Augmenting Formal Development of Control Systems with Quantitative Reliability Assessment

Anton Tarasyuk, Elena Troubitsyna, Linas Laibinis

Åbo Akademi University, Turku, Finland

April 15, 2010

Åbo Akademi

Motivation

Our goal is:

- Extend the existing Event-B framework with support for stochastic modelling
- Extend notion of Event-B refinement for quantitative (probabilistic) analysis
- Integrate the quantitative analysis of dependability into Event-B development
- Combine Event-B system development with probabilistic model checking

Åbo Akademi

Event-B

- Event-B is a notation used for developing mathematical models of discrete transition systems
- Key features of Event-B:
  - modelling notation based on set theory and predicate calculus
  - the use of refinement to represent systems at different abstraction levels
  - the use of mathematical proof to verify consistency between refinement levels
- The Rodin Platform provides automated tool support for modelling and verification in Event-B

Åbo Akademi

Model Development with Event-B

Static part of a model

- **Context** C
- **Sets** s
- **Constants** c
- **Axioms** a

$\longleftarrow$

Dynamic part of a model

- **Machine** M
- **Variables** v
- **Invariants** I
- **Events**
  - $init$
  - evt_1
  - $\ldots$
  - evt_N

$evt \mathrel{\widehat{=}}$ **when** $g(v)$ **then** $S(v)$ **end**

Language of generalised substitutions:

$$S(v) \ ::= \ x := E(v) \mid x :\in Q(v) \mid x :\mid P(v, x') \mid S_1(v) \parallel S_2(v) \mid skip$$

Parallel Composition of Events

For two events evt_1 and evt_2,

$$evt_1 \mathrel{\widehat{=}} \textbf{when } g_1 \textbf{ then } S_1 \textbf{ end}$$

$$evt_2 \mathrel{\widehat{=}} \textbf{when } g_2 \textbf{ then } S_2 \textbf{ end}$$

their parallel composition $evt_1 \parallel evt_2$ is defined as

$$evt_1 \parallel evt_2 \mathrel{\widehat{=}} \textbf{when } g_1 \wedge g_2 \textbf{ then } S_1 \parallel S_2 \textbf{ end}$$

Åbo Akademi

Event-B Refinement

- Event-B system development is based on the notion of stepwise refinement
- each development step is proved to be correct with respect to a more abstract specification
- to verify correctness the tool generates a number of Proof Obligations
- these proof obligations must be discharged
  - automatically (by the tool)
  - manually

Åbo Akademi

Incorporating Probabilities into Event-B

Event-B can be used for formal verification of various dependability attributes. However, it lacks a support for their quantitative evaluation.

Event-B can be used for formal verification of various dependability attributes. However, it lacks a support for their quantitative evaluation.

Reliability Definition

In engineering reliability is generally measured by the probability that an entity $\mathcal{E}$ can perform a required function under given conditions for the time interval $[0, t]$:

$$R(t) = P\left[\mathcal{E} \textit{ not failed over time } [0, t]\right]$$

Åbo Akademi

Incorporating Probabilities into Event-B

Event-B can be used for formal verification of various dependability attributes. However, it lacks a support for their quantitative evaluation.

Reliability Definition

In engineering reliability is generally measured by the probability that an entity $\mathcal{E}$ can perform a required function under given conditions for the time interval $[0, t]$:

$$R(t) = P\,[\mathcal{E}\ \textit{not failed over time}\ [0, t]]$$

A probabilistic extension of the Event-B framework is required

Åbo Akademi

Incorporating Probabilities into Event-B (Cont.)

S. Hallerstede and T. S. Hoang,
Qualitative probabilistic modelling in Event-B (2007)

$$S(v) \ ::= \ x := E(v) \mid x :\in Q(v) \mid x :\mid P(v, x') \mid x \oplus\mid P(v, x')$$

- qualitative probabilistic statement assigns to x a new value x' with some fixed (but unknown) probability
- it was introduced to allow the reasoning about the fairness property in Event-B
- can be placed instead of an existing nondeterministic assignment

Åbo Akademi

Incorporating Probabilities into Event-B (Cont.)

S. Hallerstede and T. S. Hoang,
Qualitative probabilistic modelling in Event-B (2007)

$$S(v) \ ::= \ x := E(v) \mid x :\in Q(v) \mid x :\mid P(v, x') \mid x \oplus\!\mid P(v, x')$$

- qualitative probabilistic statement assigns to x a new value x' with some fixed (but unknown) probability
- it was introduced to allow the reasoning about the fairness property in Event-B
- can be placed instead of an existing nondeterministic assignment
- However, this approach is not suitable for reliability or performance evaluation

Åbo Akademi

Incorporating Probabilities into Event-B (Cont.)

To reason about the system reliability we need to extend the language of generalised substitutions with a probabilistic assignment which contains a precise probabilistic information.

Incorporating Probabilities into Event-B (Cont.)

To reason about the system reliability we need to extend the language of generalised substitutions with a probabilistic assignment which contains a precise probabilistic information.

Reliability Definition

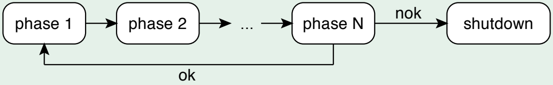
In engineering reliability is generally measured by the probability that an entity $\mathcal{E}$ can perform a required function under given conditions for the time interval $[0, t]$:

$$R(t) = P\,[\mathcal{E} \textit{ not failed over time } [0, t]]$$

Åbo Akademi

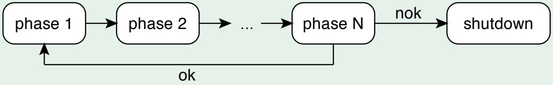
Modelling of Control Systems

Control Cycle



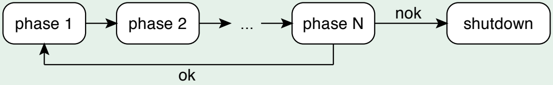
Modelling of Control Systems

Control Cycle



Assume, that the duration length of a control cycle is constant

Control Cycle



Assume, that the duration length of a control cycle is constant

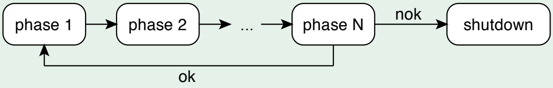
$\Downarrow$

We can evaluate the system reliability in terms of consecutive control cycles

Åbo Akademi

Modelling of Control Systems

Control Cycle



Abstract Specification of a Control System

Machine *CS*
Variables *st*
Invariants $st \in \{ok, nok\}$
Events
$init \mathrel{\widehat{=}}$ **begin** $st := ok$ **end**
$step \mathrel{\widehat{=}}$ **when** $st = ok$ **then** $st :\in \{ok, nok\}$ **end**
$shutdown \mathrel{\widehat{=}}$ **when** $st = nok$ **then** $skip$ **end**

Åbo Akademi

Modelling of Control Systems

Abstract Specification of a Control System

Machine CS
Variables st
Invariants $st \in \{ok, nok\}$
Events

$init \mathrel{\widehat{=}} \textbf{begin } st := ok \textbf{ end}$
$step \mathrel{\widehat{=}} \textbf{when } st = ok \textbf{ then } st :\in \{ok, nok\} \textbf{ end}$
$shutdown \mathrel{\widehat{=}} \textbf{when } st = nok \textbf{ then } skip \textbf{ end}$

CS traces

$$Traces(CS) = \{< step^n, shutdown > \mid \ n \in \mathbb{N}_1\}$$

Åbo Akademi

Event-B Trace Refinement

Definition

A machine M' is *trace refinement* of machine M ($M \sqsubseteq_{tr} M'$) if any trace of M' is also a trace of M, i.e., any trace that is observable for the concrete system can be also observed in the abstract system:

$$M \sqsubseteq_{tr} M' \quad \mathit{iff} \quad \mathit{Traces}(M') \subseteq \mathit{Traces}(M)$$

The proof obligations defined for standard Event-B refinement are sufficient conditions for trace refinement:

$$M \sqsubseteq M' \quad \Rightarrow \quad M \sqsubseteq_{tr} M'$$

Åbo Akademi

Probabilistic Control System

Abstract Specification of a Control System

Machine CS
Variables st
Invariants $st \in \{ok, nok\}$
Events

$init \mathrel{\widehat{=}} \textbf{begin } st := ok \textbf{ end}$
$step \mathrel{\widehat{=}} \textbf{when } st = ok \textbf{ then } st :\in \{ok, nok\} \textbf{ end}$
$shutdown \mathrel{\widehat{=}} \textbf{when } st = nok \textbf{ then } skip \textbf{ end}$

Åbo Akademi

Probabilistic Control System

Abstract Specification of a Control System

Machine CS
Variables st
Invariants $st \in \{ok, nok\}$
Events
$init \,\widehat{=}\,$ **begin** $st := ok$ **end**
$step \,\widehat{=}\,$ **when** $st = ok$ **then** $st :\in \{ok, nok\}$ **end**
$shutdown \,\widehat{=}\,$ **when** $st = nok$ **then** $skip$ **end**

Probabilistic Control System

Abstract Specification of a Control System

Machine CS
Variables st
Invariants $st \in \{ok, nok\}$
Events

$init \mathrel{\widehat{=}}$ **begin** $st := ok$ **end**
$step \mathrel{\widehat{=}}$ **when** $st = ok$ **then** $st := ok\,{}_p\!\oplus\; st := nok$ **end**
$shutdown \mathrel{\widehat{=}}$ **when** $st = nok$ **then** $skip$ **end**

Åbo Akademi

Probabilistic Control System

Abstract Specification of a Control System

Machine CS
Variables st
Invariants $st \in \{ok, nok\}$
Events
$init \mathrel{\widehat{=}}$ **begin** $st := ok$ **end**
$step \mathrel{\widehat{=}}$ **when** $st = ok$ **then** $st := ok\ {}_p\!\oplus\ st := nok$ **end**
$shutdown \mathrel{\widehat{=}}$ **when** $st = nok$ **then** $skip$ **end**

Probabilistic CS traces

$$tr \ = < step.p, step.p, \ldots, step.\bar{p}, shutdown.1 >$$

$$PTraces(CS) = \{< (step.p)^{n-1}, step.\bar{p}, shutdown.1 > \mid \ n \in \mathbb{N}_1\}$$

Probabilistic Traces

Probabilistic traces

$$tr \ = < step.p_1^{i_1}, step.p_2^{i_2}, \ldots, step.p_{n-1}^{i_{n-1}}, step_n.\bar{p}_n, shutdown.1 >$$

$$\forall j \in \mathbb{N}_1, \quad \sum_{i_j} p_j^{i_j} = 1 - \bar{p}_j$$

$$PTraces(M) = \{< step.p_1^{i_1}, \ldots, step_k.\bar{p}_n, shutdown.1 > \mid \ n \in \mathbb{N}_1\}$$

Probability of a trace

The overall probability of a probabilistic trace tr is defined as

$pr(tr) = \prod_{j=1}^{|tr|} p_j$ where $|tr|$ is the length of tr and

$$\sum_{tr \in PTraces(M)} pr(tr) = 1$$

Probabilistic Trace Refinement

Definition 1

For two Event-B models M and M', we say that M' is a probabilistic trace refinement of M, denoted $M \sqsubseteq_{ptr} M'$, iff

1. M' is a trace refinement of M' ($M \sqsubseteq_{tr} M'$)
2. for any $t \in \mathbb{N}$,

$$\sum_{\substack{tr \in PTraces(M') \\ |tr| \leq t}} pr(tr) \;\leq\; \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr).$$

Partial Probabilistic Trace Refinement

Definition 2

For two Event-B models M and M', we say that M' is a partial probabilistic trace refinement of M for $t \in [\,0, \tau\,]$ iff

1. M' is a trace refinement of M' ($M \sqsubseteq_{tr} M'$)
2. for any $t \in \mathbb{N}$ such that $t \leq \tau$,

$$\sum_{\substack{tr \in PTraces(M') \\ |tr| \leq t}} pr(tr) \;\leq\; \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr).$$

Åbo Akademi

Reliability Assessment

- $$\sum_{\substack{tr \in PTraces(M') \\ |tr| \leq t}} pr(tr) \;\leq \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr)$$

Åbo Akademi

Reliability Assessment

- $$\sum_{\substack{tr \in PTraces(M') \\ |tr| \leq t}} pr(tr) \;\leq \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr)$$
- $\mathcal{T}(M)$ is a random variable measuring the number of iterations of a control system M before the system shutdown
- $F_M(t)$ is a cumulative distribution function of $\mathcal{T}(M)$

Åbo Akademi

- $$\sum_{\substack{tr \in PTraces(M') \\ |tr| \leq t}} pr(tr) \;\leq \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr)$$
- $\mathcal{T}(M)$ is a random variable measuring the number of iterations of a control system M before the system shutdown
- $F_M(t)$ is a cumulative distribution function of $\mathcal{T}(M)$
- $$F_M(t) = \mathbf{P}[\mathcal{T}(M) \leq t] = \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr)$$
- $$F_{M'}(t) \leq F_M(t)$$

Reliability Assessment

- $$\sum_{\substack{tr \in PTraces(M') \\ |tr| \leq t}} pr(tr) \leq \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr)$$
- $\mathcal{T}(M)$ is a random variable measuring the number of iterations of a control system M before the system shutdown
- $F_M(t)$ is a cumulative distribution function of $\mathcal{T}(M)$
- $F_M(t) = \mathbf{P}[\mathcal{T}(M) \leq t] = \sum_{\substack{tr \in PTraces(M) \\ |tr| \leq t}} pr(tr)$
- $F_{M'}(t) \leq F_M(t)$
- $R_M(t)$ is a reliability function of a control system M
- $R_M(t) = \mathbf{P}[\mathcal{T}(M) > t] = 1 - \mathbf{P}[\mathcal{T}(M) \leq t] = 1 - F_M(t)$
- $R_M(t) \leq R_{M'}(t)$

The PRISM Tool

- PRISM is a probabilistic symbolic model checker
- Modelling of:
  - Discrete-Time Markov Chains (DTMCs)
  - Markov Decision Processes (MDPs)
  - Continuous-Time Markov Chains (CTMCs)
- Verification of:
  - Probabilistic Computation Tree Logic (PCTL)
  - Continuous Stochastic Logic (CSL)
- http://www.prismmodelchecker.org/

Reliability Assessment in PRISM

- PCTL logic for DTMCs and MDPs models in PRISM
- for specifying properties PRISM supports two principal operators **P** and **S**
- path properties **F**, **G**, **X**, **U**

Reliability Assessment in PRISM

- PCTL logic for DTMCs and MDPs models in PRISM
- for specifying properties PRISM supports two principal operators **P** and **S**
- path properties **F**, **G**, **X**, **U**
- $\mathbf{P}_{=?}[\mathbf{G} \leq t\ prop]$ returns the probability that the predicate *prop* remains *TRUE* in all states within the period of time t.
- $\mathcal{OP}$ is a predicate defining the set of system operating states
- $\mathbf{P}_{=?}[\mathbf{G} \leq t\ \mathcal{OP}]$ gives us the probability that the system will stay operational during the first t iterations

Åbo Akademi

Example

Abstract Specification of a Control System

Machine CS
Variables st
Invariants $st \in \{ok, nok\}$
Events

$init \,\widehat{=}\,$ **begin** $st := ok$ **end**
$step \,\widehat{=}\,$ **when** $st = ok$ **then** $st := ok\ {}_p\!\oplus\ st := nok$ **end**
$shutdown \,\widehat{=}\,$ **when** $st = nok$ **then** $skip$ **end**

Åbo Akademi

Refinement of CS

$phase \in \{env, read, det, cont\} \quad s \in \{0,1\} \quad heat \in \{on, off\}$
$tmp, tmp_{est}, cnt, N \in \mathbb{N} \quad phase = cont \Rightarrow tmp_{min} \le tmp \le tmp_{max}$

phase = env

$heat = on \rightarrow tmp := tmp + 1$

$heat = off \rightarrow tmp := tmp - 1$

$\rightarrow$

phase = read

$s = 1 \rightarrow s := 0 \;{}_f\!\oplus\; s := 1$

$s = 0 \rightarrow s := 1 \;{}_r\!\oplus\; s := 0$

$\uparrow$ $\downarrow$

phase = cont

$cnt < N$

$tmp_{est} \le tmp_{min} \rightarrow heat := on$

$tmp_{max} \le tmp_{est} \rightarrow heat := off$

$tmp_{min} \le tmp \le tmp_{max} \rightarrow skip$

$cnt \ge N \rightarrow shutdown$

$\leftarrow$

phase = det

$s = 1 \rightarrow tmp_{est} := tmp \parallel cnt := 0$

$s = 0 \rightarrow cnt := cnt + 1 \parallel$

$N := \mathbf{min}(tmp_{max} - tmp_{est},$
$tmp_{est} - tmp_{min})$

Åbo Akademi

PRSIM Counterpart

phase = read (Event-B)

$sensor_{ok} \widehat{=}$
when
$phase := read$
$s = 1$
then
$s := 0 \;{}_{f}\!\oplus\; s := 1$
$phase := det$
end

$sensor_{nok} \widehat{=}$
when
$phase := read$
$s = 0$
then
$s := 1 \;{}_{r}\!\oplus\; s := 0$
$phase := det$
end

phase = read (PRISM)

module *sensor*

$s : [0..1]$ **init** $1;$

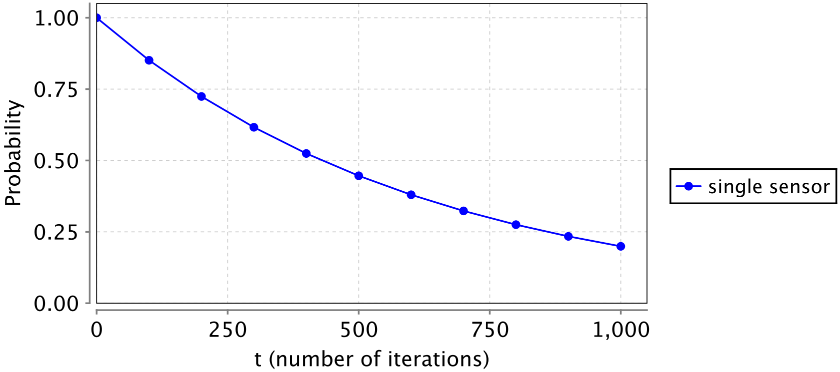
$[]\ (phase = 1)\&(s = 1) \rightarrow$
$f : (s' = 0)\&(phase' = 2)$
$+ (1 - f) : (phase' = 2);$

$[]\ (phase = 1)\&(s = 0) \rightarrow$
$r : (s' = 1)\&(phase' = 2)$
$+ (1 - r) : (phase' = 2);$

endmodule

Åbo Akademi

PRISM Verification Results



$f = 0.01, r = 0.99, tmp_{max} = 20$

The Second Refinement

$s_1 \in \{0,1\} \quad s_2 \in \{0,1\} \quad s_1 + s_2 > 0 \Leftrightarrow s = 1$

phase = read (R1)

$s = 1 \rightarrow s := 0 \;{}_f\!\oplus\; s := 1$

$s = 0 \rightarrow s := 1 \;{}_r\!\oplus\; s := 0$

$\sqsubseteq$

phase = read (R2)

$s_1 = 1 \rightarrow s_1 := 0 \;{}_f\!\oplus\; s_1 := 1$

$s_2 = 1 \rightarrow s_2 := 0 \;{}_f\!\oplus\; s_2 := 1$

$s_1 = 0 \rightarrow s_1 := 1 \;{}_r\!\oplus\; s_1 := 0$

$s_2 = 0 \rightarrow s_2 := 1 \;{}_r\!\oplus\; s_2 := 0$

phase = det (R1)

$s = 1 \rightarrow tmp_{est} := tmp \parallel cnt := 0$

$s = 0 \rightarrow cnt := cnt + 1 \parallel$

$N := \mathbf{min}(\ldots)$

$\sqsubseteq$

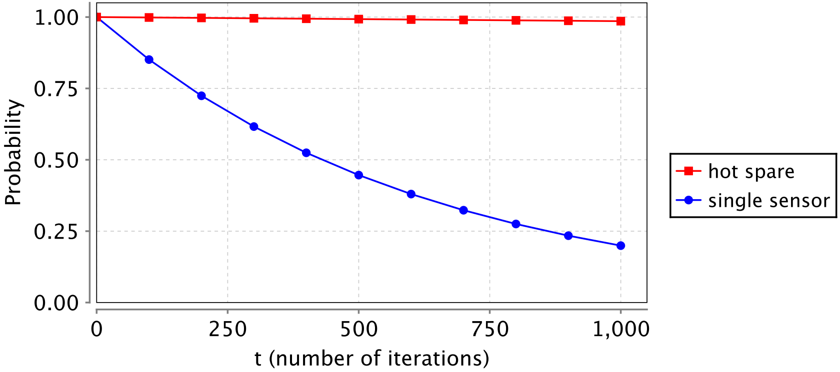
phase = det (R2)

$s_1 + s_2 > 0 \rightarrow tmp_{est} := tmp \parallel cnt := 0$

$s1_1 + s_2 = 0 \rightarrow cnt := cnt + 1 \parallel$

$N := \mathbf{min}(\ldots)$

PRISM Verification Results



$f = 0.01, r = 0.99, tmp_{max} = 20$

Summary

Conclusion

We have

- (informally) extended the Event-B generalised substitutions language with probabilistic assignment
- formulated the notion of Event-B probabilistic trace refinement
- demonstrated the benefit of combining Event-B refinement with probabilistic model checking for development of control systems

Future work

- Formally define the Event-B semantics of probabilistic assignment
- Develop a tool (plugin) for automatic translation of Event-B specifications to PRISM

Åbo Akademi