Title of the ABZ Paper / 14p.

Paulius Stankaitis, Alexei Iliasov, Alexander Romanovsky and Yamine Ait-Ameur

Centre for Software Reliability,
Newcastle University,
Newcastle upon Tyne, UK

{p.stankaitis|alexei.iliasov|alexander.romanovsky}@ncl.ac.uk

Abstract.

1 Introduction/1.5p

2 Event-B

The Event-B mathematical language used in the system development and analysis is an evolution of the classical B method [1] and Action Systems [2]. The formal specification language offers a fairly high-level mathematical language based on a first-order logic and Zermelo-Fraenkel set theory as well as an economical yet expressive modelling notation. The formalism belongs to a family of state-based modelling languages where a state of a discrete system is simply a collection of variables and constants whereas the transition is a guarded variable transformation.

```
machine M
  sees Context
  variables $v$
  invariant $I(c, s, v)$
  initialisation $R(c, s, v')$
  events
    $E_1$ = any $vl$ where $g(c, s, vl, v)$ then $S(c, s, vl, v, v')$ end
    ...
end
```

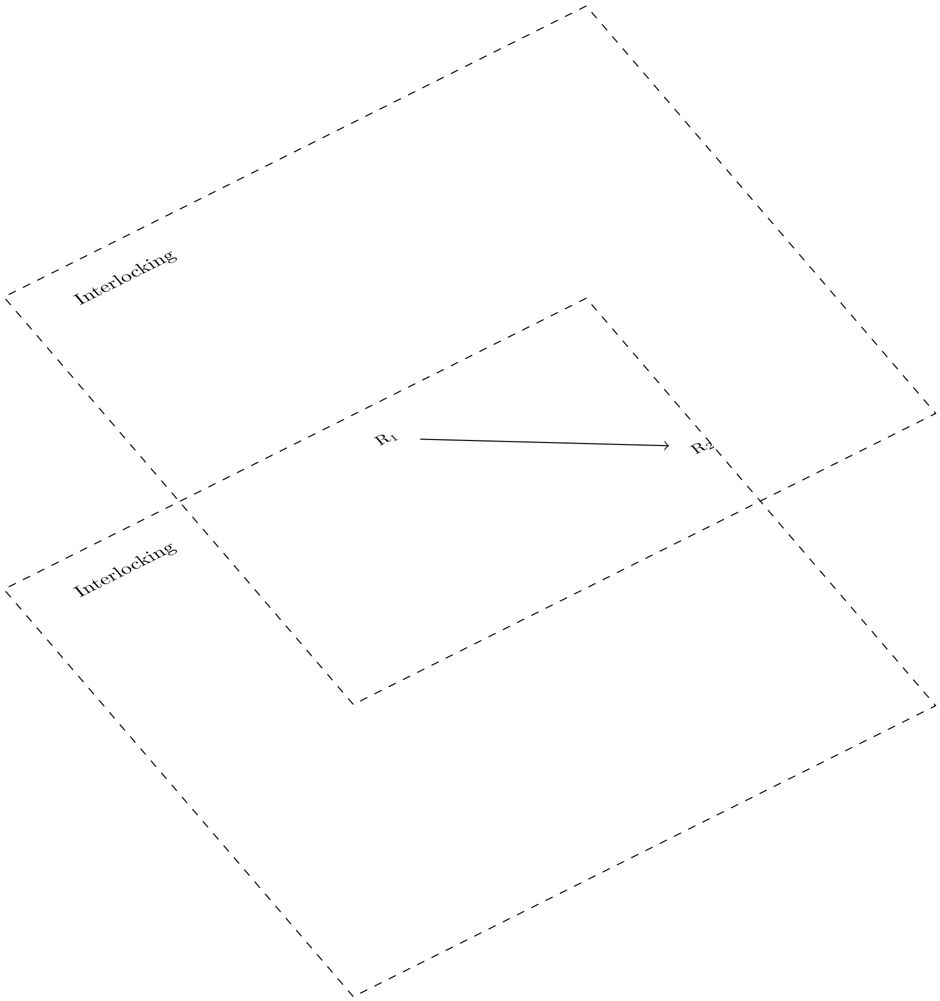
Fig. 1. Event-B machine structure.

A cornerstone of the Event-B method is the step-wise development that facilitates a gradual design of a system implementation through a number of correctness preserving refinement steps. The model development starts with a creation of a very abstract specification and the model is completed when all requirements and specifications are covered. The Event-B model is made of two key components - machines and contexts which respectively describe dynamic and static parts of the system. The context contains modeller declared constants and associated axioms which can be made visible in machines. The dynamic part of the model contains variables which are constrained by invariants and initialised by an action. The state variables are then transformed by actions which are part of events and the modeller may use predicate guards to denote when event is triggered (see Fig. 1). Specifying a model is not sufficient one must provide evidence about the correctness of the model as well. The Event-B method is a proof driven specification language where model correctness is demonstrated by generating and discharging proof obligations - theorems in the first-order logic. The model is considered to be correct when all proof obligations are discharged.

Related Work. To authors knowledge the earliest attempt to formally analyse distributed railway solid-state interlocking systems was completed by Morley [10]. In this interesting work author developed a formal model of a protocol for a cross boundary route locking and releasing mechanism. By analysing temporal properties of the model he discovered that in certain scenarios safety properties can be violated. Few years later a paper by Cimatti et al. [4] presented an industrially driven formal methods study where authors formally modelled a communication protocol for safety-critical distributed systems including distributed railway interlocking systems. Their method used Statecharts diagrams to specify high level protocol properties and the OBJECTGEODE model checker for the protocol validation. In other work a different concept of distributed railway control system was introduced by Haxthausen and Peleska [6]. Their presented engineering concept of the control system relied on a radio based communication and switch boxes - systems which can only control a single railway point. Authors formally modelled the system with the RAISE [5] specification language which allowed to develop a formal model incrementally using a refinement process and prove refinement and safety properties with available justification tools. The timing properties of the design were considered in the extended work [9]. Similar ideas for distributed railway interlocking system were also presented in [3, 7] where authors used Statecharts and Petri Nets to model and verify decentralised railway interlocking.

- write few sentences on how our work is different from theirs.
- discuss relevant papers in Event-B for example [8]

3 Protocol Description/1-1.5p.



System Requirement 1. Cross boundary route locking and releasing protocol must ensure that a cross boundary route has been reserved only to a single train at a time.

System Requirement 2. Cross boundary route locking mechanism must ensure that a locked cross boundary route has points properly positioned and signals sets.

4 Cross Boundary Interlocking Protocol Formal Model in Event-B/5p.

Environment Assumptions 1.

5 Protocol Validation with the ProB/SafeCap Platform/1.5p

6 Future Work and Conclusions/0.5p

Acknowledgments. This work is supported by an iCASE studentship (EPSRC and Siemens Rail Automation). We are grateful to our colleagues from Siemens Rail Automation for invaluable feedback.

References

1. J.-R. Abrial. *The B-book: Assigning Programs to Meanings*. Cambridge University Press, New York, NY, USA, 1996.
2. R.-J. Back. Refinement calculus, part ii: Parallel and reactive programs. In *Proceedings on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, REX workshop, pages 67–93, New York, NY, USA, 1990. Springer-Verlag New York, Inc.
3. M. Banci, A. Fantechi, and S. Gnesi. The role of formal methods in developing a distributed railway interlocking system. In *Proceedings of the 5th Symposium on Formal Methods for Automation and Safety in Railway and Automotive Systems (FORMS/FORMAT 2004)*, pages 220–230, 2004.
4. A. Cimatti, P. L. Pieraccini, R. Sebastiani, P. Traverso, and A. Villafiorita. Formal specification and validation of a vital communication protocol. In *Proceedings of the Wold Congress on Formal Methods in the Development of Computing Systems-Volume II*, FM '99, pages 1584–1604. Springer-Verlag, 1999.
5. C. George, A.E. Haxthausen, S. Hughes, R. Milne, S. Prehn, and J.S. Pedersen. *The RAISE development method.* Prentice Hall Int., 1995.
6. A.E. Haxthausen and J. Peleska. Formal development and verification of a distributed railway control system. *IEEE Transactions on Software Engineering*, 26(8):687–701, 2000.
7. X. Hei, S. Takahashi, and N. Hideo. Toward developing a decentralized railway signalling system using petri nets. In *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, pages 851–855, 2008.
8. Thai Son Hoang, Hironobu Kuruma, David Basin, and Jean-Raymond Abrial. Developing topology discovery in event-b. *Science of Computer Programming*, 74(11):879 – 899, 2009.
9. M.S. Madsen and M.M. Bæk. Modelling a distributed railway control system. Master's thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2005.
10. M.J. Morley. *Safety assurance in interlocking design.* PhD thesis, 1996.