3rd UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

Sophia Antipolis, French Riviera
20-22 October 2015

USING FAULTS FOR EFFICIENT MBT FOR A COMPLEX RAILWAY APPLICATION

Presented by Rupert Schlick

AIT Austrian Institute of Technology

THALES

Railway Interlocking - The Application

- Application Purpose:
  - Ensure safe train movement
  - Prevent collisions and derailing of rolling stock
- Experimental Evaluation
  - used a functional subset of interlocking logic following Austrian railway operation rules
  - THALES product LockTrac 6131 Elektra, approx. 250 installations, 4 countries

Railway Interlocking - Process Challenges for Testing

- Complex Application Domain
  - 30 years in service
  - country specific requirement variants
  - multiple HW and OS platforms

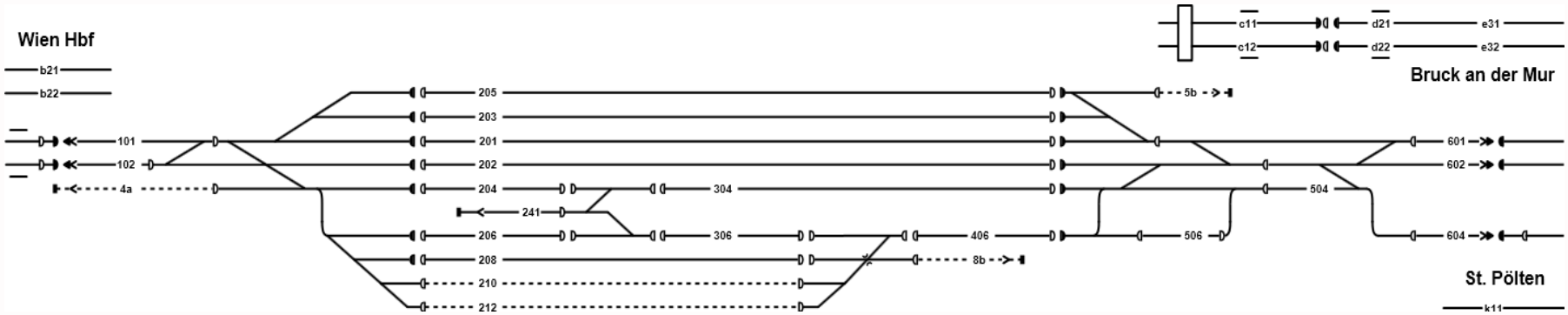
- Highly regulated domain
  - CENELEC standards, e.g EN50128 (software safety)
  - require controllable, documented test and verification process
  - traceability, certification of SW increments

Railway Interlocking - Technical Challenges for Testing

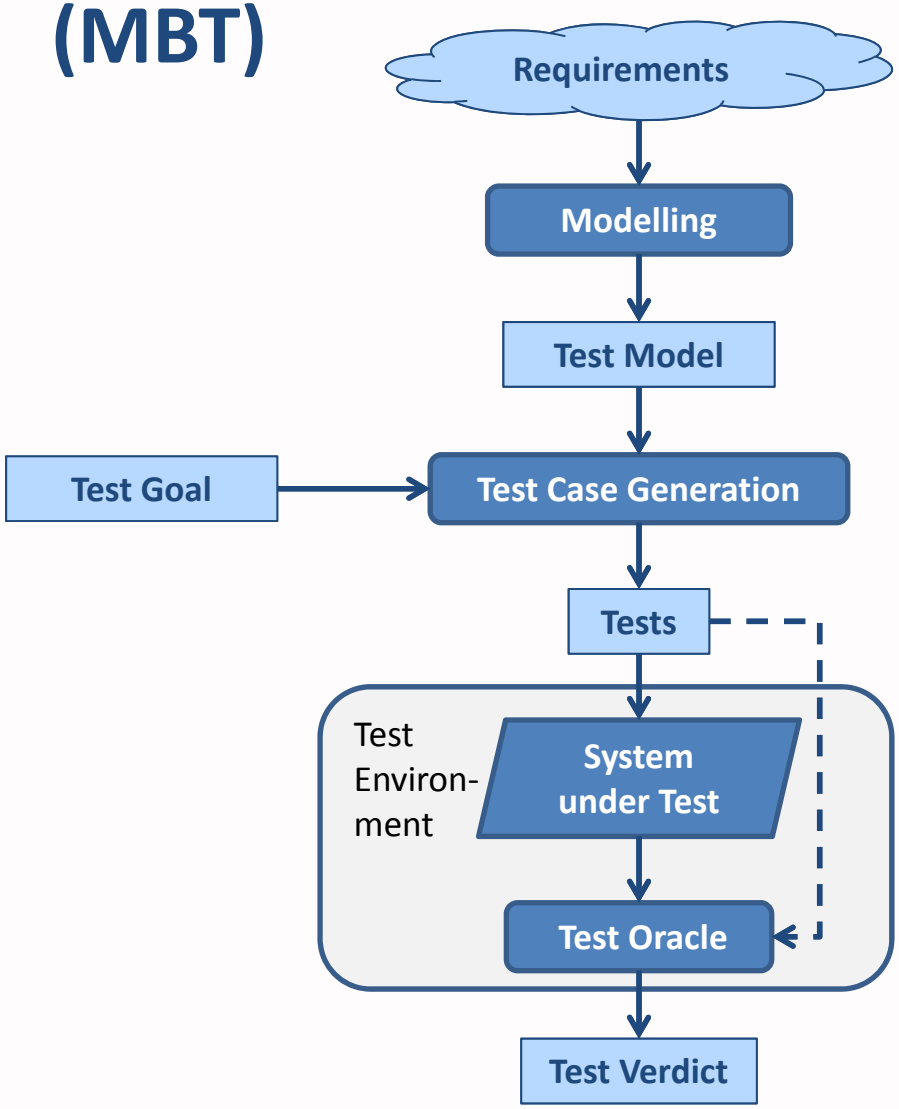
Example Rule Requirement:
IL:RULE:121: A switch shall reject any kind of moving command (both if it is a manual command and if it is an automatically generated command), if the switch holds a lock or any interlock or an interlock request.

Complexity

- 71 rule requirements in simplified eval. example
- example test station has:
  - 34 points, 56 track relais, 22 signals, 145 train routes

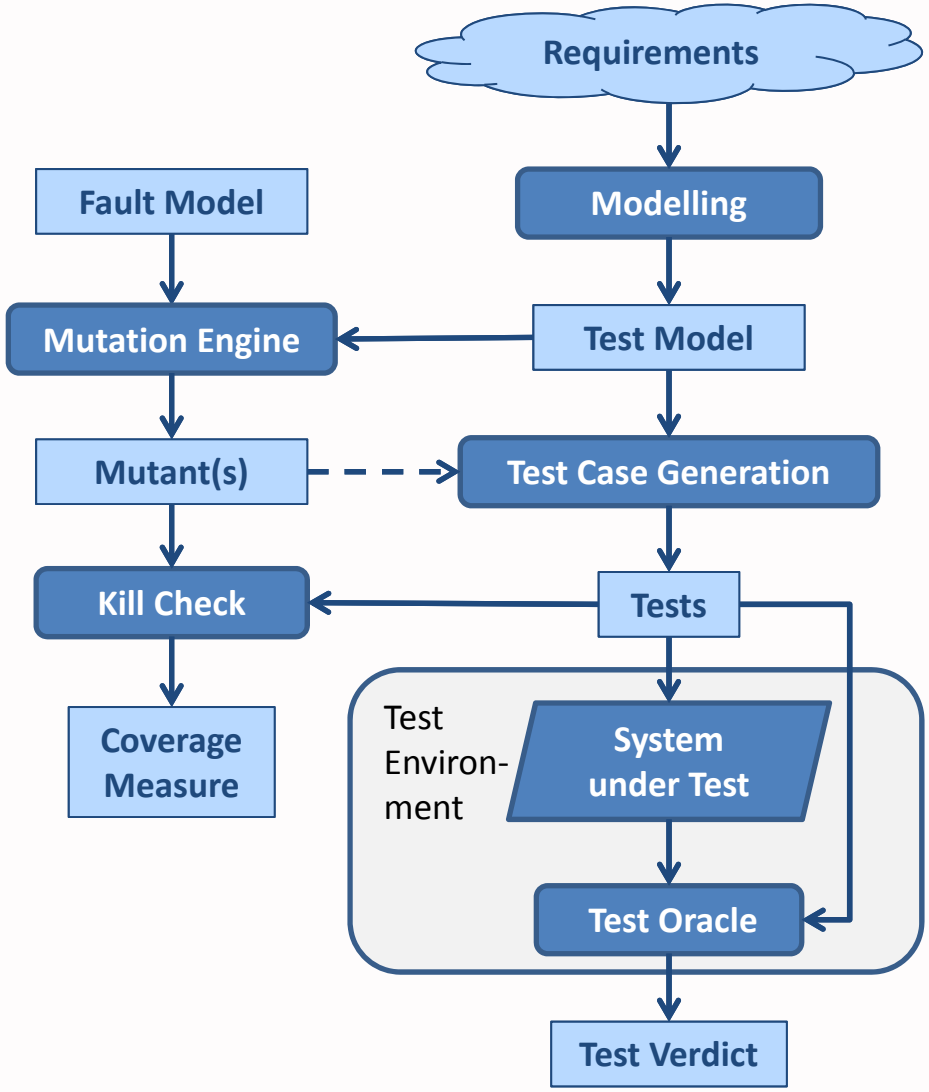


Principle of Model Based Testing (MBT)



- Test Model:
  - sequences/scenarios
  - state machines
  - formal requirements
  - usage probabilities …
- Test Goal:
  - target state (condition)
  - number of tests (random walks)
  - coverage
    - requirement
    - model structure
    - user inputs ..
- Test oracle:
  - no crash, no deadlock
  - correct behaviour (subset)
  - invariants…

Principle of fault based MBT

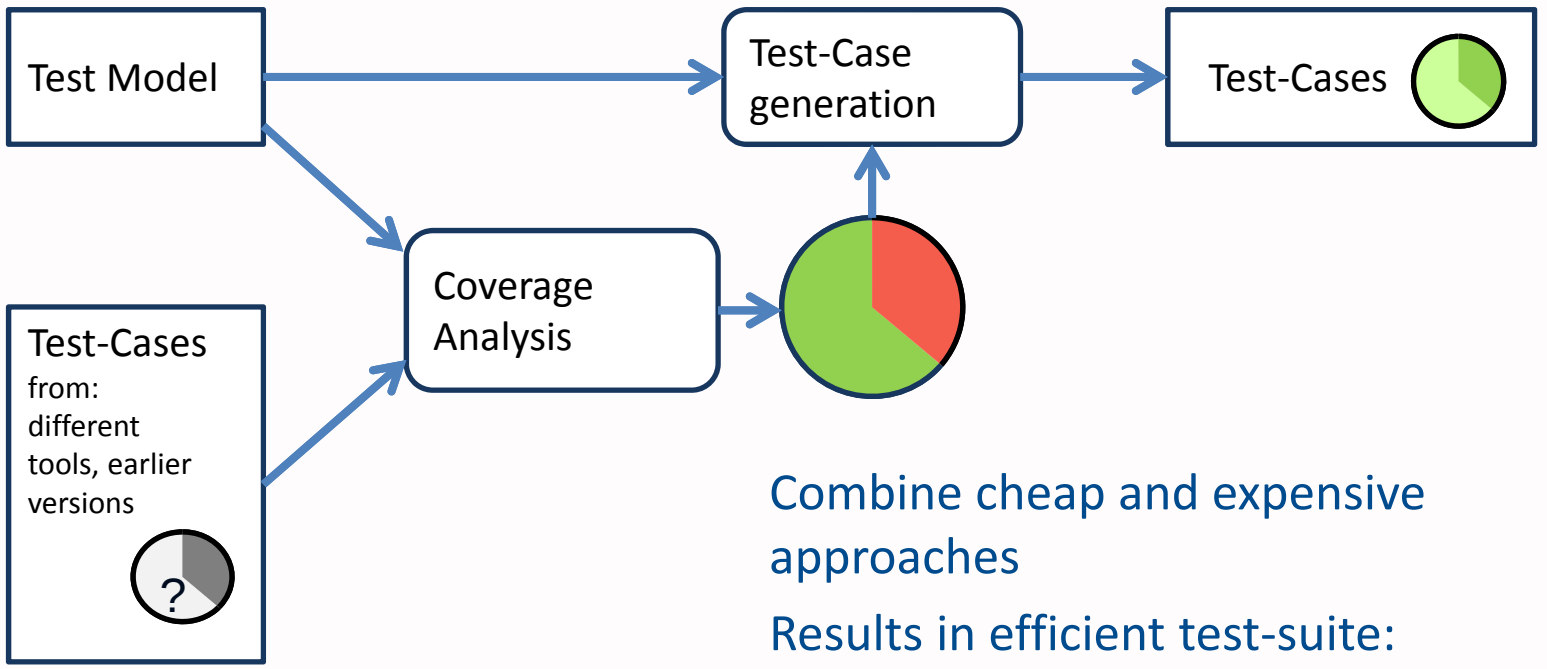


- behaviour model
- mutant: model with a small, syntactically correct change
- used for both:
  - test quality analysis
  - as a test goal (fault coverage)

Comparison with other coverage driven approaches

- structural coverage alone in state machines (e.g. transitions) is not enough -> decision, data flow
- data flow coverage not easily done in concurrent models with instances
- observability not inherent in classic coverage
- safety standards request certain coverage criteria for code

Combining Strategies



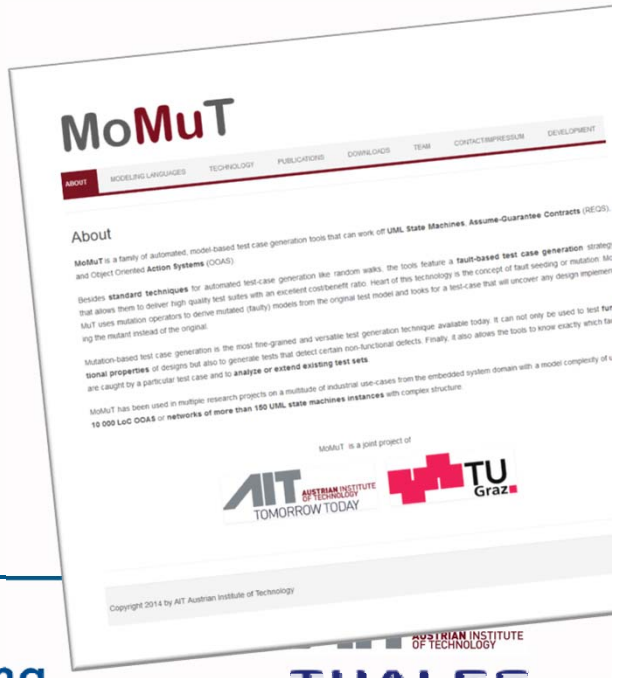
Combine cheap and expensive approaches

Results in efficient test-suite:

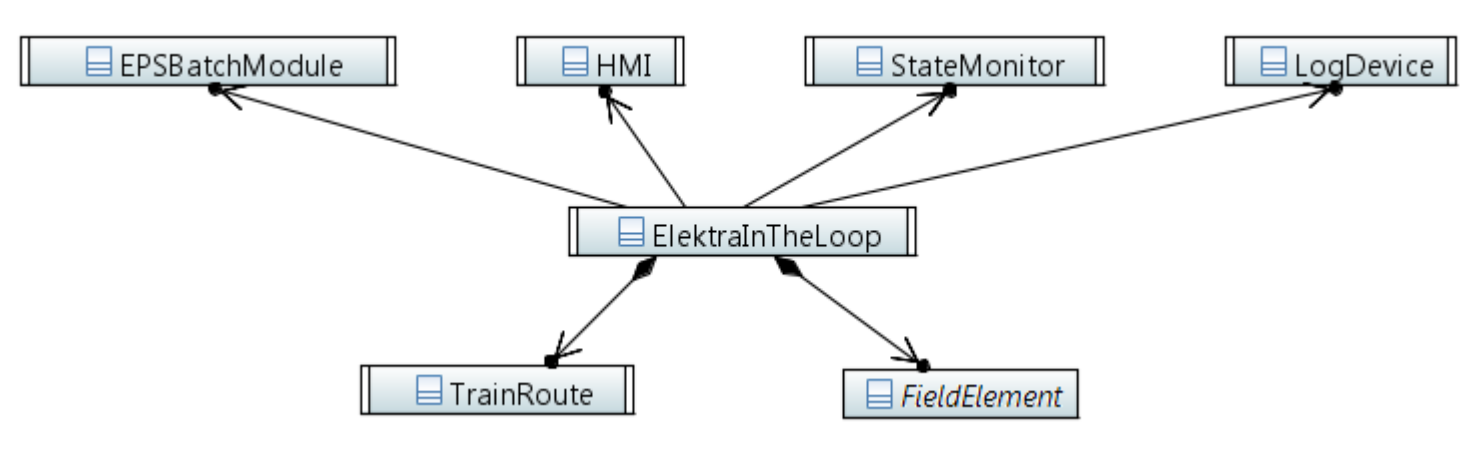
- Full coverage
- Optimized test effort
- Integration of legacy-tests

The Test Case Generator: MoMuT

- TCG engine
  - Input from different modelling tools
- Papyrus UML language front-end used in evaluation
  - Generation from UML state machines
- Other modelling languages planned:
  - DSLs from industrial users
  - Timed Automata
  - Event-B



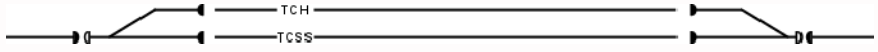
Behaviour Model

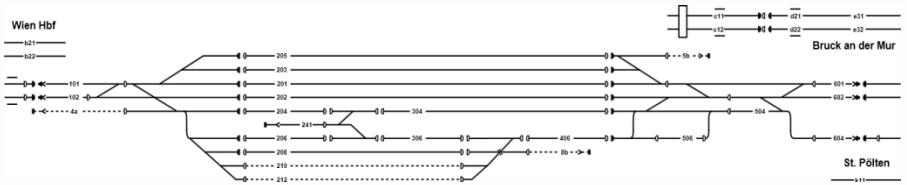


- 32 classes (4 environment, 18 field element, 10 trainroute logic)
- 18 active classes (state machines)

Example Stations + Model Size

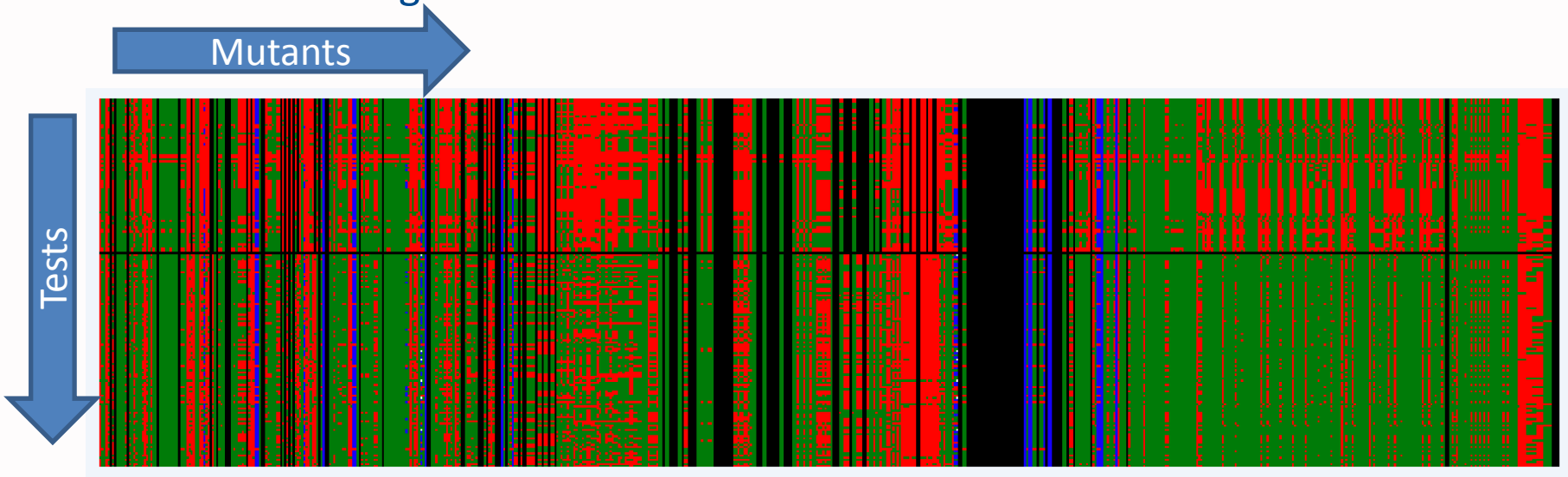
Station	MMS	LBT
Characterisation	A small meeting station	Layout used for train route tests
# track relays	4	56
# signals	8	22
# points	2	34
# train routes	10	145
# instances	125	2847
# controllable inputs	172	1652
State size / kB	22,3	> 184,9





Evaluation of generated tests

- UML mutation coverage of:
  - original tests from production use
  - random tests generated from model



- evaluation of test coverage
- option to prioritize tests
- derive traces from test to requirement

Performance/Applicability

Generates tests with overall 450 steps for MMS in 23 minutes, covering 680 of 2044 mutants

- Abstract tests including oracle and coverage information
- Not cleaned up for unreachable mutants

Use of enumerative exploration

- Just-In-Time Compilation based on LLVM 3.6
- Partial Order Reduction
- Partial Orders Encoded in Test Cases
- Exploring mutants only for needed steps (<5 steps for 99 %)
- Search based exploration driven by mutants (LBT + 10 % cov.)

Conclusion – How are the challenges addressed?

- reduced effort
  - automated test development
  - efficient tests -> affordable test run time
  - less maintenance effort
- sufficient test quality
  - better suited coverage criterion (for generation)
- certification of increments
  - only needed changes to test suite (improvement support)
- complexity can be handled
  - automated generation of tests in reasonable time

Acknowledgements

- Partners
  - Thales Austria GmbH
  - Graz University of Technology
    Institute for Software Technology
- Funding Agencies:
- Projects:

Contacts

www.MoMuT.org

Rupert Schlick,

Willibald Krenn

Department Digital Safety and Security
Business Unit Safe and Autonomous Systems

AIT Austrian Institute of Technology GmbH

Donau-City-Straße 1 | 1220 Vienna | Austria
http://www.ait.ac.at | F +43(0) 50550-4150

rupert.schlick@ait.ac.at | T +43(0) 50550-4124

willibald.krenn@ait.ac.at | T +43(0) 50550-4109

Werner Schütz

Head Methods and Tools

Thales Austria GmbH

Handelskai 92 | 1200 Vienna | Austria
http://www.thalesgroup.com/austria

werner.schuetz@thalesgroup.com
T +43(0)1-27711-3115