Universität Rostock
Traditio et Innovatio

Institut für Angewandte Mikroelektronik und Datentechnik

TWT GmbH
Science & Innovation

Executable System Level Models with SystemC

A Joint Approach of the University of Rostock and TWT

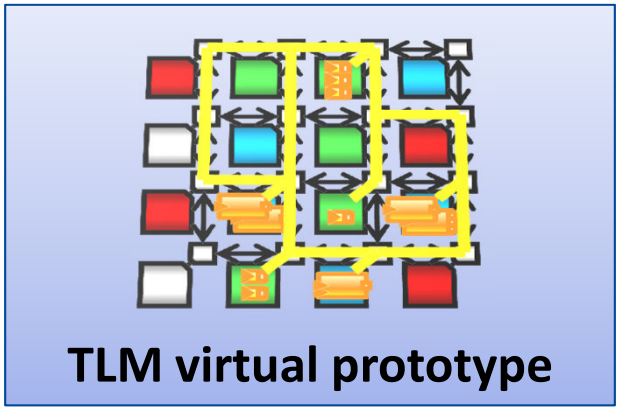
Frank Golatowski, Benjamin Beichler, Christian Haubelt (Uni Rostock)

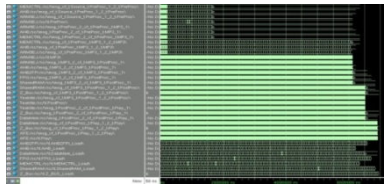
Stefan Rieger (TWT)

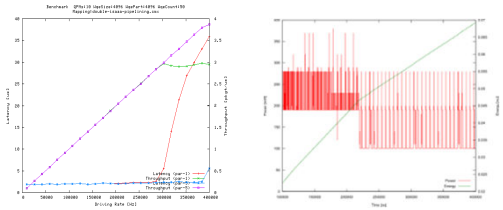
15/04/2013

What is SystemC?

- Open Source C++ Class Library
- Event-driven simulation interface
- Concurrent, communicating processes
- System-level (transaction level) as well as hard- and software modeling







What is SystemC ?

- SystemC is a modelling language initially developed to build hardware
- Due to the power of the language it has been further developed as a language to model the behavior of a system independent of implementation
- At an very early stage it is possible by means of this language to evaluate the system
  - Specification / Timing
  - Hardware (single core, multi core, quad core)
  - Software
  - Type of onboard unit (OBU)

SystemC Modelling Approach for Tool Evaluation

- Subset 026 → Subset of the Subset (D2.5)
- Braking curves (Uni Rostock)
- Communication Management (TWT)

- Using SystemC in a way to obtain a formal model
- TLM Transaction level modelling (High level)

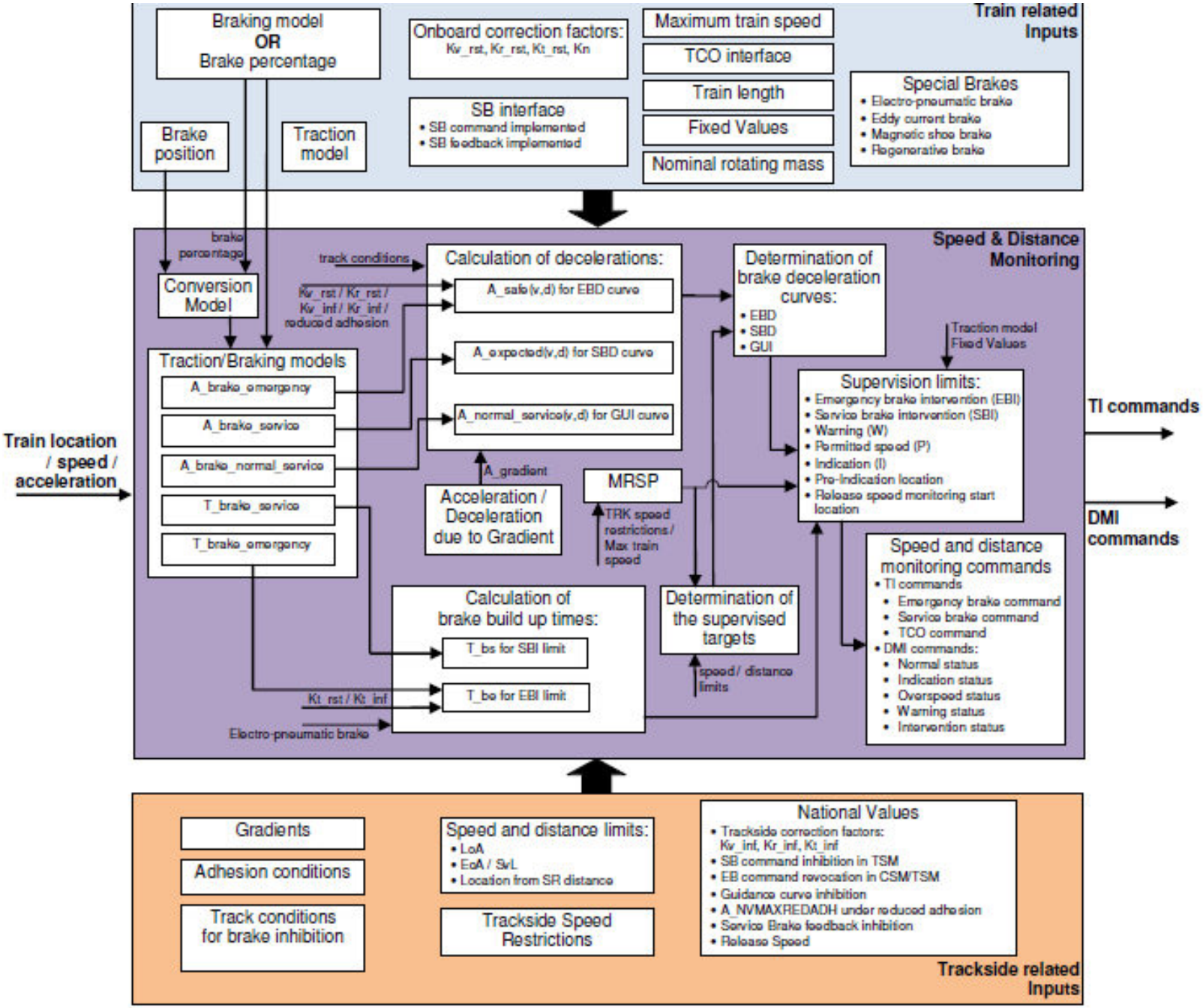
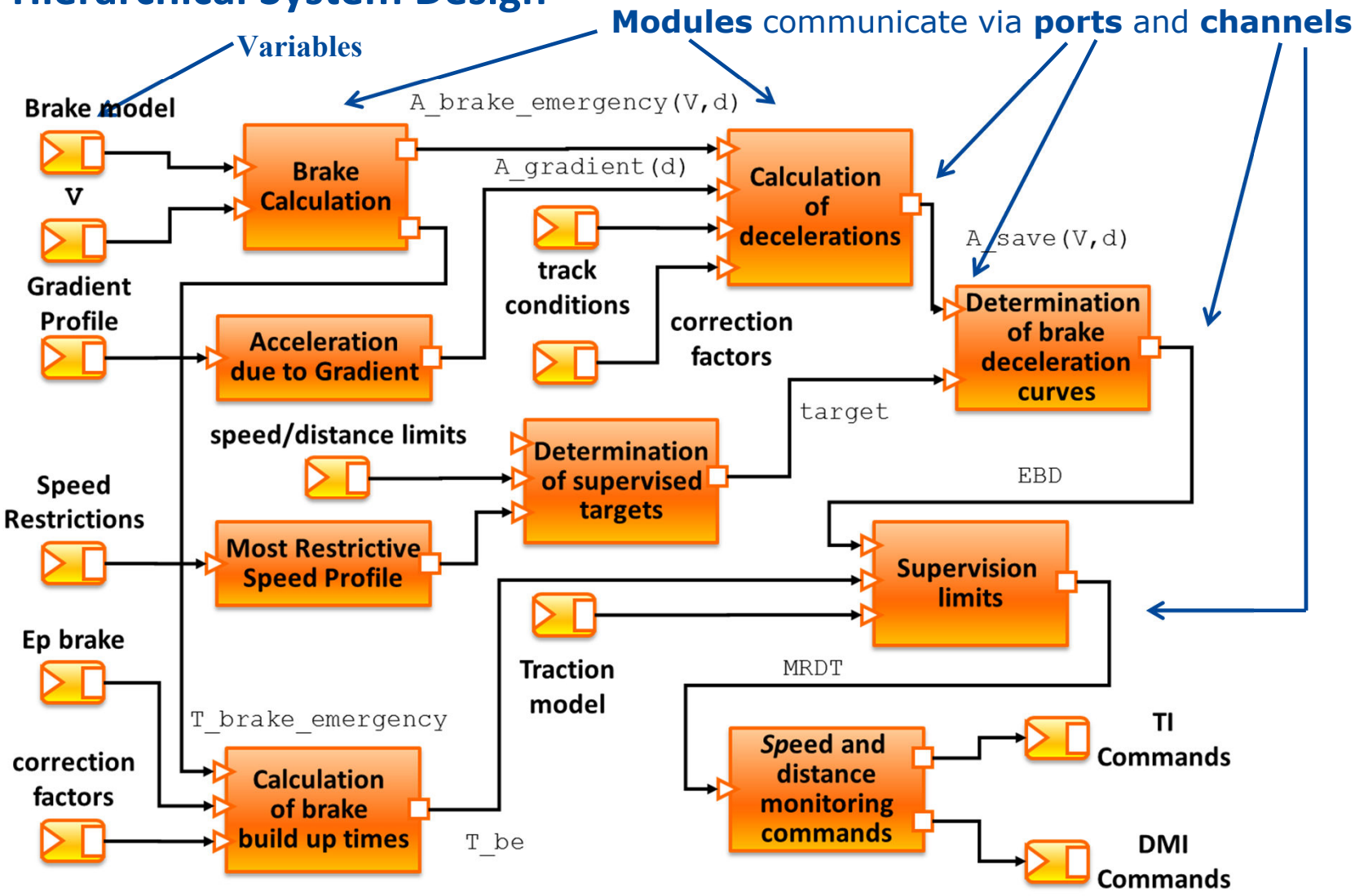


Figure 28: Speed and distance monitoring overview

Hierarchical System Design



Prototype Models in WP7

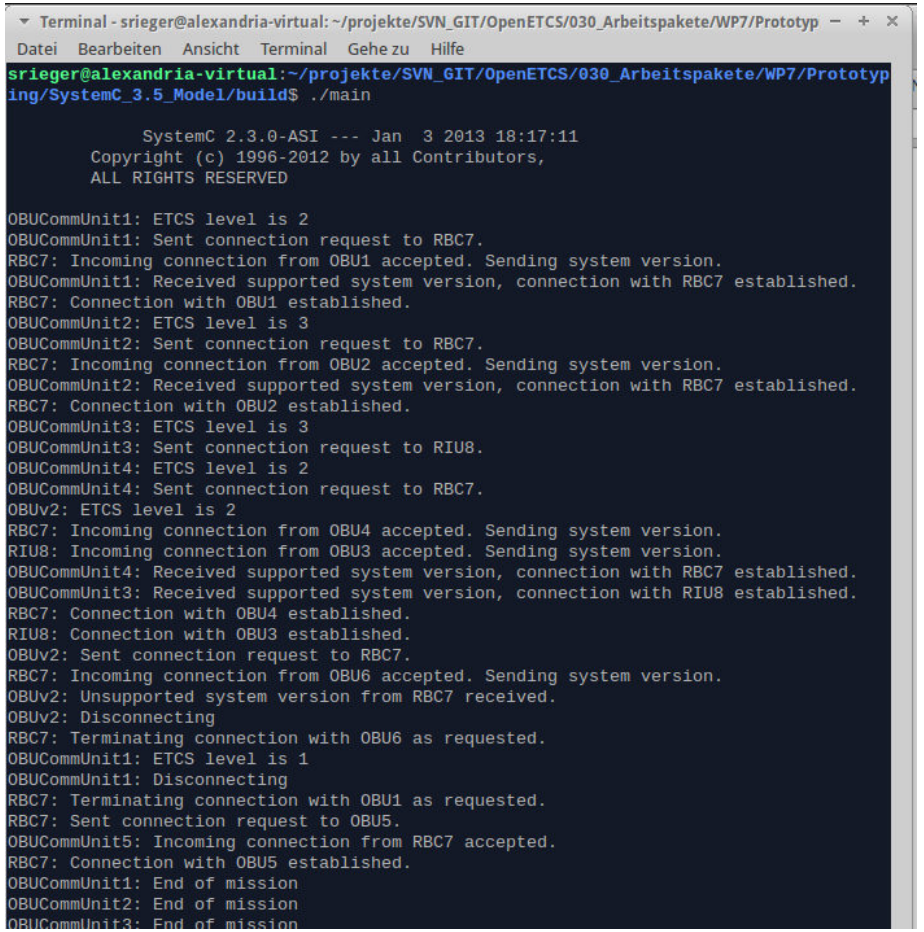
- **University of Rostock: Braking Curves**
  - **Current state: Design of the Model of braking curves**
  - **In discussion interfaces with Scade and FormalSpec**

- **TWT: Management of Communication**
  - **Current state: Model of communication setup**
  - **Arbitrary many OBUs (Trains), RBCs or RIUs**
  - **Concurrent communication setup**

Prototype for Management of Communication (by TWT)

Current state of the model:

- Model of communication setup
- Arbitrarily many OBUs (Trains), RBCs or RIUs
- Concurrent communication setup
- Switching of ETCS levels
- Still many aspects missing



Simulation run of the model

Pros

- Open Source framework & IEEE Standard 1666
- Integrated simulation kernel -> executable models
- Integrated model of time
- Concurrency
- Powerful tracing & early performance estimation
- Reusability & Maintainability
- Full power of C++ (if necessary)
- Covers multiple abstraction levels in system design

Cons

- Control-flow can get complex (due to concurrency)
- Not fully formal (currently there is no formally defined semantics)

Thank you for your attention!

QUESTIONS?