COMPUTING SCIENCE

The SafeCap Platform for Modelling Railway Safety and Capacity

Alexei Iliasov, Ilya Lopatkin and Alexander Romanovsky

TECHNICAL REPORT SERIES

No. CS-TR-1382 April 2013

TECHNICAL REPORT SERIES

No. CS-TR-1382 April, 2013

The SafeCap Platform for Modelling Railway Safety and Capacity

A. Iliasov, I. Lopatkin, A. Romanovsky

Abstract

This paper describes a tooling platform that supports reasoning about railway capacity while ensuring system safety. It uses a Domain Specific Language (DSL) that allows signalling engineers to design stations and junctions, to check their safety and to evaluate the potential improvements of capacity while applying various alteration patterns that change the railway schemas. The platform uses a combination of model checking and SMT solving to verify system safety in the most efficient and user-friendly way. It includes several plug-ins that evaluate various capacity parameters. The tool uses the Eclipse technology, including its EMF and GMF frameworks. It has been developed in close cooperation with the Invensys Rail engineers and applied in a variety of medium-scale projects, which has demonstrated its ability to help understand the effects that changes in the plans and schemas can potentially have on capacity.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

ILIASOV, A., LOPATKIN, I., ROMANOVSKY, A.

The SafeCap Platform for Modelling Railway Safety and Capacity
[By] A. Iliasov, I. Lopatkin, A. Romanovsky

Newcastle upon Tyne: Newcastle University: Computing Science, 2013.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1382)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1382

Abstract

This paper describes a tooling platform that supports reasoning about railway capacity while ensuring system safety. It uses a Domain Specific Language (DSL) that allows signalling engineers to design stations and junctions, to check their safety and to evaluate the potential improvements of capacity while applying various alteration patterns that change the railway schemas. The platform uses a combination of model checking and SMT solving to verify system safety in the most efficient and user-friendly way. It includes several plug-ins that evaluate various capacity parameters. The tool uses the Eclipse technology, including its EMF and GMF frameworks. It has been developed in close cooperation with the Invensys Rail engineers and applied in a variety of medium-scale projects, which has demonstrated its ability to help understand the effects that changes in the plans and schemas can potentially have on capacity.

About the authors

Alexei Iliasov is a Researcher Associate at the School of Computing Science of Newcastle University, Newcastle-upon-Tyne,UK. He got his PhD in Computer Science in 2008 in the area of modelling artefacts reuse in formal developments. His research interests include agent systems, formal methods for software engineering and tools and environments supporting modelling and proof.

Ilya Lopatkin is a PhD candidate and a Research Assistant in the Centre for Software Reliability. His main research interests are system dependability, fault tolerance, cyber-physical aspects of systems, formal modelling, and system verification. From 2008 - 2012 he was involved in the FP7 DEPLOY Integrated Project on Industrial Deployment of System Engineering Methods Providing High Dependability and Productivity. He is now involved in EPSRC/RSSB research project SafeCap on Overcoming the Railway Capacity Challenges without Undermining Rail Network Safety, and the TrAmS-2 EPSRC/UK platform grant on Trustworthy Ambient Systems.

Alexander (Sascha) Romanovsky is a Professor in the Centre for Software and Reliability, Newcastle University. His main research interests are system dependability, fault tolerance, software architectures, exception handling, error recovery, system structuring and verification of fault tolerance. He received a PhD degree in Computer Science from St. Petersburg State Technical University and has worked as a visiting researcher at ABB Ltd Computer Architecture Lab Research Center, Switzerland and at Istituto di Elaborazione della Informazione, CNR, Pisa, Italy. In 1993 he became a postdoctoral fellow in Newcastle University, and worked on the ESPRIT projects on Predictable Dependable Computing Systems (PDCS), Design for Validation (DeVa) and on UK-funded projects on the Diversity, both in Safety Critical Software using Off-the-Shelf components. He was a member of the executive board of EU Dependable Systems of Systems (DSoS) Project, and between 2004 and 2012 headed projects on the development of a Rigorous Open Development Environment for Complex Systems (RODIN), and latterly was coordinator of the major FP7 Integrated Project on Industrial Deployment of System Engineering Methods Providing High Dependability and Productivity (DEPLOY). He now leads work on fault tolerance in Systems of Systems within the COMPASS project and is Principal Investigator of Newcastle's Platform Grant on Trustworthy Ambient Systems.

Suggested keywords

RAILWAY INDUSTRY
SAFETY VERIFICATION
SIGNALLING SCHEMAS
DSL
ECLIPSE
SIMULATION

The SafeCap Platform for Modelling Railway Safety and Capacity

Alexei Iliasov, Ilya Lopatkin, Alexander Romanovsky

School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
alexei.iliasov@ncl.ac.uk, ilya.lopatkin@ncl.ac.uk,
alexander.romanovsky@ncl.ac.uk

Abstract. This paper describes a tooling platform that supports reasoning about railway capacity while ensuring system safety. It uses a Domain Specific Language (DSL) that allows signalling engineers to design stations and junctions, to check their safety and to evaluate the potential improvements of capacity while applying various alteration patterns that change the railway schemas. The platform uses a combination of model checking and SMT solving to verify system safety in the most efficient and user-friendly way. It includes several plug-ins that evaluate various capacity parameters. The tool uses the Eclipse technology, including its EMF and GMF frameworks. It has been developed in close cooperation with the Invensys Rail engineers and applied in a variety of medium-scale projects, which has demonstrated its ability to help understand the effects that changes in the plans and schemas can potentially have on capacity.

Keywords. Domain-specific language, Eclipse, EMF, GMF, Event-B, ProB, SMT solvers, model transformation, capacity-improving patterns

1 Introduction

Ensuring railway safety has been an area of successful application of formal methods and tools (the most famous example being the use of B by Matra for developing the automated metro line 14 in Paris [1]). There is a substantial body of work on formal verification of railway safety. For example, paper [2] defines a model-based development method for railway control systems that uses a domain-specific notation from which the models of the controllers and the domain are generated to be used to verify the safety properties by model checking. Another domain-specific language (DSL), called the Train Control Language (TCL), is in the core of the method in [3], used to verify and validate the safety of stations; this is achieved by transforming the TCL models into the Alloy models used for constraint solving. These two methods are supported with tools, while the models are created in DSLs and automatically mapped into a formal notation used for safety verification.

There are many reasons why improving railway capacity is now one of the top pri-

orities for the railway industry, including the need to satisfy passengers' requirements, tackle global warming and save running costs. Clearly, any improvements in capacity must not undermine system safety. Except for the work conducted by our collaborators in the SafeCap project [4, 5], the existing work on formal verification of railway safety does not address safety and capacity in an integrated way.

2 The SafeCap Approach

Designing railway nodes (junctions and stations) for high capacity is an art: experienced engineers know the patterns that lead to success or failure and are aware of the bottlenecks that exist within their designs. However, such an intuitive approach lacks scientific foundations, which are the basis of any advanced engineering practice. To this end, the SafeCap project [5] aims to provide a scientifically sound framework that will allow railway engineers to study railway nodes as well as design patterns, addressing safety and capacity in an integrated way

The general approach in SafeCap consists of the following steps. First, engineers model a junction or a station and evaluate its capacity. At the second step, they try to improve capacity by applying alteration patterns (in effect model transformations). These patterns can capture various changes in the route design, track layout and signalling, etc. At the next step the capacity of the modified model is evaluated. The safety of models is formally proven at every step. This approach allows the engineer to experiment with various designs to achieve better capacity.

The importance of tool-supported formal reasoning is well recognised by both engineers and academics. This is why in the SafeCap project we have been developing a tooling platform that can support domain experts, including signalling engineers, in making rigorous decisions about improving capacity. To make our tool more acceptable for industrial users, we follow the push-button approach, in which safety verification and capacity evaluation are conducted in a transparent fashion. Another choice we made in designing the platform is the exclusive use of a graphical and intuitive DSL, saving engineers from the need to understand intricate formal notations. One more feature of the tool is extensive support for representing and reusing modelling patterns that capture best practice and experience. All this should help the platform to be widely accepted by industry.

3 DSL

SafeCap offers a fairly compact core DSL. The basic element of a SafeCap schema is the definition of railway topology. The main concepts of the DSL are tracks, nodes, ambits (train detection units), routes, lines and rules. The SafeCap DSL is a *formal* language: a schema is interpreted as a hybrid transition model - a model mixing continuous and discrete behaviours. The discrete part is employed to derive static verification conditions (theorems) and, as a supplementary technique, to help discover transition traces leading to the violation of safety conditions. The continuous part refines the discrete part with the notions of train acceleration/deceleration, point switching

and driver reaction times, and so on.

Static verification conditions are logical constraints over the elements of a schema expressing requirements to schema topology, formation of routes, placement of speed limits and signalling rules of routes and points. If all such conditions are discharged, it is guaranteed that the schema is safe for any possible rail traffic. Together, the conditions form the *theory of the SafeCap schemas* [6].

4 The SafeCap Tooling Platform

The following aims were set for building the tool support. Our plan was to use an industry-strength modern technology that would allow us to create a tooling environment that is open for extensions and supports graphical modelling using our DSL. The technology should have mature support for model transformation, to allow us to deal with different representations of the schemas. We wanted to use a technology that smoothly supports tool development on various operating systems.

We have decided to use Eclipse as the basis for our tool. Eclipse is a mature and extensible IDE framework that offers a powerful and flexible customisation framework. The concept of proving extra functionality via self-contained plug-in projects makes it easier to support and maintain a large project with a team of developers.

The SafeCap platform uses the Eclipse Modelling Framework (EMF) - a versatile tool for the implementation of a custom domain-specific language. We have found it sufficient to capture the static part of SafeCap DSL.

We use the Eclipse Graphical Modelling Framework (GMF) that builds on top of the EMF to provide means for the rapid construction of graphical editors manipulating EMF models. The GMF offers a layer of abstraction above the code level to define core properties of editors. This is how the SafeCap editor is built.

Openness is the key property of the platform. To make the tool truly open and customisable, it was decided to support a scripting language that executes directly within the platform and is used to implement its main functionalities (verification, capacity assessment and improvement patterns). This frees users from having to learn Java and Eclipse API in order to customise the platform logic. The Epsilon family of languages [7] was deemed perfectly suitable for the task. Among other things, the SafeCap platform uses Epsilon to support creation and application of user-defined patterns for transformation of schemas. At the moment, these are used during modelling and capacity improvement. We plan to extend the application of patterns to support automatic search for transformations that improve capacity.

5 Safety Verification Architecture

Safety is verified by formulating railway schema properties and operational principles in a rigorous notation. The desire to have a strict mathematical foundation spanning the principal aspects of railway operation has led to a distinctive four-layered architecture of the verification back-end.

In this architecture, *the first (schema topology theory) layer* is responsible for verifying logical conditions expressed over schema physical topology (i.e., track connec-

tions, point placement) and logical topology (i.e., routes and lines as paths through a schema). These verification conditions include connectivity of track topology (no isolated pieces of track), continuity of routes and lines, etc. Typically these conditions do not uncover problems with an existing track layout, for any such defect would have a profound effect on the overall integrity - something unlikely to not have been discovered in an operational railway. It is the automated and semi-automated alteration and generation of track layouts (e.g., via the improvement patterns) that necessitates a careful inspection of these basic properties.

In *the control table theory layer* the conditions for operational safety are defined. These are derived, via formal proof, from a set of discrete (inertia-less) train movement rules into a set of theorems over the properties of control tables. We depart from the convention of associating control rules with trackside signals. Instead, we consider a more general situation where differing set of signalling rules are applied depending upon the ultimate train destination or train type. The conditions proven for a control table demonstrate such properties as the absence of potential collision (as may happen, for instance, when a proceed aspect is given while a protected part of track is still occupied) and derailment (due to incorrect point setting or point movement under a train). Certain properties, notably the control of the approach speed by means of the timed occupation of a track section, are not verified at this stage, as formalisation at this layer does not capture train inertia. Speed limit conformance and other time-related properties are formulated at the fourth layer.

The first two formalisation layers do not define the notion of a train. However, there is a link between the conditions over control tables expressed in the second layer and the notion of train movement. The definition of the latter is the purpose of *the third (discrete driving model) layer*. This layer defines principal events that are observed during railway operation: train movement, route reservation, point locking, route cancellation and so on. On the basis of these one can state operational safety principles by using safety invariants (e.g., trains are not overlapping) or by explicitly modelling possible operation faults, e.g., uncontrolled carriage movement on an adjacent route (to ensure the correctness of flank protection). Apart from its role in the validation of the first two layers, the discrete operational rules of the third layer are used to visually animate train movements over a given schema. There are two main applications for such an animation: replaying the results of model checking of discrete driving rules in order to pinpoint the source of an error in a topology or a control table; and helping an engineer to understand how trains may travel through a schema with a given set of control rules.

Train inertia is the focus of *the final modelling (inertial driving model) layer*. This layer is built from the same primitives as the rules of the third layer, with the main difference that discrete rules are defined as atomic sequences of primitive steps (for instance, a train head movement, in one step, moves the train head position, unlocks an ambit and, sometimes, resets a signal to red), whereas the more detailed inertial model accounts for the duration of every action and executes concurrent actions in a more realistic way than the lock-step fashion of the discrete model. This transition to inertial, timed transformation uncovers a wealth of concerns. The most essential one, perhaps, is whether the properties of operational safety still hold for the inertial mod-

el. There is not an unconditional answer to this question. In fact, one can show a certain railway configuration accepted by the inertial model to be unsafe. The ability to run a detailed simulation (with a fixed service pattern) that accurately deals with track gradient, engine performance, train weight, etc., gives access to rich information about realistic node performance.

The primary mechanism for verifying safety is constraint solving. After the schema is defined, the platform mechanically derives verification conditions and translates them into an Event-B model that serves as an input notation for an SMT-LIB-compliant SMT solver (Yices [8]). One downside of applying constraint solving is the absence of any useful feedback to indicate the source of a problem should an error be discovered. To compensate for this, whenever a SMT solver detects a problem, the tool runs the ProB model checker [9]. Unlike solvers, the model checker explores the state space of a discrete transition system that gives semantics to the SafeCap schemas represented as the Event-B model with the DSL axioms defined in the machine context. It is thus able to report a sequence of steps (discrete train movements, point switching, etc.) that leads to violation of a safety condition (e.g. a collision or derailment). In most cases, such a sequence can be visually replayed by the tool platform to help the user debug the schema.

This approach to safety verification is superior to the traditional analysis that uses model checkers in terms of efficiency and of the size of the models analysed. Another advantage is that the feedback received by users is expressed in the DSL.

6 Reasoning about Capacity

Depending on the objectives of modelling, the capacity assessment is conducted by calculating a value according to one of the predefined formulae or by running a detailed simulation of train movements. The platform has a range of plug-ins supporting various capacity metrics. Below we introduce some of the capacity plug-ins.

Theoretical line capacity. This criterion assesses the capability of a line to support a certain amount of traffic irrespective of signalling constraints and safety requirements. The plug-in calculates a theoretical line capacity in trains per second.

Critical section. One obvious weakness of the theoretical capacity method is its inability to capture the notion of shared or intersecting track and hence the interference of traffic on crossing lines that leads to decreased capacity. The plug-in calculates the maximum time a train on a given line occupies the critical section.

Wasted track capacity. One could take a different viewpoint and try to assess how efficiently the existing signalling makes use of the available track. Assuming a certain traffic pattern, the plug-in measures the minimum amount of free track observed during a traffic scenario. The fundamental idea is that line interference and inefficient signalling tend to increase train headways.

Cumulative travelled distance. This plug-in measures the total distance travelled by all the trains that have entered the junction. The measurement is done for a set period of time and starts after a certain delay in an attempt mitigate the effect of the initial absence of traffic. The guiding intuition here is that a more efficient layout and

signalling favour a balance between higher average speed and less wasted capacity.

Satisfaction of schedule. If there is a detailed specification of a desired traffic pattern through a junction then one may take the ability to satisfy the pattern as a measure of capacity. A service pattern defines train kinds and the times trains appear at boundary nodes of a schema during a period known as pattern duration. The plug-in checks schedule satisfaction by running a timed simulation of the hybrid model.

7 Experience of Using the Platform

To evaluate the SafeCap platform, we developed two large-scale examples modelling the existing UK stations. The primary objective for modelling the case studies was to improve and evaluate the scalability and usability of the tool. The first case study modelled is a fragment of the Thameslink line around the Kentish Town station. The fragment is 5.5 km long, with the model containing 90 ambits and 63 routes. The modelling took 37 man-hours. Our main activity was the translation of traditional railway diagrams into the SafeCap DSL using the platform.

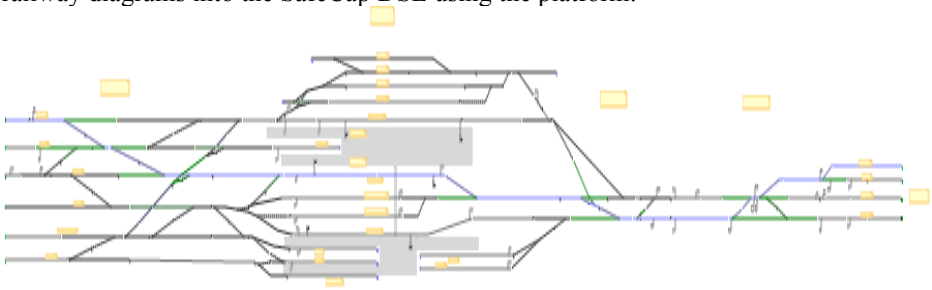

Figure 1. Carlisle central station and junctions on approaches

The second case study is the Carlisle Citadel station with the North, South, and Caldew junctions (see Figure 1). The modelled fragment is 2.6 km long and is made of 70 ambits and 79 routes. The translation activity took 45 man-hours. The safety verification of the schema topology requires 35 minutes on a modern computer and goes through 877 individual instantiated conditions. Figure 2 shows the editing mode of the SafeCap platform for this model. The main view gives a visual representation of the track layout and some signalling mark-up for the Carlisle station platforms.

The platform provides a blocking diagram tool that tries to reduce the time of schedule satisfaction by identifying and reducing the wasted track capacity. This was used to conduct a short capacity-improvement experiment (Figure 3). The total time during which a set of trains on particular lines (a service pattern) travels through the schema was taken as a measurement of capacity. Using the tool to identify a bottleneck made it possible to improve the capacity by 5%. (We should note here that we could not claim that this will always improve the capacity of the real station, as there are many factors to be considered, such as cost or validity of our assumptions.)

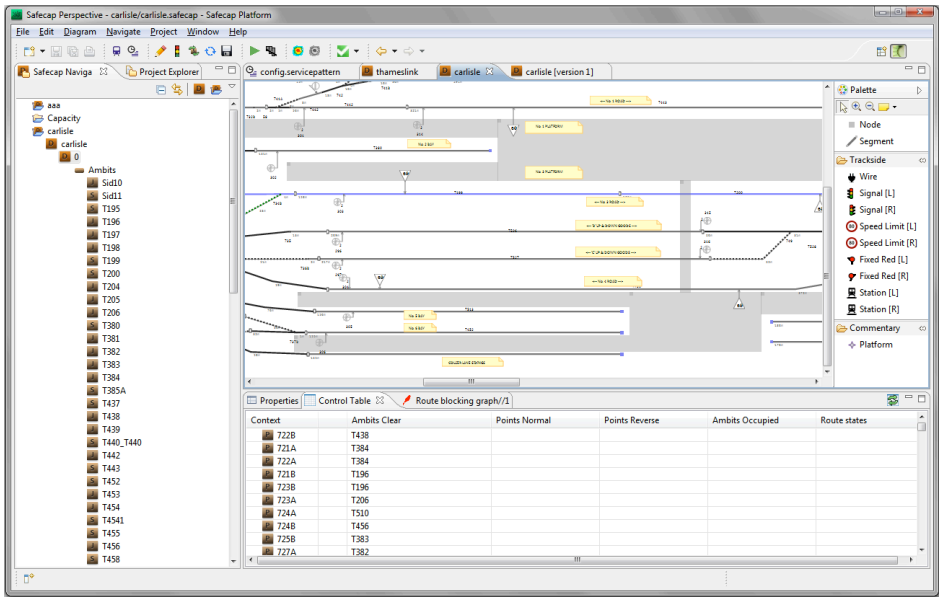

Figure 2. Model of Carlisle station

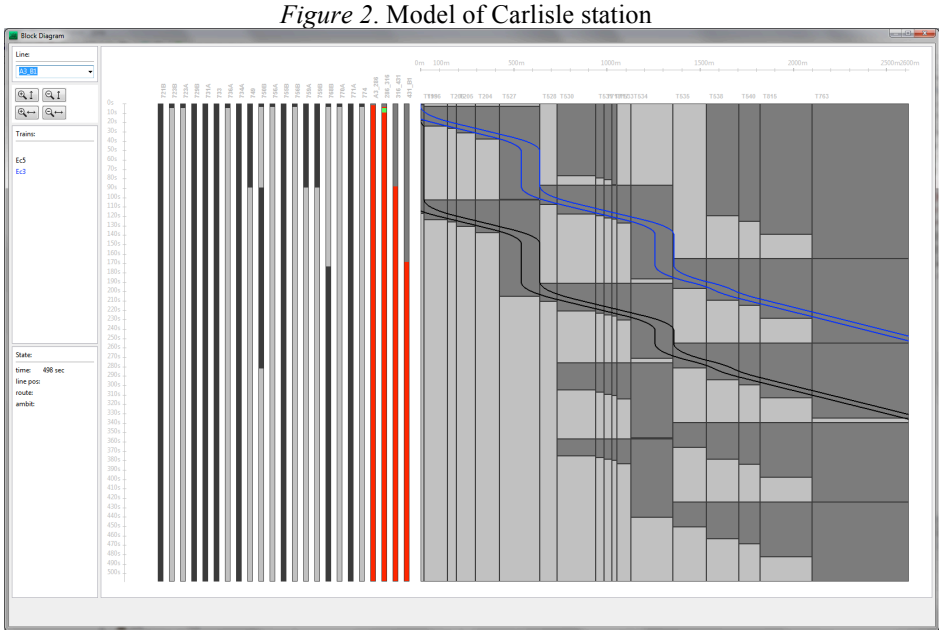

Figure 3. Blocking diagram

8 Conclusion and Future Work

The SafeCap layered architecture proved to be useful for efficient railway modelling using formal techniques developed by computer scientists. We are now training a

group of engineers to get their feedback, understand how this tooling fits into their current practice and capture their best practice in a set of capacity-improving patterns.

The open nature of the platform will allow us to substantially extend its functionality in the near future. We are already working on adding plug-ins for reasoning about energy and cost. This will be an important step in making the SafeCap approach practical, as capacity improvements should be always evaluated against their cost and energy implications. Another potential extension is to integrate tool support for reasoning about line capacity in Timed CSP [4] to allow us to add extra capabilities of capacity measuring using timed-based formal reasoning.

One of the advantages of applying formal modelling in the railway domain is that it makes it equally easy to verify the existing operational principles and novel, untested ideas. The level of confidence a formal approach brings is especially valuable in overcoming the healthy scepticism towards novelty in the field known for its conservatism. Thus, it perhaps makes sense to make a step further and claim that a set of uniform operational principles demanding the same signalling practice homogeneously deployed across a network is no longer a relevant approach at the age when most railway aspects are controlled by a computer. In-cab signalling and radio-based train position detection, like that of ETCS Level 3, can be used to implement a dynamically reconfigurable signalling logic that adapts, nearly instantaneously, to real-time capacity demands and emerging traffic patterns. Adapting our layered approach to deal with the task of dynamically generating formalisation layers will be an exciting technological challenge.

More information, including demos, examples, videos and documentation, can be found on the platform site (safecap.sourceforge.net). The tool is developed on SourceForge and publicly distributed. A users' and developers' community will be created to ensure the platform quality and to improve its applicability.

Acknowledgement. We are grateful to Simon Chadwick and Dominic Taylor for their comments on the earlier drafts of the paper.

9 References

1. P. Behm, P. Benoit, A. Faivre, J.M. Meynadier. Meteor: A successful application of B in a large project. Proc. Formal Methods 1999, LNCS-1708, 369–387. 1999.
2. A.E. Haxthausen, J. Peleska, S. Kinder. A formal approach for the construction and verification of railway control systems. Formal aspects of computing. 23, 191-219. 2011.
3. A. Svendsen, B. Moller-Pedersen, O. Haugen, J. Endresen, E. Carlson. Formalizing train control language: automating analysis of train stations. WIT TBE, 114. 2010.
4. Y. Isobe, F. Moller, H.N. Nguyen, M. Roggenbach. Safety and Line Capacity in Railways – An Approach in Timed CSP. Proc. IFM 2012, LNCS-7321, 54–68, 2012.
5. The SafeCap project: Overcoming the railway capacity challenges without undermining rail network safety. http://safecap.cs.ncl.ac.uk
6. A. Iliasov, A. Romanovsky. SafeCap domain language for reasoning about safety and capacity. Workshop on Dependable Transportation Systems. Niigata, Japan. IEEE CS. 2012.
7. Epsilon. http://www.eclipse.org/epsilon/
8. The Yices SMT solver: *yices.csl.sri.com*
9. The ProB model checker: www.stups.uni-duesseldorf.de/ProB