OnTrack: The Railway Verification Toolset
Extended Abstract

Phillip James[1(✉)], Faron Moller[1], Hoang Nga Nguyen[2], Markus Roggenbach[1], Helen Treharne[3], and Xu Wang[1]

[1] Swansea University, Swansea, Wales
{p.d.james,f.g.moller,m.roggenbach}@swansea.ac.uk,
xu.wang.comp@gmail.com

[2] University of Coventry, Coventry, England
hoang.nguyen@coventry.ac.uk

[3] University of Surrey, Guildford, England
h.treharne@surrey.ac.uk

Abstract. The verification of railway interlocking systems is a challenging task for which a number of different modelling, simulation and verification approaches have been proposed. In this paper, we present the OnTrack toolset. In OnTrack, application data for the railway domain is represented using a domain specific language. This data can be entered manually or imported from standard data formats. OnTrack then comprises of a number of different model transformations that allow the user to automatically generate formal models for a specific approach, e.g., in CASL, CSP, or CSP||B. Other transformations offer abstractions on the application data to address scalability.

OnTrack [3] is an open toolset for railway verification developed in collaboration between Swansea University and Surrey University. It has been created using the GMF framework [5] and multiple associated Epsilon [4] model transformations.

Figure 1 shows the workflow that OnTrack currently provides. Firstly, given a CAD track plan and associated control tables, a user draws a scheme plan using the OnTrack graphical front end (centre path). Scheme plans are models formulated relative to the metamodel of OnTrack's Domain Specific Language (DSL). One can also read models described in the BRaVE [1] tool format (right path), whilst reading in RailML representations (left path, dashed) is work in progress. A scheme plan is then the basis for workflows that support its abstraction or analysis. For example, a covering construction [2] is implemented as an *abstraction transformation* on the DSL level.

Scheme plans can be translated to formal specifications in various (specification) formalisms. This can be achieved in two ways. The first approach is to use a metamodel describing the formal specification language. A *represent transformation* translates a scheme plan over the DSL into an equivalent formal scheme plan over the metamodel. Various *model-to-text transformations* then turn a formal scheme plan into formal specification text ready for verification. The advantage

T. Margaria and B. Steffen (Eds.): ISoLA 2016, Part II, LNCS 9953, pp. 294–296, 2016.
DOI: 10.1007/978-3-319-47169-3_21

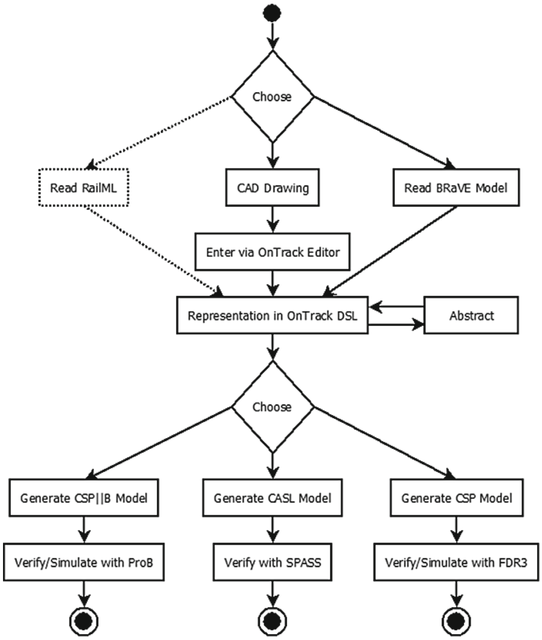

Fig. 1. The OnTrack workflow.

of this approach is that all transformations involved can be defined relative to the specification language's metamodel. The second approach is to directly generate a formal specification. Here, transformations turn a scheme plan over the DSL directly into formal specification text ready for verification.

Once a formal model has been generated, it can then be simulated or verified using the tools associated with the chosen approach. For example, ProB and FDR3 can be used for simulating and verifying CSP||B and CSP models (resp.) for safety, whereas the SPASS theorem prover can be used for verifying CASL models for safety.

Finally, OnTrack is extendable, e.g., other models from various contexts can be generated and analysed through the implementation of new model transformations. For example, the generation of Real Time Maude models is currently planned.

References

1. Douglas, H., Weston, P., Kirkwood, D., Hillmansen, S., Roberts, C.: Method for validating the train motion equations used for passenger rail vehicle simulation. J. Rail Rapid Transit (2016)
2. James, P., Moller, F., Nguyen, H., Roggenbach, M., Schneider, S., Treharne, H.: Techniques for modelling and verifying railway interlockings. STTT **16**(6), 685–711 (2014)

3. James, P., Trumble, M., Treharne, H., Roggenbach, M., Schneider, S.: OnTrack: an open tooling environment for railway verification. In: Brat, G., Rungta, N., Venet, A. (eds.) NFM 2013. LNCS, vol. 7871, pp. 435–440. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38088-4_30
4. Kolovos, D., Rose, L., Paige, R., Polack, F.: The Epsilon Book (2013)
5. Steinberg, D., Budinsky, F., Merks, E., Paternostro, M.: EMF. Pearson (2008)