

(Pro-)Seminar Softwaresystemtechnik (SST)



TECHNISCHE
UNIVERSITÄT
DARMSTADT

(SS 17, Proseminar 2 CP | Seminar 4 CP)



Einführungsveranstaltung



ES Real-Time Systems Lab

Prof. Dr. rer. nat. Andy Schürr

Dept. of Electrical Engineering and Information Technology

Dept. of Computer Science (adjunct Professor)

Johannes Bürdek

johannes.buerdek@es.tu-darmstadt.de

Tel.+49 6151 16 76089

www.es.tu-darmstadt.de

Herzlich Willkommen!

Als Proseminar (mit reduzierten Ansprüchen, 2 CP):

- B.Sc. ETiT (5. Sem.)

Als Seminar (4 CP):

- B.Sc. (5. Sem.) und M.Sc. Informatik (2. Sem.)
- Dipl. Informatik
- B.Sc. Informationssystemtechnik (5. Sem.)
- Dipl. ETiT (DT, Hauptstudium)
- entsprechende Wirtschaftsstudiengänge
- Sonstige passende Fachrichtung

Was wir von den Teilnehmern erwarten...

- Interesse am Thema + Motivation
- Wille zur Zusammenarbeit mit
 - Betreuer
 - Kommilitonen bei einer Gruppenarbeit
- Wissenschaftliches Vorgehen (unter Anleitung)
- Fristgerechte Abgabe der geforderten Arbeiten
- Teilnahme an *allen* Pflichtveranstaltung



ACHTUNG: Das Seminar ist inhaltlich und vom Umfang her anspruchsvoll!
→ Wir geben uns Mühe bei der Betreuung, und erwarten im Gegenzug von allen Teilnehmern ebenfalls **vollen Einsatz!!!**

- Grundfertigkeiten zur Erstellung einer **wissenschaftlichen Arbeit**
 - Selbständiges Erarbeiten eines Themengebietes (unter Anleitung)
 - Literaturrecherche
 - finden, lesen, verstehen, bewerten
 - Wissenschaftliches Schreiben
 - Gliedern, Zitieren, Formulieren
- Mitwirken am **Reviewprozess**
 - Verwendbares Feedback zu fremden Arbeiten geben
 - Gegenseitige Unterstützung, Schwachstellen identifizieren
- **Präsentation**
 - Aufbereiten, bewerten der Ergebnisse
 - Vorstellen der Ergebnisse
 - Techniken, Stil, Zeiteinteilung, Reden vor der Gruppe



- Heute
 - Themenvorstellung
 - Themenvergabe durch uns
- Während des Semesters
 - Erstellen einer Ausarbeitung
 - (Auf-)Schreiben von (Zwischen-) Ergebnissen
 - Vortrag vorbereiten
 - Regelmäßige Absprachen mit Betreuer!
 - Individuelle Absprachen
 - Fortschritt, Fragen, Feedback, Tipps
- Am Ende des Semesters (**10. + 11. Juli**)
 - Vortrag im Blockseminar → Präsentation + Ausarbeitung liegen bereits vor



Plagiatshinweis – „Abschreiben“ verboten!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Wir messen der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei.
- Mit der Abgabe einer Lösung (Hausaufgabe, Programmierprojekt, Diplomarbeit, etc.) bestätigen Sie, dass (Sie/Ihre Gruppe) (der alleinige Autor/die alleinigen Autoren) des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitiert haben.
- Weiterführende Informationen unter <http://www.es.tu-darmstadt.de/lehre/plagiat/>



Wichtige Links zur Veranstaltung

- **Seminarrichtlinien**
 - <http://www.es.tu-darmstadt.de/fileadmin/download/lehre/Seminarrichtlinien.pdf>
- **Plagiatshinweise**
 - <http://www.es.tu-darmstadt.de/lehre/plagiat/>
- **Schreibkurse**
 - http://www.owl.tu-darmstadt.de/owl_ueber_uns/ueber_uns_1.de.jsp
 - http://www.hda.tu-darmstadt.de/angebote_fuer_studierende_zentral/trainings_fuer_schluesselformen/angebot_startseite_stud_sk/startseite_stud_sk.de.jsp



Zeitplan

Datum	Topic	Abgabe
20.04.	Auftaktveranstaltung	-
15.05.	Gliederung der Ausarbeitung	Betreuer
12.06.	Erste Fassung der Ausarbeitung	Johannes Bürdek
19.06.	Abgabe der Reviews	Johannes Bürdek
03.07.	Endfassung der Ausarbeitung	Johannes Bürdek
03.07 - 07.07.	Probenvorträge	Betreuer
10. + 11.07.	Abschlussvortrag	-

Themenübersicht

Thema	PS
Distribution in Graph Transformation	
Controller-Synthese für (un)gezeitete Systeme	X
Effiziente Algorithmen für die Analyse von Timed Games	
On Patterns for Decentralized Control in Self-Adaptive Systems	X
Design of Domain-Specific Languages	X
Bidirectional Model Transformations with TGGs	X
Bidirectional Transformations using QVT-R	X
Data Center Topologies	
Scalable prediction of non-functional properties in software product lines: Footprint and memory consumption	



Themenvorstellungen

Distribution in Graph Transformation

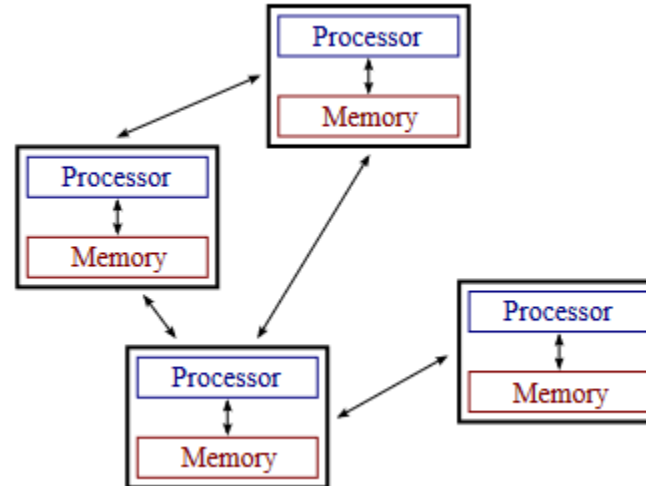
SST Seminar – SS 17

Géza Kulcsár

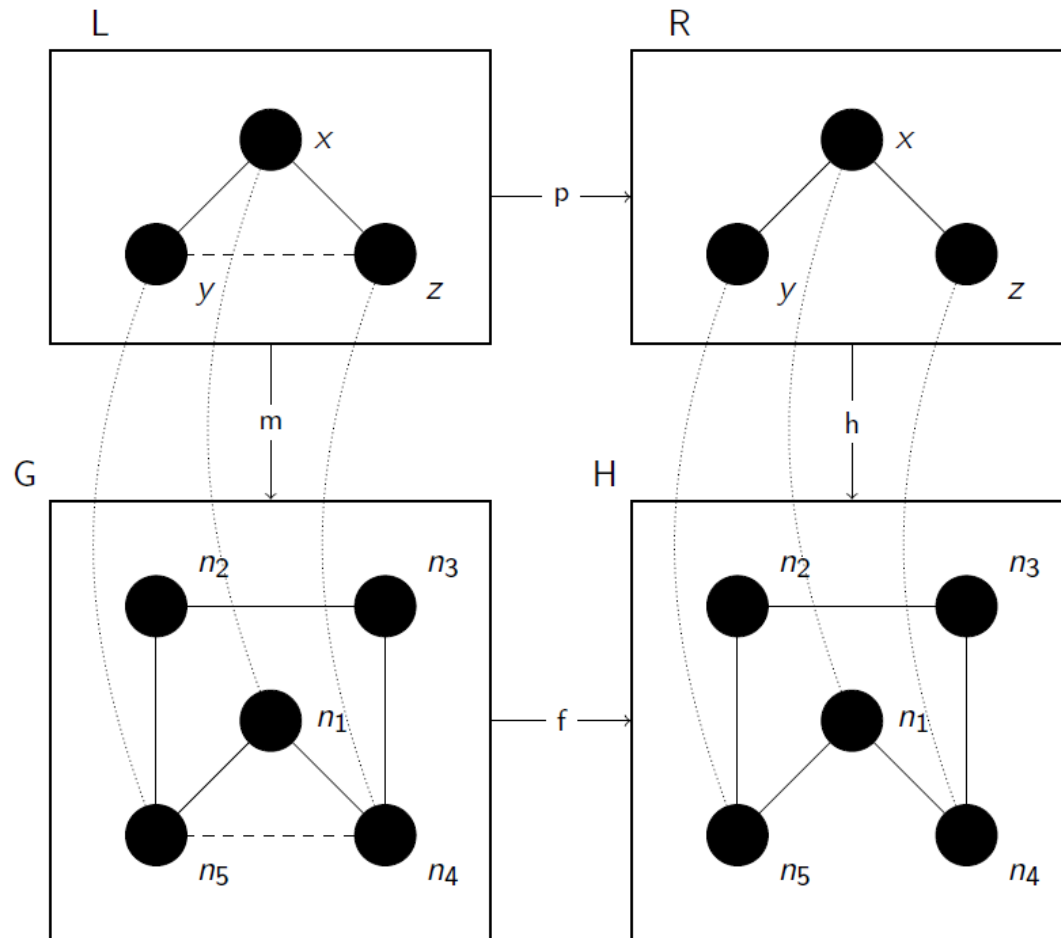


What is a distributed system?

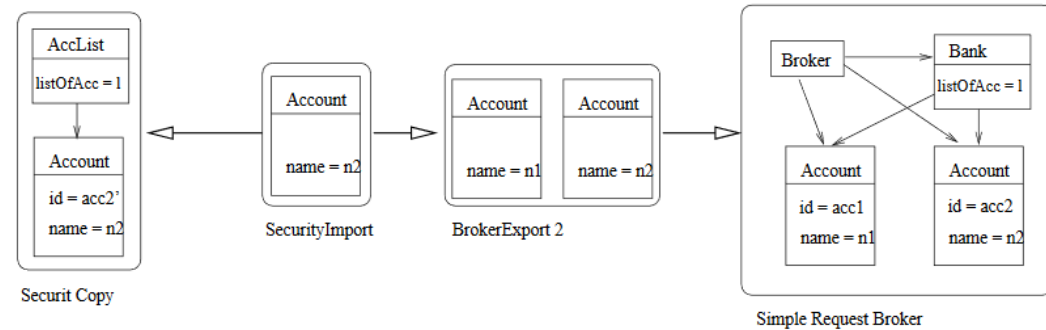
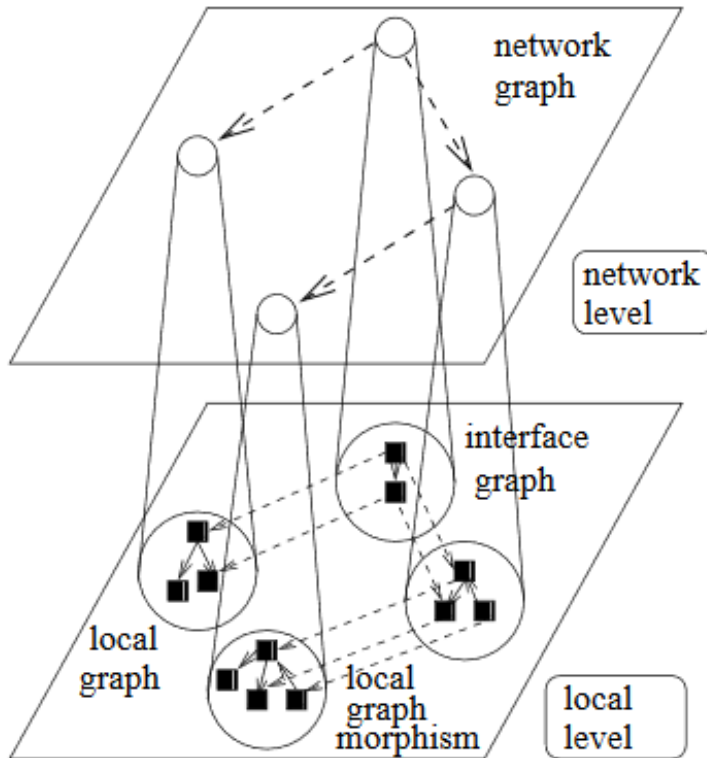
- „A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages.”



What is graph transformation?



How to distribute graph transformation?

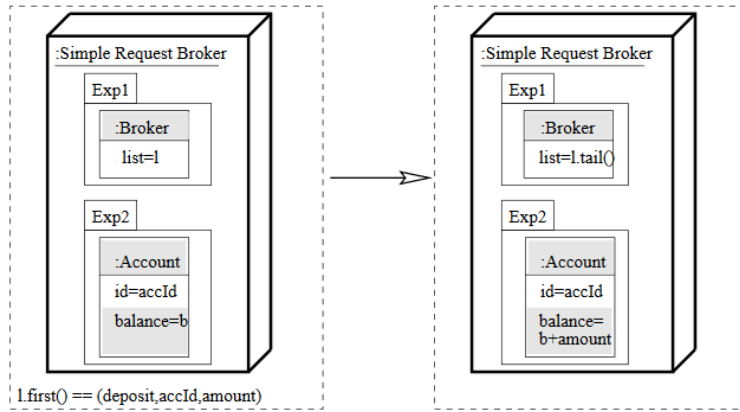


Local vs. Global

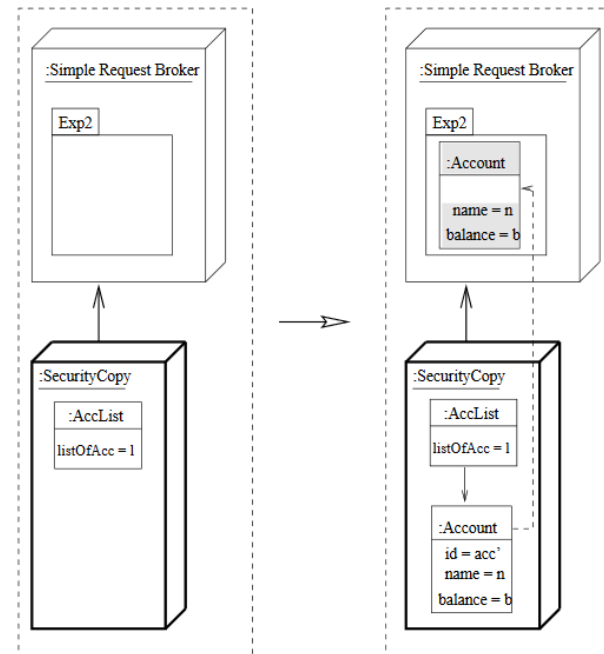
This is what the broker does:

And this is how a backup is performed:

depositRequest()



newAccount(n,b)



How to synchronize?



Overview

Goals

- Insights:
 - Modeling distributed systems
 - Distributed graph transformation
 - Notion of synchronization
 - (Optional) Investigating the tool landscape

Literature

- Andrew S. Tanenbaum: Distributed Systems – Principles and Paradigms
- *Gabriele Taentzer: Distributed Graph Transformation*

Controller-Synthese für (un)gezeitete Systeme

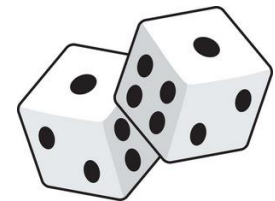
SST Seminar – SS 17
(Proseminar geeignet)

Lars Luthmann



Controller-Synthese für (un)gezeitete Systeme (Proseminar geeignet)

- Problem: Können Übergänge eines Modells eingeschränkt werden, so dass das verbleibende Verhalten bestimmte Eigenschaften erfüllt?
- Siegbedingung: Spieler A kann das Spiel innerhalb eines bestimmten Teils des Modells halten, egal welche Aktion Spieler B durchführt.
- Praktische Anwendung: Online Testing



Aufgabe:

- (Einführung in Timed Games)
- Beschreiben der Controller-Synthese
- Proseminar/Bachelor: nur ungezeitet



Effiziente Algorithmen für die Analyse von Timed Games

SST Seminar – SS 17

Lars Luthmann

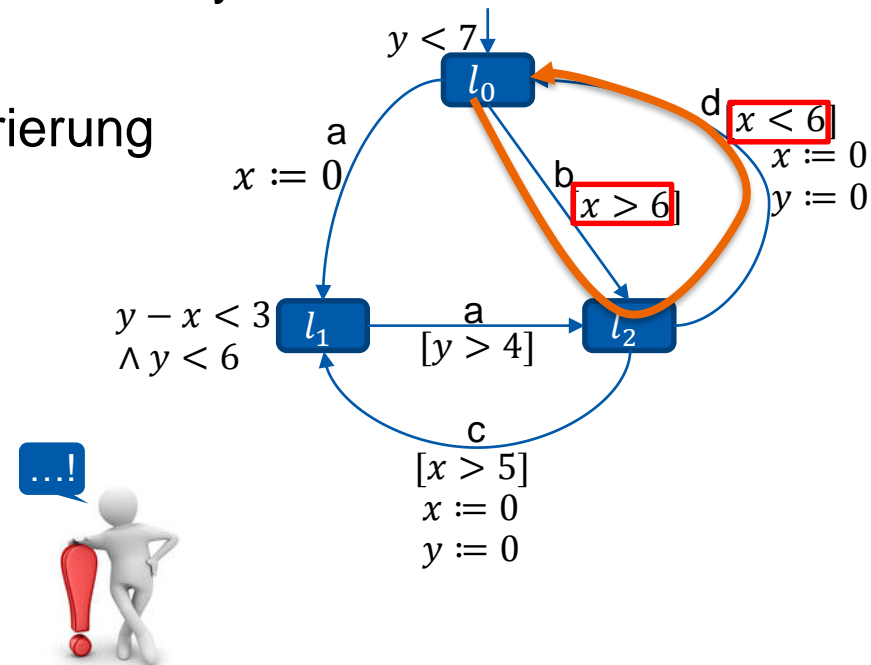


Effiziente Algorithmen für die Analyse von Timed Games

- Problem: (Wie) ist ein Zustand in einem System zu erreichen?
- Erreichbarkeit hängt von Vergangenheit des System ab
- Praktische Anwendung: Testfallgenerierung

Aufgabe:

- Einführung in Timed Games
- Beschreiben des Algorithmus
- Nur für Master-Studenten geeignet



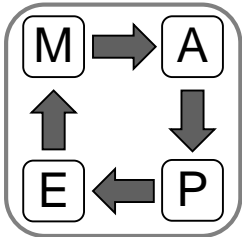
On Patterns for Decentralized Control in Self-Adaptive Systems

SST Seminar – SS 17
(Proseminar geeignet)

Markus Weckesser



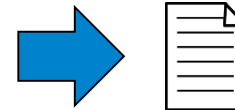
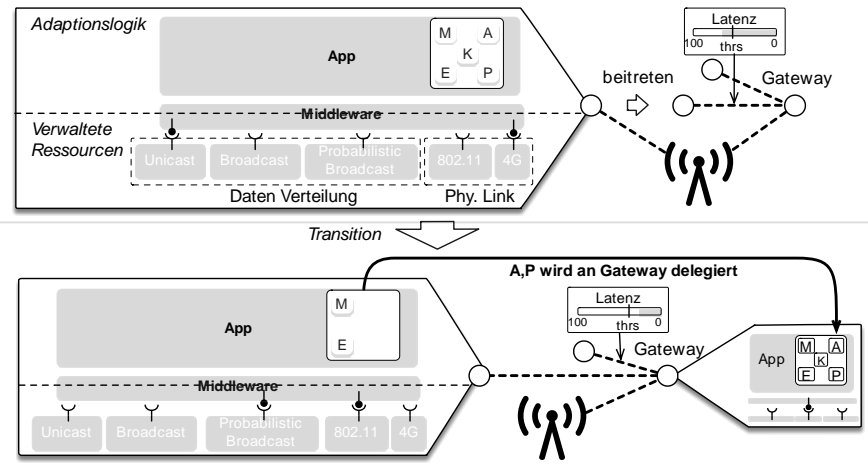
On Patterns for Decentralized Control in Self-Adaptive Systems



- **M**onitoring
- **A**nalysis
- **P**lanning
- **E**xecution



Platooning Use Case



Weyns, Danny, et al. "On patterns for decentralized control in self-adaptive systems." Software Engineering for Self-Adaptive Systems II. Springer Berlin Heidelberg, 2013. 76-107.

Ablauf

1. Paper lesen (Umfang kann nach Absprache individuell angepasst werden)
2. Paper diskutieren und offene Fragen klären
3. Interessante Fragestellungen aus Paper ggf. zusätzlich recherchieren
4. Dokumentation der Ergebnisse
5. Review-Prozess



Scalable prediction of non-functional properties in software product lines: Footprint and memory consumption

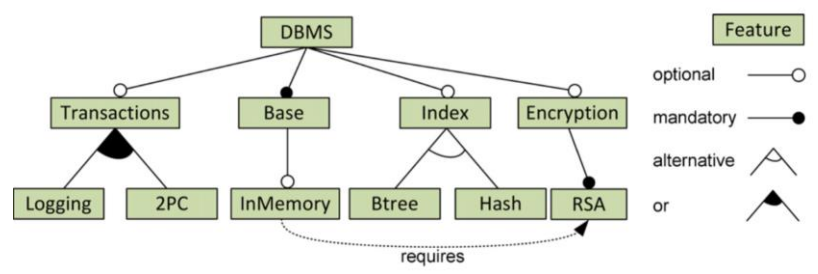
SST Seminar – SS 17

Markus Weckesser

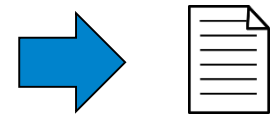
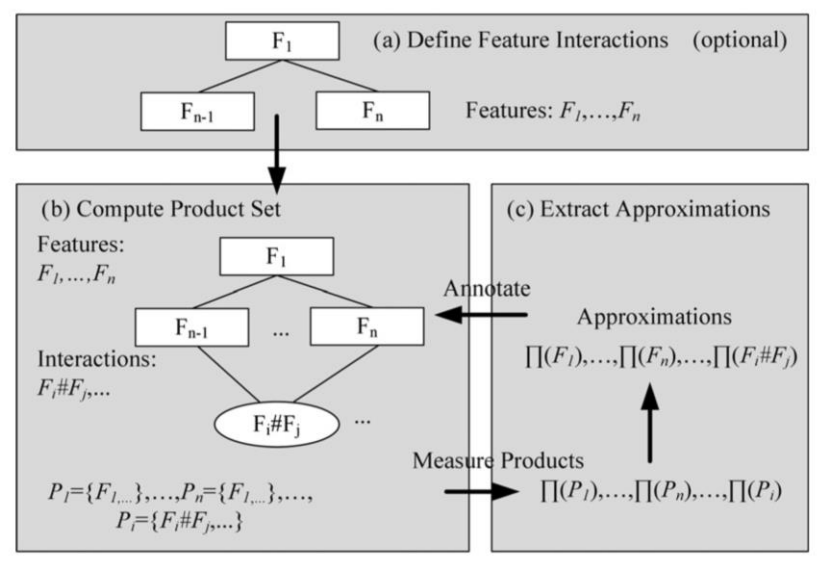


Scalable prediction of non-functional properties in software product lines: Footprint and memory consumption

Sample DBMS product line



Process of computing approximations of non-functional properties for features and feature interactions



Siegmund, Norbert, et al. "Scalable prediction of non-functional properties in software product lines: Footprint and memory consumption." Information and Software Technology 55.3 (2013): 491-507.

- ### Ablauf
1. Paper lesen (Umfang kann nach Absprache individuell angepasst werden)
 2. Paper diskutieren und offene Fragen klären
 3. Interessante Fragestellungen aus Paper ggf. zusätzlich recherchieren
 4. Dokumentation der Ergebnisse
 5. Review-Prozess



Design of Domain-Specific Languages

SST Seminar – SS 17
(Proseminar geeignet)

Lars Fritsche



1. Topic – Design of Domain-Specific Languages

How to describe an excavator?



1. Topic – Design of Domain-Specific Languages

How to describe an excavator?



```
# Initializations

# Import needed modules
import os, sys

if 'ame_apy' not in sys.modules:
    try:
        from ame_apy import *
    except ImportError:
        print 'Unable to import LMS Amesim API module. '

# Get license and initialize LMS Amesim Python API
AMEInitAPI(False)

# Create new "HydraulicExcavator" system
AMECreateCircuit('HydraulicExcavator')

# Add components

# Add "runstats" component
AMEAddComponent('runstats', 'runstats', (76, 10))

# Rotate and flip the component
AMERotateComponent('runstats')
AMERotateComponent('runstats')
AMERotateComponent('runstats')
AMEMoveComponent('runstats', (76, 10))
```



1. Topic – Goals

Getting familiar with Domain-Specific Languages

- How to design?
- How to implement?
- How to use?
- How to validate?
- How to debug?



Xtext

Literature: Markus Vlter, Sebastian Benz, Christian Dietrich, Birgit Engelmann, Mats He-lander, Lennart C. L. Kats, Eelco Visser, and Guido Wachsmuth. *DSL Engineering - Designing, Implementing and Using Domain-Specific Languages*. dslbook.org, 2013.



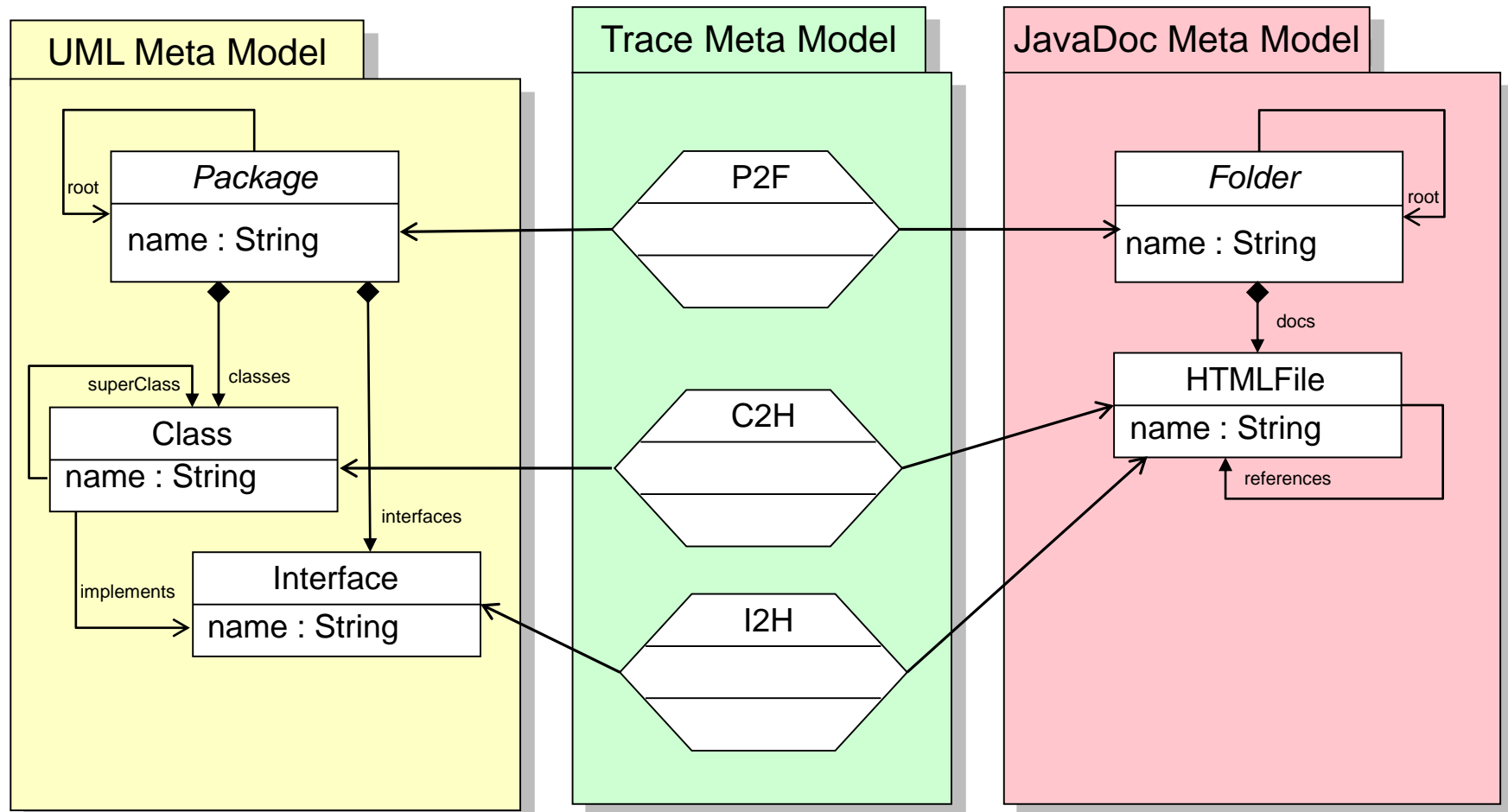
Bidirectional Model Transformations with TGGs

SST Seminar – SS 17
(Proseminar geeignet)

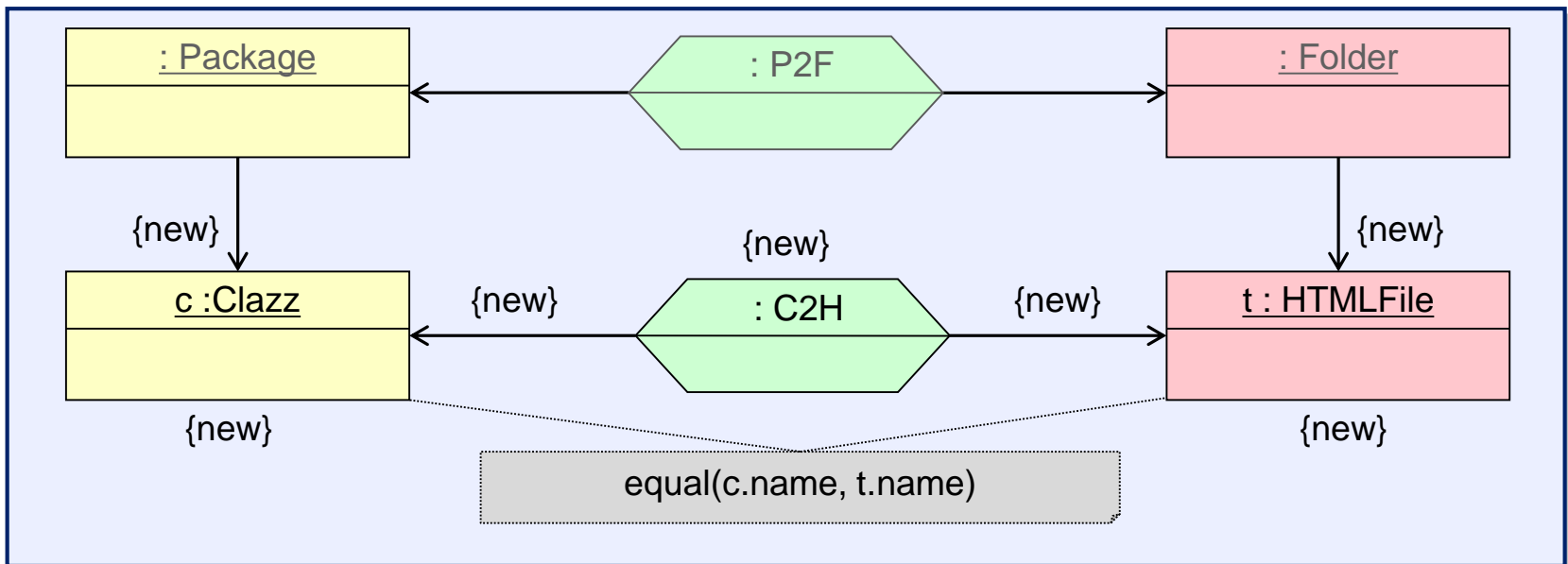
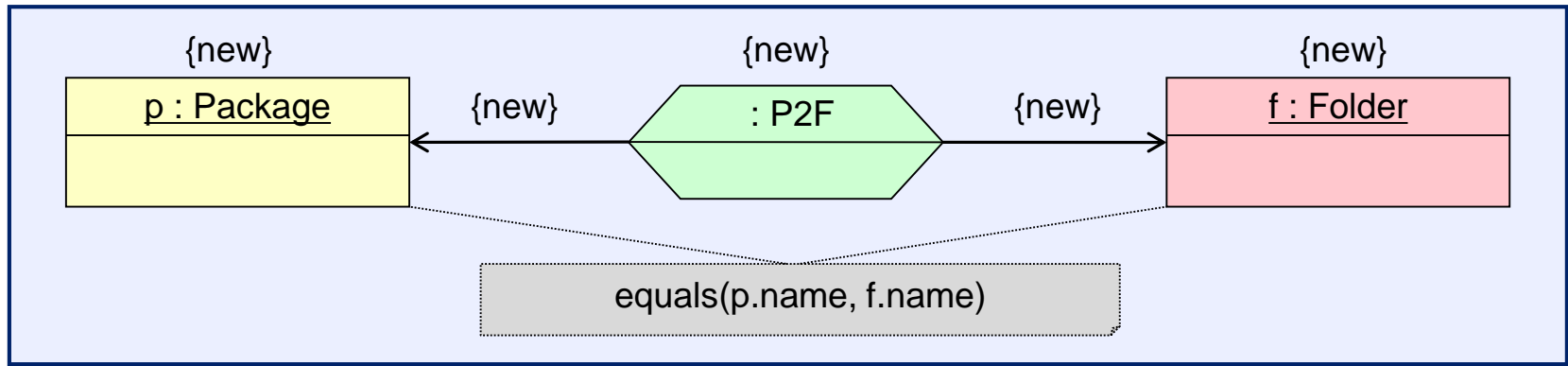
Lars Fritsche



2. Topic – Bidirectional Model Transformations with TGGs



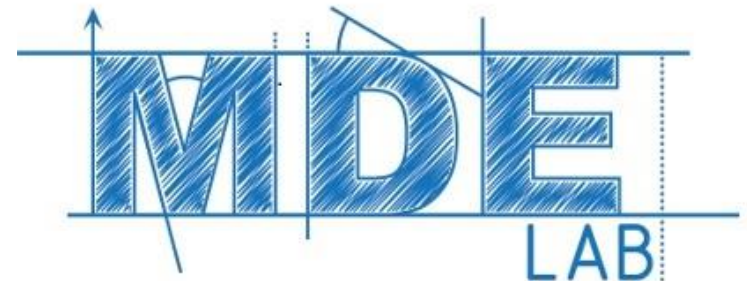
2. Topic – Bidirectional Model Transformations with TGGs



2. Topic – Goals

Getting familiar with Triple Graph Grammars

- What are patterns?
- What are graph transformations?
- How to implement a TGG? (using [MoTE](#))
- How to efficiently propagate minor changes?



Literature: Stephan Hildebrandt. *On the Performance and Conformance of Triple Graph Grammar Implementations*. PhD thesis, Hasso Plattner Institute at the University of Potsdam, 2014.



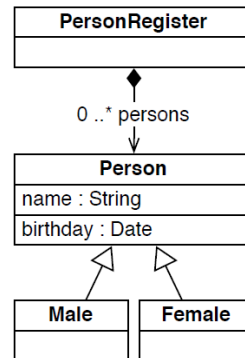
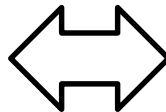
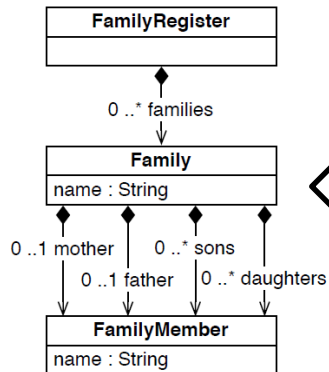
Bidirectional Transformations using QVT-R

SST Seminar – SS 17
(Proseminar geeignet)

Lars Fritsche



3. Topic – Bidirectional Transformations using QVT-R



```

transformation families2persons ( famDB : Families , perDB : Persons ) {
  top relation FamilyRegister2PersonRegister {
    enforce domain famDB familyRegister : Families :: FamilyRegister {};
    enforce domain perDB personRegister : Persons :: PersonRegister {};
    where {
      Father2Male ( familyRegister , personRegister );
      Son2Male ( familyRegister , personRegister );
      Mother2Female ( familyRegister , personRegister );
      Daughter2Female ( familyRegister , personRegister );
    }
  }
  relation Mother2Female {
    familyName , firstName , fullName : String ;
    enforce domain famDB familyRegister : Families :: FamilyRegister {
      families = family : Families :: Family {
        name = familyName ,
        mother = mother : Families :: FamilyMember { name = firstName }
      }
    };
    enforce domain perDB personRegister : Persons :: PersonRegister {
      persons = female : Persons :: Female { name = fullName }
    };
    where {
      fullName = familyName + ' , ' + firstName ;
      firstName = firstName ( fullName );
      familyName = familyName ( fullName );
    }
  }
  relation Daughter2Female {
    ... -- Like ' Mother2Female ', replace 'mother ' with 'daughters '
    when {
      female . oclIsUndefined ();
    } -- Prevents application in backward direction
    ... -- See ' Mother2Female '
  }
}

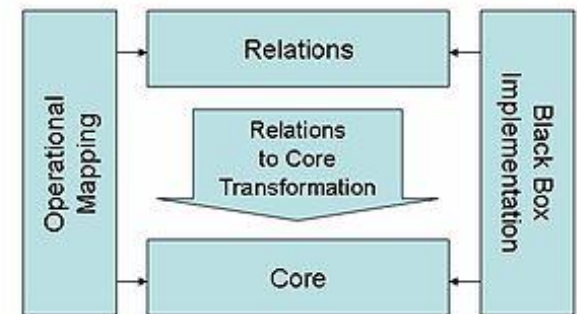
```



3. Topic – Goals

Getting familiar with Query/View/Transformation Relations

- How is this standard defined?
- How to implement a QVT specification?
- How to implement bidirectional transformations?
- What are the limitations?
- Point out interesting applications



Literature: Bernhard Westfechtel. Case-based exploration of bidirectional transformations in qvt relations. *Software & Systems Modeling*, pages 1–41, 2016.

Data Center Topologies

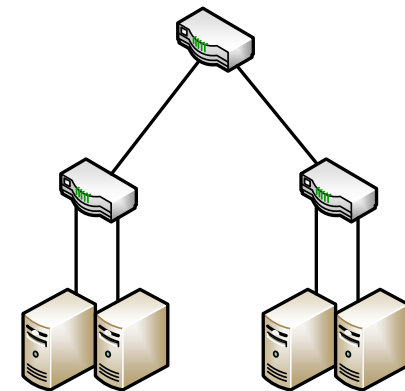
SST Seminar – SS 17

Stefan Tomaszek

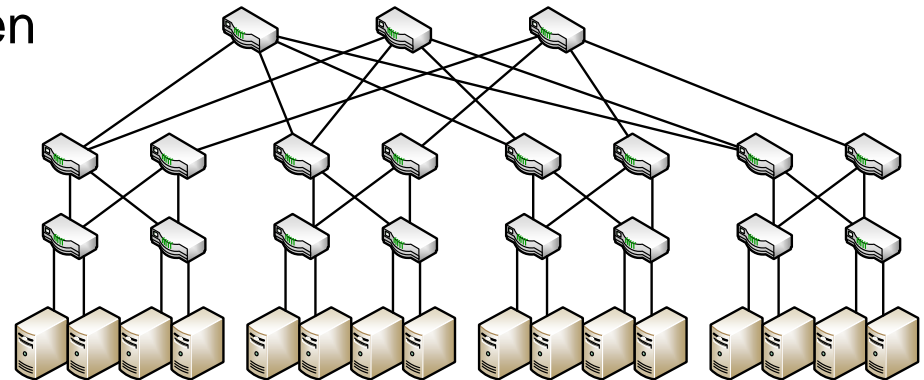


Data Center Topologies

- Rechenzentren bilden Knotenpunkte in der IT-Infrastruktur
- Rahmenbedingungen
 - Ausfallsicher
 - Redundant
 - Garantierte Bandbreiten und Latenzen
 - Hohe Auslastung
 - Generische und kostengünstige Hardware



Aufgabe: Beschreiben einer oder mehrere Topologien mit deren Vor-,
Nachteilen und Rahmenbedingungen



Wie geht es weiter...!?

E-Mail an johannes.buerdek@es.tu-darmstadt.de bis heute 24 Uhr mit

- eurem Erst-, Zweit- und Drittwunsch und
- Studiengang + Fachsemester

Unsere Aufgaben

- Wir verteilen die Themen schnellstmöglich auf die Interessenten
- Geben das Ergebnis bekannt (→ E-Mail)
- Bereiten alles vor, damit anschließend die Bearbeitung unmittelbar starten kann

Eure Aufgaben

- Warten auf Ergebnisse der Zuteilung
- Anschließend meldet ihr euch bitte unmittelbar beim Betreuer
→ Termin für ein erstes, persönliches Treffen



Fragen?

