Using CSP||B and ProB for railway modelling

Faron Moller[1], Hoang Nga Nguyen[1], Markus Roggenbach[1],
Steve Schneider[2], and Helen Treharne[2]

[1] Swansea University, Wales, UK
[2] University of Surrey, England, UK

1 Introduction

One of the goals of the UK research project SafeCap[3] (Overcoming the railway capacity challenges without undermining railway network safety) is to provide railway engineers with a formal modelling framework for analysing safety and capacity of railway systems. To this end, we have proposed a "natural modelling" approach for specifying railway networks in CSP||B [4], and we are developing the capability to model track plans of increasing complexity. We have considered a simple closed track circuit with points, the 'Mini-Alvey' [2]. We have further considered the 'Double Junction' example [3], which includes a track crossing, adjacent points, more complex route locking and open connections. Once we have a model then we are in a position to formulate and verify safety and liveness properties. Introducing more detailed behaviour, such as points in transition, manual release of routes, multi-aspect signalling and more complex driving rules is currently in development. Our approach uses the case studies to drive the development of patterns comprising a generic style for railway modelling.

In our approach, the railway models are as close as possible to the domain model, providing traceability and ease of understanding to the domain expert. This leads to a natural separation between the global modelling of the tracks in B, and the CSP encapsulation of the local views of the individual trains following the driving rules. In this poster we illustrate the modelling approach through the Mini-Alvey case study, and see how the model provides verification through model checking or informative counter example traces if verification fails.

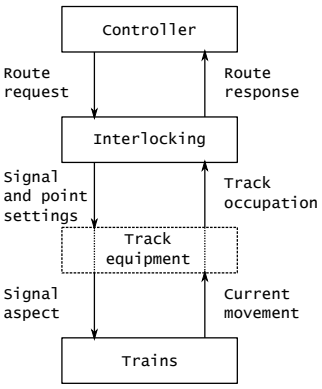
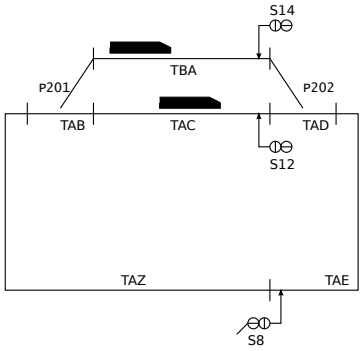


Fig. 1. Information flow.

2 The railway domain

Railways consist of (at least) four, physically different entities: see Figure 1. The *Controller* selects routes for trains and sends requests of routes to the *Interlocking*. The interlocking monitors the *Track equipment* and sends out commands to

[3] SafeCap's web site: `http://safecap.cs.ncl.ac.uk`.



Control table

Route	Normal	Reverse	Clear
8A	P201		TAZ TAB TAC
8B		P201	TAZ TAB TBA
12A	P202		TAD TAE
14A		P202	TAD TAE

Release tables

P201	Occupied
8B	TAC
8A	TBA

s

P202	Occupied
12	TAE
14	TAE

Fig. 2. Mini-Alvey.

control it with respect to the route requests and a pre-defined control table. The track equipments consists of elements such as signals, points and track circuits: signals can show green or red (the yellow aspect of a signal is not modelled at this level of abstraction since we are only interested in whether a train is authorized to enter a section); points can be in normal position (leading trains straight ahead) or in reverse position (leading trains to a different line) and track circuits detect if there is a train on a track. Finally, *Trains* have a driver who determines their behaviour.

Railways are built according to a *Track plan*. Figure 2 depicts a prominent example referred to in the literature as the Mini-Alvey track plan [5, 6]. This plan shows various tracks (TAB, TAC, TAD, ...), signals (S8, S12, S14), and points (P201, P202). This plan is accompanied with a control table describing conditions under which signals at the beginning of every route[4] can show proceed. For example, signal S12 for the route between S12 and S8 can only show proceed if point P202 is in normal (straight) position and tracks TAZ, TAB and TBA are clear. When a signal shows proceed, points on the corresponding route are locked to prevent trains from derailment. They are released according to the *Release tables* associated with each point. For example, locked P201 for route 8B from S8 to S12 will be released if TAC is occupied. In such a railway system, we are interested in verifying *Safety* properties. This means no collision (one train moving into another) and no derailment (points moving under trains, trains moving onto points from the wrong direction, trains travelling too fast).

3 A CSP||B model

The architecture of our model[5] is depicted in Figure 3. The CTRL component is a CSP description which is used to describe the driving rules of trains in order to control their movement (such as never pass a red signal) and enable the Controller to issue route requests. The Interlocking component is a B-machine which

[4] A *route* is a (directed) path which leads from one signal to the next signal.
[5] CSP||B Mini-Alvey model download: `http://www.csp-b.org/mini-alvey.zip`.

describes the general principles of an interlocking such as considering route requests by following the conditions of a control table to determine whether or not the requests are granted and monitoring the state of points and signals and the locations of trains. This component is generic and does not depend on a particular track plan. Conversely, other components are for modelling a specific track plan. ClosedContext declares track equipment such as tracks, signals and points. ClosedTopology describes the connections between tracks as well as the position of signals and points in the track plans. Then, ControlTable and ReleaseTable encode the corresponding components from the track plan. The CSP||B technical descriptions can be found in [2].

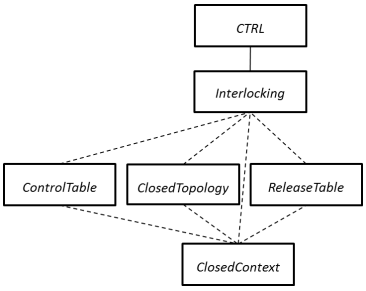


Fig. 3. CSP||B Architecture

4 Verification

Our CSP||B models can be verified using ProB [1] which supports B models that are controlled by CSP controllers. In this section we illustrate the use of ProB to verify safety properties of a railway system, represented as invariants on the *Interlocking* machine. For example, we capture the notions of no-collision and no-derailment in the invariant $pos : TRAIN \rightarrowtail\!\!\!\!\!\!\rightarrow TRACK$ on the *pos* function. This constrains no more than one train on any track circuit, and also that no train is on *nullTrack*, since $nullTrack \notin TRACK$. ProB verifies that this invariant is preserved in our model.

In the following, we consider two faulty scenarios in order to explore how the modelling and analysis exposes errors in the design. In each case ProB discovers violations of the invariant:

CSP||B model with faulty clear tracks: Suppose the control table is adjusted to contain the mistake that TAB is omitted from the tracks which should be clear to grant route $8B$. Then the following trace is produced automatically as a counter-example:

$\langle enter.albert.TAB, enter.bertie.TAE, request.B8.true,$
$nextSignal.bertie.green, move.bertie.TAE.TAZ, nextSignal.bertie.none,$
$move.bertie.TAZ.TAB\rangle$

This leads to a collision of *albert* and *bertie* on TAB.

CSP||B model with faulty points in control table: If the control table contains a mistake on the directions of points, e.g., $P202$ is normal (straight) position for route 14A. Then the check yields the following counter-example trace showing the derailment of *bertie*:

$\langle enter.albert.TAB, enter.bertie.TBA, request.A14.true,$
$nextSignal.bertie.green, move.bertie.TBA.nullTrack\rangle$

This demonstrates a violation of the safety requirement no-derailment.

5 Conclusion

This poster presents our approach to modelling in the railway domain. The "hybrid nature" of railways (namely, that some railway aspects can be directly expressed in an event-based approach while other aspects are more suited for a state-based approach) allows us to construct natural railway models in CSP||B, which are immediately understandable to the railway experts and analysable by current verification technologies.

We are developing our approach by applying it to more complex track designs, with more detailed behaviour and driving rules. We are also extending this approach in order to include the time aspect into railway models which will allow the study of both safety and capacity in an integrated way, to address the goals of the SafeCap research project.

Acknowledgement: The authors would like to thank S. Chadwick and D. Taylor from the company Invensys Rail for their support and encouraging feedback; and also Erwin R. Catesbeiana for keeping us on track.

References

1. M. Leuschel and M. Butler. ProB: an automated analysis toolset for the B method. *Int. J. Softw. Tools Technol. Transf.*, 10(2):185–203, Feb. 2008.
2. F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne. Combining event-based and state-based modelling for railway verification. Technical Report CS-12-02, Department of Computing, University of Surrey, 2012.
3. F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne. CSP‖B modelling for railway verification: the double junction case study. Technical Report CS-12-03, Department of Computing, University of Surrey, 2012.
4. S. Schneider and H. Treharne. CSP theorems for communicating B machines. *Formal Asp. Comput.*, 17(4):390–422, 2005.
5. A. Simpson, J. Woodcock, and J. Davies. The mechanical verification of solid-state interlocking geographic data. In *Formal Methods Pacific '97*. Springer, 1997.
6. K. Winter and N. Robinson. Modelling large railway interlockings and model checking small ones. In *26th ACSC*. Australian Computer Society, Inc., 2003.