Model Checking Data-Dependent Real-Time Properties of the European Train Control System

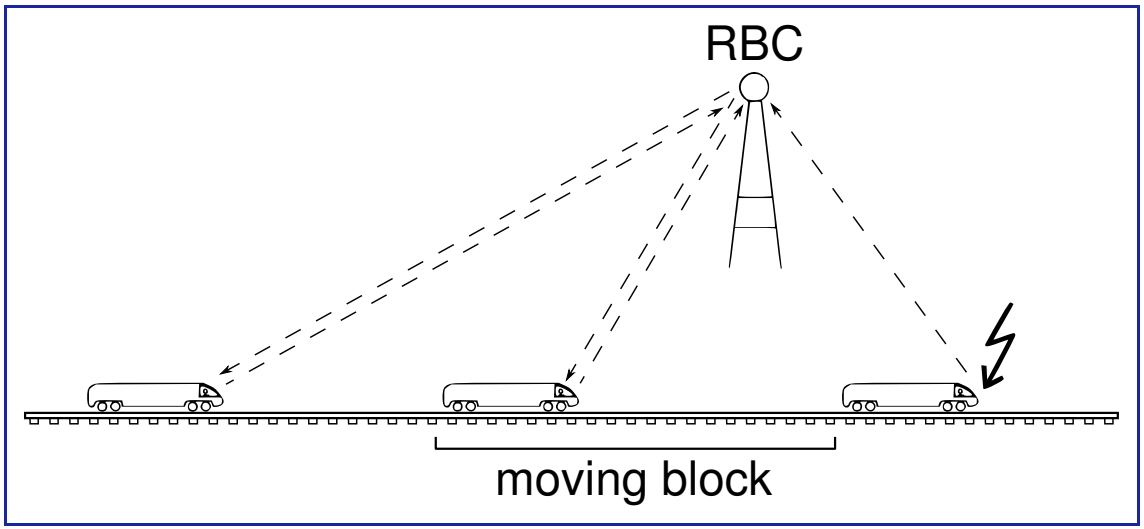
Johannes Faber
Department of Computing Science, University of Oldenburg, Germany
j.faber@uni-oldenburg.de

Joint work with Roland Meyer

Overview

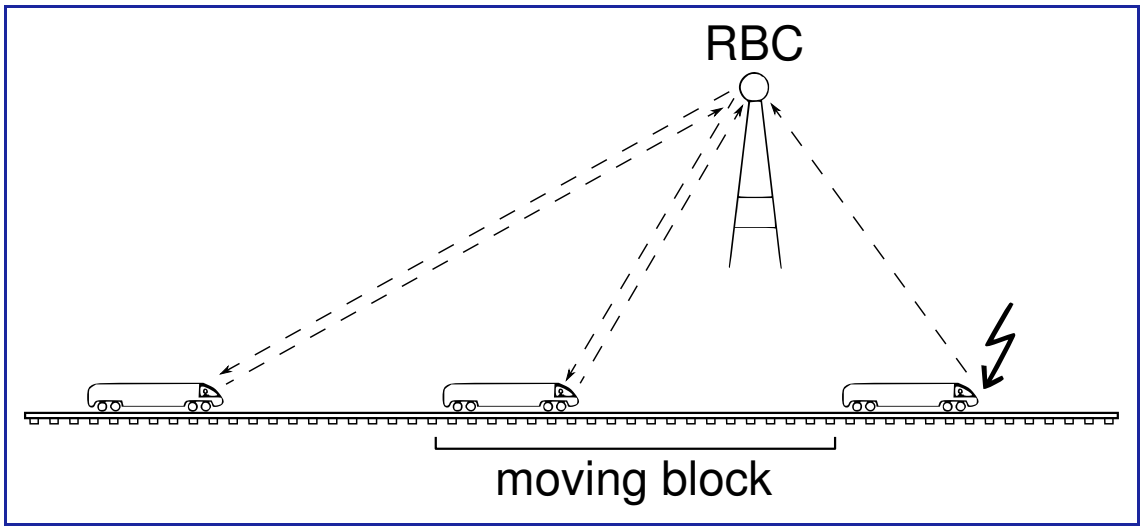
- Aim: automatic verification of complex high-level specifications with infinite state space
- Combined specification formalism (CSP-OZ-DC) integrating data, processes and time
- Real-time requirements: Duration Calculus

Case Study: Emergency Messages



- Considers treatment of emergency messages in ETCS
- Basic, potentially infinite data types ($\mathbb{R}, \mathbb{N}$)
- Events with parameters (infinite range)
- Guarantees reliability properties
  - Reaction time not exceeded
  - Trains never collide

Case Study: Emergency Messages (2)



- The case study comprises
  - Communication aspects
  - Data aspects
  - Real-time aspects

CSP-OZ-DC

The specification language CSP-OZ-DC combines

- Communicating Sequential Processes (CSP)

  $\rightsquigarrow$ *Hoare, Brookes, Roscoe*

Train:

$\texttt{main} \stackrel{c}{=} \mathit{Running} \parallel \mathit{HandleEM} \parallel \mathit{EmergencyDetection}$

CSP-OZ-DC

The specification language CSP-OZ-DC combines

- Communicating Sequential Processes (CSP)

  $\rightsquigarrow$ *Hoare, Brookes, Roscoe*

RBC:

$WarningTo(id : TrainID) \stackrel{c}{=} send.EmergencyWarning?id \rightarrow \ldots$

Train:

$HandleEM \stackrel{c}{=} receive.EmergencyWarning.ID \rightarrow \ldots$

CSP-OZ-DC

The specification language CSP-OZ-DC combines

- Communicating Sequential Processes (CSP)
- Object-Z (OZ)

  ⇝ *Spivey, Smith*

$$\begin{array}{|l} position : Position \\ sbi : Position \\ \ldots \\ \hline position \leq sbi \end{array}$$

com_*computeSBI*

$$\begin{array}{|l} \Delta(sbi) \\ eoa? : Position \\ \hline sbi' = \\ eoa? - ReleaseDist - StopDist \end{array}$$

CSP-OZ-DC

The specification language CSP-OZ-DC combines

- Communicating Sequential Processes (CSP)
- Object-Z (OZ)
- Duration Calculus (DC)

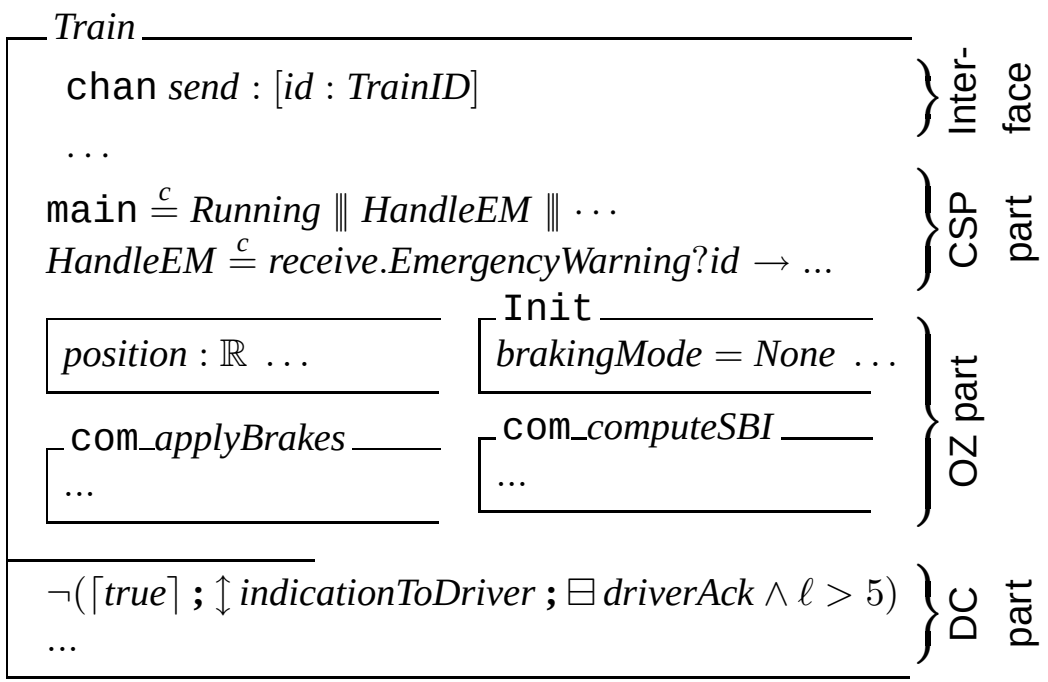
$\rightsquigarrow$ *Zhou, Hoare, Ravn, Hansen*

$$\neg\Diamond(\updownarrow \mathit{indicationToDriver} \,;\, \boxminus \mathit{driverAck} \wedge \ell > 5)$$

CSP-OZ-DC

The specification language CSP-OZ-DC combines

- Communicating Sequential Processes (CSP)
- Object-Z (OZ)
- Duration Calculus (DC)



Verification of CSP-OZ-DC

- CSP-OZ-DC is given an operational semantics in Phase Event Automata (PEA) [Hoenicke 2006]

Verification of CSP-OZ-DC

- CSP-OZ-DC is given an operational semantics in Phase Event Automata (PEA) [Hoenicke 2006]
  - Timed automata variant
  - Synchronize on data, events and time
  - Compositional

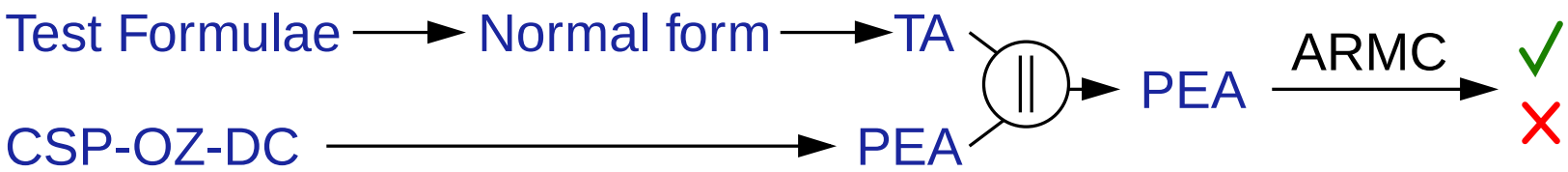
Verification of CSP-OZ-DC

- CSP-OZ-DC is given an operational semantics in Phase Event Automata (PEA) [Hoenicke 2006]
  - Timed automata variant
  - Synchronize on data, events and time
  - Compositional

$$\mathcal{A}(\textit{CSP-OZ-DC}) = \mathcal{A}(\textit{CSP}) \parallel \mathcal{A}(\textit{OZ}) \parallel \mathcal{A}(\textit{DC})$$

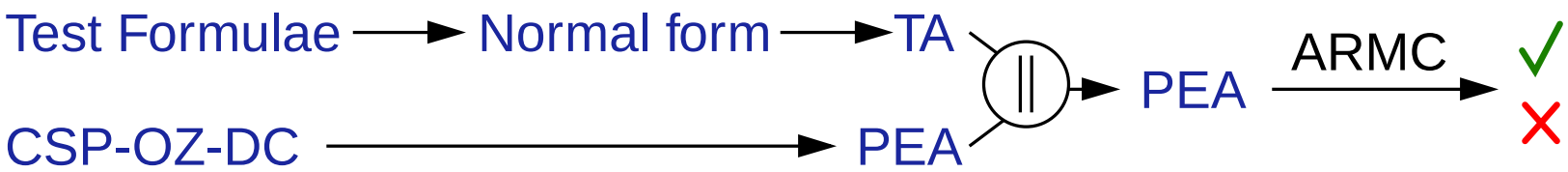
$$\mathcal{A}_1 \models \phi \quad \Rightarrow \quad \mathcal{A}_1 \parallel \mathcal{A}_2 \models \phi$$

Verification of CSP-OZ-DC (2)



- Aim: Verification of $\mathcal{A} \models \phi$
  - Translation of DC *Test Formulae* into PEA [Meyer, Faber, Rybalchenko 2006]

Verification of CSP-OZ-DC (2)



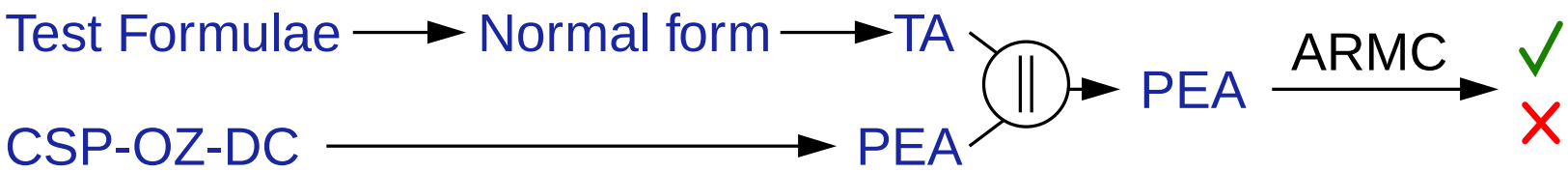
- Aim: Verification of $\mathcal{A} \models \phi$
  - Translation of DC *Test Formulae* into PEA [Meyer, Faber, Rybalchenko 2006]

DC Test Formulae - Example

$$\neg\Diamond(\updownarrow \mathit{detectEmergency} \,;\, \boxminus \mathit{applyEB} \wedge \boxminus \mathit{driverAck} \wedge \ell > 8)$$

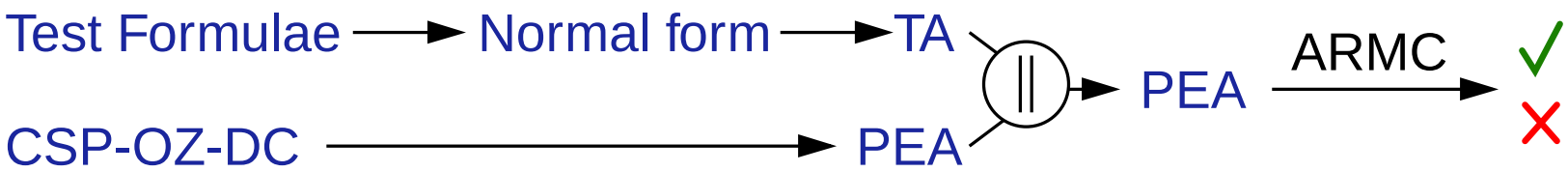
$$(\mathit{true} \,;\, \lceil \mathit{Track.position\_1} > \mathit{Track.position\_0} - \mathit{Length} \rceil \,;\, \mathit{true}) \wedge \boxminus \mathit{driverAck}$$

Verification of CSP-OZ-DC (2)



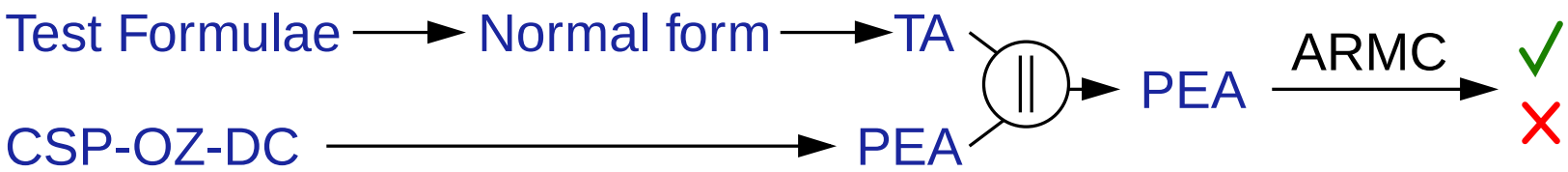
- Aim: Verification of $\mathcal{A} \models \phi$
  - Translation of DC *Test Formulae* into PEA [Meyer, Faber, Rybalchenko 2006]
  - Decomposition of DC properties by means of a normal form

Verification of CSP-OZ-DC (2)



- Aim: Verification of $\mathcal{A} \models \phi$
  - Translation of DC *Test Formulae* into PEA [Meyer, Faber, Rybalchenko 2006]
  - Decomposition of DC properties by means of a normal form
  - Check for reachability of "bad" states in $\mathcal{A} \parallel \mathcal{A}(\phi)$

Verification of CSP-OZ-DC (2)



- Aim: Verification of $\mathcal{A} \models \phi$
  - Translation of DC *Test Formulae* into PEA [Meyer, Faber, Rybalchenko 2006]
  - Decomposition of DC properties by means of a normal form
  - Check for reachability of "bad" states in $\mathcal{A} \parallel \mathcal{A}(\phi)$
  - Representation of PEA in Transition Constraint Systems (TCS)
  - Abstraction refinement model checking: ARMC [Rybalchenko 2002]

Results

Task	Locs	Trans	Vars	Refs	TA	ARMC
Delivery	122	>18000	20	32	50s	86m
Delivery (decomp. 1)	14	366	14	8	2.7s	13.9s
Delivery (decomp. 2)	17	173	10	17	2.2s	1.9s
Delivery (decomp. 3)	12	71	9	9	1.9s	0.7s
Delivery (decomp. 4)	17	156	12	17	2.2s	2.6s
Delivery (decomp. 5)	7	28	4	3	1.6s	0.1s

Results

Task	Locs	Trans	Vars	Refs	TA	ARMC
Delivery	122	**>18000**	20	32	50s	**86m**
Delivery (decomp. 1)	14	366	14	8	2.7s	13.9s
Delivery (decomp. 2)	17	173	10	17	2.2s	1.9s
Delivery (decomp. 3)	12	71	9	9	1.9s	0.7s
Delivery (decomp. 4)	17	156	12	17	2.2s	2.6s
Delivery (decomp. 5)	7	28	4	3	1.6s	0.1s

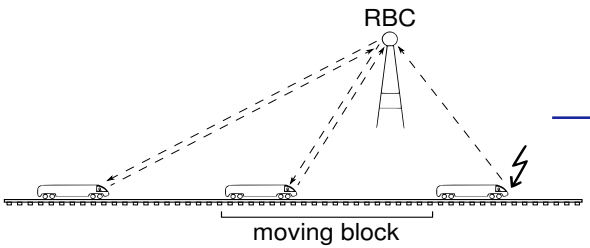
Results

Task	Locs	Trans	Vars	Refs	TA	ARMC
Delivery	122	>18000	20	32	50s	86m
Delivery (decomp. 1)	14	366	14	8	2.7s	**13.9s**
Delivery (decomp. 2)	17	173	10	17	2.2s	**1.9s**
Delivery (decomp. 3)	12	71	9	9	1.9s	**0.7s**
Delivery (decomp. 4)	17	156	12	17	2.2s	**2.6s**
Delivery (decomp. 5)	7	28	4	3	1.6s	**0.1s**

Results (2)

- First large-scale application example for the declarative, high-level language CSP-OZ-DC involving
  - Infinite data
  - Real-time
  - Communication aspects (infinite parameter types)
  - Object-orientation
- Focusing on message passing and infinite data types
- State explosion problem
  ⇝ Decomposition for CSP-OZ-DC/PEA/DC needed

Ongoing/Future Work



Several variants

CSP-OZ-DC

- Slicing
- Task analysis

PEA

- FTA
- Decomposition

TCS

- Data types
- ARMC

References

- J. Hoenicke

  Combination of Processes, Data, and Time

  PhD thesis, University of Oldenburg, 2006.

- J. Hoenicke and P. Maier

  Model-checking of Specifications Integrating Processes, Data and Time

  In Proc. Formal Methods (FM), Eds. J.S. Fitzgerald, I.J. Hayes, and A. Tarlecki, LNCS 3583, 2005.

- S. Jacobs and V. Sofronie-Stokkermans

  Applications of Hierarchical Reasoning in the Verification of Complex Systems

  In Proc. Fourth Intl. Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR), 2006.

References (2)

- R. Meyer, J. Faber and A. Rybalchenko

  Model Checking Duration Calculus: a Practical Approach

  In Proc. Intl. Coll. on Theoretical Aspects of Computing (ICTAC), Eds. K. Barkaoui, A. Cavalcanti, and A. Cerone, LNCS, 2006. To appear.

- I. Brückner and H. Wehrheim

  Slicing an Integrated Formal Method for Verification

  In Proc. Seventh Intl. Conference on Formal Engineering Methods (ICFEM), LNCS 3785, 2005.

References (3)

- B. Cook, A. Podelski and A. Rybalchenko

  Terminator: Beyond safety (tool description)

  In Proc. Conf. on Computer Aided Verification, LNCS, 2006. To appear.

- A. Rybalchenko

  A Model Checker Based on Abstraction Refinement

  MSc thesis, University of Saarbrücken, 2002.

- ARMC

  `http://www.mpi-inf.mpg.de/~rybal/armc`.

- Verification tools for PEA

  `http://csd.informatik.uni-oldenburg.de/projects/epea.html`.
  `http://csd.informatik.uni-oldenburg.de/~moby/`.