Technische Universität Braunschweig

Institut für Programmierung und Reaktive Systeme

# Konzeption und Evaluation eines delta-orientierten modellbasierten Testverfahrens für Softwareproduktlinien

**Sascha Lity**

**Betreuer: Malte Lochau, Ina Schaefer**

34. TAV, Aachen, 14. Februar 2013

# Agenda

Motivation

Delta State Machines

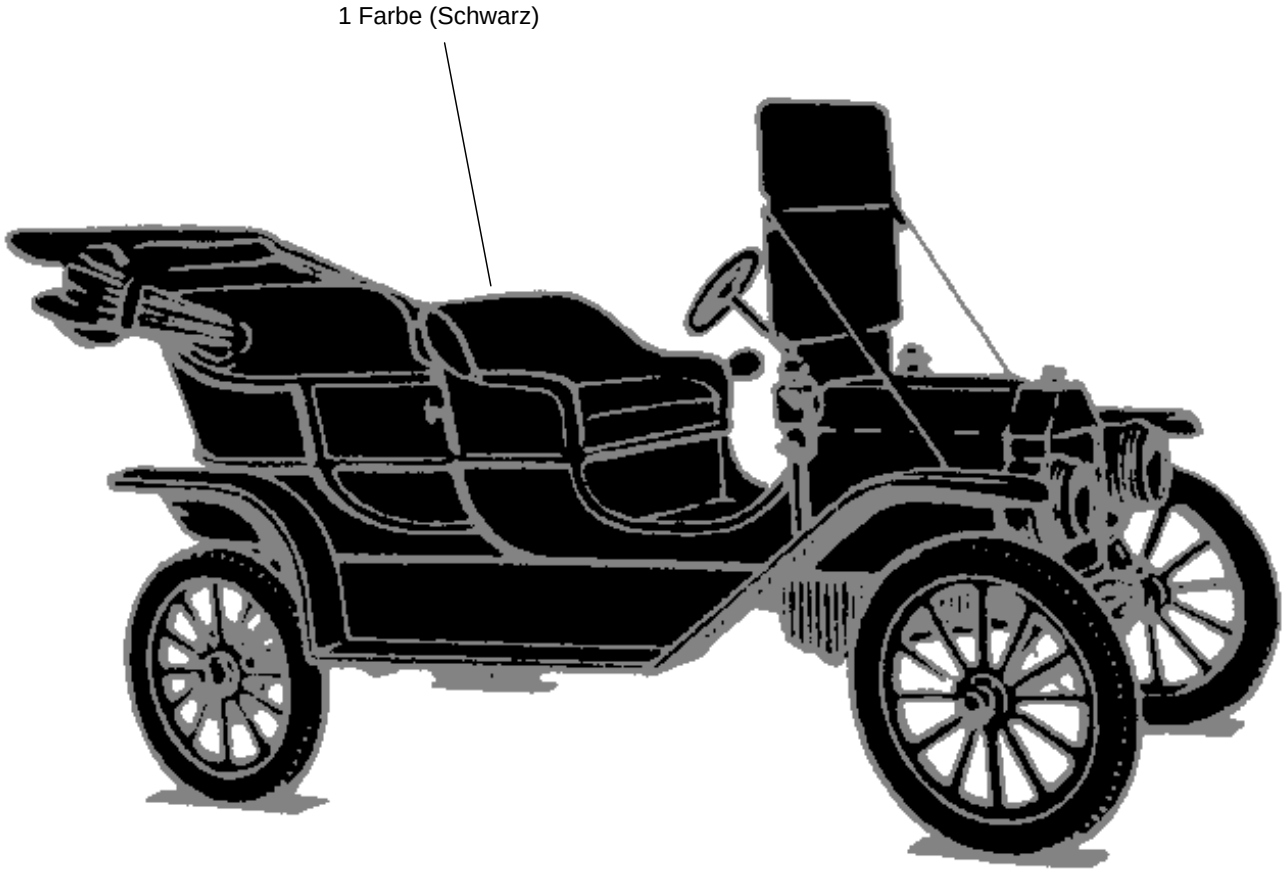
Delta-orientiertes modellbasiertes Testen

Werkzeugunterstützung eDeltaMBT

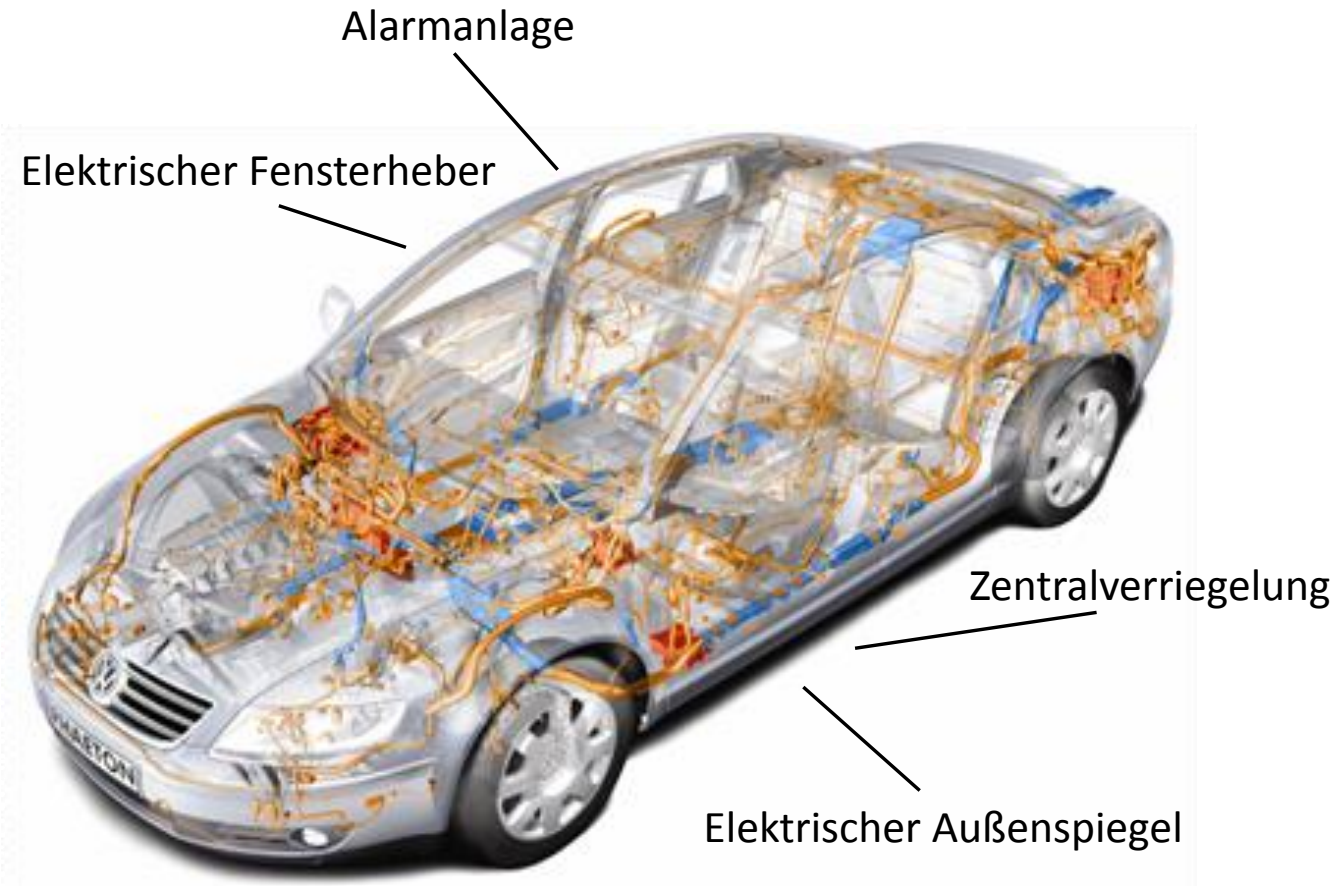
Evaluation

Zusammenfassung

# Motivation: Auto 1908



# Motivation: Auto 2011




Bildquelle: `http://www.verkehr.fraunhofer.de/de/automotive`

# Softwareproduktlinien

*[. . . ] explicit specification of commonality and variability between variants in a family of similar [software] products by means of features [PBL05].*

- Gemeinsame Domäne
- Gemeinsame Plattform
- Variabilität
- Systematische Wiederverwendung

Bildquelle: `http://www.rittersportolympia.de`

# Testen von Softwareproduktlinien

## Herausforderungen

- Variabilität
- Hohe Anzahl möglicher Produktvarianten
- Redundantes Testen
- Kosten

# Testen von Softwareproduktlinien

## Herausforderungen

- Variabilität
- Hohe Anzahl möglicher Produktvarianten
- Redundantes Testen
- Kosten

⇒ Single-Software-Testansätze nicht praktikabel [OMR10, LOGS]

# Testen von Softwareproduktlinien

## Herausforderungen

- Variabilität
- Hohe Anzahl möglicher Produktvarianten
- Redundantes Testen
- Kosten

⇒ Single-Software-Testansätze nicht praktikabel [OMR10, LOGS]

## SPL-Testphilosophien [OMR10]

- Contra-SPL-Philosophy
- Subset-Heuristics
- Reuse-Techniques

# Ziele der Arbeit

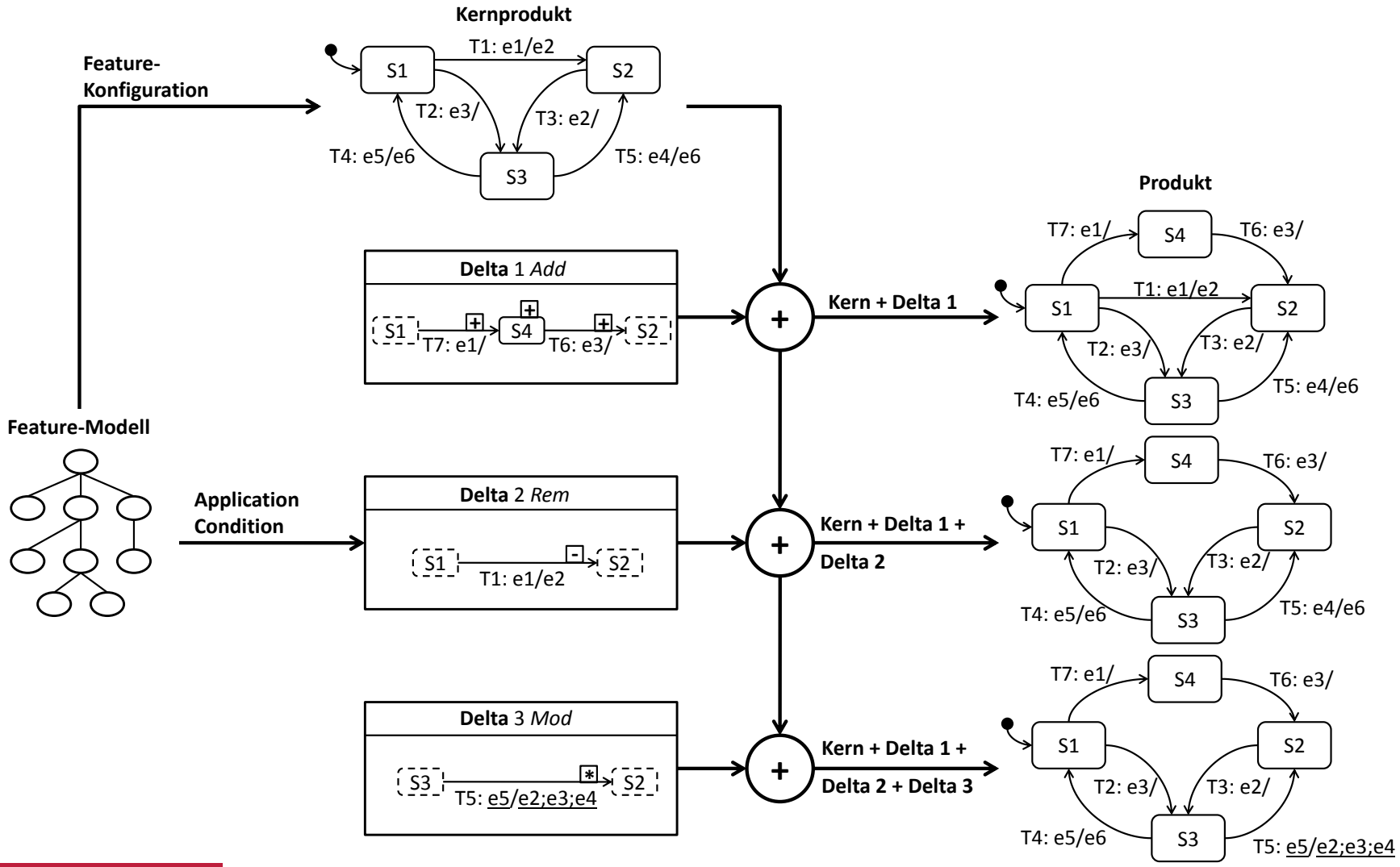
- Konzeption eines Delta-orientierten Modellbasierten Testverfahrens
  - Kombination der SPL-Testphilosophien [OMR10]
    - Subset-Heuristics
    - Reuse-Techniques
  - Kombination der Testtechniken
    - Modellbasiertes Testen [BJK+05]
    - Regressionstesten [Eng10]
- Werkzeugunterstützung
  - Modellierung und Generierung von variablen Testmodellen
  - Generierung von Testfällen
- Evaluation des Testverfahrens anhand einer Fallstudie

# Delta State Machines

- Adaption der Delta-Modellierung [Sch10] für State Machines
- Transformationeller Ansatz
- Modellierung und Generierung von variablen State Machines
  - Anwendung bei der Produktentwicklung (Code-Generierung)
  - Anwendung als Testmodell
- Deltas definieren Transformationen
  - Hinzufügen und Entfernen von
    - Zuständen
    - Transitionen
  - Modifikation von Transitionsbeschriftungen

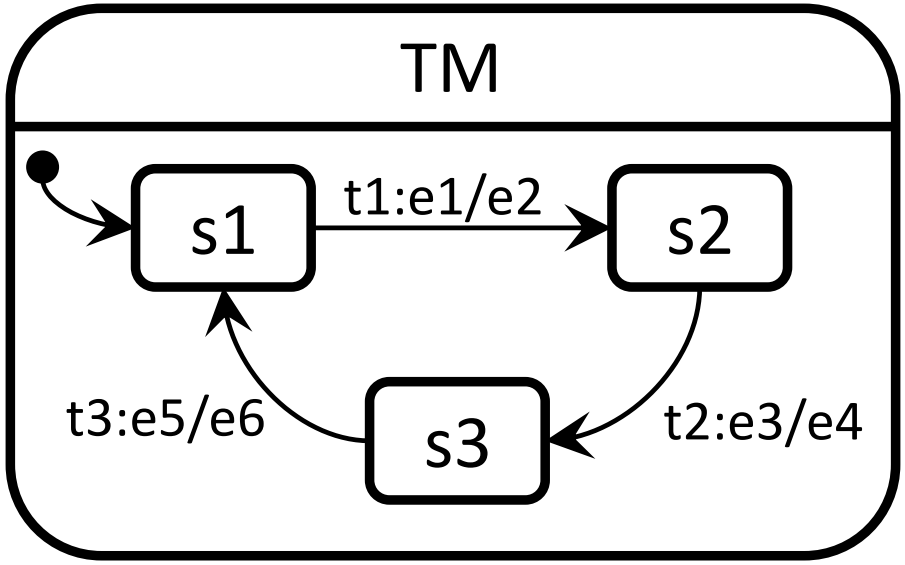
$\delta$

# Anwendung der Delta State Machines



# Testmodell

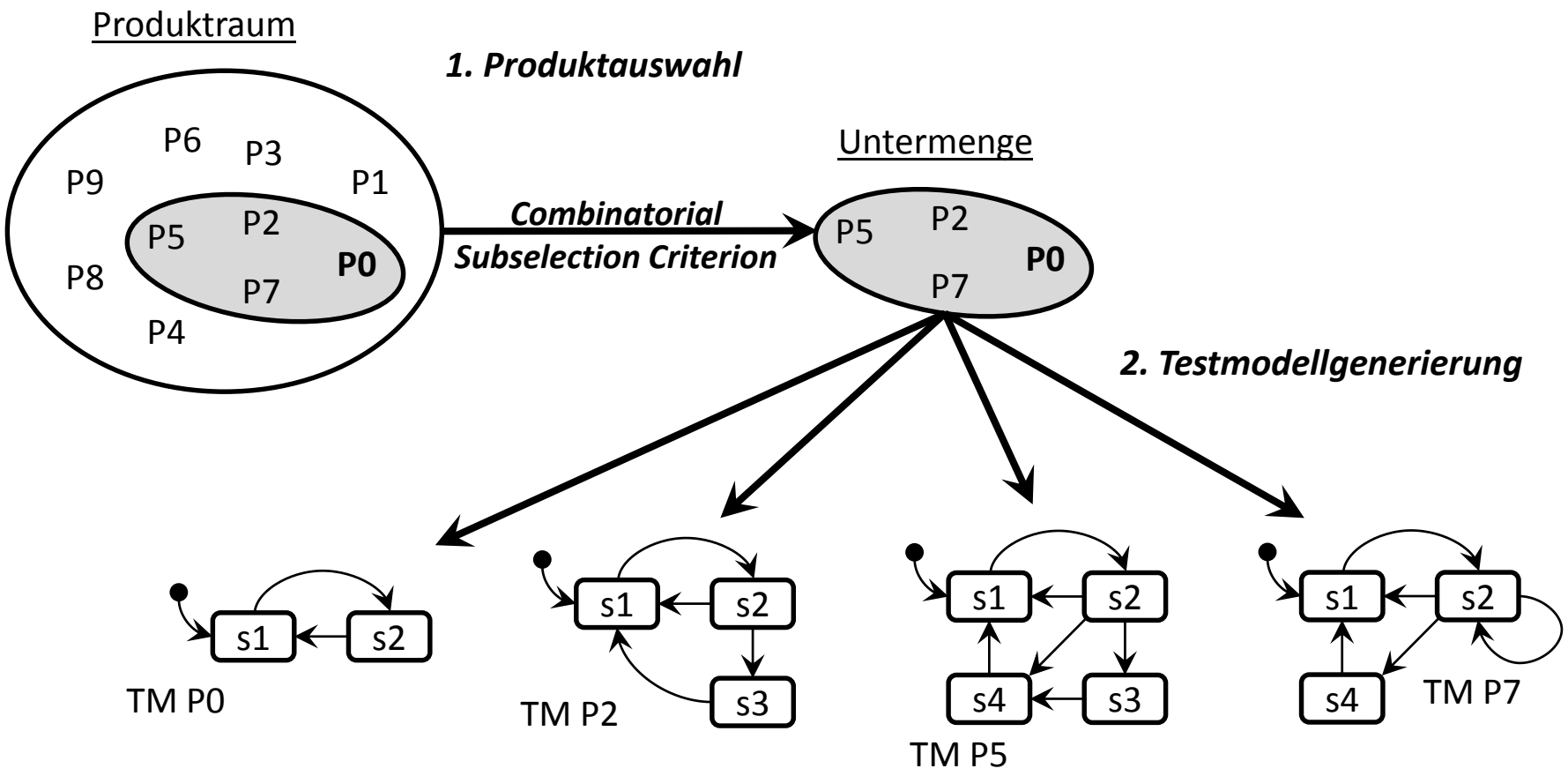
- Deterministische State Machine als Testmodell
- Testfälle besitzen einen Testlauf im Testmodell
- Abdeckungskriterium für Definition von Testzielen
- Testsuite fasst Testfälle zusammen



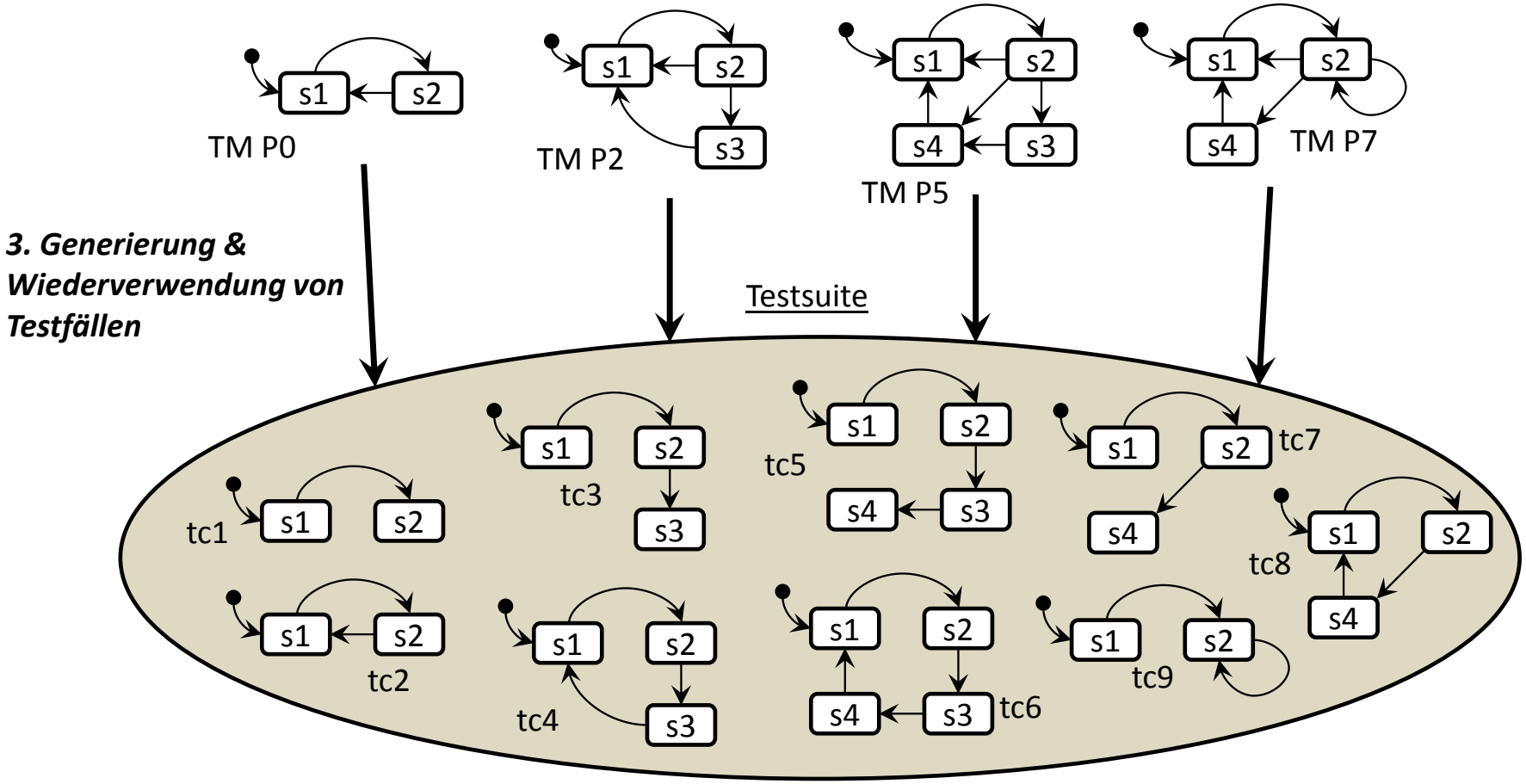
# Teststrategie

- Ermöglicht inkrementellen Testprozess
- **Wiederverwendung** von Testfällen und Testergebnissen
- Ziel: Reduzierung des Testaufwands
- Definition von 5 Aktivitäten
  1. Produktauswahl (Oster et al. [OMR10])
  2. Testmodellgenerierung
  3. Generierung und Wiederverwendung von Testfällen (Cichos et al. [COLS11])
  4. Testfallkategorisierung
  5. Testfallselektion (Cichos et al. [CLOS12])

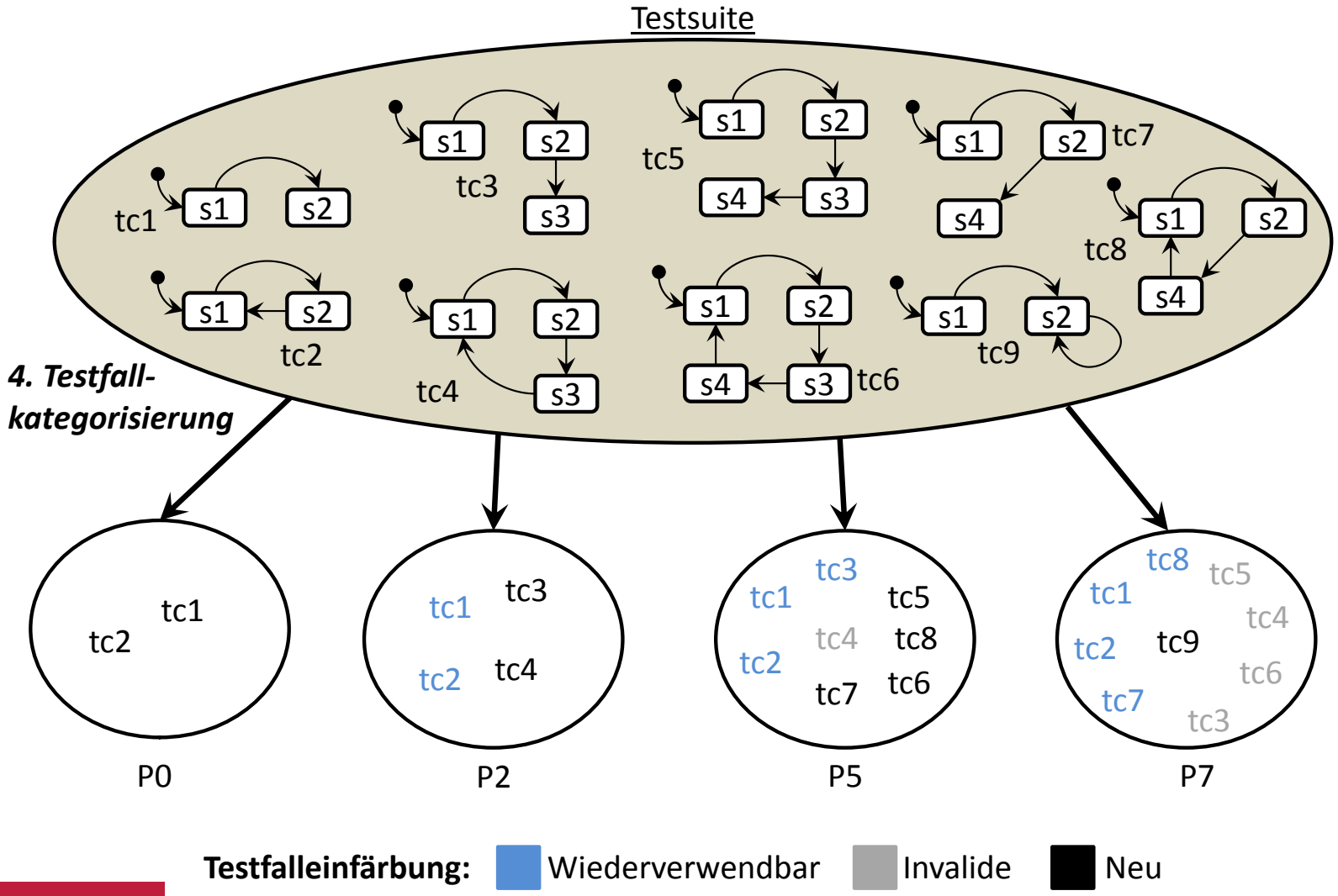
# Produktauswahl und Testmodellgenerierung



# Generierung und Wiederverwendung von Testfällen

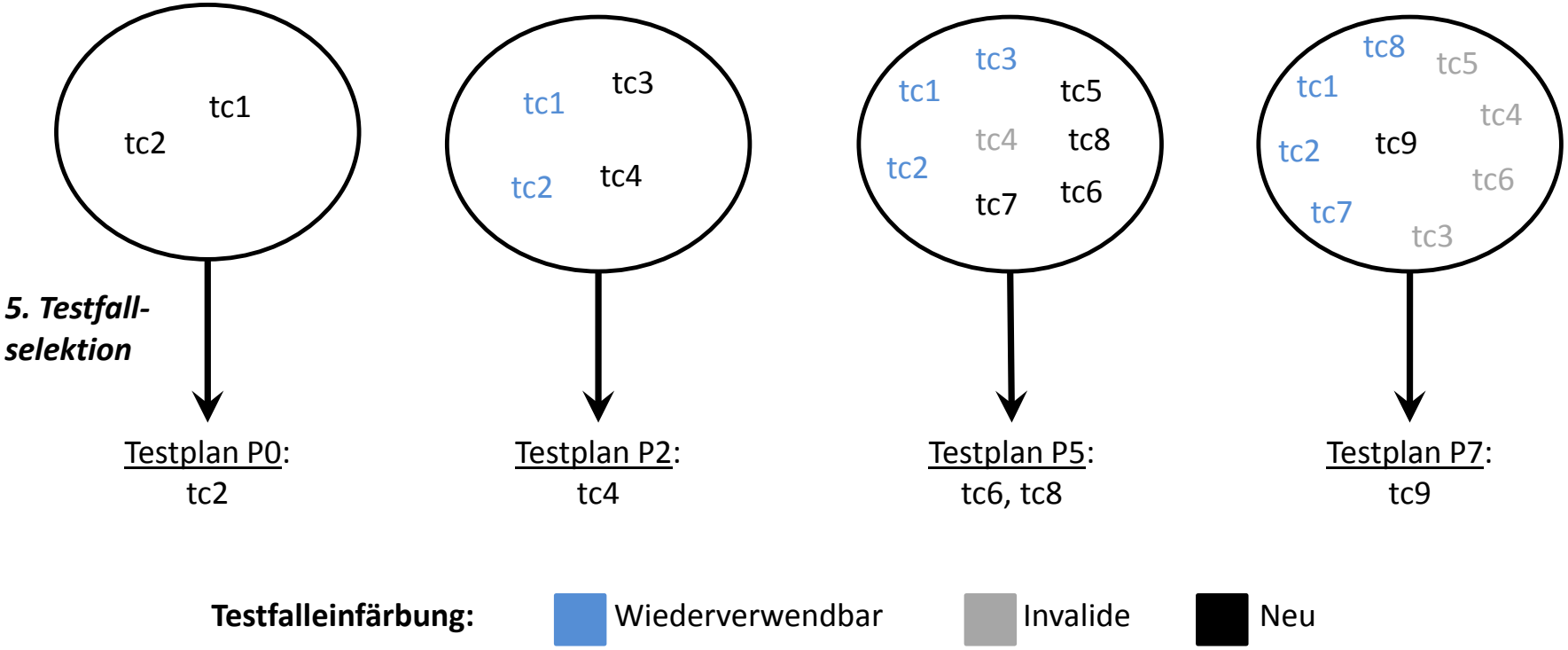


# Testfallkategorisierung

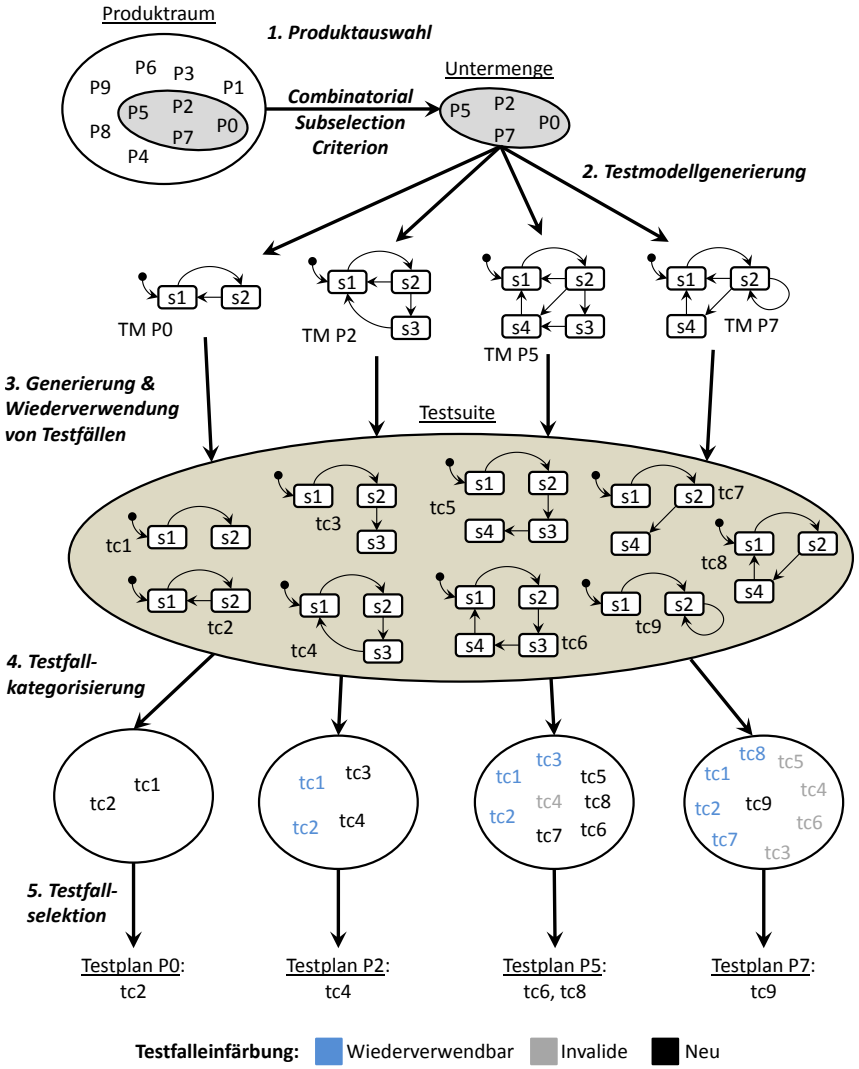



**Testfalleinfärbung:** Wiederverwendbar Invalide Neu

# Testfallselektion



# Testverfahren



# Werkzeugunterstützung

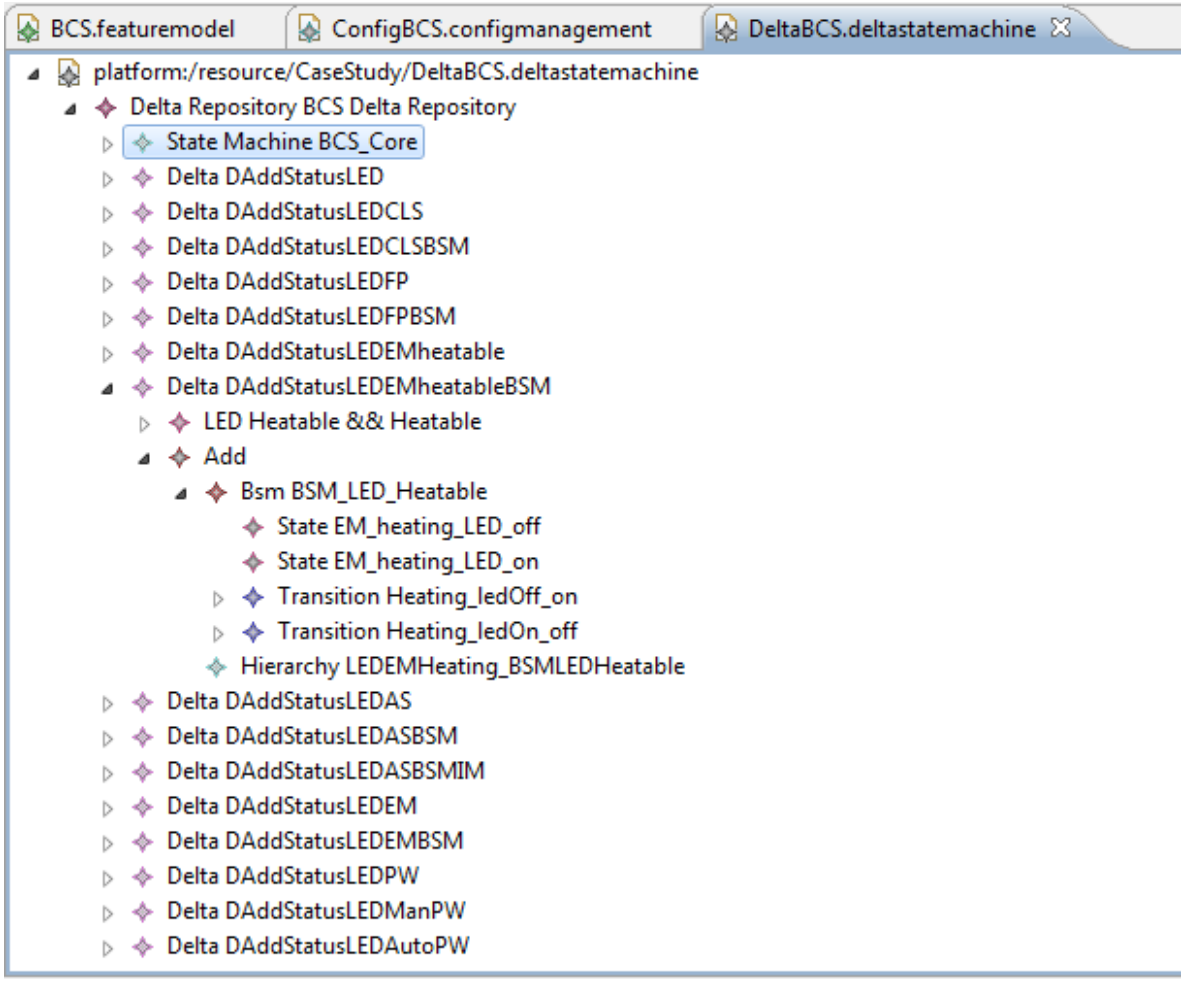
Ziele:

- Modellierung von Delta State Machines
- Generierung von variablen Testmodellen
- Automatische Testfallgenerierung

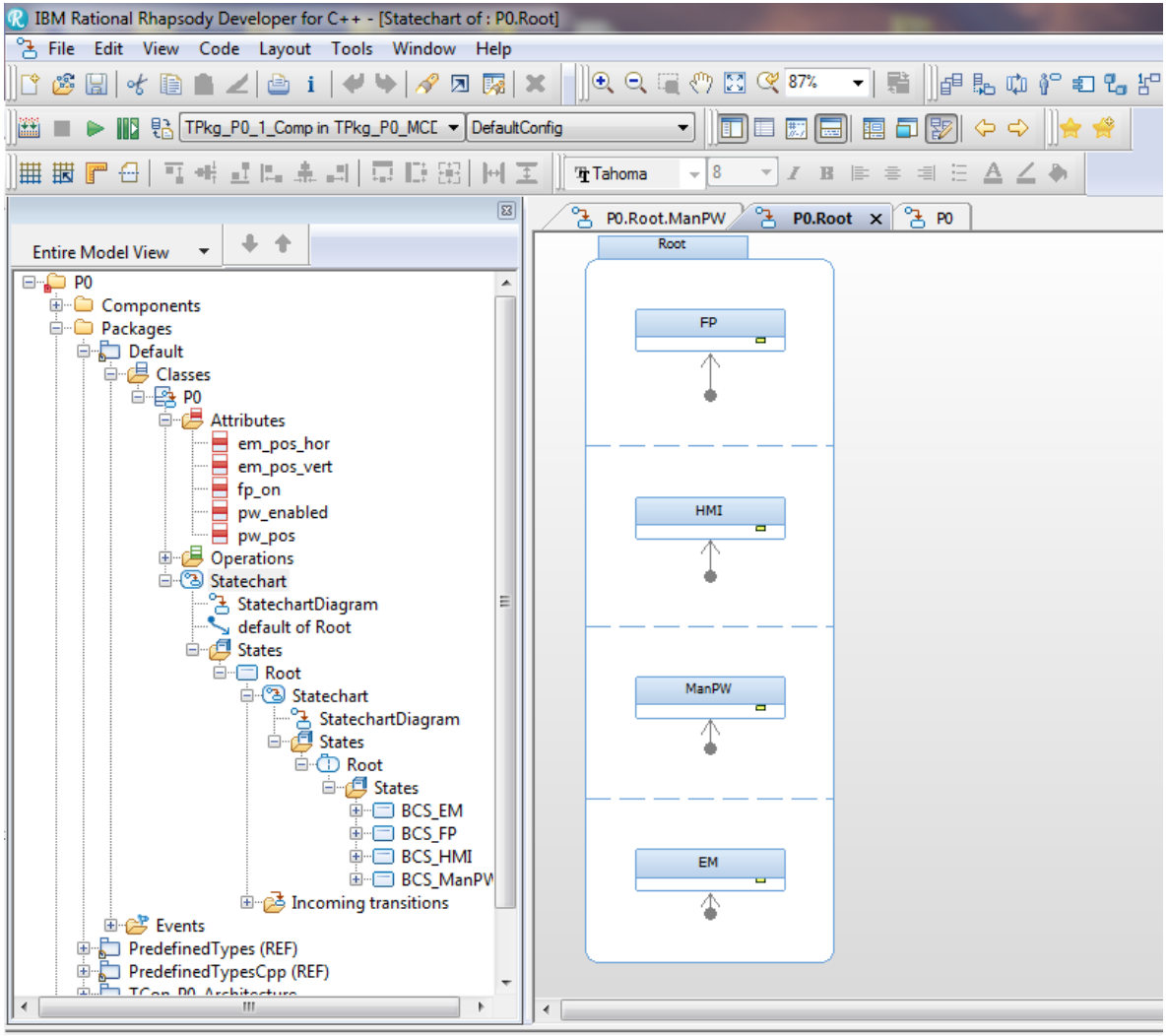
Umsetzung:

- Realisierung eines Eclipse Plug-ins
- Modellierung von
  - Feature-Modellen
  - Feature-Konfigurationen
  - Delta State Machines
- Generierung von variablen Testmodellen
- Anbindung an IBM Rational Rhapsody für die Testfallgenerierung

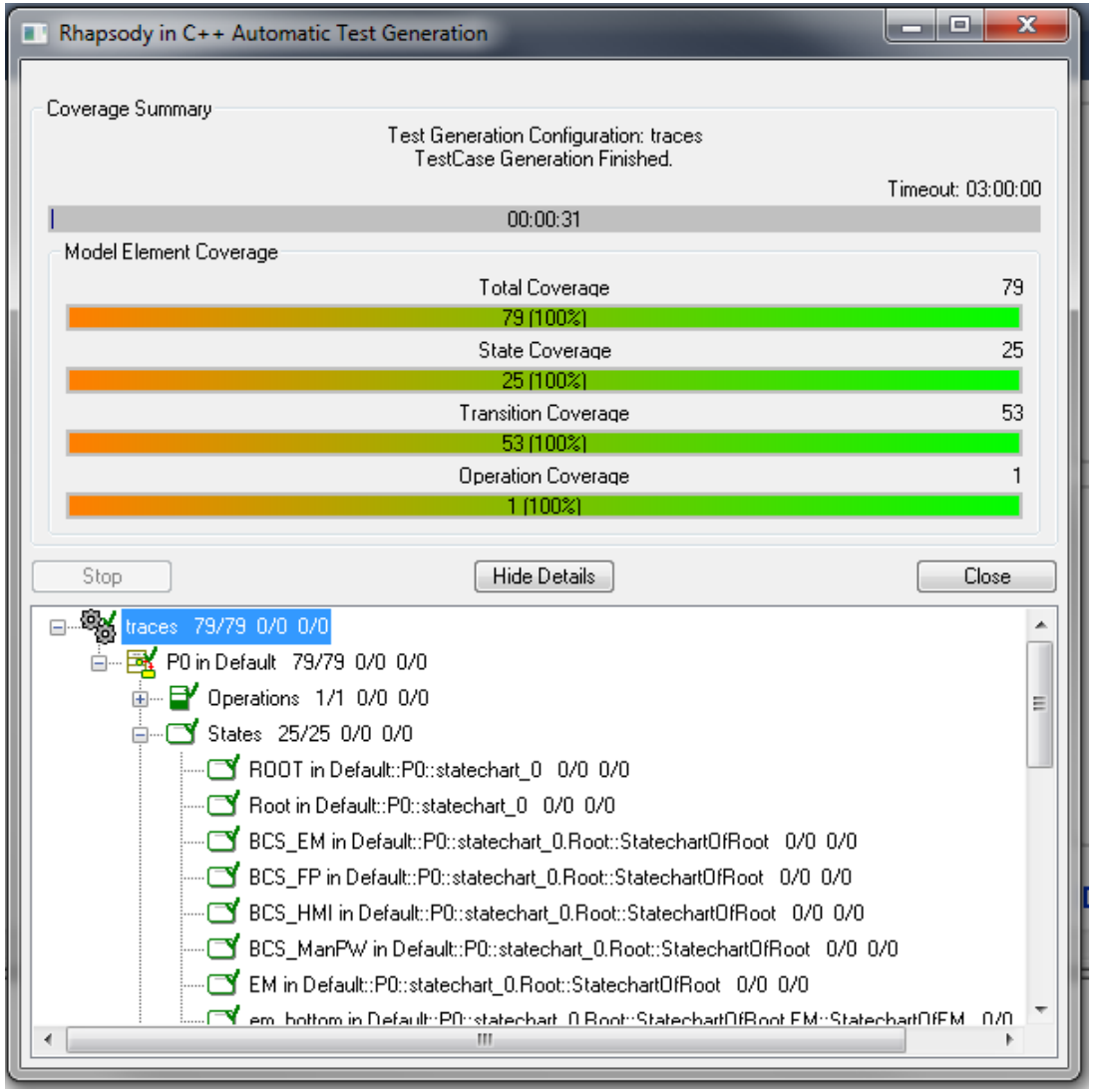
# Modellierung mit Delta State Machines



# Testmodell in Rhapsody



# Testfallgenerierung mit ATG



# Evaluation des Testverfahrens

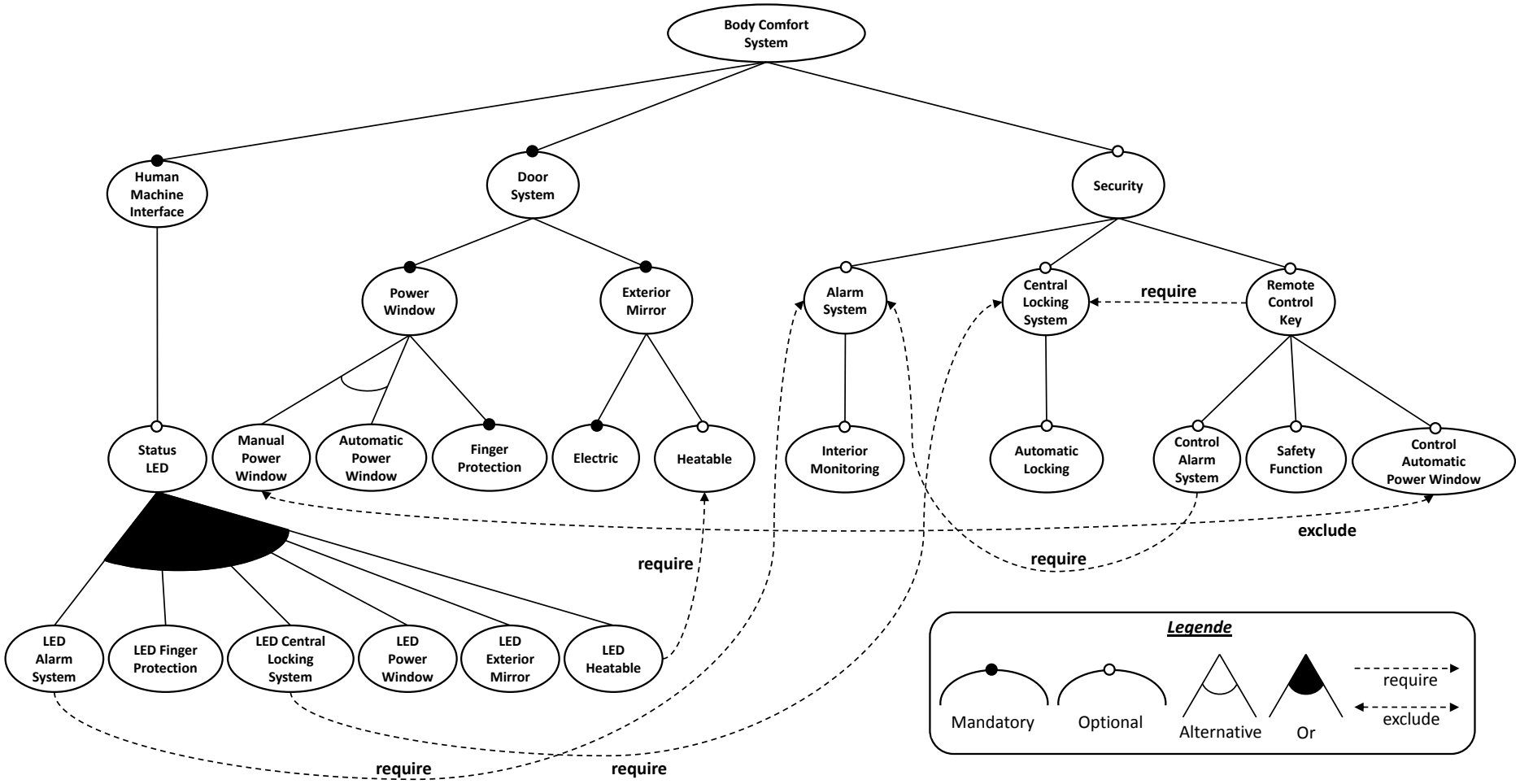
Ziele der Evaluation:

- Anwendung der Delta State Machines
- Überprüfung der Anwendbarkeit des konzipierten Testverfahrens
- Vergleich der Testergebnisse mit den Ergebnissen des MoSo-PoLiTe [OZLG11, Zin11] Testansatzes

Umsetzung der Fallstudie:

- Erweiterung der Implementierung
- Fallbeispiel duch Delta State Machines nachmodelliert
- Testverfahren angewendet und Ergebnisse ausgewertet

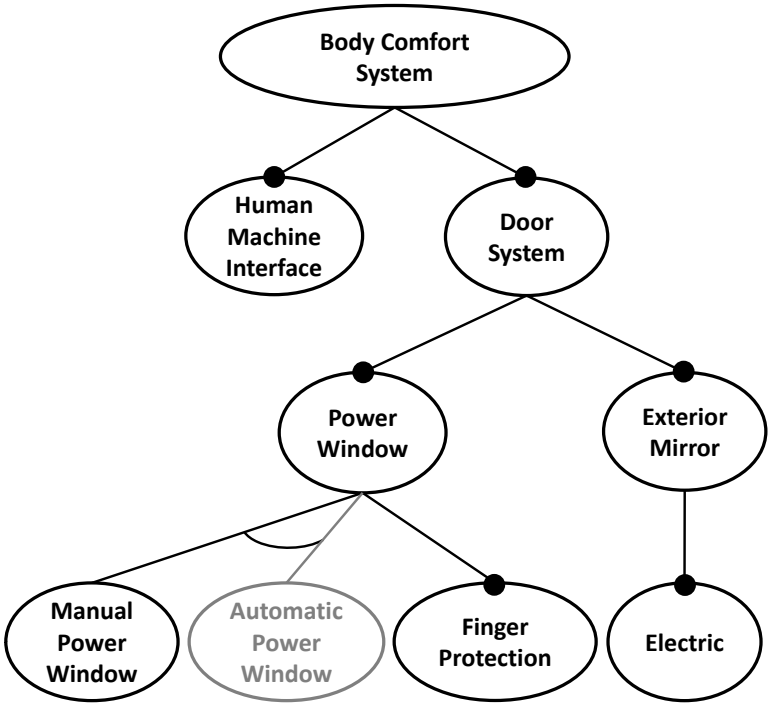
# Body Comfort System



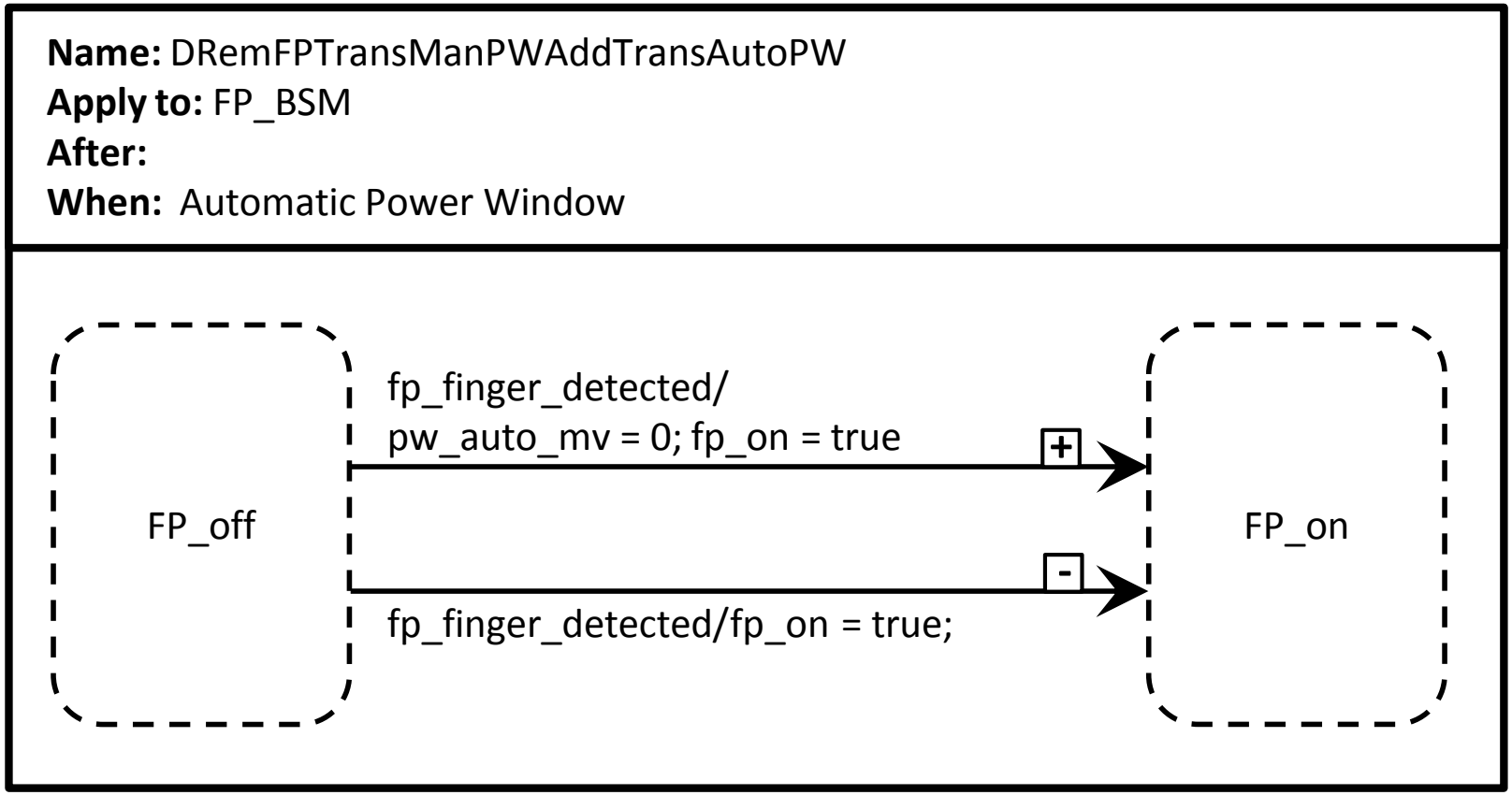
# Ergebnisse (1/4)

| | Anzahl |
|---|---|
| Kern | 1 |
| Delta | 40 |

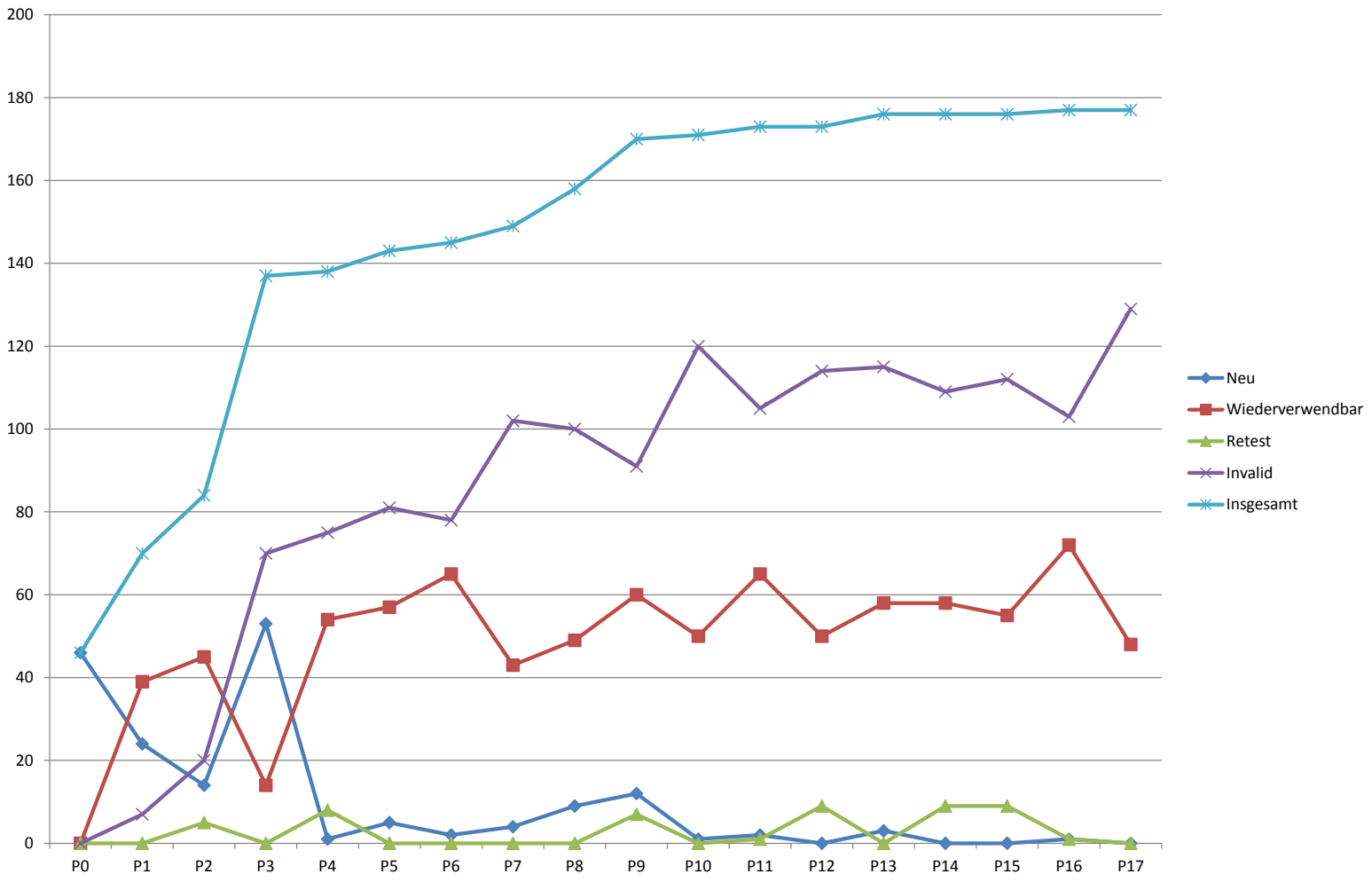
| | Anzahl |
|---|---|
| Hierarchie | 8 |
| Basic State Machine | 9 |
| Zustände | 20 |
| Transitionen | 44 |



# Ergebnisse (2/4)

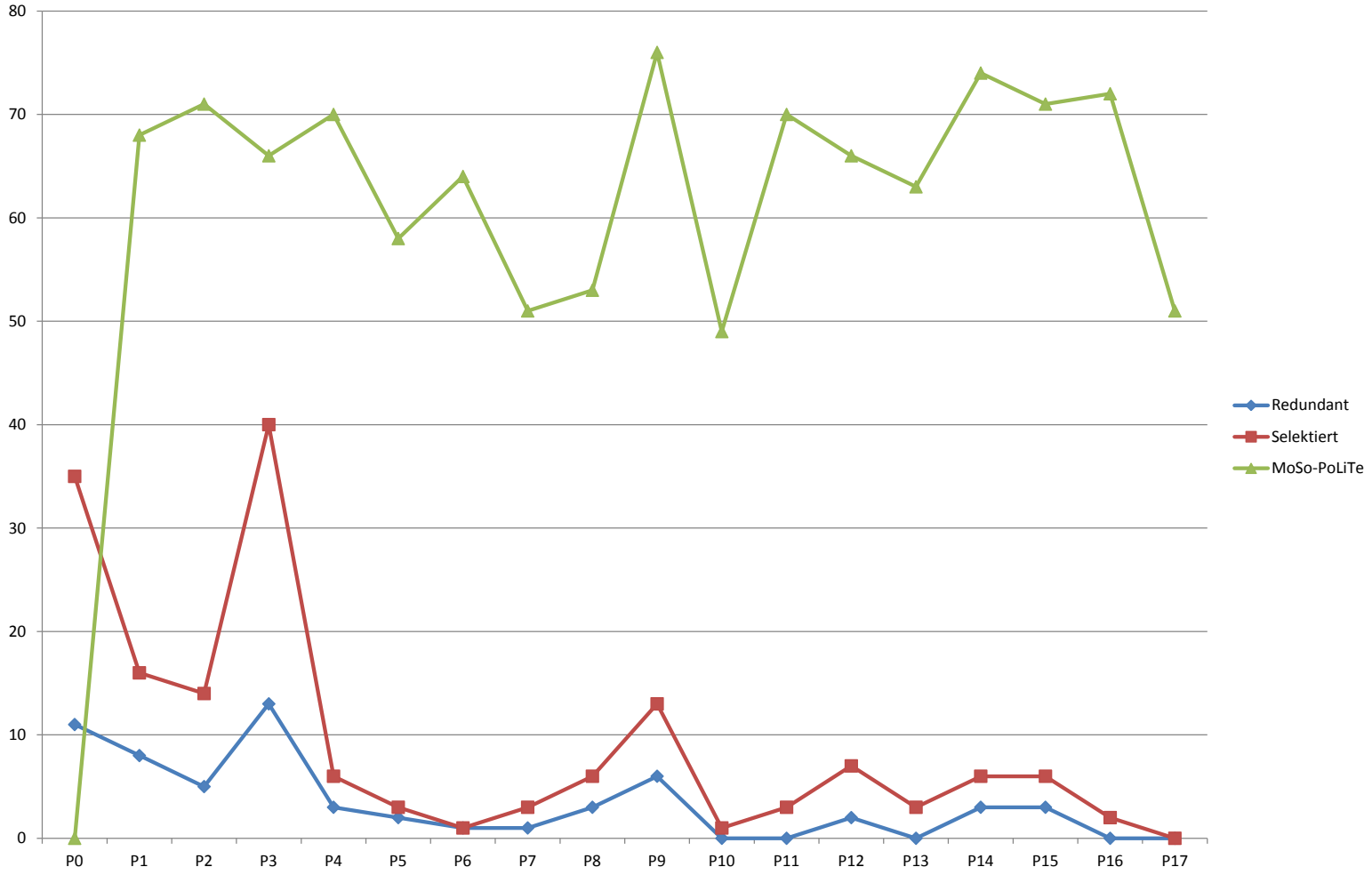


# Ergebnisse (3/4)




Ergebnisse der Testfallkategorisierung

# Ergebnisse (4/4)




Ergebnisse der Testfallselektion

# Zusammenfassung

- Herausforderungen des Testens von Produktlinien angehen
- Adaption der Delta-Modellierung für State Machines zur Modellierung und Generierung von variablen Testmodellen
- Delta-orientiertes modellbasiertes Testverfahren durch Kombination von SPL-Testphilosophien und Testtechniken
- Werkzeugunterstützung durch ein Eclipse Plug-in
- Evaluation anhand einer Komfortsystem-Produktlinie

# Zusammenfassung

- Herausforderungen des Testens von Produktlinien angehen
- Adaption der Delta-Modellierung für State Machines zur Modellierung und Generierung von variablen Testmodellen
- Delta-orientiertes modellbasiertes Testverfahren durch Kombination von SPL-Testphilosophien und Testtechniken
- Werkzeugunterstützung durch ein Eclipse Plug-in
- Evaluation anhand einer Komfortsystem-Produktlinie

$\Rightarrow$ Vielversprechender Ansatz, der als Grundlage verwendet werden kann, um die Herausforderungen anzugehen

# Ausblick

- Evaluation durch weitere Softwareproduktlinien
- Erweiterung der Delta State Machines
- Adaption der Delta-Modellierung für weitere Modellierungssprachen
- Erweiterung der Implementierung

Danke für Ihre Aufmerksamkeit. Gibt es noch Fragen?

TU Braunschweig, Institut für Programmierung und Reaktive Systeme
Sascha Lity
*Email*: lity@ips.cs.tu-bs.de

# Referenzen I


Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, and Alexander Pretschner.
*Model-Based Testing of Reactive Systems: Advanced Lectures (Lecture Notes in Computer Science)*.
Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

Harald Cichos, Malte Lochau, Sebastian Oster, and Andy Schürr.
Reduktion von testsuiten für software-produktlinien.
erscheint 2012.

Harald Cichos, Sebastian Oster, Malte Lochau, and Andy Schürr.
Model-based coverage-driven test suite generation for software product lines.
In *Proceedings of the 14th international conference on Model driven engineering languages and systems*, MODELS'11, pages 425–439, Berlin, Heidelberg, 2011. Springer-Verlag.

Emelie Engström.
*Exploring Regression testing and software product line testing - research and state of practice*.
Lic dissertation, Lund University, May 2010.

# Referenzen II


Malte Lochau, Sebastian Oster, Ursula Goltz, and Andy Schürr.
Model-based pairwise testing for feature interaction coverage in software product line engineering.
*Software Quality Journal*, pages 1–38.
10.1007/s11219-011-9165-4.

Sebastian Oster, Florian Markert, and Philipp Ritter.
Automated incremental pairwise testing of software product lines.
In *Proceedings of the 14th international conference on Software product lines: going beyond*, SPLC'10, pages 196–210, Berlin, Heidelberg, 2010. Springer-Verlag.

Sebastian Oster, Marius Zink, Malte Lochau, and Mark Grechanik.
Pairwise feature-interaction testing for spls: potentials and limitations.
In *Proceedings of the 15th International Software Product Line Conference*, SPLC '11, pages 6:1–6:8, New York, NY, USA, 2011. ACM.

Klaus Pohl, Günter Böckle, and Frank J. van der Linden.
*Software Product Line Engineering: Foundations, Principles and Techniques*.
Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

# Referenzen III

Ina Schaefer.
Variability modeling for model-driven development of software product lines.
*Fourth International Workshop on Variability Modelling of Software-intensive Systems (VaMoS 2010)*, January 2010.

Marius Zink.
Anwendung von moso-polite in einer automotive spl.
Masterarbeit, Technische Universität Darmstadt, June 2011.