven convergent in standard Event-B and then propose a set of sufficient conditions, expressed as POs, that allow proving that a set of events is almost-certainly convergent in probabilistic Event-B.

Convergence in Standard Event-B. In order to prove that a set of events is convergent in Event-B, one has to show that it is not possible to keep executing convergent events infinitely, and therefore that a non-convergent event is eventually performed from any state. The classical solution is therefore to introduce a natural number expression $\mathsf{V}(\bar{v})$, called a *variant*, and show that all convergent events strictly decrease the value of this variant. As a consequence, when the variant hits zero, it is guaranteed that no convergent event can be performed. In practice, this is expressed using two POs:

1. **Numeric variant.** Under the guard $\mathsf{G}_i(\bar{t},\bar{v})$ of each convergent event e_i, the variant $\mathsf{V}(\bar{v})$ is greater than 0.

   $\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \vdash \mathsf{V}(\bar{v}) \in \mathsf{NAT}$ (event/var/NAT)

2. **Convergence.** The action $\mathsf{S}_i(\bar{t},\bar{v})$ of each convergent event e_i *must always* decrease the variant $\mathsf{V}(\bar{v})$.

   $\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \vdash \forall\, \bar{v}'.\ \mathsf{S}_i(\bar{t},\bar{v},\bar{v}') \Rightarrow \mathsf{V}(\bar{v}') < \mathsf{V}(\bar{v})$ (event/VAR)

Almost-certain Convergence in the litterature. In the context of probabilistic Event-B, instead of proving that a given set of events necessarily converges as in standard Event-B, we are interested in showing that a given set of events *almost-certainly* converges. In other words, we are interested in showing that, in all states of the system where convergent events can be executed, the probability of eventually taking a non-convergent event or reaching a deadlock is 1 (the probability of infinitely executing convergent events is 0).

This property has already been investigated in [9] and [12], in the context of events having probabilistic actions but where non-determinism is still present between events. In this context, Hallerstede and Hoang propose in [9] sufficient conditions for a set of events to almost-certainly converge. These conditions can be summarized as follows: As in standard Event-B, one needs to exhibit a natural number expression $\mathsf{V}(\bar{v})$ called a variant, but unlike in the standard setting, only one resulting valuation of the execution of *each* convergent event needs to decrease this variant. Indeed, in this case, the probability of decreasing the variant is strictly positive. Unfortunately, using such a permissive condition is not sufficient: there might also be a strictly positive probability of increasing the variant. Therefore, Hallerstede and Hoang require the introduction of

another natural number expression $\mathsf{U}(\bar{v})$ which must maximise the variant $\mathsf{V}(\bar{v})$ and never increase. The proposition from [9] is refined in [12], where Hoang requires in addition that the probabilities considered in probabilistic assignments are bounded away from 0. This is ensured by requiring that the set of values that can be returned by a probabilistic assignment is finite.

Adaptation to fully probabilistic Event-B. We now show how to adapt the results proposed in [9] and [12] to our fully probabilistic Event-B setting. Since there are no non-deterministic choices between enabled events, it is not anymore necessary to require that *all* enabled events in a given configuration may decrease the variant. We therefore start by relaxing the condition proposed in [9]: we only require that, in all configurations where a convergent event is enabled, there is *at least one* convergent event for which at least one resulting valuation decreases the variant.

1. **Almost-certain convergence.** In all configurations where at least one convergent event is enabled, there must exist at least one valuation $\bar{v}$' obtained after the execution of one of these enabled events which decreases the variant.

$$\mathsf{I}(\bar{v}) \wedge ((\mathsf{G}_1(\bar{t},\bar{v}) \wedge \mathsf{W}_1(\bar{v}) > 0) \vee \ldots \vee (\mathsf{G}_i(\bar{t},\bar{v}) \wedge \mathsf{W}_i(\bar{v}) > 0)) \vdash$$
$$(\exists\, \bar{v}'.\ \mathsf{G}_1(\bar{t},\bar{v}) \wedge \mathsf{W}_1(\bar{v}) > 0 \wedge \mathsf{S}_1(\bar{t},\bar{v},\bar{v}') \wedge \mathsf{V}(\bar{v}') < \mathsf{V}(\bar{v})) \vee \ldots \vee$$
$$(\exists\, \bar{v}'.\ \mathsf{G}_i(\bar{t},\bar{v}) \wedge \mathsf{W}_i(\bar{v}) > 0 \wedge \mathsf{S}_i(\bar{t},\bar{v},\bar{v}') \wedge \mathsf{V}(\bar{v}') < \mathsf{V}(\bar{v})) \quad \text{(model/pVar)}$$

As in [9], we also require that convergent events can only be enabled when the variant is positive and that the variant is bounded above. In order to simplify the reasoning, we propose to use a constant bound U, as in [12].

2. **Numeric variant.** Convergent events can only be enabled when the variant is greater or equal to 0.

$$\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \wedge \mathbf{W}_i(\bar{v}) > \mathbf{0} \vdash \mathsf{V}(\bar{v}) \in \mathsf{NAT} \quad \text{(event/var/pNAT)}$$

3. **Bounded variant.** Convergent events can only be enabled when the variant is less or equal to U.

$$\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \wedge \mathbf{W}_i(\bar{v}) > \mathbf{0} \vdash \mathsf{V}(\bar{v}) \leq \mathsf{U} \quad \text{(event/pBOUND)}$$

Finally, the finiteness of the set of values that can be returned by a probabilistic assignment is already ensured by the syntax for enumerated probabilistic assignments and by PO (event/assign/pWD3) for predicate probabilistic assignments and their non-emptyness is ensured by the standard feasability POs.

Inadequacy of adapted POs. Unfortunately, as we deal with potentially infinite-state systems, POs 1–3 presented above are not anymore sufficient for proving that the probability of eventually executing a non-convergent event or reaching a deadlock state is 1. Indeed, although the probability of decreasing the variant is always strictly positive because of PO (model/pVar) and although the number of values that can be returned by a given probabilistic assignment is always finite, the combination of event weights and parameter choice can make this value infinitely small in some cases. In this case, it is well known that almost-certain reachability/convergence is not ensured. This problem is a direct consequence of the unboundedness of the weights of convergent events, which, by getting arbitrarily big, cause the probability of decreasing the variant to get arbitrarily small. Examples illustrating this fact are given in [5].

Additional Proof Obligations. We therefore adapt classical results from infinite-state DTMC to our setting and propose sufficient conditions in terms of proof obligations to prove the almost-certain convergence of a given set of events. Informally, the following POs ensure that the probability of decreasing the variant cannot get infinitely small by requiring that both the weights of convergent events and the number of potential values given to parameters in convergent events are bounded.

4. **Bounded weight.** The weight of all convergent events must be bounded above by a constant upper bound BW.

$$\mathsf{I}(\bar{v}) \wedge \mathsf{G}_i(\bar{t},\bar{v}) \vdash \mathsf{W}_i(\bar{v}) \leq \mathsf{BW} \quad \text{(event/wght/BOUND)}$$

5. **Bounded parameters.** The number of potential values for parameters in convergent events must be bounded above by a constant upper bound BP.

$$\mathsf{I}(\bar{v}) \vdash \mathsf{card}(\{\bar{t} \mid \mathsf{G}_i(\bar{t},\bar{v})\}) \leq \mathsf{BP} \quad \text{(event/param/BOUND)}$$

We now formally prove that the conditions presented above are sufficient for guaranteeing the almost-certain convergence of a given set of events in a probabilistic Event-B model.

Theorem 1. Let $M = (\bar{v}, I(\bar{v}), V(\bar{v}), \mathsf{PEvts}, \mathsf{Init})$ be a probabilistic Event-B model and $\mathsf{PEvts}_c \subseteq \mathsf{PEvts}$ a set of convergent events. If M satisfies the above POs (1-5), then the set PEvts_c almost-certainly converges.

Proof-sketch. *We consider the DTMC semantics $[\![M]\!]$ of the probabilistic Event-B model M and use the global coarseness property of infinite-state DTMC [15] to show that, from all states of $[\![M]\!]$, the probability of eventually taking a non-convergent event or reaching a deadlock is* 1. *The full proof is presented in [5].* □

7. CONCLUSION

As suggested by Morgan *et al.* in [17], the ideal probabilistic extension of Event-B should allow using probabilities as a refinement of non-deterministic choices in all places where such choices exist. In Event-B, non-determinism occurs in several places and, to the best of our knowledge, existing works on extending Event-B with probabilities have only focused on refining non-deterministic assignments into probabilistic assignments [9, 20, 22] while leaving other sources of non-determinism such as the choice between enabled events or the choice between admissible parameter values untouched.

In this paper, we have proposed a fully probabilistic extension of Event-B where probabilistic choices are introduced as replacement of *all* non-deterministic choices, be it between enabled events, parameter values or assignments as suggested by Morgan *et al.* in their seminal work. Our long term goal is to produce a probabilistic extension of Event-B where the developer can choose at his convenience where to refine non-deterministic choices with probabilities and where to keep non-deterministic choices intact. However, this long-term goal is clearly ambitious and will require several years of study to be achieved. In this paper, we have therefore focused on a more reasonable objective, restricting ourselves to purely probabilistic systems where probabilities appear in the last step of refinement. Although the long-term goal presented above is not yet achieved, this is clearly a first step in the right direction.

In particular, we have introduced new notations and semantics, along with novel and adapted POs dedicated to the *consistency* of probabilistic Event-B models. We have shown that, when these POs are satisfied, the semantics of a probabilistic Event-B model is a discrete time Markov chain. Finally, we have provided sufficient conditions, expressed in terms of POs, to show that a probabilistic Event-B model satisfies the *almost-certain convergence* of a given set of events, which is a necessary step for adressing refinement in the future.

In parallel, we have started the development of a prototype plugin for the Rodin Platform. This plugin currently allows the specification of fully probabilistic Event-B models and the semi-automatic generation of a probabilistic Event-B model from a standard Event-B model. It also supports the generation of several consistency proof obligations on probabilistic Event-B models.

Future work. As the development in Event-B is intrinsically based on a refinement process, we plan on studying the refinement of probabilistic Event-B models, including (but not restricting to) the "probabilisation" of non-deterministic models, the introduction of new probabilistic events, and, the merge and the split of probabilistic events. We also plan to study how to handle Event-B models combining non-deterministic and probabilistic events as well as the (probabilistic) refinement of such models.

Most of the properties of interest that are verified in standard Event-B are safety-related. They are most of the time expressed by means of invariants and discharged as POs. We therefore plan to consider *probabilistic invariants*, i.e. invariants related to probabilistic distributions [13]. In addition, critical systems must also satisfy some liveness properties. In this paper, we have studied the *almost-certain convergence* of a given set of events, but other probabilistic liveness properties could be considered. Indeed, the verification of other liveness properties on standard Event-B models using refinement and proof obligations have been considered in [10, 4]. We will pursue these works and extend them to the verification of probabilistic liveness properties on probabilistic Event-B models.

8. REFERENCES

[1] J.-R. Abrial. *Modeling in Event-B: system and software engineering*. Cambridge University Press, 2010.

[2] J.-R. Abrial, M. Butler, S. Hallerstede, T. S. Hoang, F. Mehta, and L. Voisin. Rodin: an open toolset for modelling and reasoning in event-b. *International journal on software tools for technology transfer*, 12(6):447–466, 2010.

[3] J.-R. Abrial, D. Cansell, and D. Méry. A mechanically proved and incremental development of ieee 1394 tree identify protocol. *Formal aspects of computing*, 14(3):215–227, 2003.

[4] J.-R. Abrial, D. Cansell, and D. Méry. Refinement and reachability in event-b. In *ZB 2005: Formal Specification and Development in Z and B*, volume 3455 of *LNCS*, pages 222–241. Springer, 2005.

[5] M. A. Aouadhi, B. Delahaye, and A. Lanoix. Moving from Event-B to Probabilistic Event-B. Research Report hal-01316610, LINA-University of Nantes, 2016.

[6] D. Bert and F. Cave. Construction of finite labelled transition systems from b abstract systems. In *Integrated Formal Methods*, volume 1945 of *LNCS*, pages 235–254. Springer, 2000.

[7] M. Butler and I. Maamria. Practical theory extension in event-b. In *Theories of Programming and Formal Methods*, volume 8051 of *LNCS*, pages 67–81. 2013.

[8] W. W. Chu and C.-M. Sit. Estimating task response time with contentions for real-time distributed systems. In *Real-Time Systems Symposium, 1988., Proceedings.*, pages 272–281. IEEE, 1988.

[9] S. Hallerstede and T. S. Hoang. Qualitative probabilistic modelling in event-b. In *Integrated Formal Methods*, pages 293–312. Springer, 2007.

[10] T. Hoang and J.-R. Abrial. Reasoning about liveness properties in event-b. In *Formal Methods and Software Engineering*, volume 6991 of *LNCS*, pages 456–471. Springer, 2011.

[11] T. S. Hoang. *The development of a probabilistic B-method and a supporting toolkit*. PhD thesis, The University of New South Wales, 2005.

[12] T. S. Hoang. Reasoning about almost-certain convergence properties using event-b. *Sci. Comput. Program.*, 81:108–121, 2014.

[13] T. S. Hoang, Z. Jin, K. Robinson, A. McIver, and C. Morgan. Probabilistic invariants for probabilistic machines. In *ZB 2003: Formal Specification and Development in Z and B*, pages 240–259. Springer, 2003.

[14] T. S. Hoang, Z. Jin, K. Robinson, A. McIver, and C. Morgan. Development via refinement in probabilistic b - foundation and case study. In *International Conference of B and Z Users*, pages 355–373. Springer, 2005.

[15] R. Mayr, N. B. Henda, and P. A. Abdulla. Decisive markov chains. *Logical Methods in Computer Science*, 3, 2007.

[16] A. McIver, C. Morgan, and T. S. Hoang. Probabilistic termination in b. In *International Conference of B and Z Users*, pages 216–239. Springer, 2003.

[17] C. Morgan, T. S. Hoang, and J.-R. Abrial. The challenge of probabilistic event b—extended abstract—. In *ZB 2005: Formal Specification and Development in Z and B*, pages 162–171. Springer, 2005.

[18] C. Morgan and A. McIver. Abstraction, refinement and proof for probabilistic systems. *Monographs in Computer Science. Springer*, 2005.

[19] R. Motwani and P. Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.

[20] A. Tarasyuk, E. Troubitsyna, and L. Laibinis. Reliability assessment in event-b development. *NODES 09*, page 11, 2009.

[21] A. Tarasyuk, E. Troubitsyna, and L. Laibinis. Towards probabilistic modelling in event-b. In *Integrated Formal Methods*, pages 275–289. Springer, 2010.

[22] A. Tarasyuk, E. Troubitsyna, and L. Laibinis. Integrating stochastic reasoning into event-b development. *Formal Aspects of Computing*, 27(1):53–77, 2015.

[23] K. S. Trivedi, S. Ramani, and R. Fricks. Recent advances in modeling response-time distributions in real-time systems. *Proceedings of the IEEE*, 91(7):1023–1037, 2003.

[24] A. Villemeur. *Reliability, Availability, Maintainability and Safety Assessment, Assessment, Hardware, Software and Human Factors*, volume 2. Wiley, 1992.

[25] E. Yilmaz. *Tool support for qualitative reasoning in Event-B*. PhD thesis, Master Thesis ETH Zürich, 2010, 2010.