Building Event-B Interlocking Theories: Lessons Learned using the Theory Plug-in

Yoann Guyot, Renaud De Landtsheer, Christophe Ponsard
CETIC Research Center, Charleroi, Belgium
{yoann.guyot, renaud.delandtsheer, christophe.ponsard}@cetic.be

A computer-based interlocking is a railway signalling system that automatically controls the objects of a railway network, such as signals or points, in order to let trains move on its tracks without colliding with other trains nor derailing. A number of approaches based on the Event-B method [1] have already been proposed to model and prove interlocking systems [2, 3]. However its use in the industrial world is still a challenge. First, engineers responsible for specifying these systems are generally not fluent in using formal methods such as the Event-B method. Representing railway specific concepts such as routes, tracks, signals or points using primitive Event-B constructs such as sets, relations and functions quickly leads to models that are both hard to understand and manage. Second, the loss of the rail structure of the problem makes the proof obligations difficult to discharge, hence requires a lot of manual proof-work, also restricting its use to specialists.

A relevant way of making the approach practical for railway engineers is to raise the level of abstraction from the set theory of Event-B to the interlocking domain and also to provide efficient, yet generic enough, proof automation at this level. This can be achieved using the Theory plug-in of the Rodin Platform for system modelling in Event-B [4].

This talk presents our experience and discusses a number of open questions on the use of the Theory plug-in in the context of a work-in-progress aiming at defining a set of interlocking theories ready to be used by signalling engineers. These theories are made of train-specific constructs with a set of theorems and proof rules. They form a domain specific language (DSL) for modelling interlocking systems that has a fully formal semantics enabling to carry out verification activities but also to perform animation and to generate interlocking systems form the model based on a number of standard Rodin plug-ins [5].

Based on the well-known train example of [1], we progressively factorized key domain concepts such as blocks, routes, trains, points, and signals into new theories. Inspired by [6], we have first defined theories for manipulating chains and subchains, in order to be able to express routes properties and relations in a dedicated theory for routes. We notably introduced a new route reservation theory which lets the user manipulate the set of all possible route reservations in the modelled network by providing operators such as:

- **validRouteRes** to select the set of all valid route reservations on a network made of the given blocks and routes

- **compatibleRoutesOnly** to guarantee that two incompatible routes will never be reserved at the same time
- **res_blocks** to get the set of reserved blocks of a given route
- **isReserved** to check whether a given route is reserved.

On the proving side, the definition of theorems partly automates the discharging of proof obligations and also cuts down the effort of manual proving thanks to the ability to reason at the domain level. However, we are still facing a major challenge related to the number of proofs because about only forty percent of the total proof obligations are currently discharged automatically in our model as well as in our theories. In the last part of the talk, we highlight possible directions to tackle this, such as enriching the theorems and defining proof tactics. This might also initiate some discussion about future developments of the Theory plug-in itself.

Acknowledgment

This work was partly funded by the Walloon Region under the INOGRAMS project (grant nr 7171).

References

[1] Jean-Raymond Abrial. *Modeling in Event-B*. Cambridge University Press, 2010.

[2] Minh-Thang Khuu, Laurent Voisin, and Luis-Fernando Meija. Modeling a Safe Interlocking Using the Event-B Theory Plug-in. *Proceedings of the 5th Rodin User and Developer Workshop*, 2014.

[3] Michael Leuschel, Jens Bendisposto, and Dominik Hansen. Unlocking the Mysteries of a Formal Model of an Interlocking System. *Proceedings of the 5th Rodin User and Developer Workshop*, 2014.

[4] Michael Butler and Issam Maamria. *Theories of Programming and Formal Methods: Essays Dedicated to Jifeng He on the Occasion of His 70th Birthday*, chapter Practical Theory Extension in Event-B. Springer Berlin Heidelberg, 2013.

[5] Rodin Community. Rodin Plug-ins. http://wiki.event-b.org/index.php/Rodin_Plug-ins.

[6] Luis-Fernando Mejia, Minh-Thang Khuu, and Asieh Salehi. Application in Railway Domain. *ADVANCE Project Deliverable*, 2014.