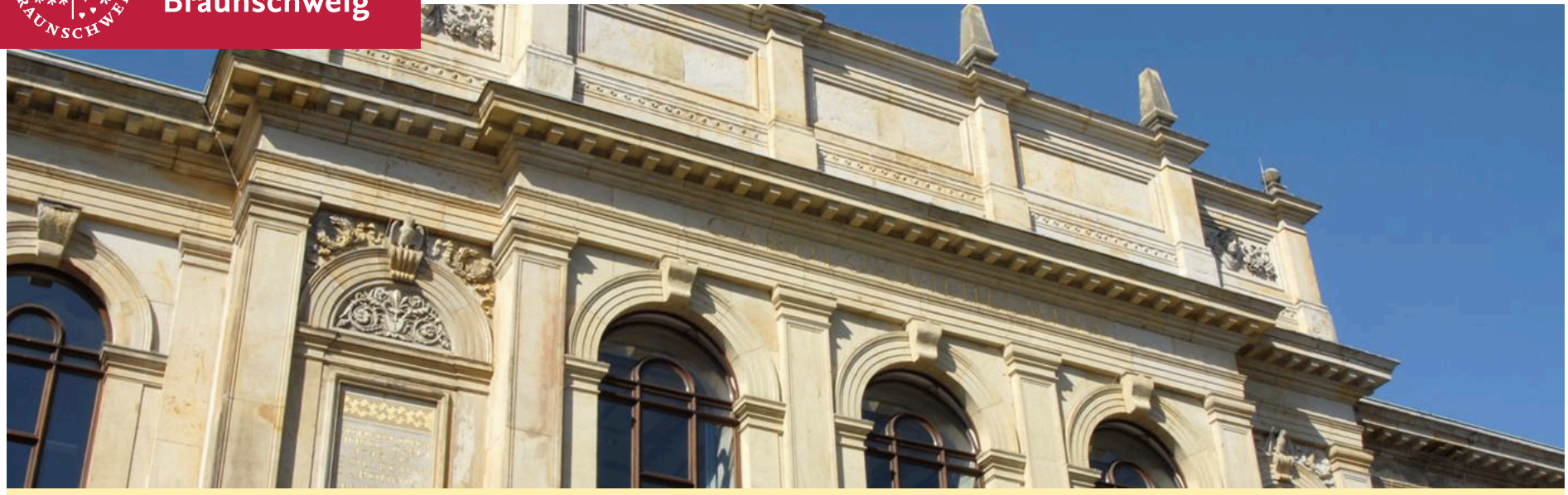
Technische Universität Braunschweig

ISF



Incremental Model-based Testing of Delta-oriented Software Product Lines

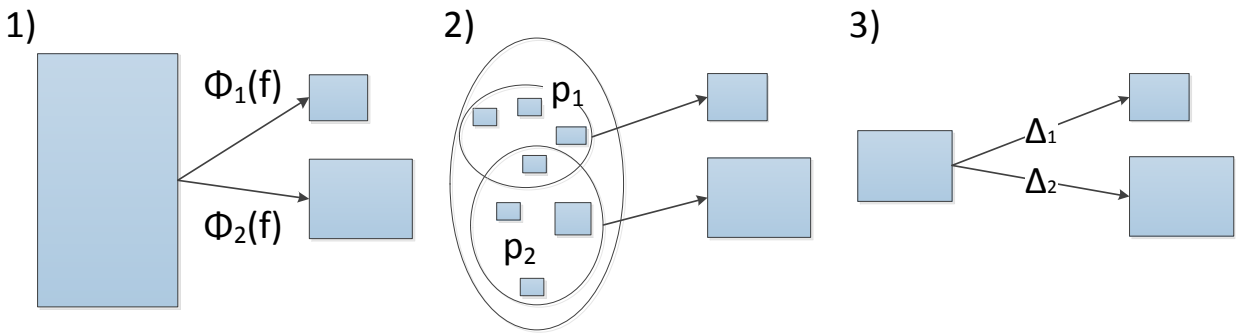
Ina Schaefer
(joint work with Malte Lochau and Sascha Lity)

IFIP WG 2.11 Meeting, Halmstad, 26 June 2012

Software Product Line Engineering

Reuse Approaches in SPL Engineering:

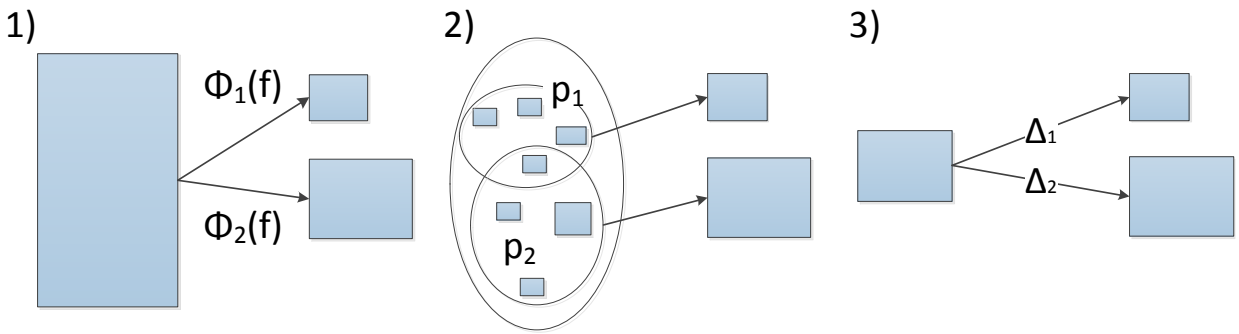
1. Annotation: artifact selection from 150% core product
2. Composition: combine core artifacts
3. Transformation: apply deltas to core product



Software Product Line Engineering

Reuse Approaches in SPL Engineering:

1. Annotation: artifact selection from 150% core product
2. Composition: combine core artifacts
3. Transformation: apply deltas to core product

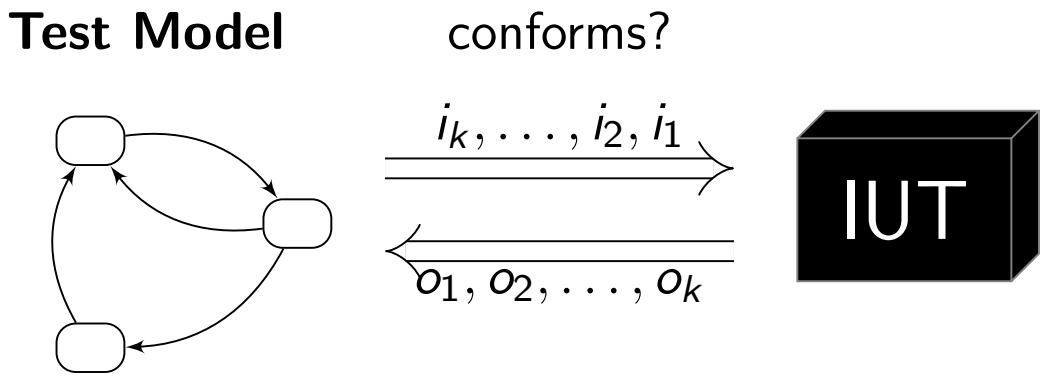


⇒ But how to reuse **test artifacts** for SPL quality assurance?

Agenda

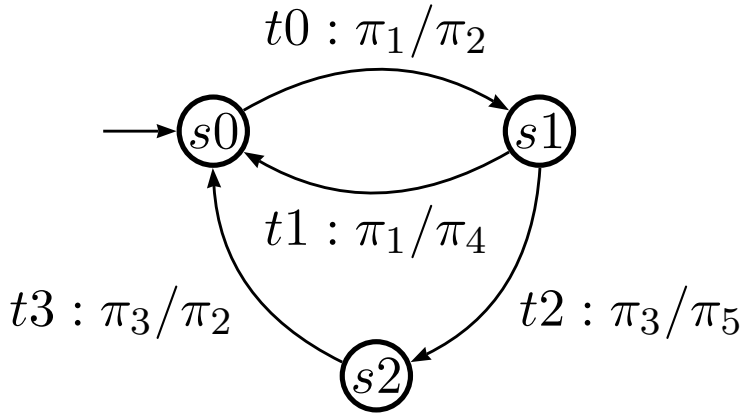
- Model-based Testing
- Model-based SPL Testing
- Incremental Model-based SPL Testing
  - Delta-oriented Test Models
  - Delta-oriented Test Regression

Model-based Testing



- Black-box assumption for implementation-under-test (IUT)
- Automated derivation and application of test suites from a behavioral test model [UL07]

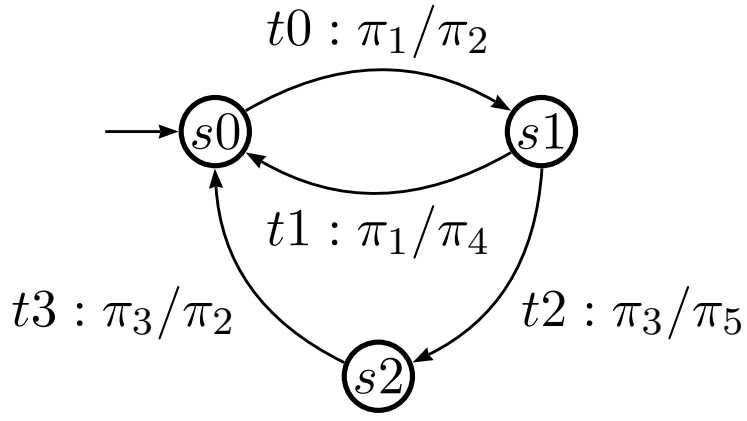
State Machine Test Model



State machines as test models

- S is a finite set of states
- $s_0 \in S$ is an initial state
- $L \subseteq \Pi_I \times \Pi_O$ is a set of transition labels
- $T \subseteq S \times L \times S$ is a transition relation

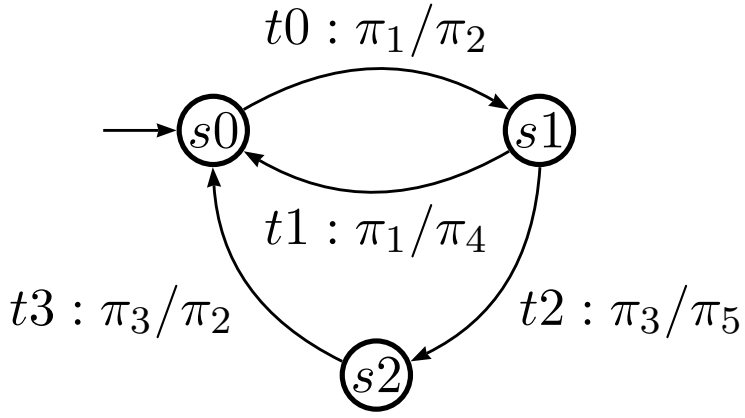
State Machine Test Case



State machine test artifacts:

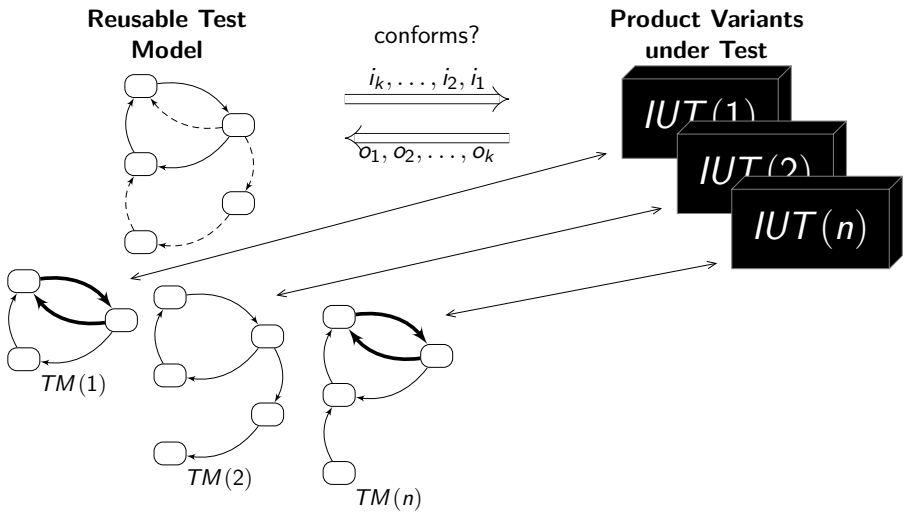
- Test case: $tc = (t_0, t_1, \ldots, t_k) \in T^*$ is a sequence of k transitions
- Test run: $exec(tc, tm) = (l_0, l_1, \ldots, l_k) \in L^*$
- Test result: *iut* **passes** $tc :\Leftrightarrow exec(tc, iut) \approx_{te} exec(tc, tm)$

State Machine Test Suite



- Infinite set of all valid test cases: $TC(tm) \subseteq T^*$
- Derive finite test suites: $ts \subseteq TC(tm)$
- Coverage criteria C: finite sets $tg = C(tm)$ of test goals such that $\forall g \in tg : \exists tc \in ts : covers(tc, g)$

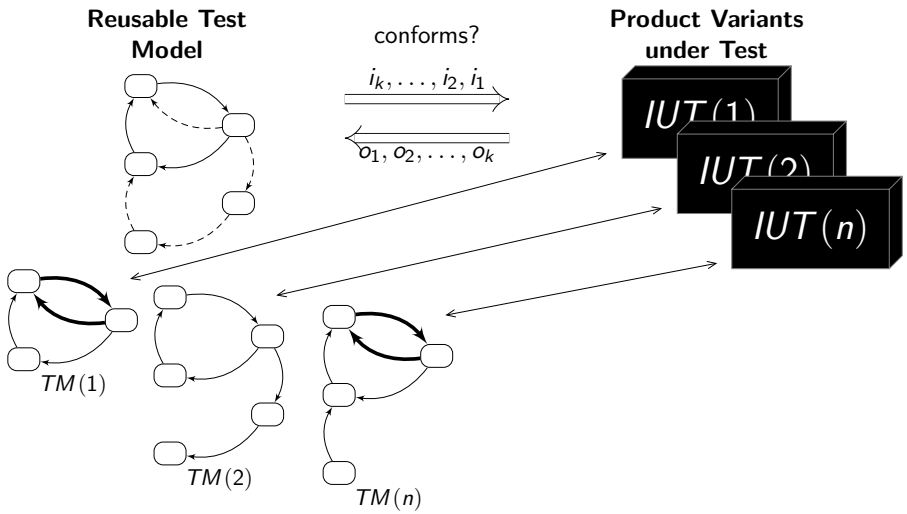
Model-Based Testing of Software Product Lines



Testing a family of similar software product variants

1. Reusable SPL test model [Oli08]
2. Selection of representative products-under-test [OWES10]

Model-Based Testing of Software Product Lines

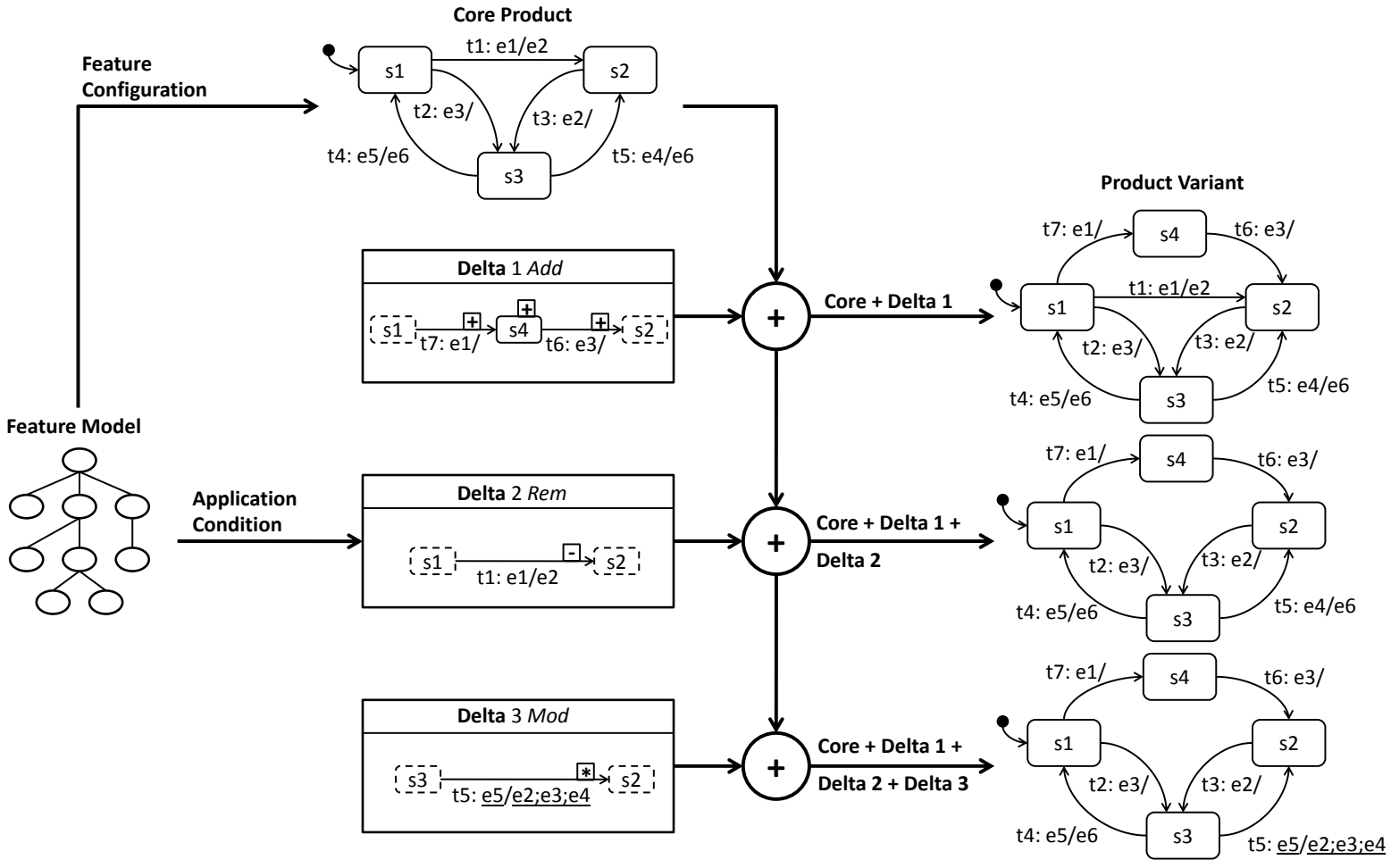


Testing a family of similar software product variants

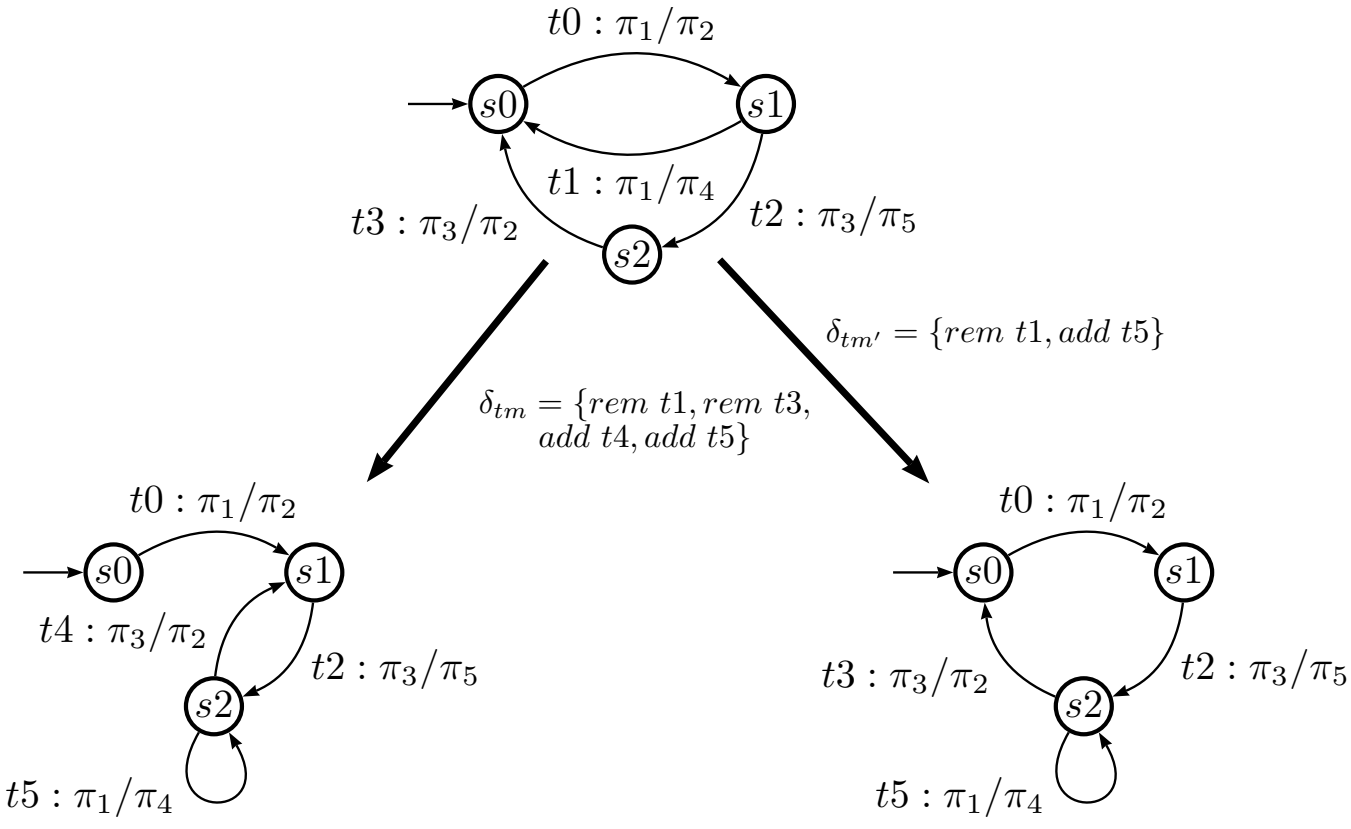
1. Reusable SPL test model [Oli08]
2. Selection of representative products-under-test [OWES10]

$\Rightarrow$ **Incremental test artifact reuse between products-under-test**

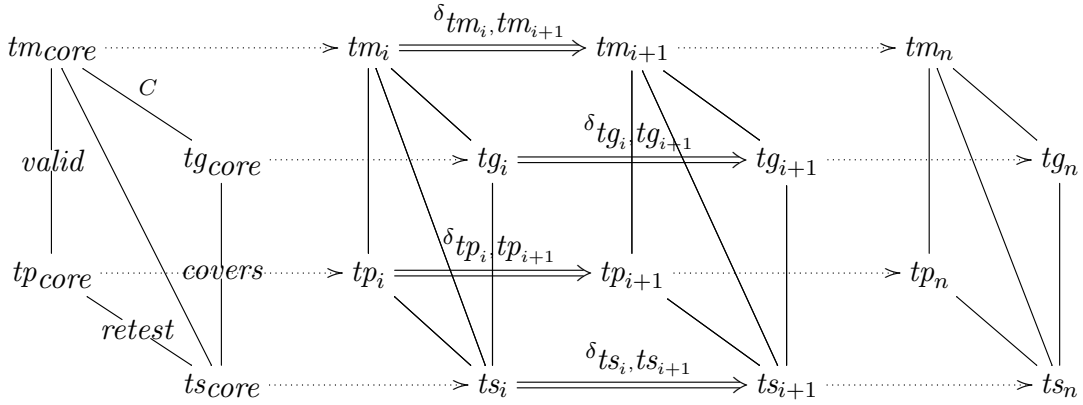
Delta-Oriented SPL Engineering [CHS11]



State Machine Delta – Example



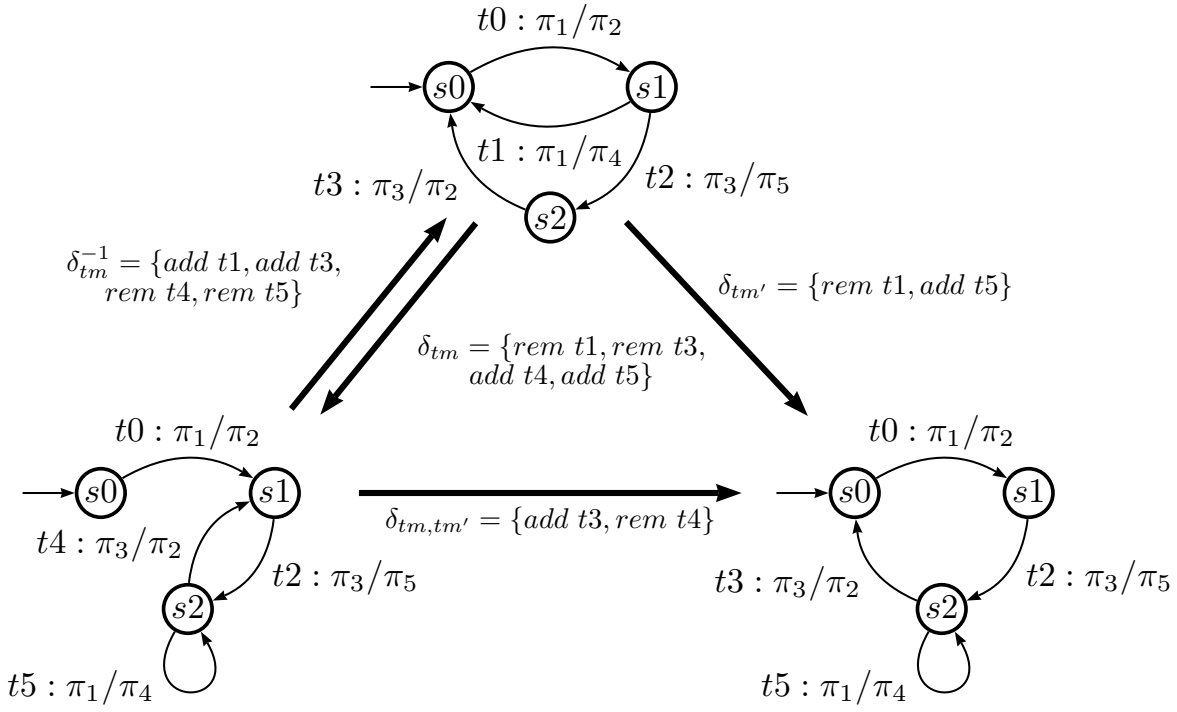
Incremental SPL Testing



- Starting with test artifacts of the core product
  $ta_{core} = (tm_{core}, tg_{core}, ts_{core}, tp_{core})$
- Incremental evolution of product test artifacts
  $ta_p = (tm_p, tg_p, ts_p, tp_p) \xrightarrow{\delta_{ta,ta'}} ta_{p'} = (tm_{p'}, tg_{p'}, ts_{p'}, tp_{p'})$

Test Model Regression Delta

State machine regression delta $\delta_{tm,tm'} \subseteq Op$ captures the difference between test model variant tm and tm'.



Regression Delta Construction

Proposition (State Machine Regression Delta Construction)

For two state machine deltas $\delta_{tm} \subseteq Op$ and $\delta_{tm'} \subseteq Op$, the regression delta is given as

$$\delta_{tm,tm'} = (\delta_{tm} \backslash \delta_{tm'})^{-1} \cup (\delta_{tm'} \backslash \delta_{tm})$$

- Invert delta operations exclusive to *tm*,
- Keep delta operations shared with *tm′*
- Add delta operations exclusive to *tm′*

$\Rightarrow$ Basis for **test artifact evolution**

Test Goal Delta

Test goal regression delta $\delta_{tg,tg'}$

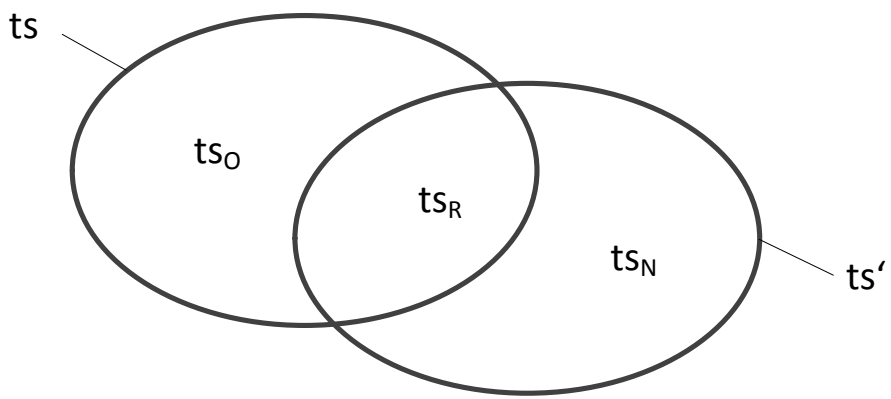
- for each $\{rem\ e\} \subseteq \delta_{tm,tm'} : e \in tg \Rightarrow \{rem\ e\} \subseteq \delta_{tg,tg'}$
- for each $\{add\ e\} \subseteq \delta_{tm,tm'} : e \in C(tm') \Rightarrow \{add\ e\} \subseteq \delta_{tg,tg'}$

$\Rightarrow$ Limited to C0/C1-like coverage criteria

Test Suite Delta

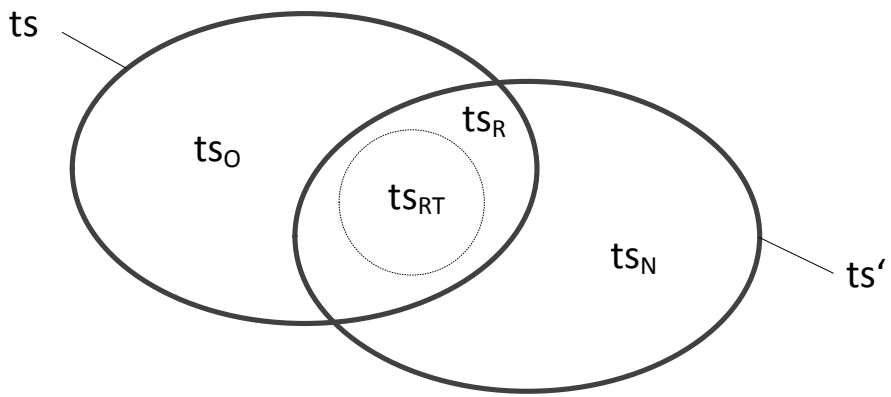
Test suite regression delta $\delta_{ts,ts'}$

- Obsolete test cases $ts_O = ts \backslash ts_R$
- Reusable test cases $ts_R = ts \cap ts'$
- New test cases $ts_N = ts' \backslash ts$



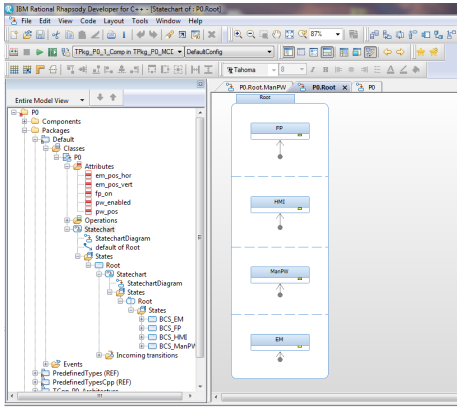
Test Plan Delta

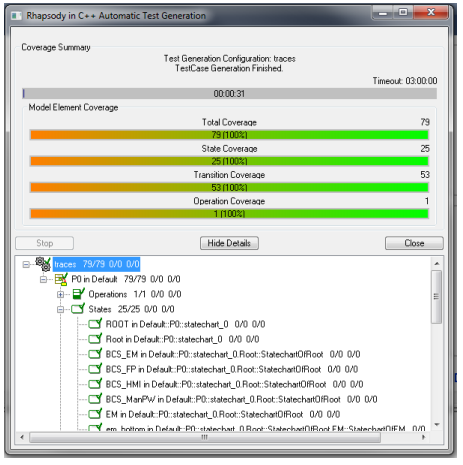
- Test plan: $tp_p = ts_{RT} \cup ts_N$
- Regression Test Selection for ts_{RT}: Retest-all, retest-random, test model slicing, ...



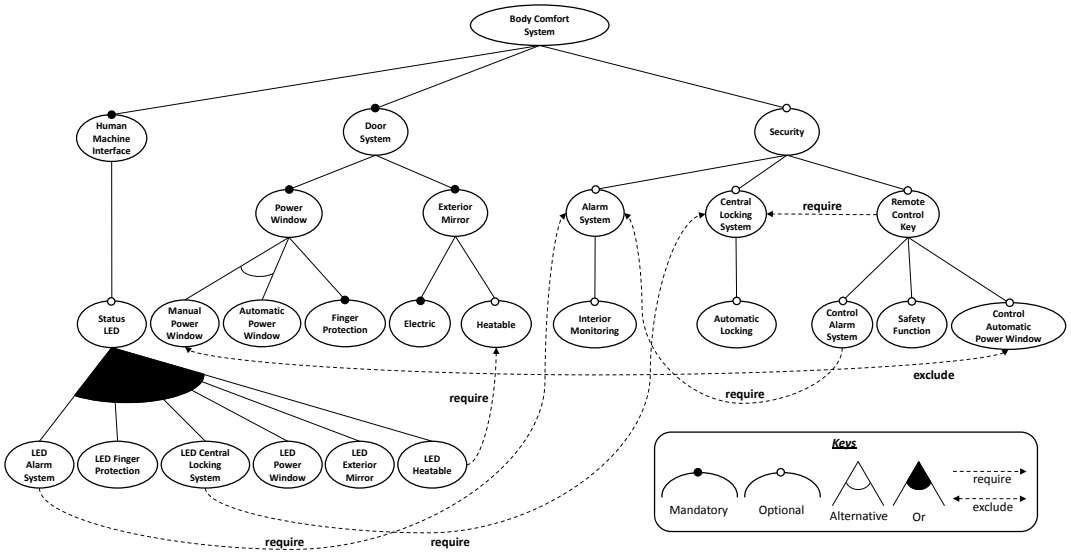
Implementation and Sample Tool Chain

- EMF-based Eclipse plug-in for modeling delta-oriented state machines
- Domain feature model for product configuration and derivation
- Test model import into IBM Rational Rhapsody
- Add-on ATG for automated test case generation and execution





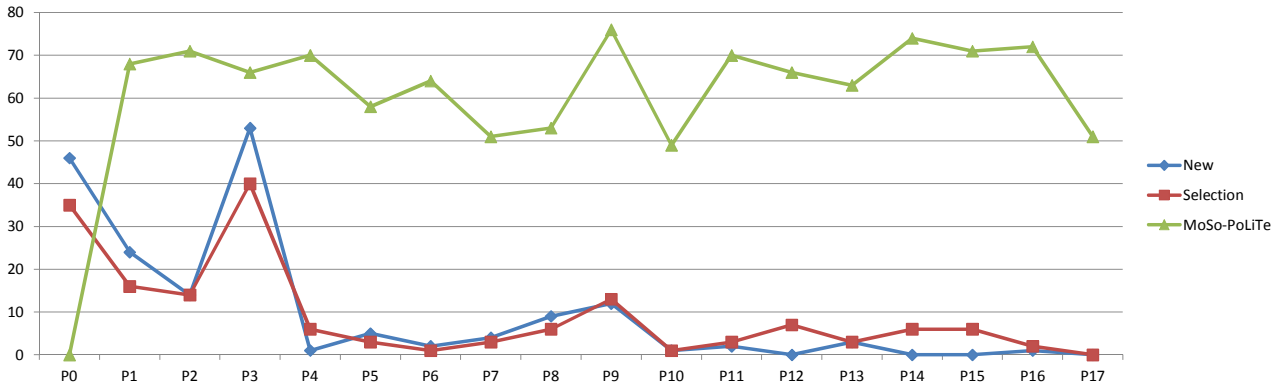
Evaluation



Automotive Case Study: Body Comfort System

- 26 features, 11616 product variants
- Delta test model with core model and 40 delta modules, 105 states and 107 transitions

Evaluation – Results



- Subset selection of 17 representative products-under-test for pairwise feature interaction coverage
- Product-by-product testing: 64 test cases generated and executed per product on average
- Incremental testing: 10 test cases generated per product, 9 selected for execution on average

Conclusions

- Efficient incremental model-based SPL testing approach based on regression delta analysis.
- Future Work:
  - Test goal evolution for path-oriented coverage criteria, e.g., MC/DC
  - Perform industrial case studies
  - Scalability: combination with delta-oriented architecture models

References I

Gilles Bernot.
Testing against formal specifications: A theoretical view.
In S. Abramsky and T. Maibaum, editors, *TAPSOFT '91*, volume 494 of *Lecture Notes in Computer Science*, pages 99–119. Springer Berlin , Heidelberg, 1991.
10.1007/3540539816 63.

D. Clarke, M. Helvensteijn, and I. Schaefer.
Abstract Delta Modeling.
Mathematical Structures in Computer Science, 2011.
(to appear).

R. DeNicola and M. C. B. Hennessy.
Testing equivalences for processes.
Theoretical Computer Science, pages 83–133, 1984.

E. M. Olimpiew.
Model-Based Testing for Software Product Lines.
PhD thesis, George Mason University, 2008.

References II

S. Oster, A. Wübbeke, G. Engels, and A. Schürr.
Model-Based Software Product Lines Testing Survey.
In *Model-based Testing for Embedded Systems*. 2010.
to appear.

S. Oster, I. Zorcic, F. Markert, and M. Lochau.
MoSo-PoLiTe - Tool Support for Pairwise and Model-Based Software Product Line Testing.
In *VAMOS'11*, 2011.

M. Utting and B. Legeard.
Practical Model-Based Testing. A Tools Approach.
M. Kaufmann, 2007.