DOPSY group

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Testen von Echtzeiteigenschaften für verteilte Ablaufsteuerungen

M. Jurisch, K. Beckmann

Labor für Verteilte Systeme

Hochschule RheinMain

Echtzeit 2015

Matthias Jurisch
Kai Beckmann
{Vorn.Nachn}@hs-rm.de

Labor für Verteilte Systeme
Distributed Systems Lab

http://wwwvs.cs.hs-rm.de

Agenda

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Motivation
- Background
- Problembeschreibung
- Abstrakte Events
- Bewertung
- Fazit und Ausblick

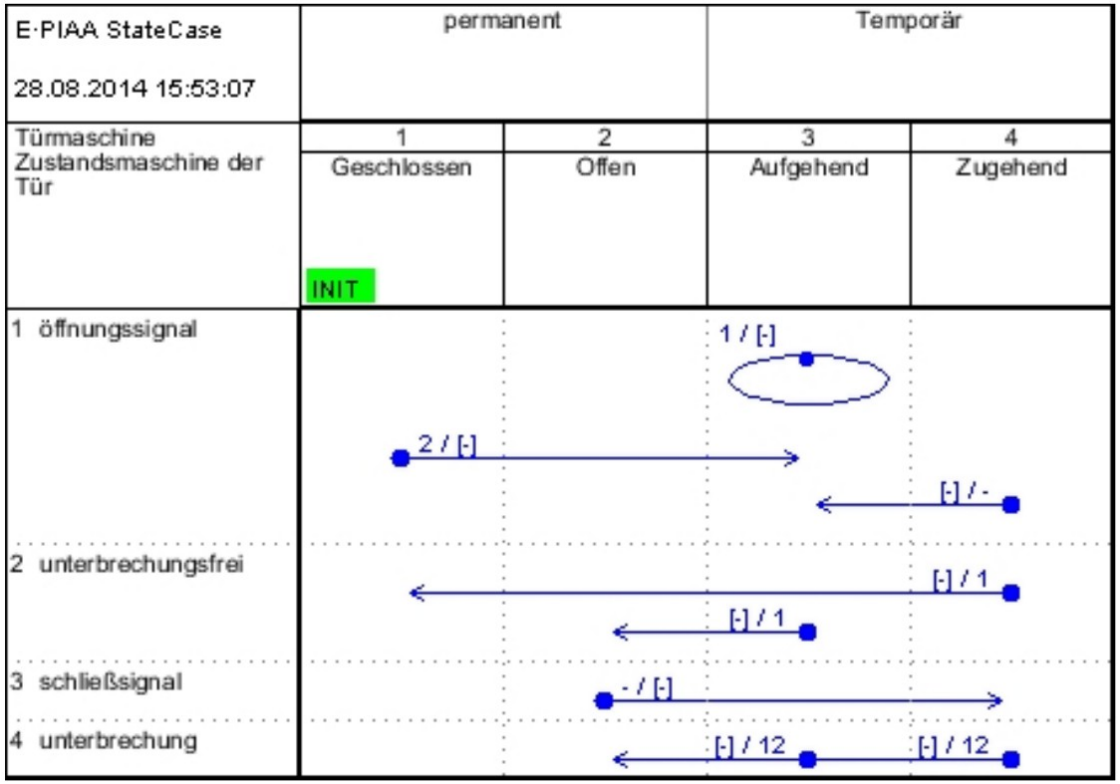
Motivation

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Software-Qualität in der Automatisierungsbranche
  - Entscheidend für Erfolg von Unternehmen
  - Aber: Kosten für moderne Verfahren oft zu hoch für KMUs
  - Bedarf nach kostengünstiger, an Problemdomäne angepasster Lösung
- Projekt „Testframework für Automatisierungsanwendungen“
  - Kooperation zwischen Hochschule RheinMain und Eckelmann AG aus Wiesbaden
  - Modellgetriebenes Testen ermöglichen
    - Tests werden modelliert
    - Modell ist Grundlage der Testausführung
  - Anpassung an Umgebung der Tester
  - Integration in Entwicklungsprozess
  - Mittel: Erstellung eines Testframeworks

StateCase

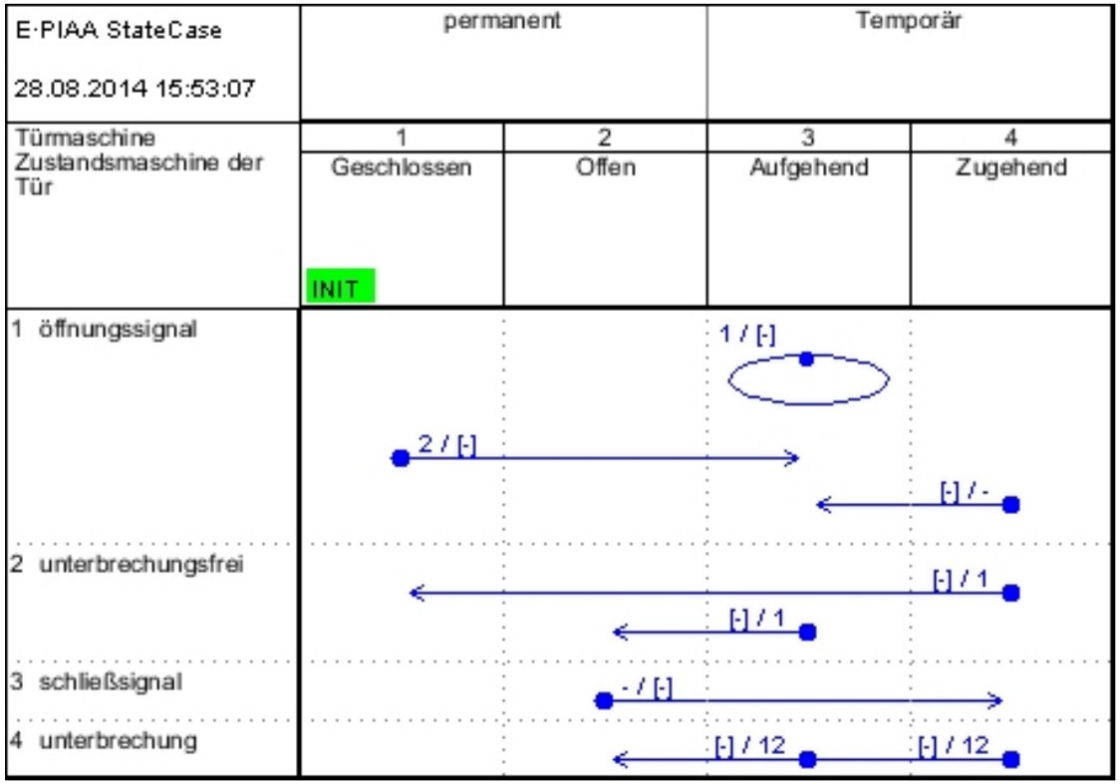
Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim



- Formulierung von „Sequential Function Tables“(SFTs)
- Intern entwickeltes Werkzeug von Eckelmann
- Wird im Unternehmen sehr breit eingesetzt
- Viel Know-How in Entwicklung des Werkzeugs gesteckt
- Codegenerierung aus SFTs
  - Verschiedene Sprachen
  - Erzeugt Skelett des Automaten

Problem

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim



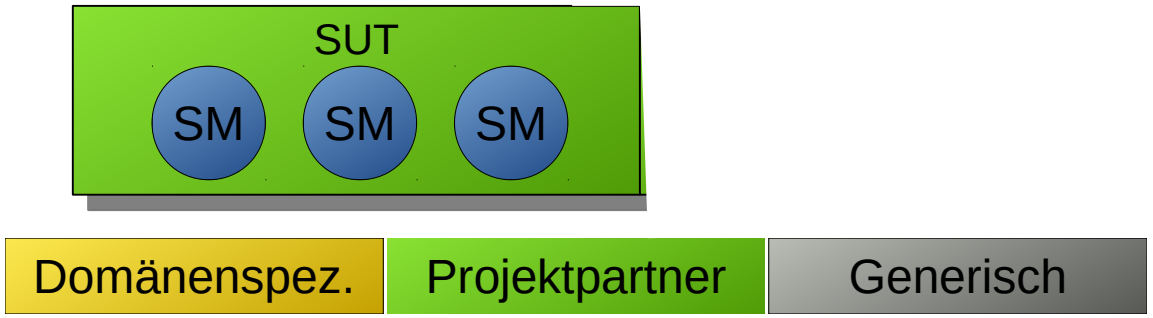
- Typisches System besteht aus mehreren Automaten
- Keine Beziehungen zwischen Automaten modellierbar
- Beim Füllen der Lücken im Template können Fehler auftreten
- Automatisiertes Testen Notwendig
- Wiederverwerten bestehender Modelle
- Formulieren von gültigen Ausführungspfaden als Pfadausdrücke

Das Testframework

Hochschule **RheinMain**
University of Applied Sciences
Wiesbaden Rüsselsheim

- Tool zum Testen von mit StateCase modellierten Systemen

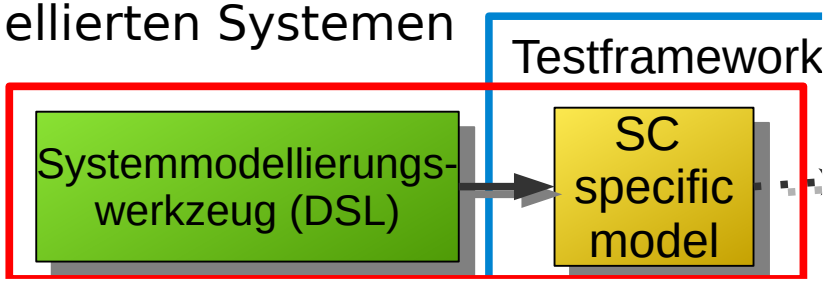
Systemmodellierungs-werkzeug (DSL)



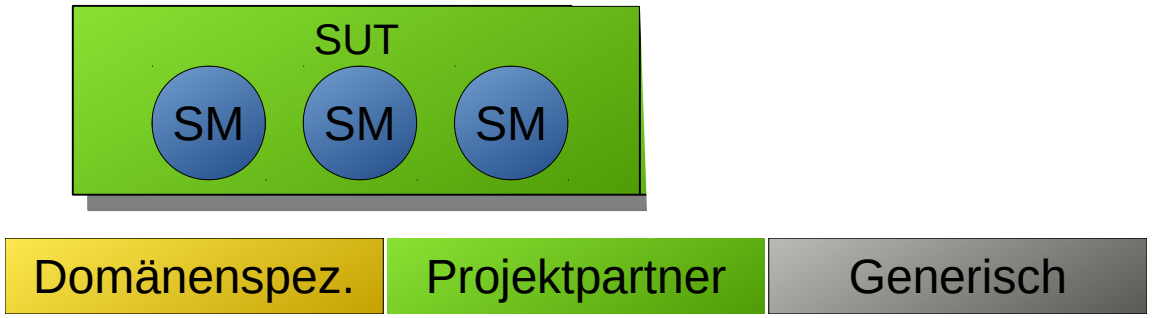
Das Testframework

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Tool zum Testen von mit StateCase modellierten Systemen



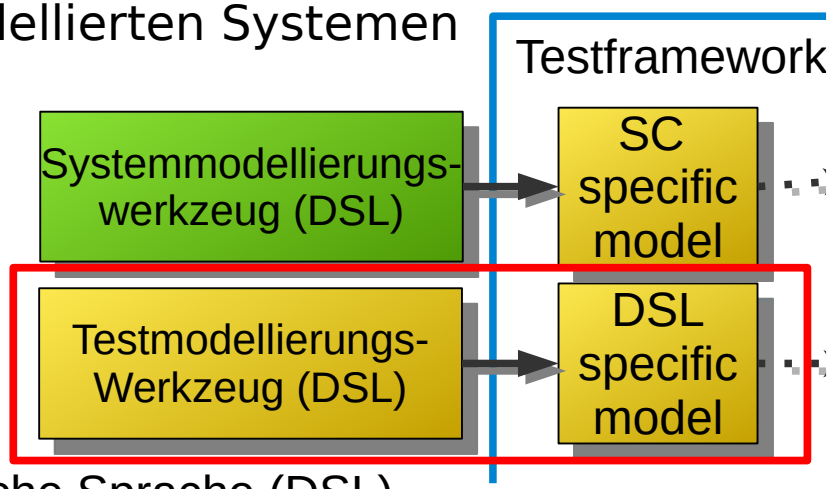
- Import der Automaten aus StateCase
- Verwendung von Modelltransformationen



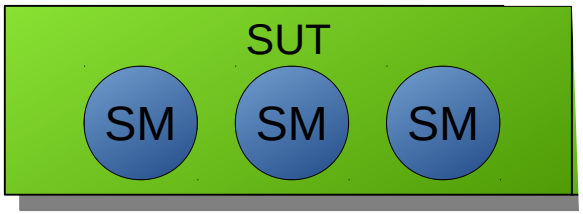
Das Testframework

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Tool zum Testen von mit StateCase modellierten Systemen



- Domänenspezifische Sprache (DSL) zur Formulierung von Tests
  - An Bedürfnisse von Testern angepasst
- „Globale Pfade“
  - Formulierung von Zustandsfolgen
  - Über mehrere Automaten



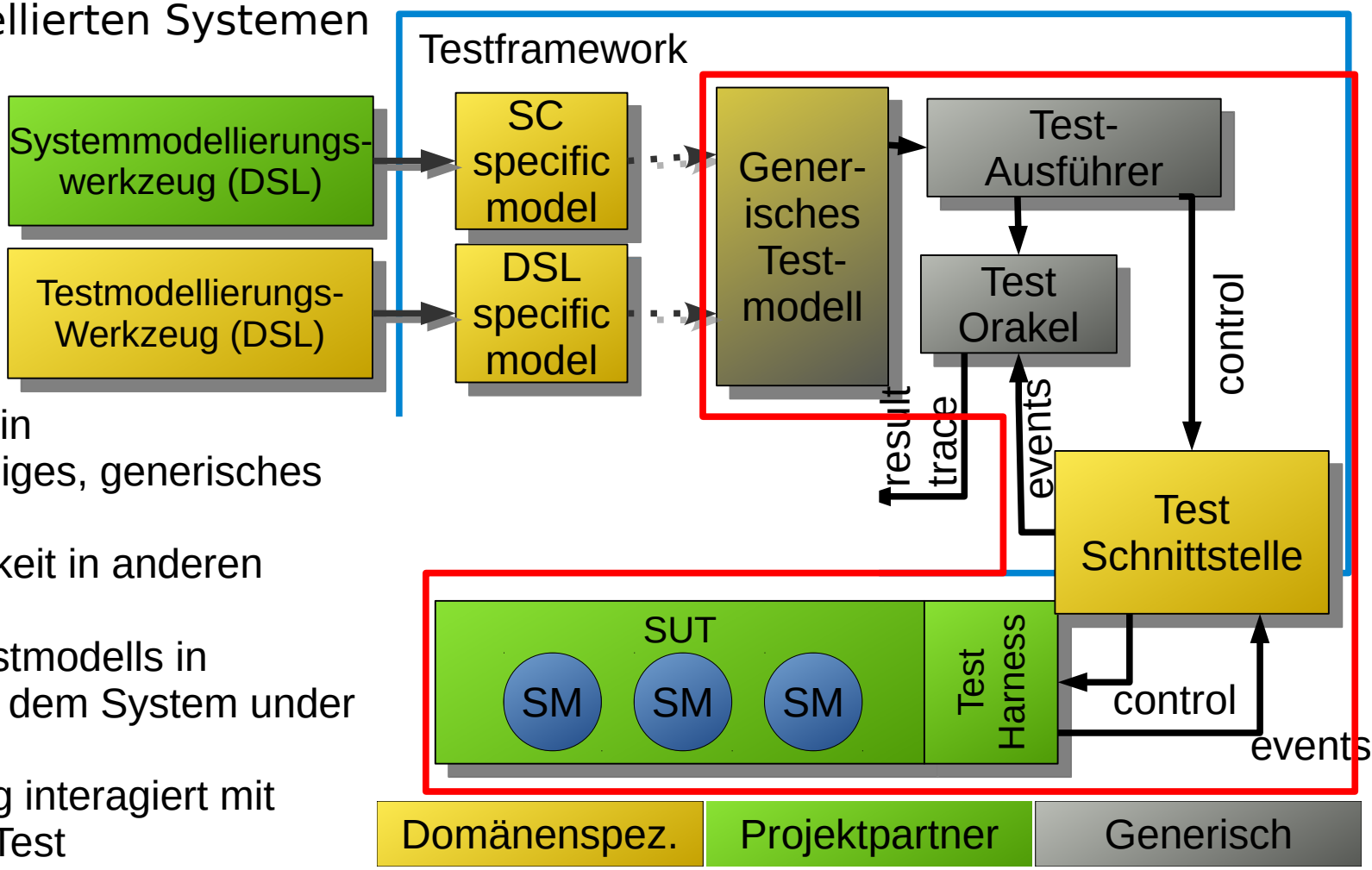
Beispiel

```
Schalter.Gedrueckt ->
Tuer.Aufgehend -> * ->
Tuer.Offen
```

Das Testframework

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Tool zum Testen von mit StateCase modellierten Systemen

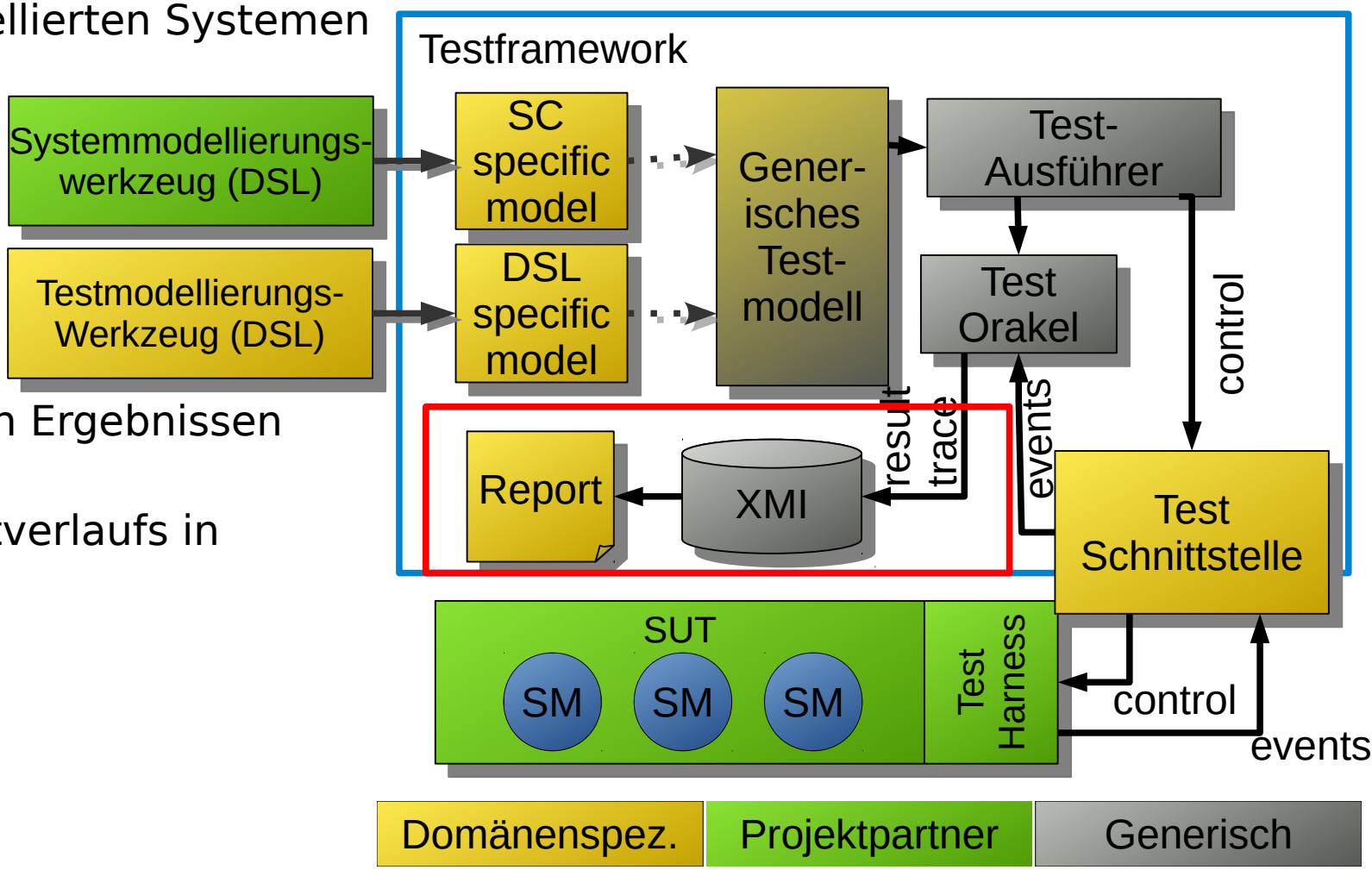


- Transformation in ein domänenunabhängiges, generisches Testmodell
- Wiederverwendbarkeit in anderen Domänen
- Ausführung des Testmodells in Kommunikation mit dem System under Test (SUT)
  - Testauswertung interagiert mit System under Test
  - Verarbeitung des Systemverhaltens im laufenden Betrieb

Das Testframework

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Tool zum Testen von mit StateCase modellierten Systemen
- Bereitstellung von Ergebnissen als Testreport
- Abbilden des Testverlaufs in einem Testtrace



Ziele

- Bisher nicht überprüfbar:
  - Zeitliche Anforderungen
  - Besonders in technischen Systemen relevant
- Ziel:
  - Testen von zeitlichen Anforderungen
  - Formulierung auf SFT-Ebene
  - Integriert mit bisherigen Komponenten des Testframeworks
  - Testen von verteilten SFTs
- Relevante Ereignisse:
  - Betreten, Verlassen von Zuständen
  - Auftreten von Transitionen
  - Betrachtung von verteilten Zuständen
  - „Globale Pfade”

Related Work

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- UML Testing Profile (UTP)
  - Allgemeines UML-Profil für Testmodellierung
  - Nur sehr rudimentäres Zeitkonzept
- Testing and Test Control Notation (TTCN-3)
  - Teststandard aus dem Umfeld des Protocol Testing
  - Fokussiert auf das Testen an Kommunikationsports
  - Einfaches Timer-Konzept
- Computer-Aided Specification and Testing (CAST) [Wahler et. al. 2012]
  - Ansatz zum Testen von technischen Anwendungen von ABB und ETH Zürich
  - Formulierung von Bedingungen über Variablen eines OPC-Servers
  - Auch zeitliche Formulierungen möglich

Grundidee

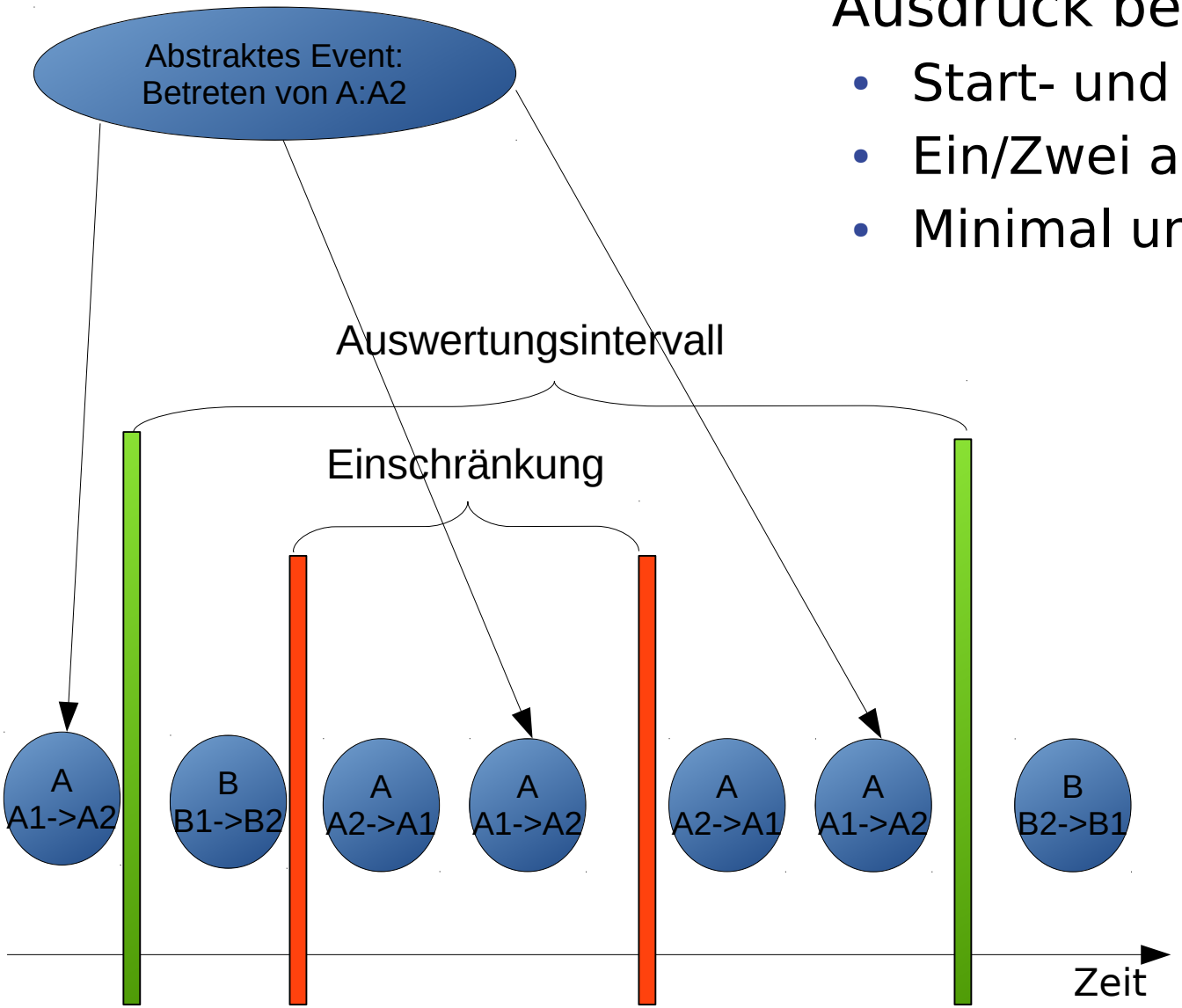
- Formulierung von *abstrakten Events*
  - Alle zu beobachtenden Ereignisse
  - Betreten/Verlassen von Zuständen
  - Auftreten von Transitionen
  - Wahr/Falschwerden von Prädikaten über Zuständen
  - Beenden eines Pfadausdrucks
- Darauf bauen Ausdruckstypen auf
  - Deadlines: Einschränkung einzelner abstrakter Events
  - Dauern: Einschränkung von Dauern zwischen zwei abstrakten Events

Mögliche Ausdrücke über Trace

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Ausdruck besteht aus

- Start- und Endzeitpunkt
- Ein/Zwei abstrakte Events
- Minimal und/oder Maximalwert



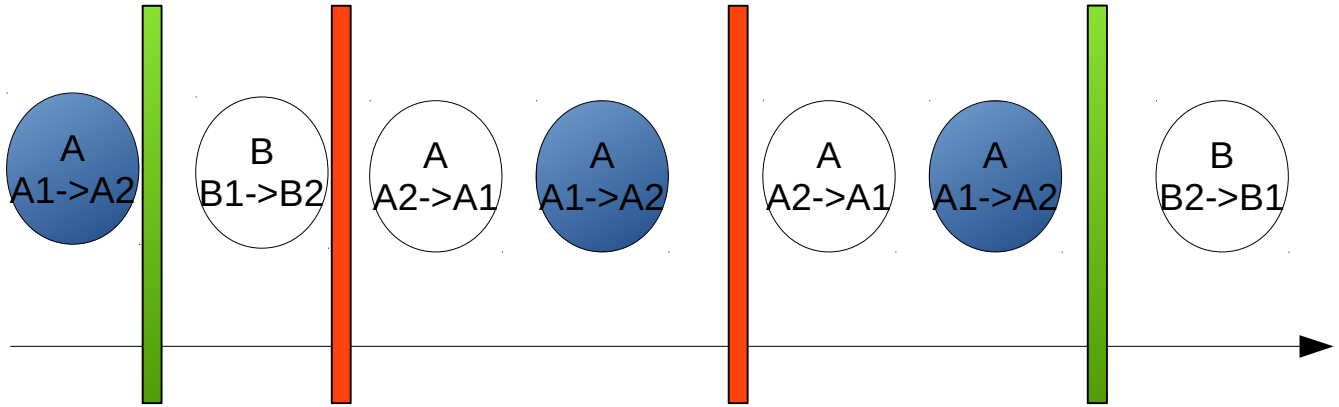
Start und Ende der Auswertung

Minimal- und Maximalwert der Einschränkung

Mögliche Ausdrücke über Trace

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Typ	Beschreibung	Deadlines	Dauern
`Eventually`	Mindestens ein Zutreffen im Intervall	☑	☑
`ExactlyOnce`	Genau ein Zutreffen im Intervall	☑	☑
`Always`	Aussage ist im Intervall immer korrekt	☑	☑
`JitterToAverage`	Abweichung einer Dauer zum Mittelwert immer innerhalb der Spezifikation	☒	☑
`CycleToCycleJitter`	Abweichung zwischen aufeinanderfolgenden Dauern immer innerhalb der Spezifikation	☒	☑
`SumOfDurations`	Summe der Dauern in einem Intervall innerhalb der Spezifikation	☒	☑



Formulierung von Uhren

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Uhren erlauben Vergleichbarkeit von zeitlichen Werten
- Uhren haben eine Startzeitpunkt in der Ausführung

Typ	Beschreibung
`SystemClock`	Systemzeit
`TestCaseClock`	Zeit, die seit Test-Case-Anfang vergangen ist
`AbstractEventClock`	Zeit, die seit dem letztmaligen Auftreten von einem bestimmten abstrakten Event vergangen ist

Beispiel

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

Beispiel mit Quantoren

```
Always {
   enter Tür.Aufgehend          //Abstraktes Event
   max 600 s TestCaseClock      //Maximalwert der Einschränkung

   start 0 s TestCaseClock      //Start des Intervalls
   stop 1800 s TestCaseClock    //Ende des Intervalls
}
```

Beispiel mit Akkumulation

```
SumOfDurations {
   from enter Tür.Offen         //Dauerbeschreibung durch
   to leave Tür.Offen           //zwei abstrakte Events
   max 600 s                    //Maximalwert der Einschränkung

   start 0 s TestCaseClock      //Start des Intervalls
   stop 1800 s TestCaseClock    //Ende des Intervalls
}
```

Verteilte Uhren

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

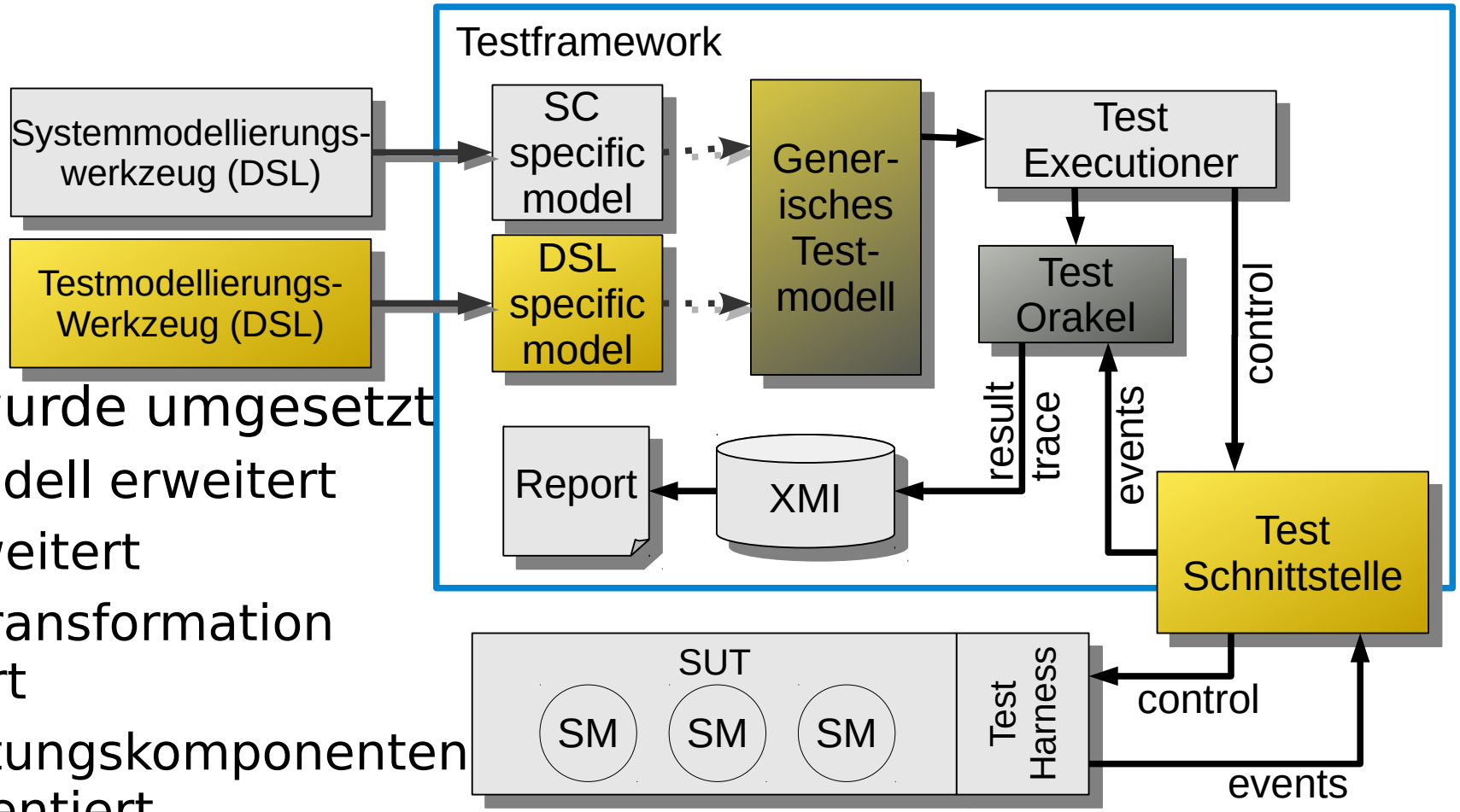
- Zeitstempel im Trace stammen von unterschiedlichen Rechnern
- Rechneruhren können synchronisiert werden
  - Fehler g: Abweichung von “idealer” Uhr
  - Fehlerbetrachtung kann in Testauswertung einbezogen werden

Typ	Deadlines	Dauern
`Eventually`	g	$2g$
`ExactlyOnce`	g	$2g$
`Always`	g	$2g$
`JitterToAverage`		$4g$
`CycleToCycleJitter`		$4g$
`SumOfDurations`		$\frac{g}{c}\sqrt{2n}$

Umsetzung

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Konzept wurde umgesetzt
  - Metamodell erweitert
  - DSL erweitert
  - Modelltransformation erweitert
  - Auswertungskomponenten implementiert
  - Analyse der zeitlichen Messfehler wurde berücksichtigt



Evaluation

- Tests von Echtzeitconstraints möglich
  - Wird verwendet um mehrere SPS im Verbund zu testen
- Formulierbar mit den Testern bekannten Modellelementen
- Beispiel des Projektpartners konnte getestet werden
  - Betrachtete Zykluszeiten liegen im Millisekundenbereich
- Genauigkeit: Vorgegeben durch Genauigkeit der Uhrensynchronisation
- Performance des Systems bei hoher Last noch nicht betrachtet

Fazit und Ausblick

Hochschule RheinMain
University of Applied Sciences
Wiesbaden Rüsselsheim

- Modell für zeitliche Ausdrücke im Rahmen von Testfällen
- Fehlerschranken
  - Berechnet für gegebenen maximalen Offset g
  - Obere Schranke konstant (mit einer Ausnahme)
- DSL um Modell zu formulieren
  - Angepasst an Modellwelt der Tester/Entwickler → Feedback
  - Angepasst an Testprozess
  - Eingebettet in bisherige Projektarbeiten
- Prototypisch evaluiert
  - Beispiel von Projektpartner getestet
  - Umsetzung mit Zeitsynchronisation
- Nächster Schritt:
  - Übertragung auf weitere Use Cases