Towards the Verification of Bidirectional Railway Models in CSP||B

Phillip James[1], **Faron Moller**[1], **Hoang Nga Nguyen**[2], **Markus Roggenbach**[1], **Steve Schneider**[3], **Helen Treharne**[3]

[1]Swansea University; [2]University of Nottingham; [3]University of Surrey

Abstract: We develop proof support for bi-directional railway modelling.

Keywords: Railway verification; CSP||B; model checking.

Formal verification of railway control software has been identified as one of the "Grand Challenges" of Computer Science [Jac04]. In a number of publications [MNR+12, MNR+13, JMN+14b, JMN+b] we have – in cooperation with our industrial partner Siemens Rail Automation UK – developed a scheme-plan verification approach based on the formal method CSP||B. Characteristics of our approach include: a railway Domain Specific Language (DSL); model transformations between DSL representations and formal models; general abstractions on the DSL level proven to be correct relative to the formal models; and ongoing implementations in our OnTrack tool [JTT+13]. In this, uni-directional scheme-plans have been our main focus to ease both modelling and verification. In this paper we focus on how to extend our approach towards the verification of bi-directional scheme plans, for example, an end station as shown in Figure 1. Here, trains can move from from the entry track with Signal 5 to the two platforms A and B; furthermore, trains can leave platform A and platform B towards the Exit track. In our example, track TC6 is used in opposing directions: one direction for moving into platform A, and one for moving out of platform A.

Uni-directional train control systems are considerably simpler than their bi-directional counterparts. Uni-directional systems have to prevent the setting of a route that is currently in use, i.e., a train is travelling on it. In Figure 1 an example of a route would be the sequence of units $\langle TC9, TC8, TC7, TC6, TC3\rangle$ from signal S5 to signal S7. Uni-directional systems further have to prevent conflicting routes from being set at the same time. Routes are said to be conflicting when they share a unit. In Figure 1, e.g., the route from signal S5 to signal S7 and the route from signal S5 to signal S8 are conflicting, as they for instance share track TC9. Bi-directional systems additionally have to deal with the direction of travel. This leads to the new challenge of preventing opposing routes from being set at the same time. Routes are called opposing when they share a unit but for use in opposite directions. In Figure 1, e.g., the route from signal S5 to signal S7 and the route from signal S2 to the exit track are conflicting, as they use TC6 in opposite directions. In interlocking design, this challenge is addressed by route locking and release of units in sequence behind the train (therefore sometimes also called sequential release).

Recently we have extended our modelling approach for bi-directional scheme plans [JMN+14a, JMN+a] to include directional information which models sequential release. This increases code complexity (as events have to take care of directions). Also, in model checking, the state space becomes considerably larger. E.g., our single junction [MNR+13] using uni-directional modelling has 8,646 states, however has 196,284 states when using bi-directional modelling (both models include two trains), showing that our new bi-directional models are of a larger complexity. The single junction under discussion consists of 15 tracks, one point, and six signals. For

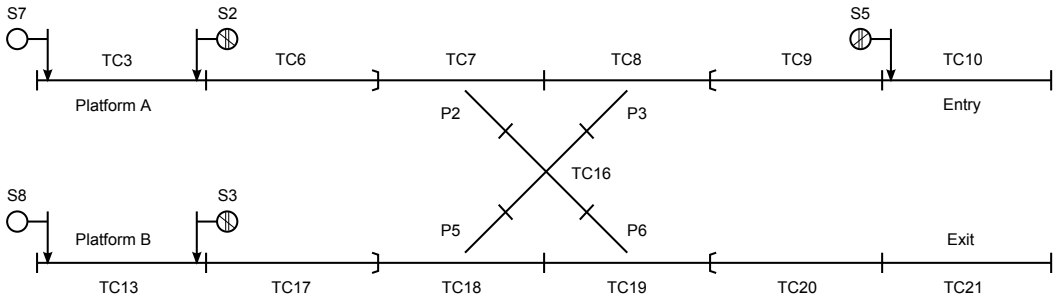

Figure 1: An end station with two platforms.

comparison: our bi-directional model of the end station shown in Figure 1 has 32,369 states. The end station consists of eight tracks, one crossing, four points, three route signals and two fixed signals.

In the unidirectional world, we developed three abstractions [JMN+14b] that were key for verification: finitisation, covering, and topological abstraction, without which model checking large models is not feasible. Here we focus on the finitisation abstraction for bi-directional rail systems.

Safety in our models is dependent on the number of trains that are introduced into the model. This motivates the following definition of safety.

Definition Let $ERROR = \{collision, derailment, runthrough, grantedRouteOccupied, offRoute\}$ be the set of errors of interest. We make the following definitions.

- A scheme plan *SP* is *n-e-free* (for $n \in \mathbb{N}_{>0}$ and $e \in ERROR$) iff e never occurs in a run of our bi-directional model of *SP* involving at most n trains.
- A scheme plan *SP* is *safe* iff it is *n-e*-free for all $n \in \mathbb{N}_{>0}$ and $e \in ERROR$.

The contribution of this paper is to provide the technique of finitisation for bi-directional railway models, i.e., to reduce the proof of safety to a proof of 2-*e-freedom* for $e \in ERROR$. In particular, we show the following.

Theorem 1 A scheme plan *SP* is safe if it is 2-$collision$-free, 1-$derailment$-free, 1-$runthrough$-free, 1-*grantedRouteOccupied*-free and 1-*offRoute*-free.

The proof of this theorem is by induction over the length of a run of our model. Here, we have to argue about each line of the code. Due to the higher complexity of the code, this proof is considerably longer – due to the necessity to argue about sequential release, it is also more difficult than its uni-directional counterpart.

Theorem 2 If a scheme plan *SP* is *l-e*-free then *SP* is *k-e*-free for any $k < l$.

Corollary A scheme plan *SP* is safe if it is 2-*e*-free for all $e \in ERROR$.

This corollary allows us to reduce the safety proof for a scheme plan to one model checking run based on two trains. Based on this technique, we successfully proved our single junction as

well as the end station from Figure 1 to be safe (when equipped with appropriate control and release tables).

The bi-directional modelling is of interest for the wider community since it is a step towards modelling of the European Train Control System, which is also based on sequential release. Similarly, the theoretical result of finitisation is of interest, as it provides further justification for applying bounded model checking to rail control systems. It is future work to adapt covering and topological abstraction for bi-directional modelling as well.

Acknowledgement The authors would like to thank Simon Chadwick, Siemens Rail Automation, for helpful discussions on sequential release, and Erwin R. Castesbeiana (Jr.) for pointing out the right directions.

Bibliography

[Jac04] R. Jacquart (ed.). *IFIP 18th World Computer Congress, Topical Sessions*. Chapter TRain: The Railway Domain - A Grand Challenge. Kluwer, 2004.

[JMN+a] P. James, F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, H. Treharne. Decomposing scheme plans to manage verification complexity. In *FORMS/FORMAT 2014*. To appear.

[JMN+b] P. James, F. Moller, H. Nguyen, M. Roggenbach, S. Schneider, H. Treharne. On Modelling and Verifying Railway Interlockings: Tracking Train Lengths. *SCP*. To appear.

[JMN+14a] P. James, F. Moller, H. Nguyen, M. Roggenbach, S. Schneider, H. Treharne, M. Trumble, D. Williams. Verification of Scheme Plans Using CSP||B. In *SEFM'13*. LNCS 8368. Springer, 2014.

[JMN+14b] P. James, F. Moller, H. Nguyen, M. Roggenbach, S. Schneider, H. Treharne. Techniques for modelling and verifying railway interlockings. *STTT*, 2014.

[JTT+13] P. James, M. Trumble, H. Treharne, M. Roggenbach, S. Schneider. OnTrack: An Open Tooling Environment for Railway Verification. In *NASA Formal Methods'13*. LNCS 7871. Springer, 2013.

[MNR+12] F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, H. Treharne. Railway modelling in CSP||B: the double junction case study. In *AVoCS'12*. ECEASST 53. 2012.

[MNR+13] F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, H. Treharne. Defining and Model Checking Abstractions of Complex Railway Models Using CSP||B. In *HVC'12*. LNCS 7857. 2013.