Stochastic Analogues of Invariants: Martingales in Stochastic Event-B

Richard Banach[1]
[1]*School of Computer Science, University of Manchester, Manchester, U.K.*
banach@cs.man.ac.uk

Keywords: Model Based Frameworks, Event-B, Stochastic Extensions, Invariants, Martingales.

Abstract: In conventional formal model based development frameworks, invariants play a key role in controlling the behaviour of the model (when they contribute to the definition of the model) or in verifying the model's properties (when the model, independently defined, is required to preserve the invariants). However, when variables take values distributed according to some probability distribution, the possibility of verifying that system behaviour is, in the long term, confined to some acceptable set of states can be severely diminished because the system might, in fact, with low probability fail to be thus confined. This short paper proposes martingales as suitable analogues of invariants for capturing suitable properties of non-terminating systems whose behaviour is with high probability good, yet where a small chance of poor behaviour remains. The idea is explored in the context of the well-known Event-B framework.

1 INTRODUCTION

In conventional formal model based development frameworks invariants play a key role in controlling the behaviour of the model. In frameworks in which they are part of the definition of the model (such as Z (Spivey, 1992; ISO-Z, 2002)) they restrict the behaviour of the rest of the model. In most frameworks though, and typically in the various dialects of the B-Method (Abrial, 1989; Abrial, 1996; Abrial, 2010), invariants are proposed independently of the system model and then the behaviour of the system as defined must be proved to conform to the invariants — i.e. the reachable set, as defined by the system model, must be a subset of any stated invariant set of states. This is normally done by an inductive technique which proves that every state change allowed by the system model results in an after-state that is within the invariant set, provided the before-state was also in the invariant set.

All of this works fine when variable values are assigned either deterministically, or in more abstract settings, nondeterministically. Considering pure nondeterminism, no restriction is placed on the occurrence of any allowed value, so any system invariant must include all the possibilities permitted by all the nondeterminism in the model in order that the model preserves the invariant. However, in practice, systems often have smaller families of states that are visited frequently, with other permitted states visited much less frequently. A simple invariant cannot capture the distinction between the smaller 'frequently visited' subset and the infrequently visited 'outlying states'.

The distinction between frequently visited and infrequently visited is quantified in probabilistic terms. When the values taken by variables are labelled with probabilities (implicitly also labelling the transitions that cause those values to be adopted), then in principle, we can make precise statements regarding the likelihood that the system trajectory remains in a certain region of the state space. Useful byproducts of this are statements that the long term behaviour of the system is predominantly confined to a relatively small portion of the state space, despite the probability of its visiting other states being non-zero (though small).

This short paper proposes the *martingale* concept from stochastic calculus (see e.g. (Resnick, 1992; Grimmett and Stirzaker, 2001)) as a suitable analogue for an invariant in comparable situations. For example, it can be used to capture such properties of non-terminating systems as when their behaviour is with high probability 'good', yet where a limited probability of 'poor' behaviour remains. The idea is explored in the context of the well-known Event-B framework, using a small case study drawn from the Mondex Electronic Purse problem (Stepney et al., 2000) — this was the first problem in the Verification Grand Challenge, active for a number of years now (Jones et al., 2006; Woodcock, 2006; Woodcock and Banach, 2007; Jones and Woodcock (eds.), 2008).

In the rest of this paper we do the following. In Section 2 we give a brief description of Event-B, for purposes of orientation. In Section 3 we describe a fragment of the Mondex Purse problem, originally developed in Z, using Event-B. In Section 4 we give a short introduction to martingales, in the form we will need them later. In Section 5 we focus our martingale discussion on the Mondex problem introduced earlier. Section 6 concludes, and looks forward to the further development of the ideas presented here, especially in their application to hybrid and cyber-physical systems.

2 EVENT-B

In this section, we briefly review Event-B machines and their ingredients. Explicit examples of these things will appear below.

In a nutshell, an Event-B MACHINE has a *name*, it SEES one or more *static contexts*, and it owns some VARIABLES. The latter are allowed to be updated via EVENTS, but are required to always satisfy the INVARIANTS (this requirement generating verifications conditions a.k.a. proof obligations (POs)). The events can declare their own *parameters* (which are bound variables which can be used to reduce internal nondeterminism, or to carry input values or output values). Furthermore, each event has one or more *guards* which define when the event is enabled, and one or more *actions* which defines the state change to be implemented by the event, specified via *before-after predicates* (or notations such as assignment for simpler cases). Among the events there is an *INITIALISATION*, whose guard must be true.

The semantics of Event-B machines, is expressed via a number of POs. These must be provable in order for the machine in question to be well defined. We quote the main ones of interest to us, mentioning the others more briefly. See (Abrial, 2010; Abrial et al., 2010) for full details.

For a machine A to be well defined the *initialisation, feasibility* and *correctness* POs must hold:

$$Init_A(u') \Rightarrow I(u') \tag{1}$$

$$I(u) \land G_{Ev_A}(u) \Rightarrow (\exists u' \bullet Ev_A(u,u')) \tag{2}$$

$$I(u) \land G_{Ev_A}(u) \land Ev_A(u,u') \Rightarrow I(u') \tag{3}$$

In (1), $Init_A$ is the initialisation event and (1) says that the value u' of A's state variable u established by $Init_A$ satisfies A's invariant $I(u')$. Then (2) says that event Ev_A is *feasible*; i.e. when it is enabled in an invariant state $(I(u) \land G_{Ev_A}(u))$ then an after-state u' exists so that $Ev_A(u,u')$ can complete successfully. Likewise, (3) says that for Ev_A, if A's invariant $I(u)$, and Ev_A's guard $G_{Ev_A}(u)$, both hold in the before-state u of the event, and Ev_A's before-after relation $Ev_A(u,u')$ also holds (for an after-state u' whose existence is established by (2)), then the after-state will satisfy the invariant I once more (i.e. we have *invariant preservation*). In (1), (2) and (3) we have suppressed mention of the details of the static contexts seen by machine A, and have suppressed mention of any local bound variables or of inputs or outputs. These would be easy to include in the before-after relation if desired. Aside from (1), (2) and (3), the initialisation must also be feasible, and for non-terminating systems we must also have *deadlock freedom*:

$$I(u) \Rightarrow \bigvee_k G_{Ev_{A,k}}(u') \tag{4}$$

This says that if the invariant $I(u)$ holds in a state u, then some event $Ev_{A,k}$ is enabled. See (Abrial, 2010; Abrial et al., 2010) for more details. Non-terminating systems are of particular interest in this paper.

3 THE MONDEX PURSE

The Mondex Electronic Purse (Stepney et al., 2000), produced by the NatWest Development Team, is a system of Smartcard-based electronic purses carrying currency for electronic commerce applications. Clearly, this is a security-critical application. For this reason, the developers of Mondex (formerly a part of NatWest Bank U.K.), employed state of the art methods to ensure the implementation was as robust as possible. At the time of its creation (in the late 1990s), the Mondex Purse achieved an ITSEC (D.T.I., 1991) rating of E6. This requires a formal abstract model, a formal concrete model, and a proof of correspondence between them. (In the case of Mondex, the correspondence proof was a proof of refinement.) This was the the highest possible dependability standard achievable at the time, and the development was a trailblazer for showing that fully formal techniques could be applied within realistic time and cost limitations to industrial scale applications.

The abstract model of the Mondex Purse system describes a world of purses which exchange value through atomic transactions, and specifies the security properties: purse authentication, preservation of overall system value, and correct processing of both transferred and lost value. The concrete model describes a distributed system of purses, transferring value via an insecure and lossy medium using an n-step protocol. Security features are implemented locally on each purse. In the field each purse is self-sufficient, logging any lost value from failed transactions locally, for later central archiving and reconciliation.

We have neither the space nor the need to describe all of the above fully. A good account, from the vantage point of the work done on Mondex in the Verification Grand Challenge, can be gained from (Jones and Woodcock (eds.), 2008) in particular.

For us, it will be sufficient to focus on one detail of Mondex's operation, the event —to use Event-B terminology— that updates the sequence number that each purse maintains to prevent replay attacks. This event is introduced at the intermediate level of modelling in (Stepney et al., 2000) and is in fact an abstraction of a number of internal operations of a real purse that need the *no-replay* property. (Evidently an event that literally does nothing other than increase a sequence number is rather pointless in a real application.) In Event-B notation it can be captured in the following small machine:

```
MACHINE Mondex0          Increase
VARIABLES                  ANY inc
  seqno                    WHERE $inc \in \mathbb{N} \wedge$
INVARIANTS                   $0 < inc$
  $seqno \in \mathbb{N}$   THEN
EVENTS                       seqno :=
  INITIALISATION               $seqno + inc$
    BEGIN                  END
      seqno := 0         END
    END
```

In *Mondex*0, having been initialised to 0, the variable *seqno* increases by an arbitrary positive integer at each *Increase* event (the arbitrariness being intended to prevent attacks based on predicting the next sequence number). More realistically, the increment will not really be arbitrary, but will be drawn from a finite range, say $1 \leq inc \leq 10$, yielding:

```
MACHINE Mondex1            THEN
...   ...   ...   ...        seqno :=
  Increase                     $seqno + inc$
    ANY inc                END
    WHERE $inc \in \mathbb{N} \wedge$   END
      $1 \leq inc \leq 10$
```

In such a case all that we could say with certainty would be that *seqno* was not greater than 10 times the number of *Increase* calls, nor smaller than one times the number of *Increase* calls, a fact that we could express via an invariant if we introduced a variable to count the number of calls to *Increase*.

However, it is reasonable to presume that individual occurrences of *inc* in a run would be uniformly distributed over 1..10 for example. In that case, it is equally reasonable to deduce that, on average, *seqno* would increase by $5\frac{1}{2}$ on each *Increase* call. In fact, a closer probabilistic analysis reveals that the more calls to *Increase* are made, the closer, on average, the behaviour of *seqno* adheres to that behaviour. But it is impossible to express such a fact via a straightforward state invariant. This observation brings us to the topic of martingales.

4 MARTINGALES

In this section we give a brief introduction to martingales. We start with the idea of a stochastic process. A stochastic process is a family of random variables $\mathcal{X} = X_0, X_1, X_2, \ldots$ indexed by 'some analogue of time'. If the analogue of time is a conventional discrete set, such as the natural numbers, then the family can be written in a form similar to $\mathcal{X}$ and we speak of a discrete stochastic process. Alternatively, if the analogue of time is, say, a segment of the reals, then we need a continuous index, and we speak of a continuous stochastic process. (We will not need the continuous case in this paper.)

With the family $\mathcal{X}$ we associate another family of random variables $\mathcal{S} = S_0, S_1, S_2, \ldots$ indexed in the same way. The family $\mathcal{S}$ is a *martingale* with respect to the family $\mathcal{X}$ if two conditions hold. Firstly, the expectation of the absolute value of each random variable S_n is finite: $\mathbb{E}(\mid S_n \mid) < \infty$. Secondly, the expectation of each 'next' S_{n+1}, conditional on the values of all the X_js up to n, is equal to the value of S_n: $\mathbb{E}(S_{n+1} \mid X_0 \ldots X_n) = S_n$. (Of course it is possible for a family $\mathcal{X}$ to be a martingale with respect to itself, but in most cases of interest this does not quite work, and we typically need to apply some function to the members of $\mathcal{X}$ before a martingale is derived: $S_n = \phi_n(X_0 \ldots X_n)$.)

The intuition behind the martingale concept may be briefly described as follows. The outcomes of the random variables $S_0 \ldots S_n$, being random, are unpredictable. But having arrived at these outcomes, there are no grounds for expecting the outcome of the next one in the sequence S_{n+1}, to be different, on average, to the value that has been arrived at thus far. This comment also helps to explain why we usually need two sequences, $\mathcal{X}$ and $\mathcal{S}$, to define a martingale. The sequence most conveniently to hand in an application, $\mathcal{X}$ say, will typically display a 'drift' of some kind that prevents it from being a martingale in its own right; and so, to obtain a martingale, we modify it in an appropriate way (using functions ϕ_n say) to get a martingale $\mathcal{S}$.

Why are martingales (which are intensely studied in probability theory) of interest? The main reason is that under rather mild restrictions, martingales enjoy significant convergence properties. (N. B. The precise details of these properties and of the needed restrictions vary according to the specific convergence theorem being discussed — we do not delve into the technical details in this paper.) Although the details vary from result to result, in general, the restrictions involve uniform bounds on a low order moment of the martingale, and promise convergence (in a suitable sense) of the martingale sequence to a limiting random variable S_∞, i.e. $\mathcal{S} \to S_\infty$ in a suitable manner.

The convergence to a limiting behaviour given by S_∞ is the feature of a martingale $\mathcal{S}$ that we identify as being analogous to an invariant property. It allows us to neglect transient details in favour of concentrating on the long term properties of the behaviour that are often of the greatest interest for practical applications. Of course, whether it is permissible to neglect such transient details will depend critically on the application in question, and refers very much to the requirements pertaining to the transients and to their significance in the application as a whole. When it is appropriate to do so, the martingale approach can be expected to give a much tighter quantification of system behaviour than an invariant based approach in which the invariants must accommodate every possible outlier, no matter how extreme it is or how rarely it might occur.

5 THE MONDEX SEQUENCE NUMBER AS MARTINGALE

With the preceding background, we reexamine our Mondex sequence number case study. Bearing in mind the remarks just above concerning requirements, we first accept that being interested in the sequence number in terms of its average behaviour is reasonable. We know from the *Mondex*0 and *Mondex*1 models, that the sequence number is guaranteed to increase at each call of *Increase*, which would (in an application of more realistic size) help to ensure the key *no-replay* safety property.

With this aspect confirmed, we can look at quantifying the average behaviour via a martingale. We know from our previous discussion that we expect that the *seqno* variable will increase by about $5\frac{1}{2}$ on each call of *Increase*, so we want to turn this behaviour into a martingale. Obviously the variable *seqno* by itself will not do, since it evidently does not approach any limit. This illustrates rather well the remark made in Section 4, that often, the most selfevident variables do not provide a martingale directly, but that some adjustment is needed before the requisite convergence properties emerge.

In our case, we can take the X_j random variables as the successive values of *seqno*, and one way to obtain convergence is to introduce a suitable analogue of time. For us, a 'tick' variable *tk* that is incremented by 1 on each *Increase* call will do the job. Now we can divide the successive values of *seqno* by the corresponding value of *tk*, and the resulting sequence of values will converge to the mean of a single trial, namely $5\frac{1}{2}$, as required. These successive values can be taken as the S_j values corresponding to our X_j values. We are not quite done though. Martingale convergence requires that low order moments of S_j are uniformly bounded.

Looking at the successive values of $seqno/tk$, and noting that they are successive sums of independent identically distributed random variables, each drawn from a finite distribution, we can use the Central Limit Theorem to deduce that although the mean of $seqno - 5\frac{1}{2}tk$ tends to 0, the standard deviation will grow as the square root of the number of trails, i.e. as the value of $\sqrt{tk}$. So $seqno/tk - 5\frac{1}{2}$ will tend to 0, and its standard deviation will tend to 0. In other words the limiting distribution for $seqno/tk$ will converge (in probability) to a point mass centred on $5\frac{1}{2}$. This gives us a martingale derived from the *seqno* variable's behaviour.

However, the preceding is not the only way of dealing with this example. The Central Limit Theorem states that, in distribution, the successive values of *seqno* satisfy, in the limit, $(seqno - n\mu)/\sqrt{n\sigma^2} \to N(0,1)$, where n is the number of increments, σ^2 is the variance of the distribution of a single trial, and $N(0,1)$ is the standard normal distribution. Applying this to our situation, we create a martingale as follows.

The X_j are as before, but this time the S_j are defined as $seqno/\sqrt{tk} - 5\frac{1}{2}\sqrt{tk}$. By what we said above, this will converge to $N(0,1)/SD$, where SD is the standard deviation of the single trial distribution. This gives us a second martingale based on the same case study. The structure of the second martingale displays a feature that occurs commonly, namely a 'drift' term, this being $5\frac{1}{2}\sqrt{tk}$, the term that has to be subtracted from the 'main' term $seqno/\sqrt{tk}$ in order to gain stability in the convergence process.[1]

It remains to package these insights into a form compatible with the structure of Event-B models. We introduce a fresh syntactic component into an Event-B machine, the MARTINGALES clause (allowing for more than one, by analogy with the INVARIANTS clause). In its most primitive form a syntactic martingale of this kind will consist of four expressions: the first being the 'raw' stochastic expression of interest, and the second being the drift term that has to be subtracted from it. Next, we would have the mean of the limiting distribution, followed by an additional term

[1] In much of the martingale literature it would be referred to as the *compensator*.

that quantifies the standard deviation of the limiting distribution.

In this way the first martingale we derived would be expressed thus:

```
MARTINGALES
  [ $seqno/tk$ ; 0 ; $5\frac{1}{2}$ ; 0 ]
```

and the second thus:

```
MARTINGALES
  [ $seqno/\sqrt{tk}$ ; $5\frac{1}{2}\sqrt{tk}$ ; 0 ; $1/SD$ ]
```

The model *Mondex*2 below shows how these details have been incorporated into the earlier model. The *tk* variable has been included in the model,[2] and the details of the two martingales explored above have been packaged into the MARTINGALES clause, each martingale's components being enclosed in square brackets, and each listed, separated by semicolons, in the order: *raw* ; *drift* ; *mean* ; *standard deviation*.

```
MACHINE Mondex2
VARIABLES
  seqno
  tk
INVARIANTS
  $seqno \in \mathbb{N}$
  $tk \in \mathbb{N}$
MARTINGALES
  [ $seqno/tk$ ;
    0 ; $5\frac{1}{2}$ ; 0 ]
  [ $seqno/\sqrt{tk}$ ;
    $5\frac{1}{2}\sqrt{tk}$ ; 0 ; $1/SD$ ]
EVENTS
  INITIALISATION
    BEGIN
      $seqno := 0$
      $tk := 0$
    END
  Increase
    ANY inc
    WHERE $inc \in \mathbb{N} \wedge$
      $inc \in \mathcal{U}[1..10]$
    THEN
      $seqno :=$
        $seqno + inc$
      $tk := tk + 1$
    END
END
```

The account just developed may be compared with the earlier treatment of the sequence number issue that appeared in (Banach et al., 2005). In (Banach et al., 2005), the authors considered the Mondex sequence number issue from a different perspective, though based on very similar technical details.

In that work, we were interested in stepping from a relatively ideal formal description of some of the requirements (including deidealising an ideal model of the sequence number) to a more realistic one.[3] So, there, the idea was to quantify the difference between the idealised and more realistic treatments of the sequence number, in order to produce evidence that the parameters chosen for the real implementation were appropriate. The simplest way of doing that involved a calculation very similar in character to the one in this paper. However, there, the whole process necessarily resided outside of the formal elements of the system model. In this paper, the more formal treatment via martingales opens the door to a more generic and systematic treatment of similar problems.

[2]An alternative approach would be to incorporate such an 'analogue of time' variable as part of the framework itself.

[3]In fact the cited paper was just the first element in a wider study of the imperfect treatment of some Mondex requirements in the formal models of (Stepney et al., 2000), see (Banach et al., 2005; Banach et al., 2006a; Banach et al., 2006b; Banach et al., 2007).

6 CONCLUSIONS

The familiar invariant based way of specifying and reasoning about programs has been used for a long time to bring greater dependability to the systems constructed using those techniques. However, many properties of such systems are stochastic in nature rather than absolute. For those aspects, the invariant based approach tends to give extremely conservative results, too conservative to be useful in practice. (This is particularly so in the case of variables with distributions which are not bounded; for example if the increments of *seqno* were drawn from an exponential distribution. In such a case the only invariant one could write regarding such a variable would be true.) What is desirable for these situations is a more stochastic analogue of the concept of program invariant. In this paper we have advanced the idea that the martingale concept from the theory of stochastic processes is a plausible such analogue.

In the preceding sections, using a simple example, we have proposed that martingales can give us an analogue of invariants in situations where the long term behaviour of a digital system tends to stability, even though the short term behaviour may admit fluctuations of much greater amplitude.

Of course, probabilistic behaviour in transition systems is not new and has been studied intensively. From an extensive literature we cite only (Heerink and Tretmans, 1996; McIver and Morgan, 2005; van Breugel and Worrell, 2001). In general, the stochastic approach does not sit very comfortably with the familiar nondeterminism and invariant based one, which uses nondeterminism to circumscribe unknown aspects of system behaviour (rather than probability).

In the majority of the existing probabilistic work (that focuses on the probabilistic quantification of program behaviour), the focus is on detailed correctness notions, in other words, despite permitting probabilistic behaviour, the concern is with the properties of individual steps. This contrasts with our focus, which permits transient behaviour to be disregarded, in favour of long term averaged properties.

Of course, the discussion in this paper is but a first step in the incorporation of such ideas into a generic formal context. A particularly fertile possible application area for such ideas is in the formal description of cyberphysical systems, with their unavoidable involvement of continuous physical processes (Sztipanovits, 2011; Willems, 2007; Summit Report, 2008; National Science and Technology Council, 2011). There, the use of martingales for the description of long term noisy physical components is even more compelling than for the purely discrete case, due to the fact that the relevant stochastic processes typically enjoy an 'independent increments on disjoint intervals of time' property, something well captured via the theory of ideal continuous stochastic processes. This makes the martingale behaviour of stochastic physical variables in stable dynamical situations into a convincing metaphor for observed phenomena. However, a proper treatment of these will need an excursion into the more challenging continuous version of martingale theory. This remains as future work. Also, with the experience of a more fully worked out proposal, there will be more clarity regarding what are the most useful verification conditions that should be generated to support martingale use in formal model based development.

REFERENCES

Abrial, J.-R. (1989). A Formal Approach to Large Software Construction. In van de Snepscheut, editor, *Mathematics of Program Construction*, volume 375 of *LNCS*, pages 1–20. Springer.

Abrial, J.-R. (1996). *The B-Book: Assigning Programs to Meanings*. Cambridge University Press.

Abrial, J.-R. (2010). *Modeling in Event-B: System and Software Engineering*. Cambridge University Press.

Abrial, J.-R., Butler, M., Hallerstede, S., Hoang, T., Mehta, F., and Voisin, L. (2010). Rodin: an open toolset for modelling and reasoning in Event-B. *Int. J. Software Tools for Technology Transfer*, 12(6):447–466.

Banach, R., Jeske, C., Poppleton, M., and Stepney, S. (2006a). Retrenching the Purse: Finite Exception Logs, and Validating the Small. In Hinchey, editor, *Proc. NASA/IEEE SEW-30*, pages 234–245, Layola College Graduate Center, Columbia, MD.

Banach, R., Jeske, C., Poppleton, M., and Stepney, S. (2006b). Retrenching the Purse: Hashing Injective CLEAR Codes, and Security Properties. In Steffen, Margaria, and Philippou, editors, *ISOLA-06*, Paphos, Cyprus. IEEE.

Banach, R., Jeske, C., Poppleton, M., and Stepney, S. (2007). Retrenching the Purse: The Balance Enquiry Quandary, and Generalised and (1,1) Forward Refinements. *Fundamenta Informaticae*, 77:29–69.

Banach, R., Poppleton, M., Jeske, C., and Stepney, S. (2005). Retrenching the Purse: Finite Sequence Numbers and the Tower Pattern. In *Formal Methods 2005*, pages 382–398.

D.T.I. (1991). Information Technology Security Evaluation Criteria. http://www.cesg.gov.uk/site/iacs/itsec/media/formal-docs/Itsec.pdf.

Grimmett, G. and Stirzaker, D. (2001). *Probability and Random Processes*. O.U.P., 3rd edition.

Heerink, L. and Tretmans, J. (1996). Formal Methods in Conformance Testing: A Probabilistic Refinement. In *Testing of Communicating Systems*, pages 261–276. Springer.

ISO-Z (2002). *Information Technology – Z Formal Specification Notation – Syntax, Type System and Semantics: International Standard*. ISO/IEC 13568. http://www.iso.org/iso/en/ittf/PubliclyAvailableStandards/c021573_ISO_IEC_13568_2002(E).zip.

Jones, C., O'Hearne, P., and Woodcock, J. (2006). Verified Software: A Grand Challenge. *IEEE Computer*, 39(4):93–95.

Jones, C. and Woodcock (eds.), J. (2008). FAC Special Issue on the Mondex Verification. *Formal Aspects of Computing*, 20(1):1–139.

McIver, A. and Morgan, C. (2005). *Abstraction, Refinement and Proof for Probabilistic Systems*. Springer.

National Science and Technology Council (2011). Trustworthy Cyberspace: Strategic plan for the Federal Cybersecurity Research and Development Program. http://www.whitehouse.gov/sites/default/files/microsites/ostp/fed_cybersecurity_rd_strategic_plan_2011.pdf.

Resnick, S. (1992). *Adventures in Stochastic Processes*. Birkhauser.

Spivey, J. (1992). *The Z Notation: A Reference Manual*. Prentice-Hall, second edition.

Stepney, S., Cooper, D., and Woodcock, J. (2000). An Electronic Purse: Specification, Refinement and Proof. Technical Report PRG-126, Oxford University Computing Laboratory.

Summit Report (2008). Summit Report: Cyber-Physical Systems. http://iccps2012.cse.wustl.edu/_doc/CPS_Summit _Report.pdf.

Sztipanovits, J. (2011). Model Integration and Cyber Physical Systems: A Semantics Perspective. In Butler and Schulte, editors, *Proc. FM-11*. Springer, LNCS 6664, p.1, http://sites.lero.ie/download.aspx?f =Sztipanovits-Keynote.pdf. Invited talk, FM 2011, Limerick, Ireland.

van Breugel, F. and Worrell, J. (2001). Towards Quantitative Verification of Probabilistic Transition Systems. In *Proc. ALP-01*, pages 421–432. Springer LNCS.

Willems, J. (2007). Open Dynamical Systems: Their Aims and their Origins. Ruberti Lecture, Rome. http://homes.esat.kuleuven.be/~jwillems/Lectures/2007/Rubertilecture.pdf.

Woodcock, J. (2006). First Steps in the The Verified Software Grand Challenge. *IEEE Computer*, 39(10):57–64.

Woodcock, J. and Banach, R. (2007). The Verification Grand Challenge. *JUCS*, 13(5):661–668.