fortiss
innovation in software and systems

Reusing Test Cases on Different Levels of Abstraction in a Model Based Development Tool

25.03.2012, Tallinn, Estonia

Jan Olaf Blech Dongyue Mou Daniel Ratiu

fortiss GmbH
An-Institut der Technischen Universität München

Motivation

- Manufacturers
  - Provide an abstract specification
  - With **abstract test-cases**
- Sub-contractors
  - Responsible for development
  - Refine and implement the specification

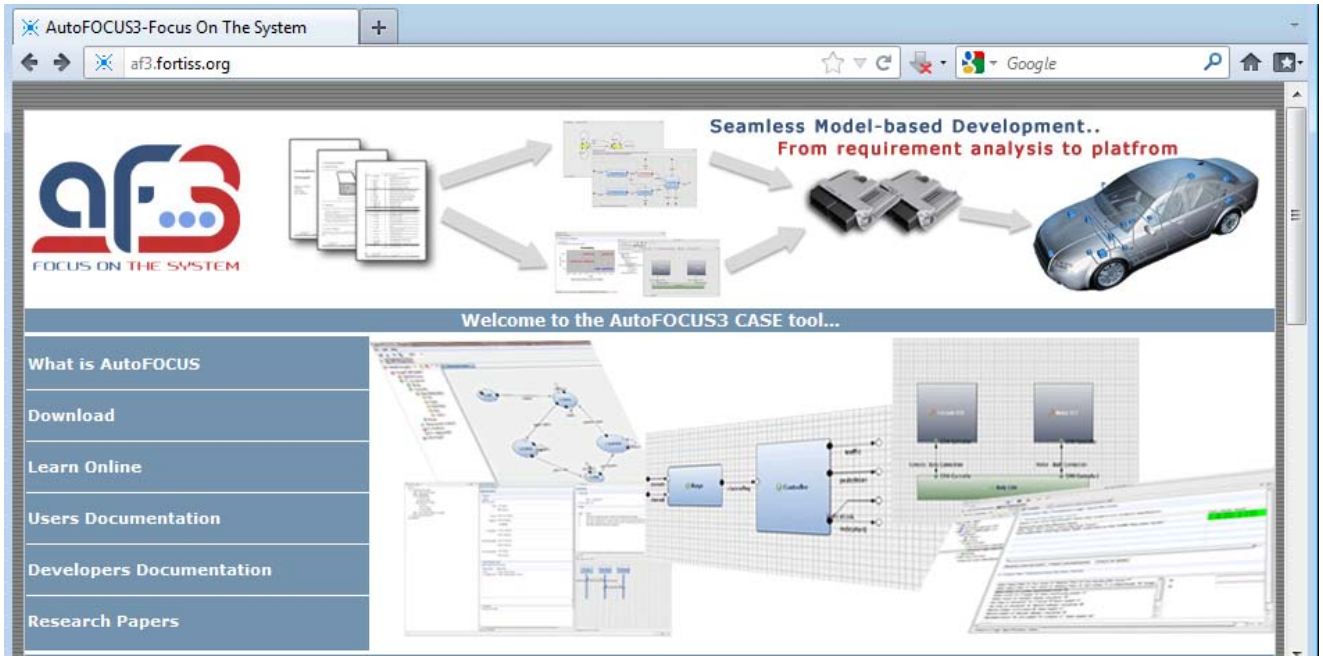
How to use abstract test-cases in a concrete context?

Modeling Environment

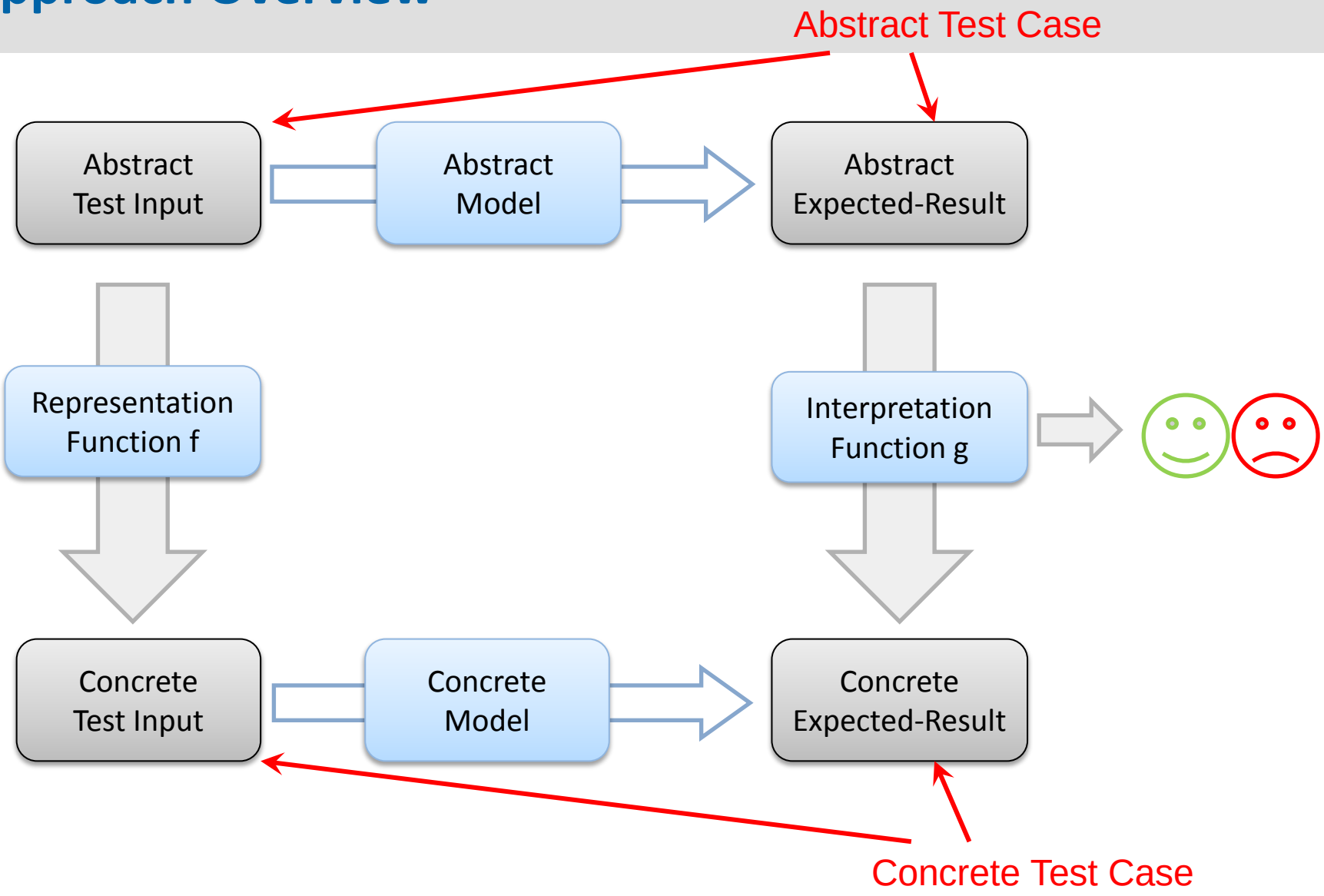
- AutoFocus
  - Seamless model-based development tool
  - Using stream processing function concept
  - Binding formal techniques together
    - Modular Component
    - State machines
    - Mode automaton
    - Temporal logics
    - Requirements analysis
    - Test suite generation
    - ……

AutoFocus 3

- Web Site
  - http://af3.fortiss.org
  - Current Version: AutoFocus Phoenix 2.1-RC2 (15.03.2012)



Approach Overview



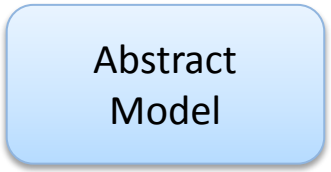
Differences from Abstract to Concrete Model

- Reduced behavior space
  - Abstract model contains more execution paths
  - Concrete model may only support a subset of all paths

- Increased complexity
  - Abstract state is normally very simple
    - » Boolean value, Abstract token, ...
  - Concrete state is much more complex
    - » 32bit integer, list type, …

Abstract Model

- Focus on limited logical behavior
- No implementation details



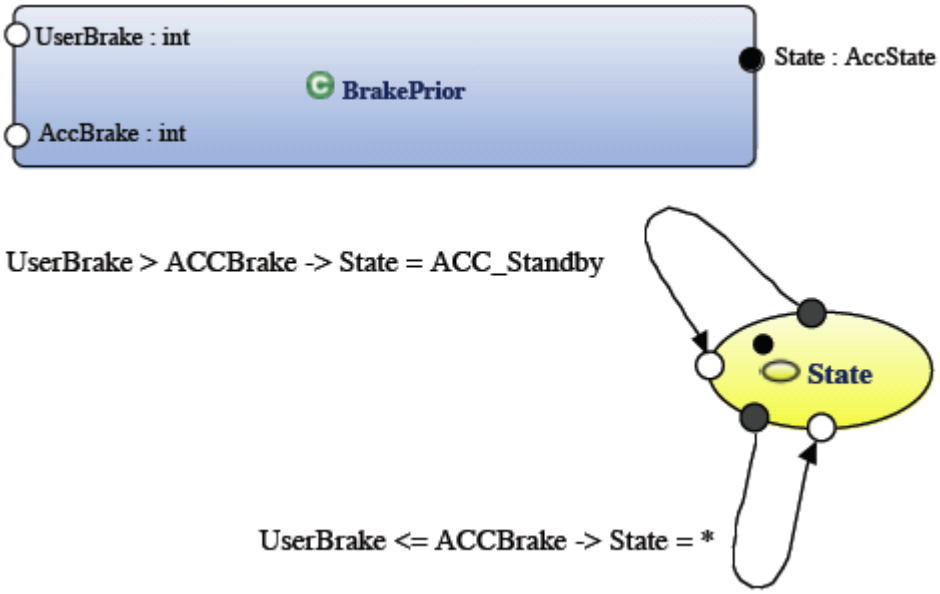
- Advantages
  - Easy to model
  - Easy to validate
  - Easy to understand

Example – Abstract Model

- Text Requirement
  - *Braking by the driver shall deactivate the ACC function at least if the driver initiated brake force demand is higher than the ACC initiated brake force. (ISO 15622)*

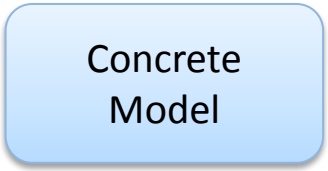
Example – Abstract Model (Cont.)

- Formal Requirement



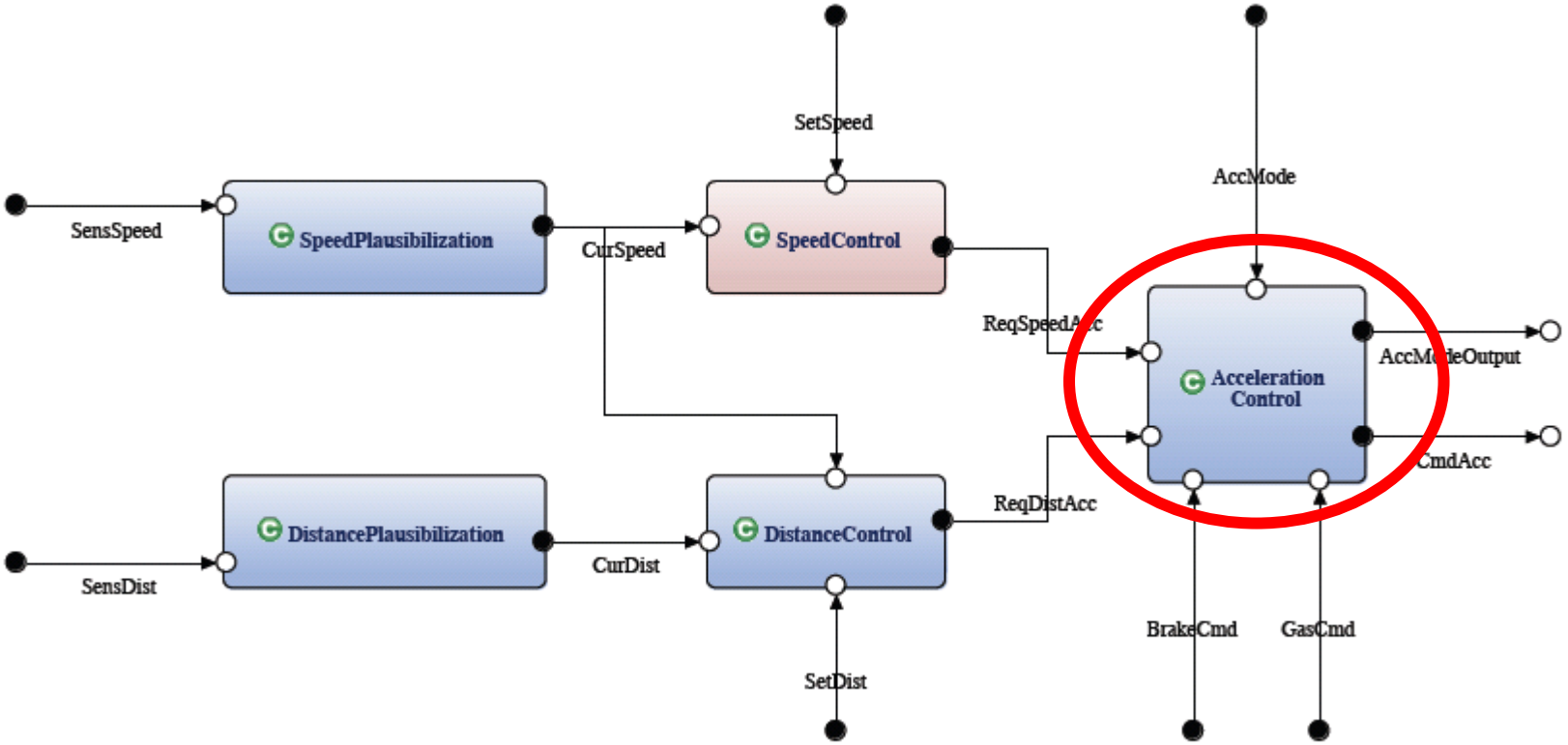
Concrete Model

- Implementation
  - Refines the specification
  - Under multiple constraints
  - Adds substantial complexity



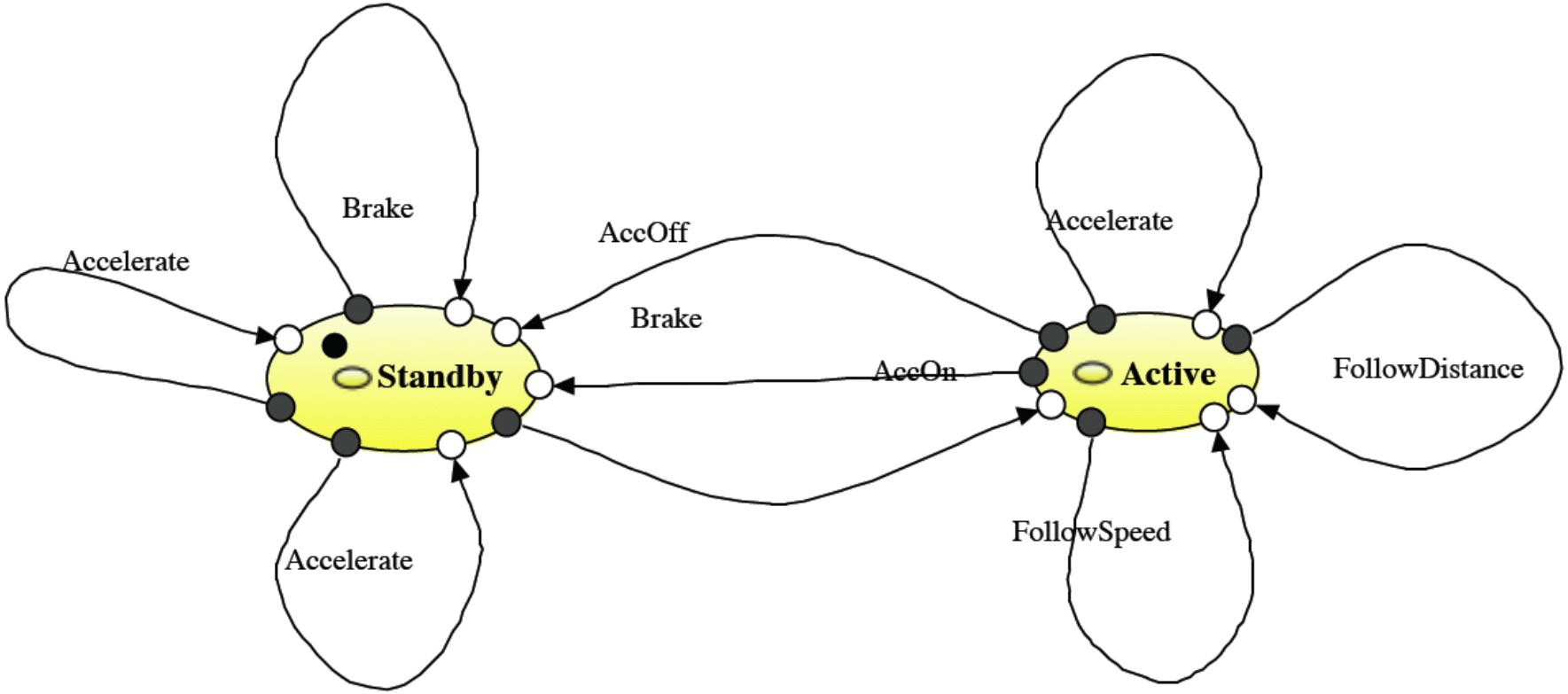
Example – Concrete Model

- ACC System Model

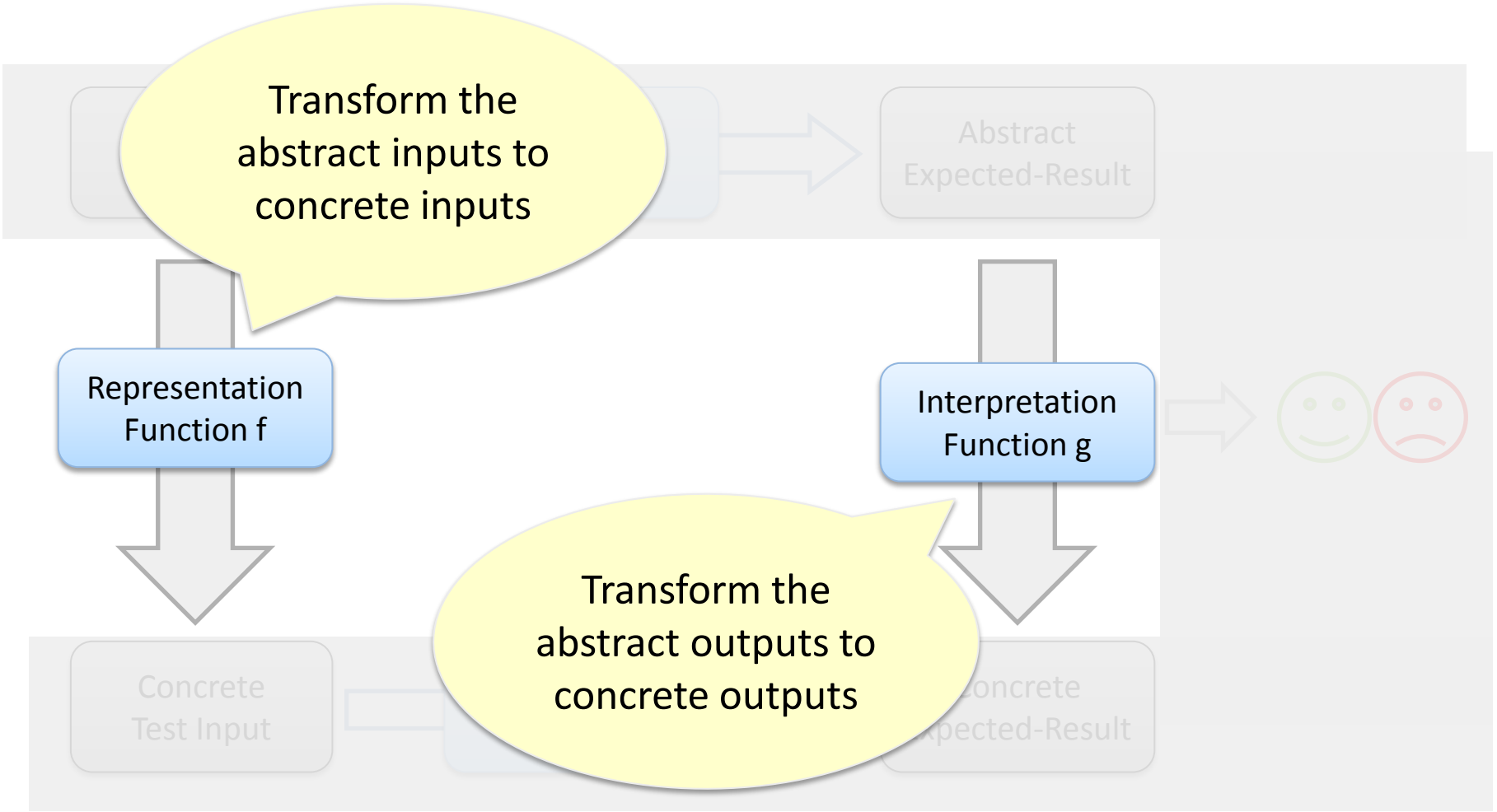


Example – Concrete Model (2)

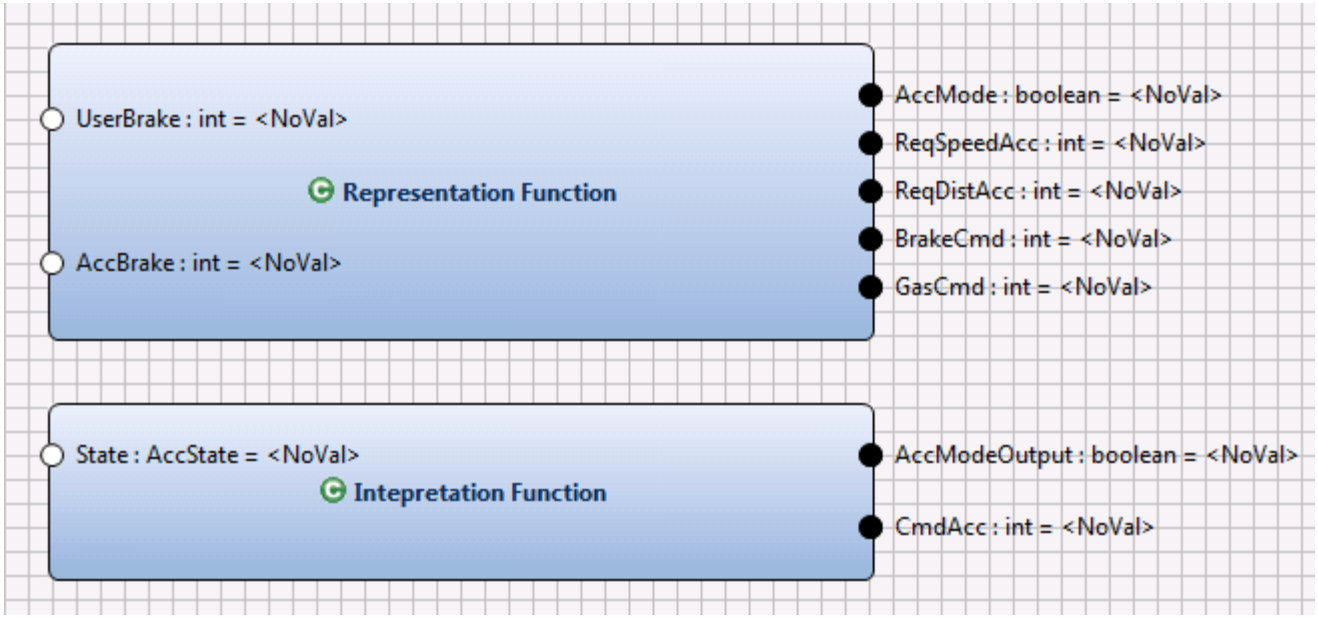
- State Automaton for Acceleration Control



Refinement Functions

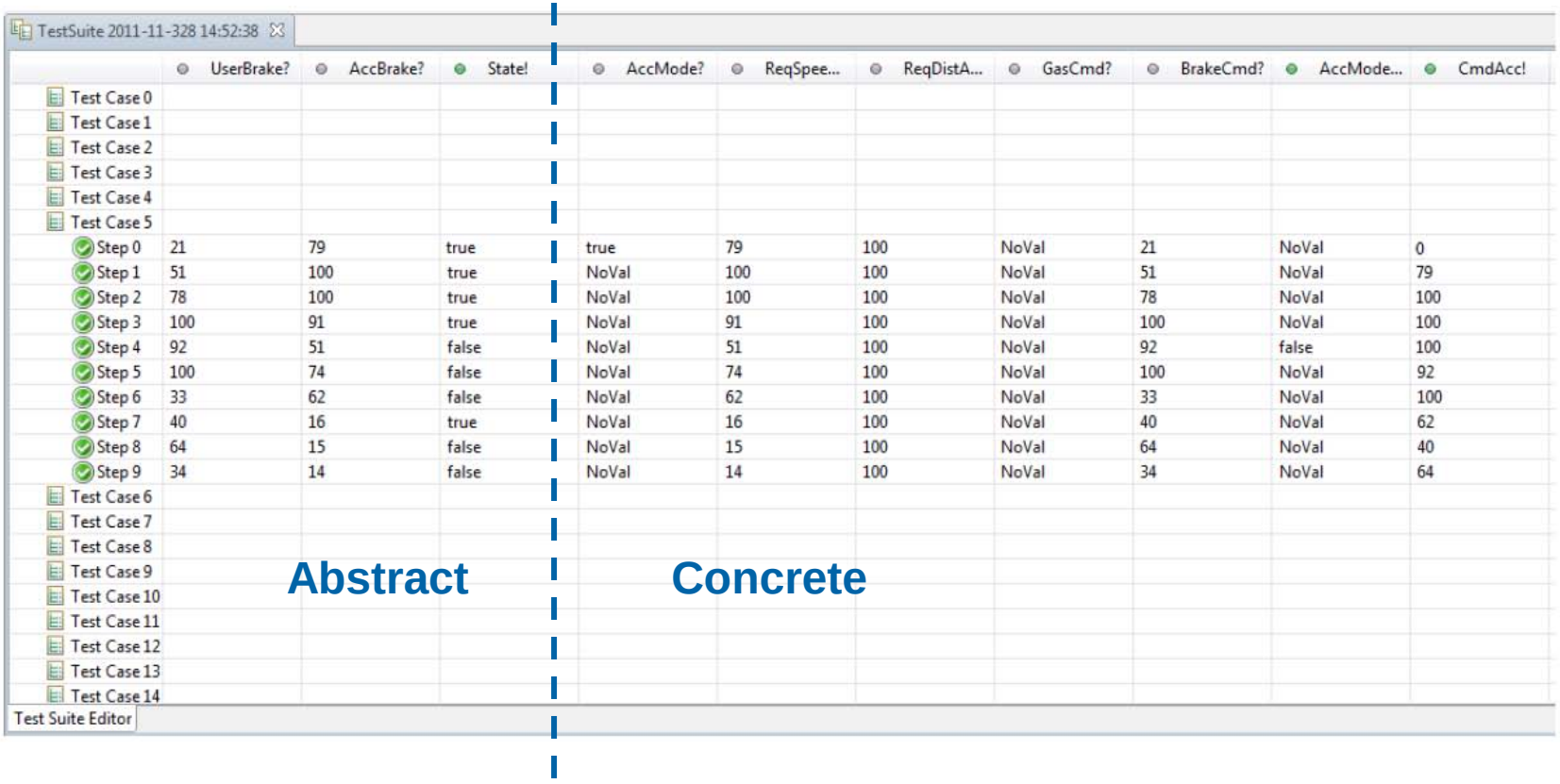


Example - Refinement Functions



Example - Results

- Expected Results
  - Combined view of abstract and concrete test data



	UserBrake?	AccBrake?	State!	AccMode?	ReqSpee...	ReqDistA...	GasCmd?	BrakeCmd?	AccMode...	CmdAcc!
Test Case 0										
Test Case 1										
Test Case 2										
Test Case 3										
Test Case 4										
Test Case 5										
Step 0	21	79	true	true	79	100	NoVal	21	NoVal	0
Step 1	51	100	true	NoVal	100	100	NoVal	51	NoVal	79
Step 2	78	100	true	NoVal	100	100	NoVal	78	NoVal	100
Step 3	100	91	true	NoVal	91	100	NoVal	100	NoVal	100
Step 4	92	51	false	NoVal	51	100	NoVal	92	false	100
Step 5	100	74	false	NoVal	74	100	NoVal	100	NoVal	92
Step 6	33	62	false	NoVal	62	100	NoVal	33	NoVal	100
Step 7	40	16	true	NoVal	16	100	NoVal	40	NoVal	62
Step 8	64	15	false	NoVal	15	100	NoVal	64	NoVal	40
Step 9	34	14	false	NoVal	14	100	NoVal	34	NoVal	64
Test Case 6										
Test Case 7										
Test Case 8										
Test Case 9										
Test Case 10										
Test Case 11										
Test Case 12										
Test Case 13										
Test Case 14										

Conclusion

- Proposed and implemented a method
  - To reuse test cases cross abstract and concrete domains
  - Part of a solution to support development with a manufacturer/sub-contractor relation

Future Work

- Automatic test suite generation for formal requirements
- Improvement of refinement functions for expressing of timing constraints and macro behaviors

Thank You

Dongyue Mou

fortiss GmbH – An-Institut der Technischen Universität München
Guerickestr. 25 | 80805 München | Germany
Tel. +49 89 360 35 22 – 19 | Fax +49 89 360 35 2250
mou@fortiss.org | www.fortiss.org