

# Model-Based Testing With RT-Tester

SysML – Formal Methods

Jan Peleska and Cécile Braunstein  
University of Bremen  
2013-04-16

# The MBT Approach

# Our MBT Approach

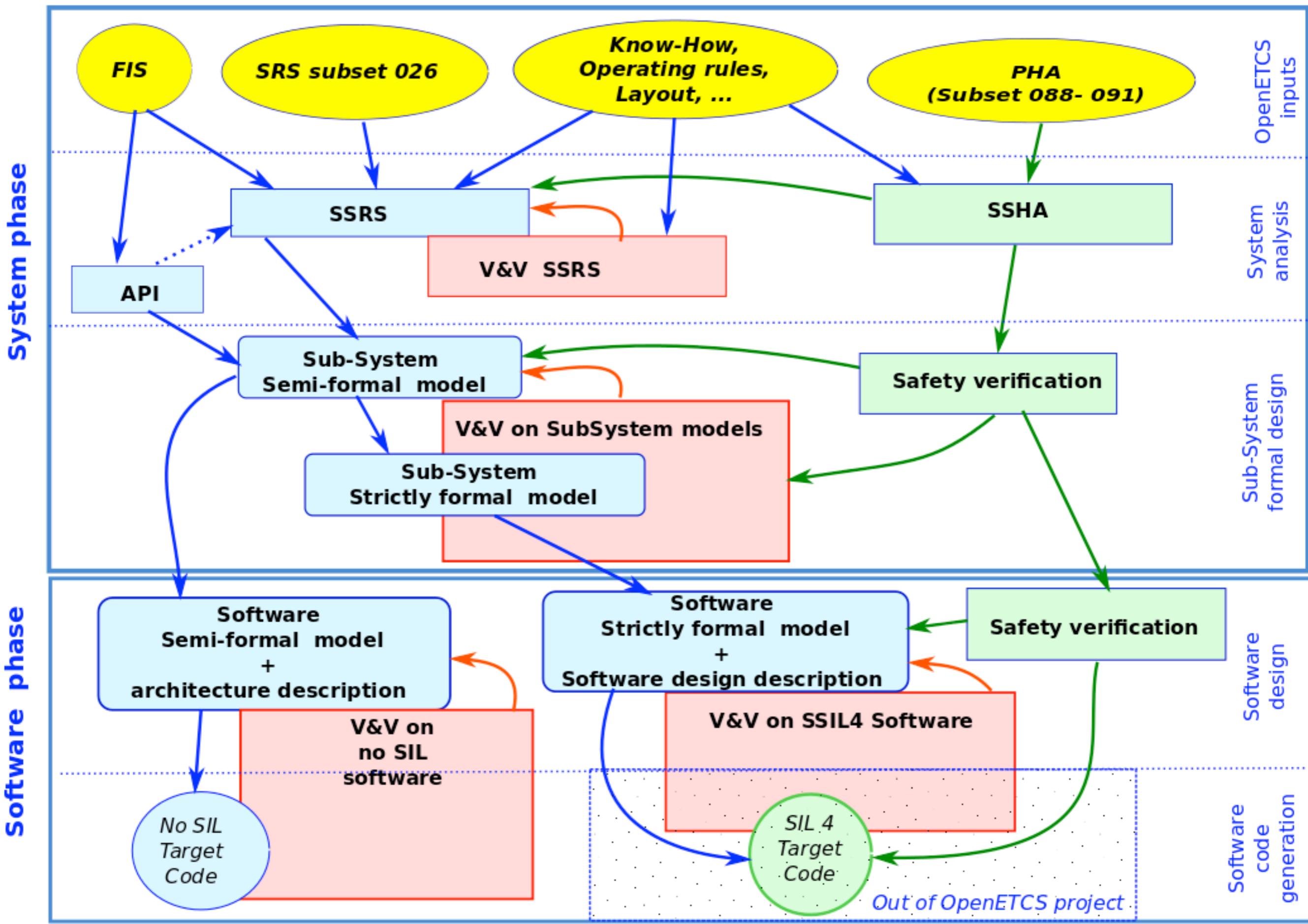
Instead of writing test procedures,

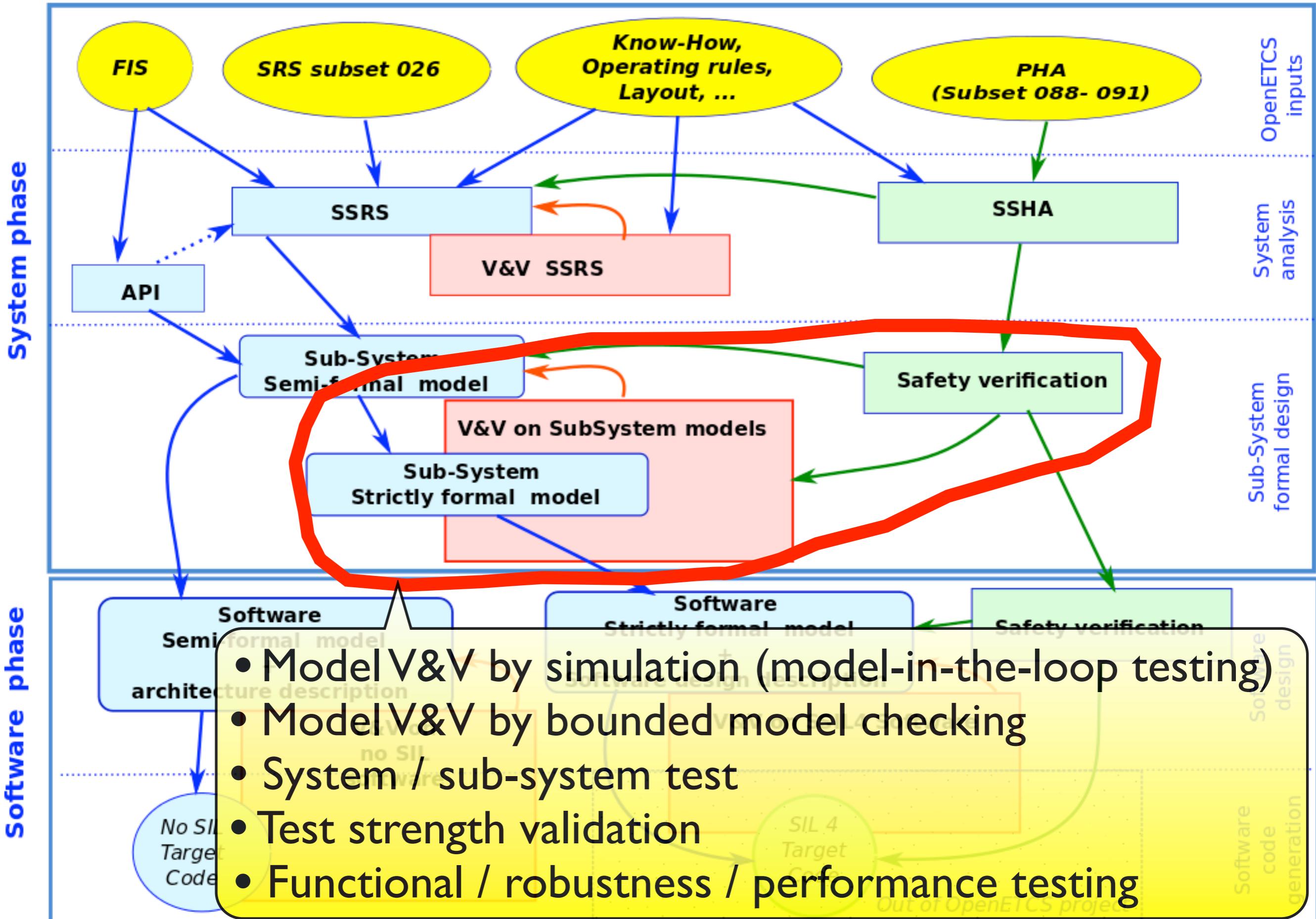
- develop a **test model** specifying expected behaviour of SUT
- use **generator** to identify “relevant” test cases from the model and calculate concrete test data
- generate **test procedures** fully automatic
- perform **tracing** requirements ↔ test cases in a fully automatic way

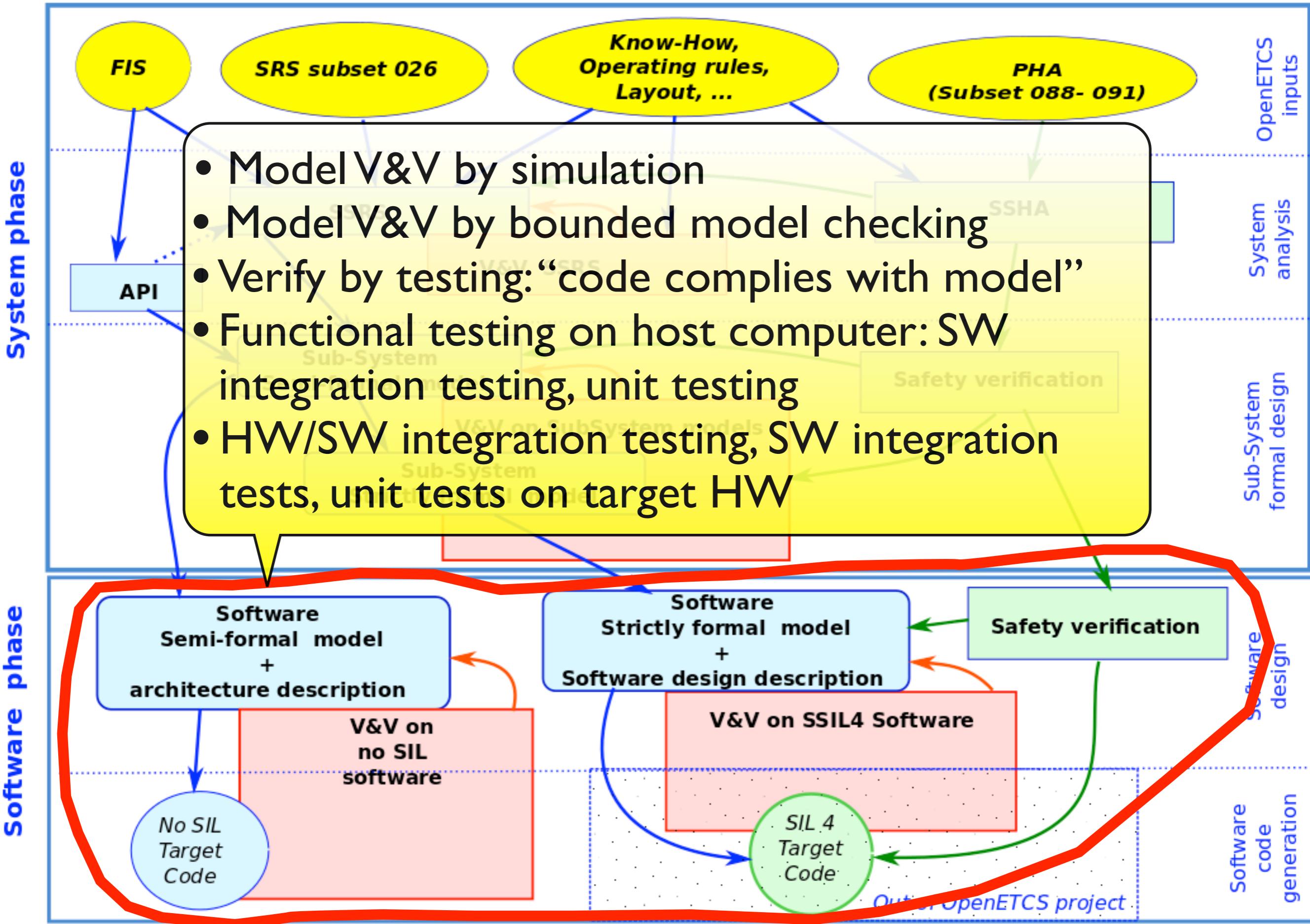
# Our MBT Approach

- Support all test levels – from unit to system integration testing
- Software tests and hardware-in-the-loop tests
- Test projects may combine hand-written test procedures with automatically generated procedures

# **RT-Tester in the openETCS Tool Chain**







# Demonstration

# Demonstration

- UML/SysML model with requirements tracing
  - Requirements have been broken down into atomic requirements
- Automated requirements-based test procedure generation
- Automated simulation generation
- Test execution and evaluation

# Demonstration – Case Study With UML/SysML

- Simplified version of the turn indication and emergency flashing function in Daimler vehicles
- Full model available under

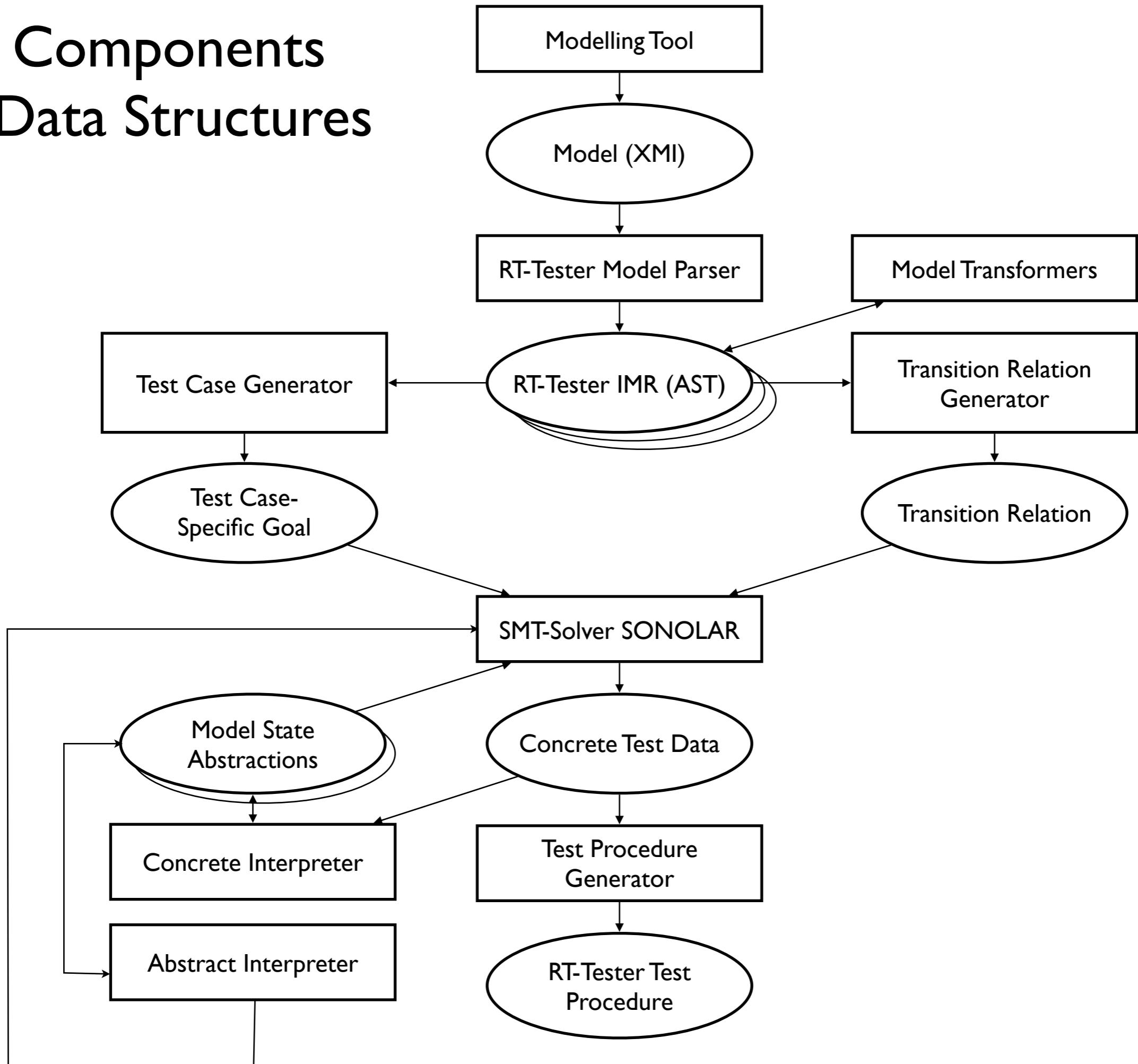
<http://www.mbt-benchmarks.org>

→ Benchmarks

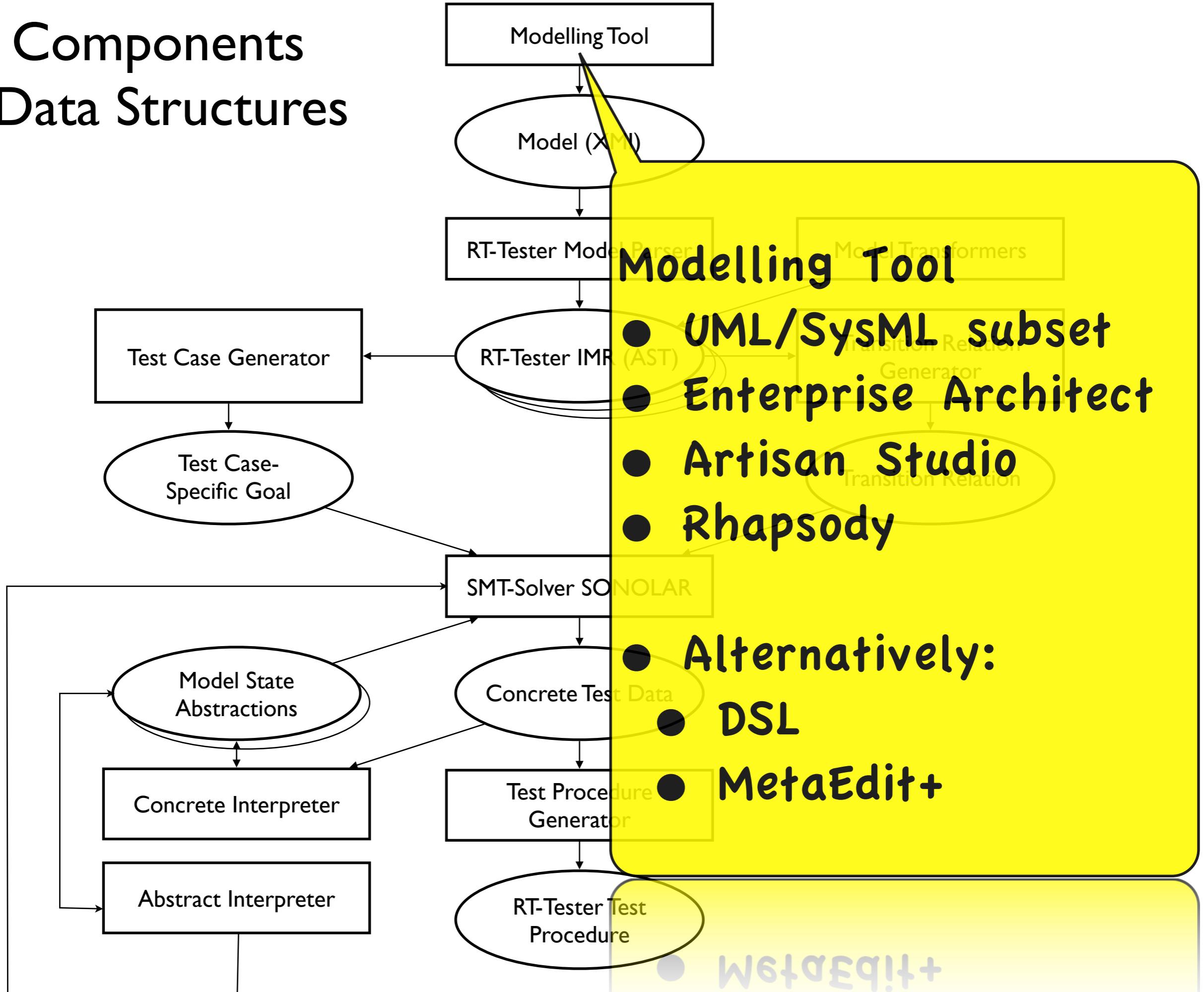
→ Turn Indicator Model Rev. 1.4

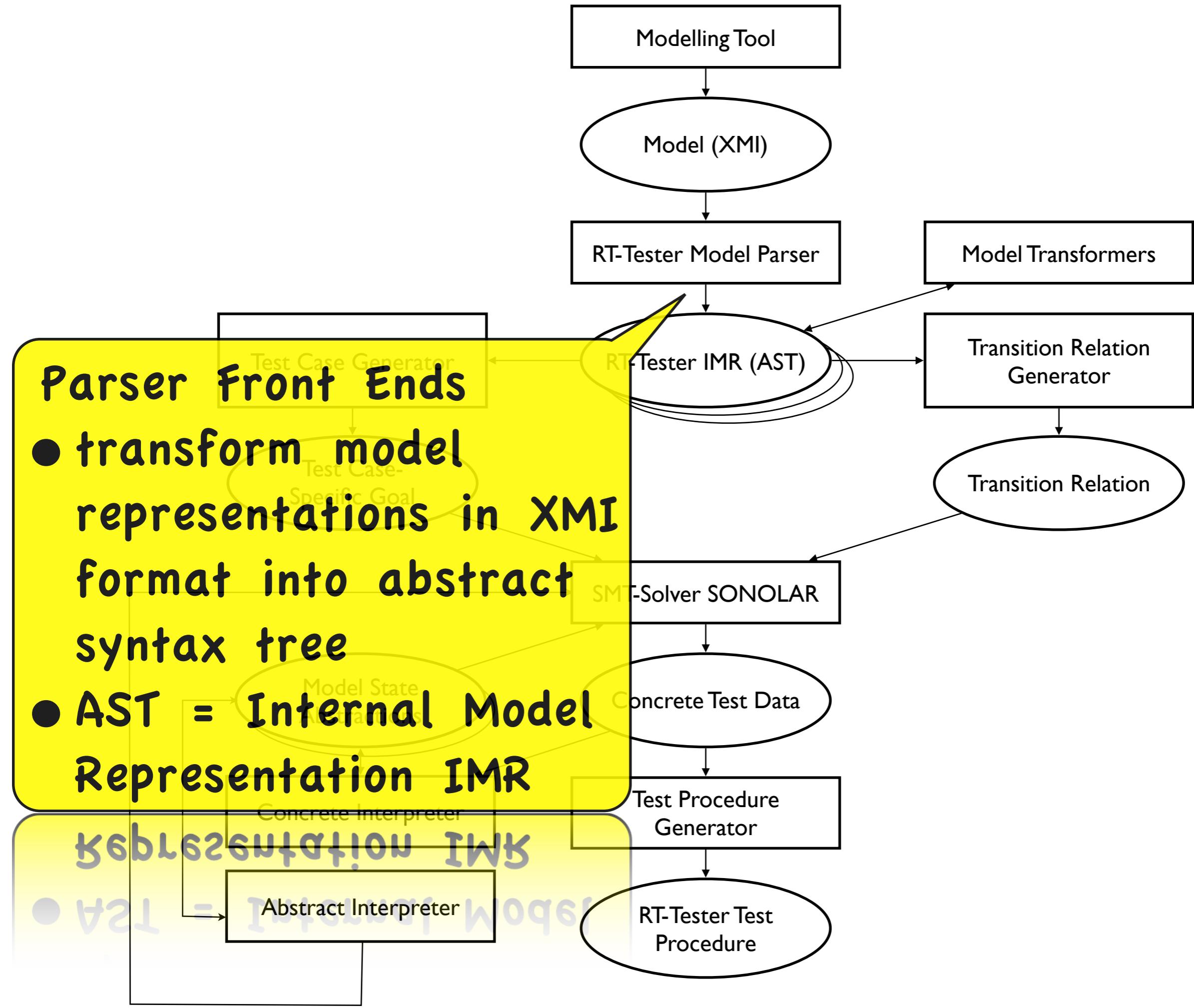
# RT-Tester Internals – an Overview

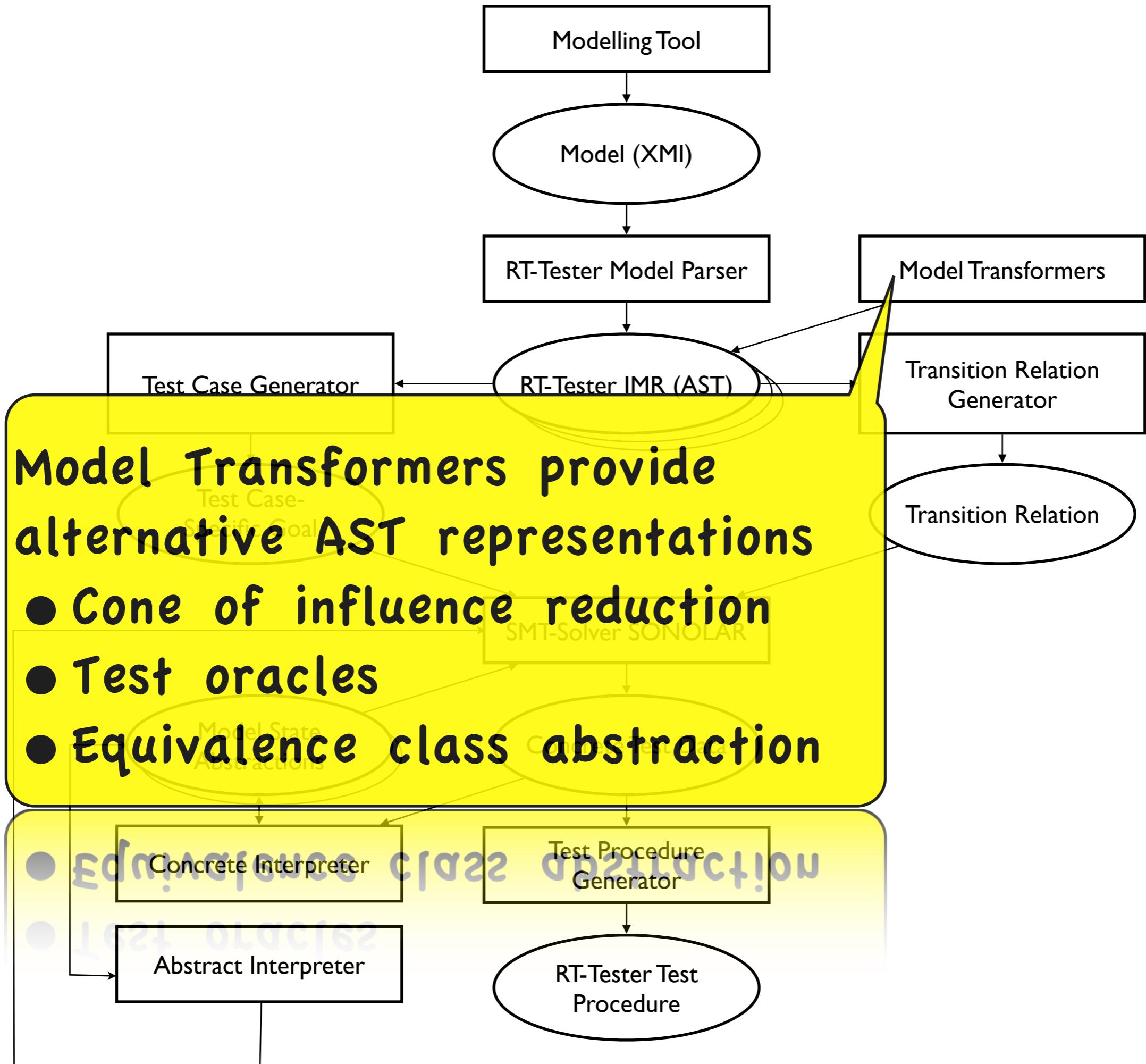
# Tool Components and Data Structures

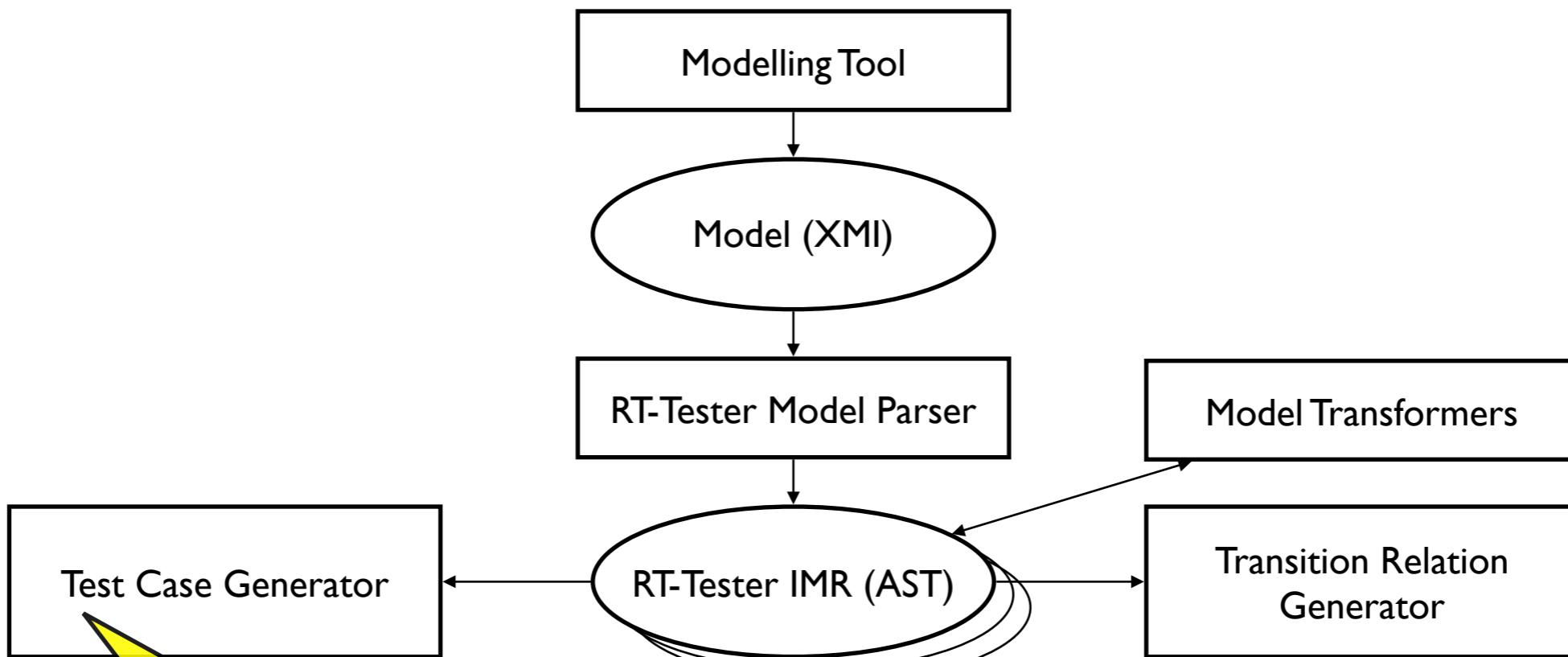


# Tool Components and Data Structures





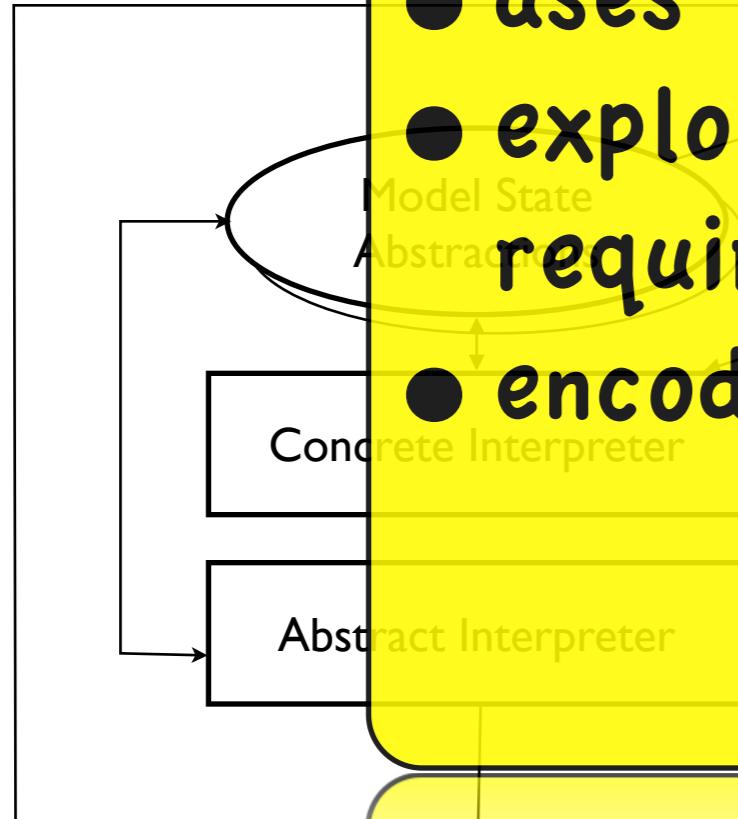




## Test Case Generator

- identifies “relevant” test cases
- uses ASTs as identification basis
- exploits traceability information from requirements to model elements
- encodes test case goals as propositions

$$G(s_0, s_1, \dots, s_c)$$



# Transition Relation Generator

- encodes operational semantics of the model by relating pre-states to post states

$$\Phi(s, s')$$

Modelling Tool

Model Transformers

Transition Relation Generator

Transition Relation

SMT-Solver SONOLAR

Model State Abstractions

Concrete Test Data

Concrete Interpreter

Test Procedure Generator

Abstract Interpreter

RT-Tester Test Procedure

Test Case Generator

RT-Tester IMR (AST)

Model (UML)

RT-Tester Model Parser

# SMT-Solver

- calculates solution of test goals which are compatible the transition relation

$$J(s_0) \wedge \bigwedge_{i=0}^n \Phi(s_i, s_{i+1}) \wedge G(s_0, \dots, s_{n+1})$$

Test Case-Specific Goal

RT-Tester Model Parser

Model Transformers

Transition Relation Generator

Transition Relation

SMT-Solver SONOLAR

Model State Abstractions

Concrete Test Data

Concrete Interpreter

Test Procedure Generator

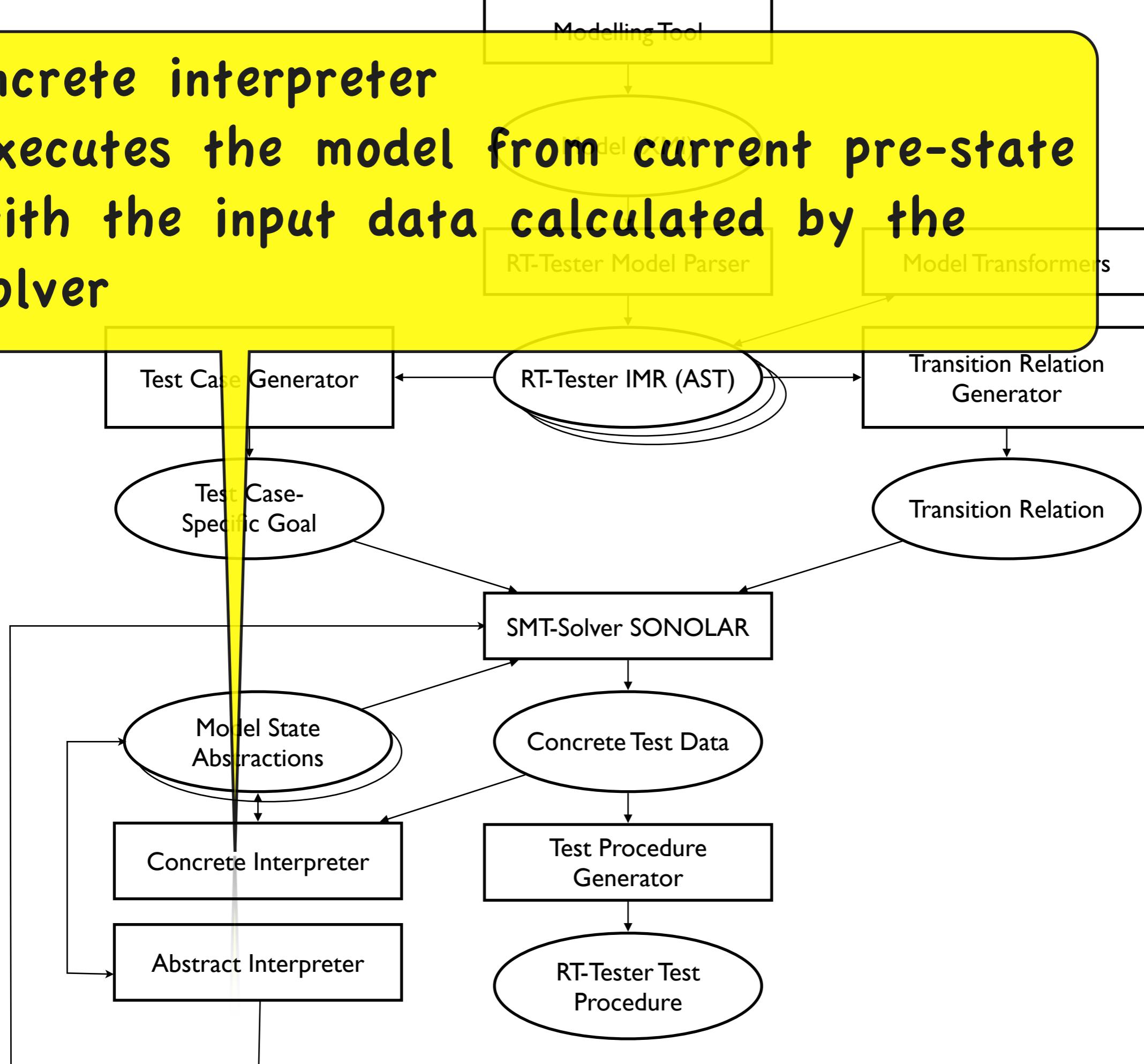
Abstract Interpreter

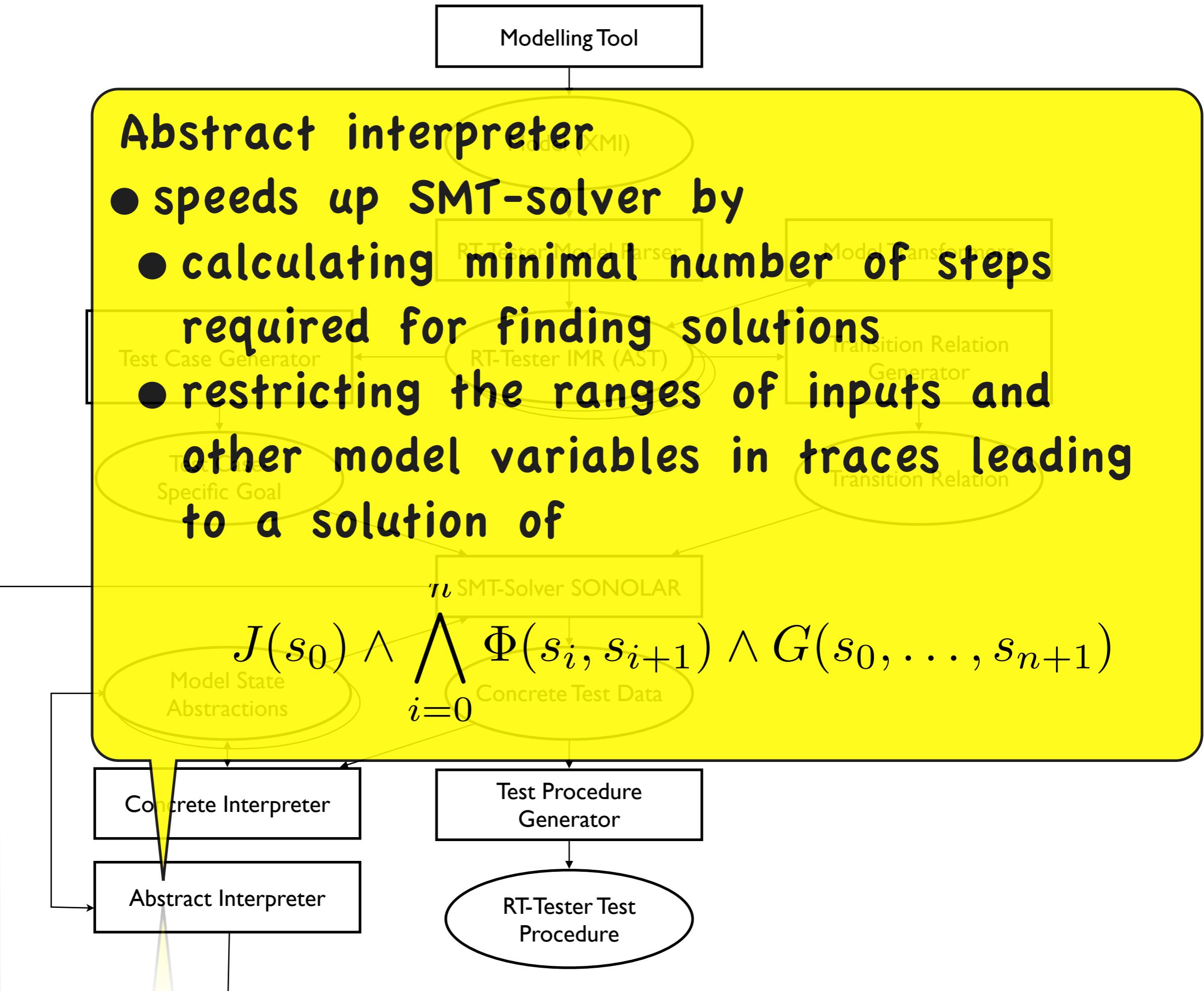
RT-Tester Test Procedure

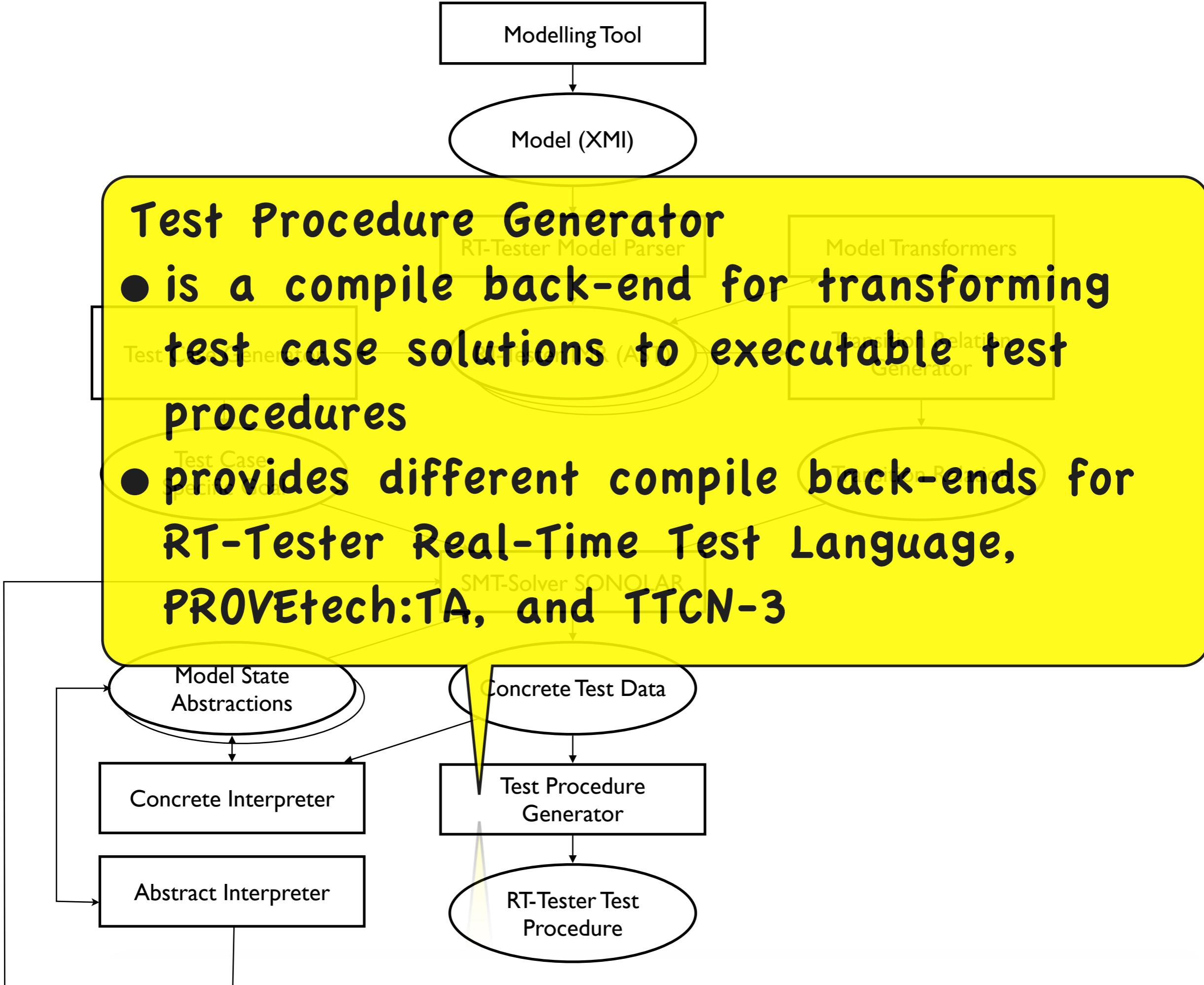
Can handle  
Boolean, Integer,  
Float, Array data  
types

# Concrete interpreter

- executes the model from current pre-state with the input data calculated by the solver







# Effort Figures

- Development costs: > 120 person years
- Lines of code core system: 500000
- Lines of code model-based test generator: 300000
- Additionally: graphical user interface and test management system

# Conclusion

# RT-Tester – Unique Selling Points

- Automated requirements tracing based on SysML test models
- Fully automated test case, test data, and test procedure generation for complex concurrent real-time models
- Test data generation can handle bit-precise integer and floating point arithmetics
- Justified test strategies compliant with standards

# RT-Tester – Unique Selling Points

- Comprehensive **tool qualification** according to
  - ISO 26262
  - RTCA DO178C – RTCA DO330
  - CENELEC EN50128:2011
- Automated **software testing** for **SCADE** software
- **Proven product**, applied in railway, avionic, and automotive domains
- **Open interfaces** to cooperating tools (DOORS, other test tools ...)

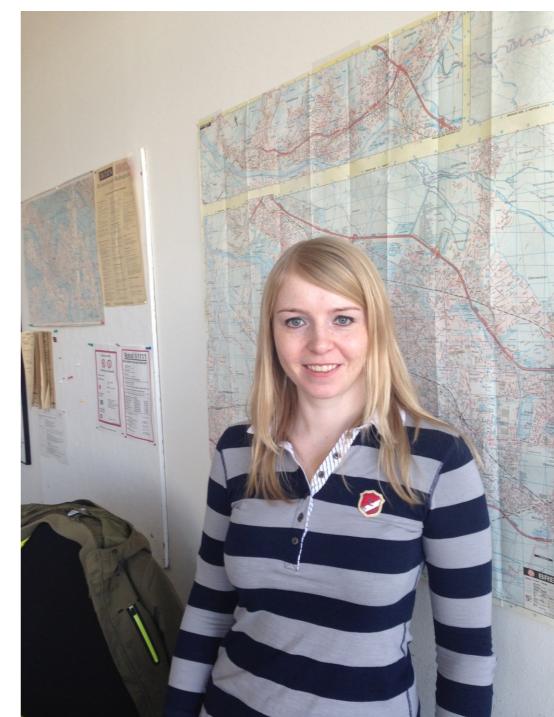
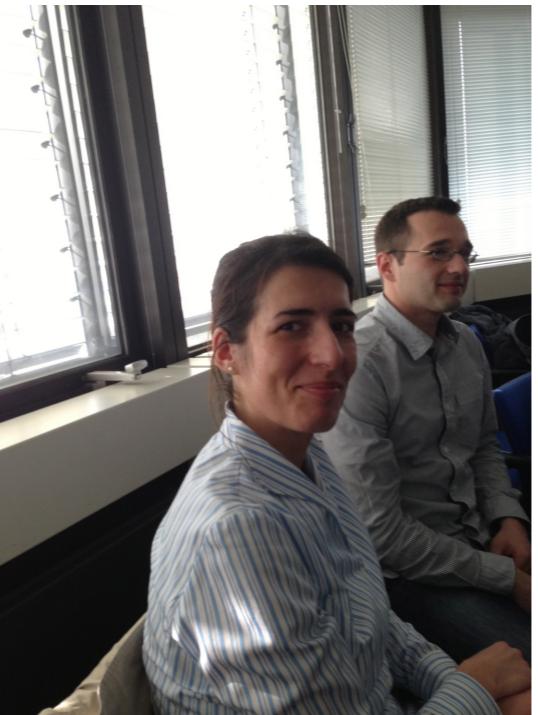
# RT-Tester And Open Source

- SMT-Solver SONOLAR will become open source in 2013
- Open source Eclipse GUI is currently tested internally, will be available with baseline I of the openETCS tool chain
- RT-Tester is freely available for universities and other research institutions
- Licenses have to be obtained for commercial use

# RT-Tester And Open Source

- Product is maintained and distributed by Verified Systems International GmbH
- All RT-Tester-related research about model-based testing is published by University of Bremen
- Open source business model is currently under investigation

# Contributors ...



# Contributors ...

