S3 Benchmark
"Brake risk assessment for ETCS train platoons"

AVACS S3*

[1] Albert-Ludwigs-Universität Freiburg, Fahnenbergplatz, 79085 Freiburg, Germany
[2] Carl von Ossietzky Universität Oldenburg, 26111 Oldenburg, Germany
[3] Universität des Saarlandes, 66041 Saarbrücken, Germany

Abstract. The upcoming *European Train Control System* (ETCS) standard level 3 allows high-speed trains to follow each other at close distances. To achieve this, the trains continuously request position dependent *Movement Authorities* (MAs) via GSM-R based wireless communication from track side *Radio Block Centers* (RBCs). The grant of an MA is decisive for moving on. We present a STATEMATE model which is used to investigate the risk of breaking maneuvers in train platoons that may be caused by an error prone wireless communication infrastructure, namely delays in the communication of MAs.

1 General Description

ETCS and GSM-R (Global System for Mobile communications - Railway), an adaptation of GSM wireless protocol, are designed to replace the multitude of incompatible (safety) systems used by European Railways and enable dense, fast transnational railway service. Different ETCS application levels are defined to meet the requirements of particular routes.
Central element in level 3 is the "moving block principle". Each train continuously receives (position dependent) MAs from the radio block center. Thus, the distance control does no longer rest upon the grant of an MA for one statically partitioned track section but becomes floating by addressing a "moving block". This allows train headway control to come close to an operation mode of braking distance spacing.
In the described case study, we investigate the risk of (unnecessary) braking maneuvers in a platoon of trains, that is, of trains that follow each other at small distances. The GSM-R based MA communication is considered error-prone. Failures in the GSM-R cause (stochastic) delays in sending and receiving of messages which may result in braking maneuvers of the trains in the platoon.
At the current stage, the purpose of the described model is to study and demonstrate the strength and limitations of the *S3 tool chain* [2] rather than providing new insight into the case. In particular we deviate from the concrete ETCS specification and set the focus on the STATEMATE designs scalability by varying the number of trains within a platoon.

* `http://www.avacs.org`

2 The S3 Tool Chain and its Formalisms

In the following, we will briefly sketch the S3 tool chain, in order to (i) provide an idea of the intermediate models derived from the top-level STATEMATE design and to (ii) embed the model in its verification context, that is, to show, which kind of properties we are analyzing.
For a detailed description of the S3 tool chain and the modeling formalisms being used, we have to refer the reader to [2], where we also published the described case-study. A slight variant of the study was presented in [3]. The established tool chain allows to determine *timed reachability properties* of a STATEMATE design based uniform continuous-time Markov decision process (uCTMDP). Thus, the chain enables to analyze properties such as:

"The probability to enter a safety critical system state within a mission time of 3 hours is at most 10^{-6}*."*

The inputs to the timed reachability analysis are (i) a STATEMATE design, (ii) a safety requirement that determines a set of safety critical system states and (iii) a set of Statechart transitions we will also refer to as *failure modes* in the following (if they represent failure behavior). The set of Statechart transitions serves later as synchronization points for the fourth input: (iv) stochastic delays. These are derived from (failure) probability distributions.
A rough overview of the tool chain, its inputs and intermediate modeling formalisms is depicted in Fig. 1.

3 Statemate Model

An overview of the system architecture is depicted in Fig. 2: 3 trains are moving on the track. Each of them communicates with the RBC, which is divided into 3 local handlers. Each handler is responsible for the communication with one dedicated train.

For our case, we assumed the RBC to operate as follows: It receives the current position of each moving train. To authorize a train to move on, it sends an authorization message. The idea is that the RBC only sends a moving authorization once it has received the position from the preceding train. Since a train is only allowed to send its new position if it is moving, each train can only move if the previous trains did already receive a "move" before. A special case has to be observed for the leading train, since there is no predecessor train the moving authorization for this train is always valid.

3.1 Statemate Description and Failure Modes

Initially, the RBC is idle (state `IDLE`). Upon receiving a position information from the train in front, i.e., event `MOVE_FROM_PRED`, it tries to transmit a moving authorization. Depending on the environmental circumstances, this either fails

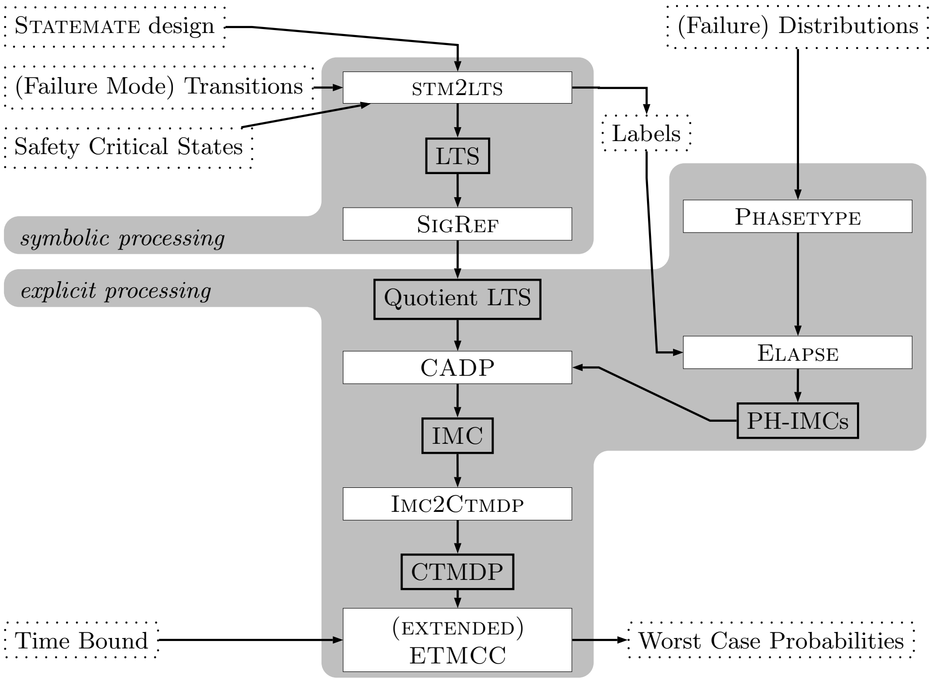

Fig. 1. S3 Tool Chain - Main Processing Steps

or succeeds (conditions `TRANS_FAILS` or `TRANS_SUCCEEDS`). The moving authorization will be submitted as an event (`MOVE`) to the parallel state which represents the train. If a train successfully transmits its position report to the RBC, an affirmative signal (`MOVE_TO_NEXT`) is sent to the next train.

In Fig. 3 and 4, some actions are prefixed with `E.wait` and some are not. All prefixed actions denote delayed actions. They are preserved during minimization and will later be associated with phase-type distributions. In particular, two types of errors can affect the communication between the RBC and the train. The occurrence of `ERROR_STARTS` indicates errors in the transmitted date. The condition `CONN_LOSS_STARTS`, on the other hand, signals a connection loss. At the end of error and connection loss, the conditions `ERROR_ENDS` and `CONN_LOSS_ENDS`, respectively, are set.

The train consists of two parallel activities, which are modeled in Statemate by an AND-node (see Fig. 4). The lower node controls the movement of the train. Upon getting a `MOVE` event from the RBC, the train is in the `MOVING` state until the `BRAKE` condition is set. The train then waits in the `BRAKING` state until a new moving authorization arrives. The upper node controls the position reports. If the lower node is in state `MOVING`, a new position is reported (via the `POSITION` event). Afterwards, the train has to wait in the state `REPORT_SENT` for a new `REPORT` event, which indicates, that all necessary information for a new report

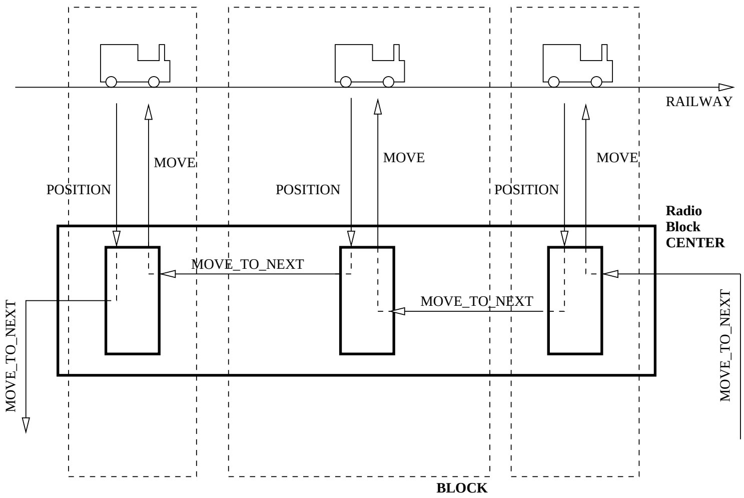

Fig. 2. Architecture

has been collected. It then changes to the `REPORT_READY` state, from which it can send a new position report (provided that it is in the `MOVING` state).

3.2 Safety Requirements

We consider all system states as *unsafe*, where the system occupies the node `BRAKING`.

3.3 Failure Mode Distributions

The failure mode distributions used are taken from [4], interpreted for multiple trains. Some of the delays associated with the failure modes are distributed according to exponential distributions, others are given by deterministic distributions. Deterministic distributions are best approximated by Erlang distributions with appropriate number of phases [1]. An Erlang distribution consists of several exponential distributions of the same rate arranged in series. The number of the exponential distributions are the phases of the Erlang distribution. The deterministic distributions in the model are approximated directly by Erlang distributions with n stages. We made some experiments to understand the sensitivity of the numerical results and of the state space sizes on different values of n.

The delay of `TRANS_SUCCEEDS`, indicating the delay to establish a GSM-R connection, is at most 5 seconds with 95 % and at most 7.5 seconds with 99.9 % probability. We approximated this delay by our prototype tool. Fig. 5 depicts the absorbing Markov chain obtained from the approximation. To simplify the figure, the chain is not uniform, i.e., self-loops are omitted.

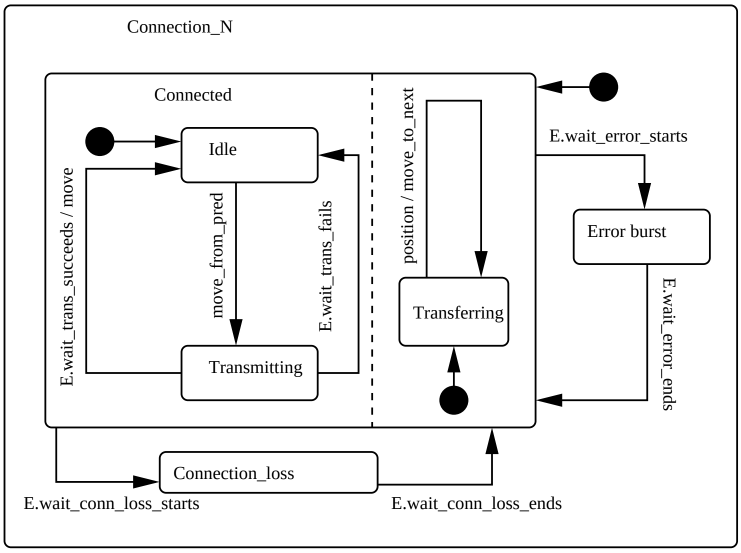

Fig. 3. Model of the Connection between the Train and the RBC

4 Verification Results

In this section, we give some statistics we obtained from experiments on the ETCS case study where we vary the number of consecutive trains. The delays of events `BRAKE` and `REPORT` are distributed by deterministic delays of 25 and 5 seconds, respectively. They were approximated by Erlang distributions. The different settings we use are determined by the number of phases (namely 1, 5 and 10) in the approximating Erlang distributions.

Table 1 gives an overview of the computation time and the model sizes for the symbolic part of our tool chain, as generated by Stm2Lts. We display the bit-vector sizes for states and transitions of the generated LTS, with and without cone-of-influence reduction that we apply to shrink the model to the analysis relevant behavior. The bit-vector size corresponds to a potential state space of the model, where a bit-vector size of x gives a potential of 2^x in the number of states. We also show the actual reachable state space, and the result of symbolic branching minimization, as generated by SigRef, as well as the overall computation time (in seconds) in the table.

In Table 2 and 3, we report results concerning the construction and minimization of the model. Experimental results are displayed for monolithic (Table 2) and compositional (Table 3) construction. For each type of construction, we report the size of the *largest intermediate state space* we needed to handle, the construction time (Generation) and the Minimization time in seconds. The state spaces of the final results are also provided. For the compositional approach, we report the accumulated time (G+M) over all steps.

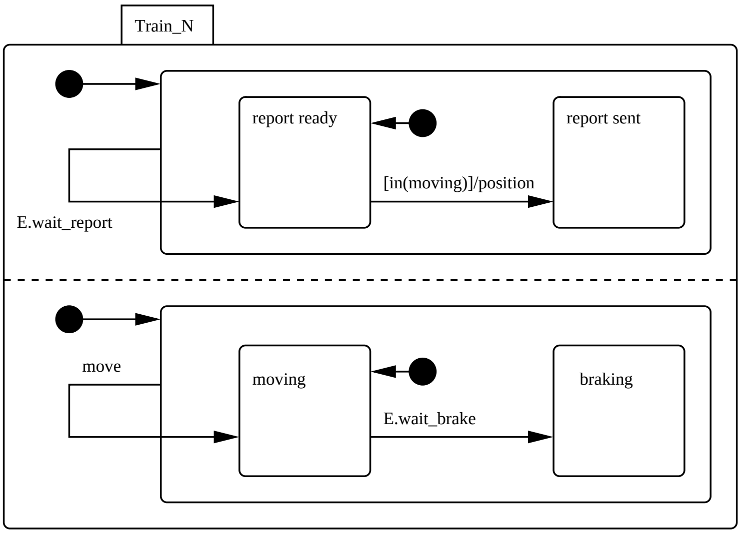

Fig. 4. Model of the Train Internals

The advantage of using compositional construction in terms of space and time is apparent. Stepwise minimization keeps the size of state spaces low. This, in turns, reduces the duration of the minimization time in the next step, and so on, thus saving significant amount of time.

Statistical results for the transformation from IMC to CTMDP are displayed in Table 4. We give the number of states and transitions for the quotient IMC and the resulting CTMDP, together with the computation time required for this transformation. The column depicting the number of CTMDP transitions deserves a special comment. Since transitions in CTMDPs are triples (s, l, R) with a function R assigning rates to successor states, representing one transition may in the worst case already require space in the order of the number of states. Of course, this is not the case, the functions are very sparse. The numbers denoted in brackets are the average number of nonzero entries per transition.

The runtime of the extended ETMCC model checker is shown in the last two columns of Table 4. The computation time needed to compute the worst case probability to reach the set of safety critical states has been computed for time bounds of 10 and 180 seconds, respectively. Since the timed reachability

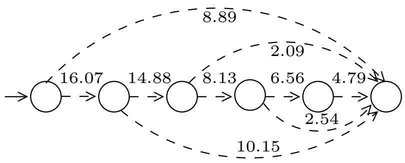

Fig. 5. Phase-type approximation of the delay of `TRANS_SUCCEEDS`

algorithm is implemented prototypically so far, we are actually quite satisfied with its performance.

Table 1. Symbolic Steps: Statemate Safety Analysis and Minimization Statistics

	Stm2Lts										SigRef		
Trains	Without COI					With COI					Branching Bisimulation		
	Potential		Reachable		Time	Potential		Reachable		Time	Min. Result		Time
	s bits	t bits	s	t	(sec.)	s bits	t bits	s	t	(sec.)	s	t	(sec.)
2	18	12	253	11132	6.9	16	12	121	5324	0.3	25	359	0.07
3	30	22	10585	3217840	30.2	28	22	5041	1532464	1.9	79	2065	1.16
4	42	32	444529	768146112	897.5	40	32	211681	365784768	6	79	2969	43.19
5	54	42	18670200	167284992000	18677	52	42	8890560	79659417600	6.1	79	4341	1150.94

Table 2. Monolithic Construction for ETCS with 2 Trains

Phases	Monolithic Construction			
	States	Transitions	G Time (sec.)	M Time (sec.)
1	33600	518464	12	3
5	302400	4142016	22	402
10	1016400	13521376	46	5154

Table 3. Explicit Steps: Composition and Minimization Statistics

Trains	Phases	Compositional Construction			Final Quotient IMC	
		States	Transitions	G + M Time (sec.)	States	Transitions
2	1	600	2505	42	355	1590
	5	10000	53625	61	5875	39500
	10	37500	207500	511	20000	154750
3	1	3240	16064	58	1375	5225
	5	64440	354100	813	36070	159119
	10	249480	1382900	10666	113650	533500
4	1	2870	11260	53	1435	5475
	5	57950	260350	420	30575	141000
	10	224900	1022700	7391	119650	558500

Table 4. Explicit Steps: CTMDP Transformation and Analysis Statistics

Trains	Phases	Quotient IMC		Uniform CTMDP		Time	Time for Analysis of Formula (sec.)	
		States	Transitions	States	Transitions	(sec.)	$\sup_D \Pr_D(s, \overset{\leq 10}{\rightsquigarrow} B)$	$\sup_D \Pr_D(s, \overset{\leq 180}{\rightsquigarrow} B)$
2	1	358	1593	227	352 (1.75)	3.39	0.06	0.44
	5	5878	39503	3127	3752 (4.60)	3.67	0.54	7.00
	10	22003	154753	11252	12502 (5.52)	4.70	2.23	31.15
3	1	1378	5228	787	1347 (1.10)	3.61	0.14	2.01
	5	36073	159113	21722	35942 (1.55)	4.99	6.24	89.39
	10	113653	533503	56452	90402 (1.84)	8.46	17.95	254.29
4	1	1438	5478	817	1457 (1.01)	3.53	0.16	2.28
	5	30578	141003	15477	26577 (1.57)	4.86	4.43	62.83
	10	119653	558453	59452	101402 (1.64)	8.40	19.94	280.88

References

1. J. Abate, G. L. Choudhury, and W. Whitt. Calculation of the GI/G/1 waiting-time distribution and its cumulants from pollaczek's formulas. *AEÜ Hirzel Verlag*, 47, 5/6:311–321, 1993.
2. Eckard Böde, Marc Herbstritt, Holger Hermanns, Sven Johr, Thomas Peikenkamp, Reza Pulungan, Ralf Wimmer, and Bernd Becker. Compositional performability evaluation for statemate. In *3rd International Conference on the Quantitative Evaluation of Systems, QEST 2006, Riverside (USA)*, pages 167–178. IEEE Computer Society, 2006.
3. Marc Herbstritt, Ralf Wimmer, Thomas Peikenkamp, Eckard Böde, Michael Adelaide, Sven Johr, Holger Hermanns, and Bernd Becker. Analysis of Large Safety-Critical Systems: A quantitative Approach. Reports of SFB/TR 14 AVACS 8, SFB/TR 14 AVACS, Feb 2006. ISSN: 1860-9821, http://www.avacs.org.
4. H. Hermanns, D.N. Jansen, and Y.S. Usenko. From StoCharts to MoDeST: a comparative reliability analysis of train radio communications. In *5th International Workshop on Software and Performance, WOSP 2005*, pages 13–23. ACM Press, 2005.