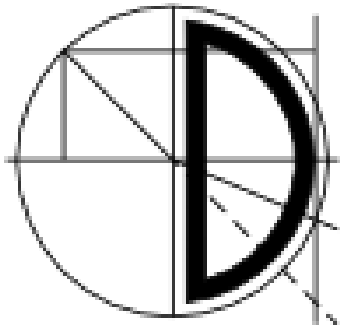
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

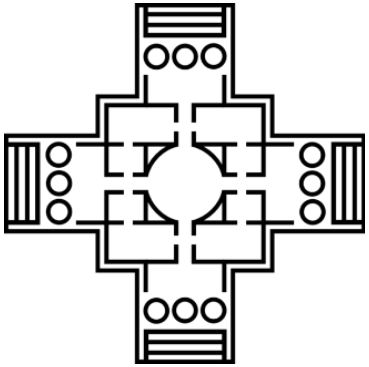
Automated Transformation from Descartes Modeling Language to Palladio Component Model

Jürgen Walter, Simon Eismann, Adrian Hildebrandt

Dept. of Computer Science, University of Würzburg

Symposium on Software Performance,
Nov 6th 2015, Munich, Germany





Differences PCM and DML?

Motivation

Benefits Transformation

- Tooling can be reused
- Improves understanding of differences
- Flexibility to change

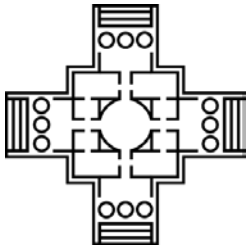
Benefits Automation

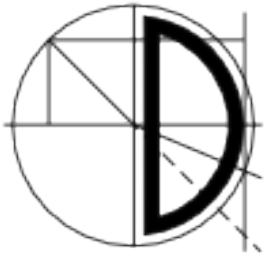
- Automated transformation is faster, less error-prone and less expensive compared to manual extraction

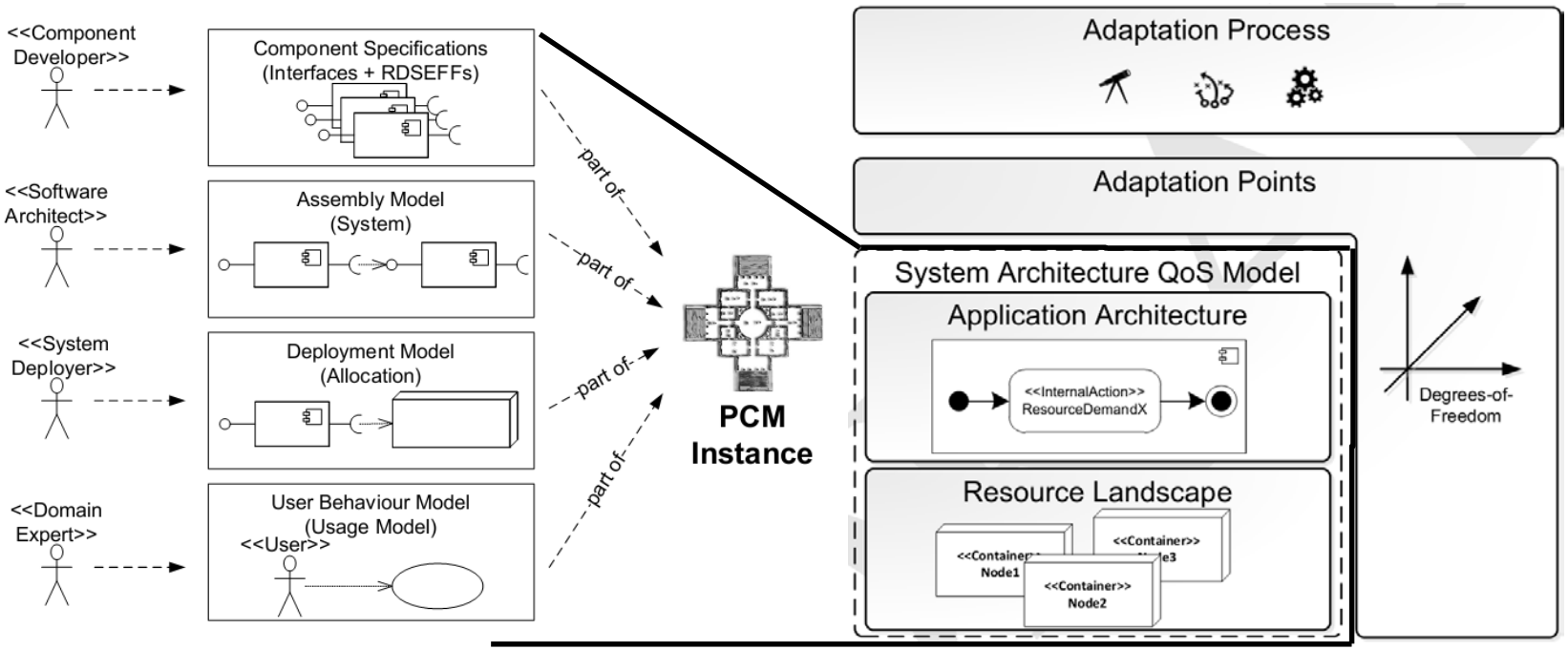
Project Overview

Problem	• Tooling has to be developed for each formalism • Comparison challenging • Manual transformation error-prone and time consuming
Idea	• Model transformation of DML to PCM
Benefit	• Reuse of existing PCM tooling for DML • Improved understanding of differences and similarities • Simplified change of formalism
Action	• Implement a model-to-model transformation

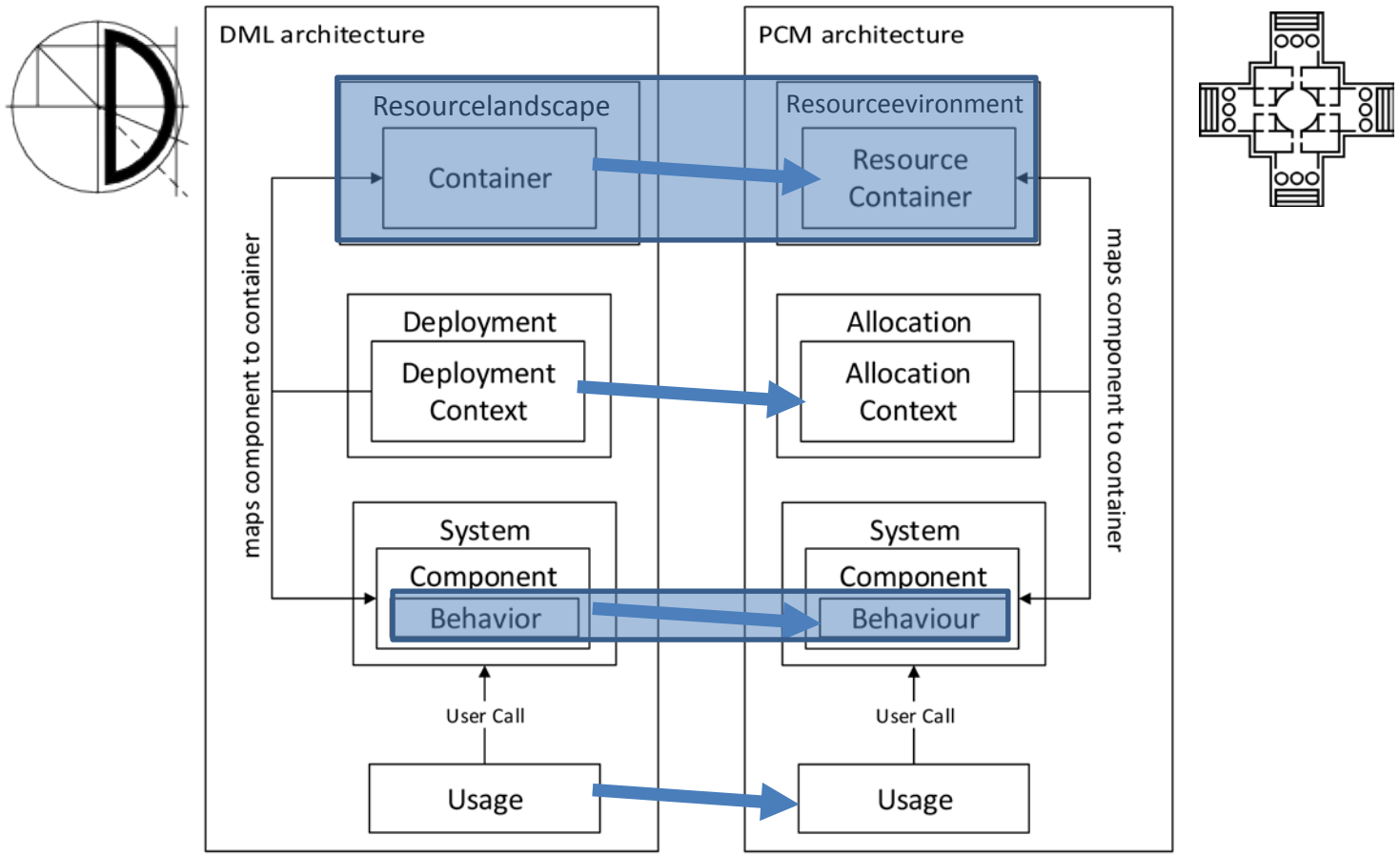
Architectural Performance Modeling







Transformation Overview



Resource Landscape

platform:/resource/MediaStore/default.resourcelandscape

- Distributed Data Center distributedDataCenter
  - Data Center smallDataCenter
    - Computing Infrastructure AppServer
    - Storage Infrastructure DBServer
    - Networking Infrastructure aName

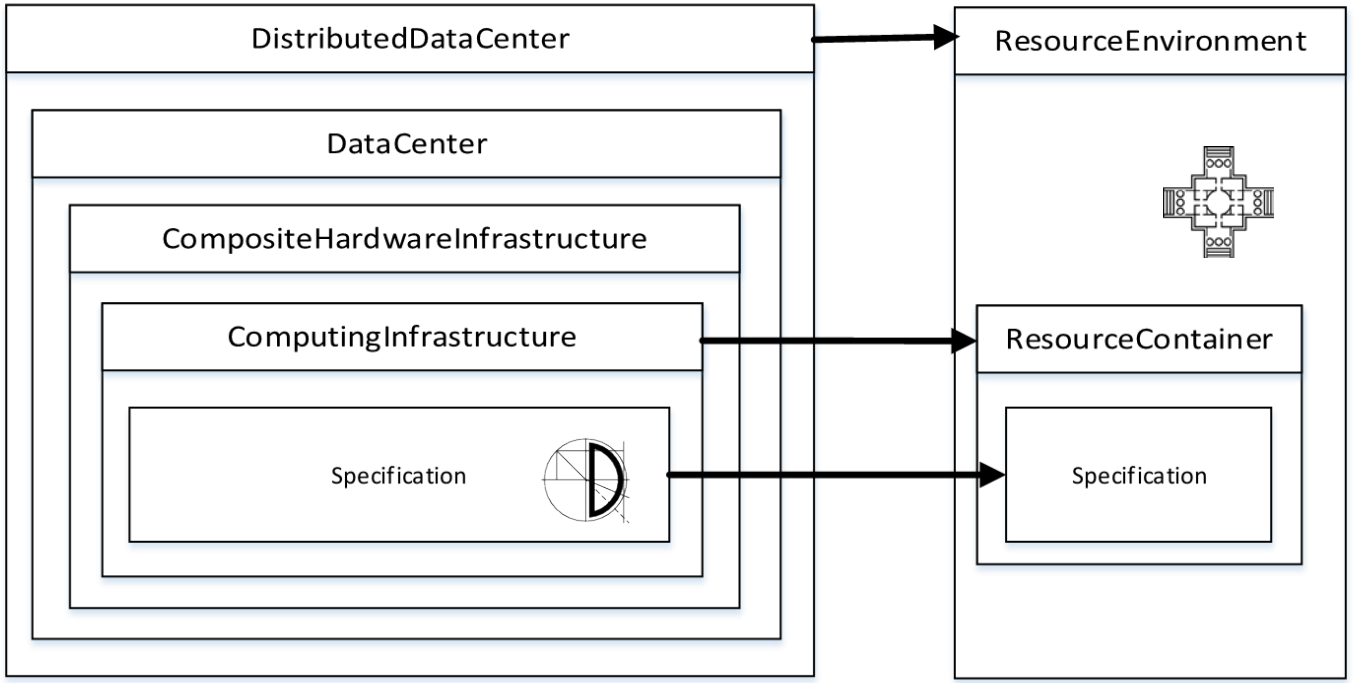
Hierarchical resource landscape

platform:/resource/MediaStoreExample/mediastore.resourceenvironment

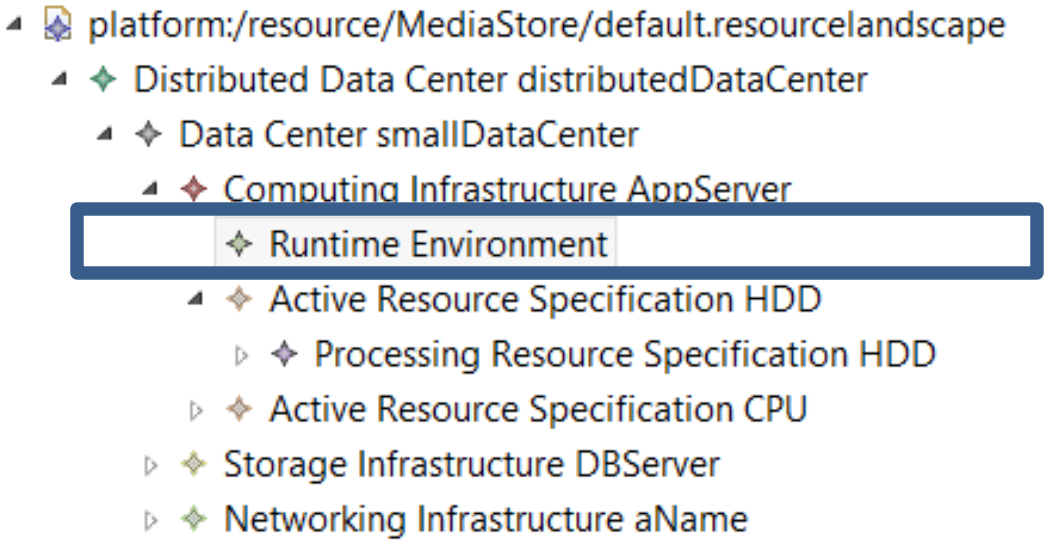
- Resource Environment aName <ResourceEnvironment>
  - aName <LinkingResource> [ID: __oYXADq_EeCCbpF63PfiyA]
  - AppServer <ResourceContainer> [ID: _5uTBUBpmEdyxqpPYxT_m3w]
  - DBServer <ResourceContainer> [ID: _tVi40Dq_EeCCbpF63PfiyA]

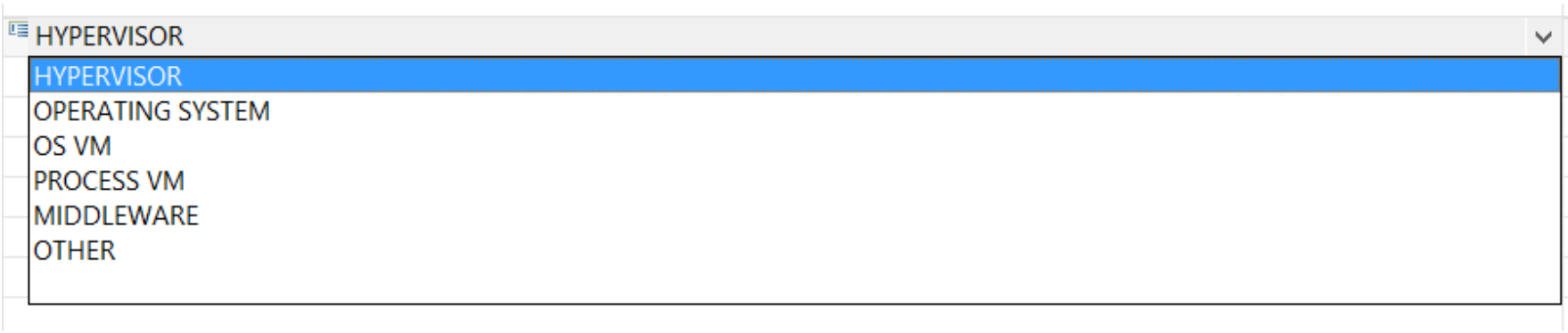
Information loss

Resource Landscape

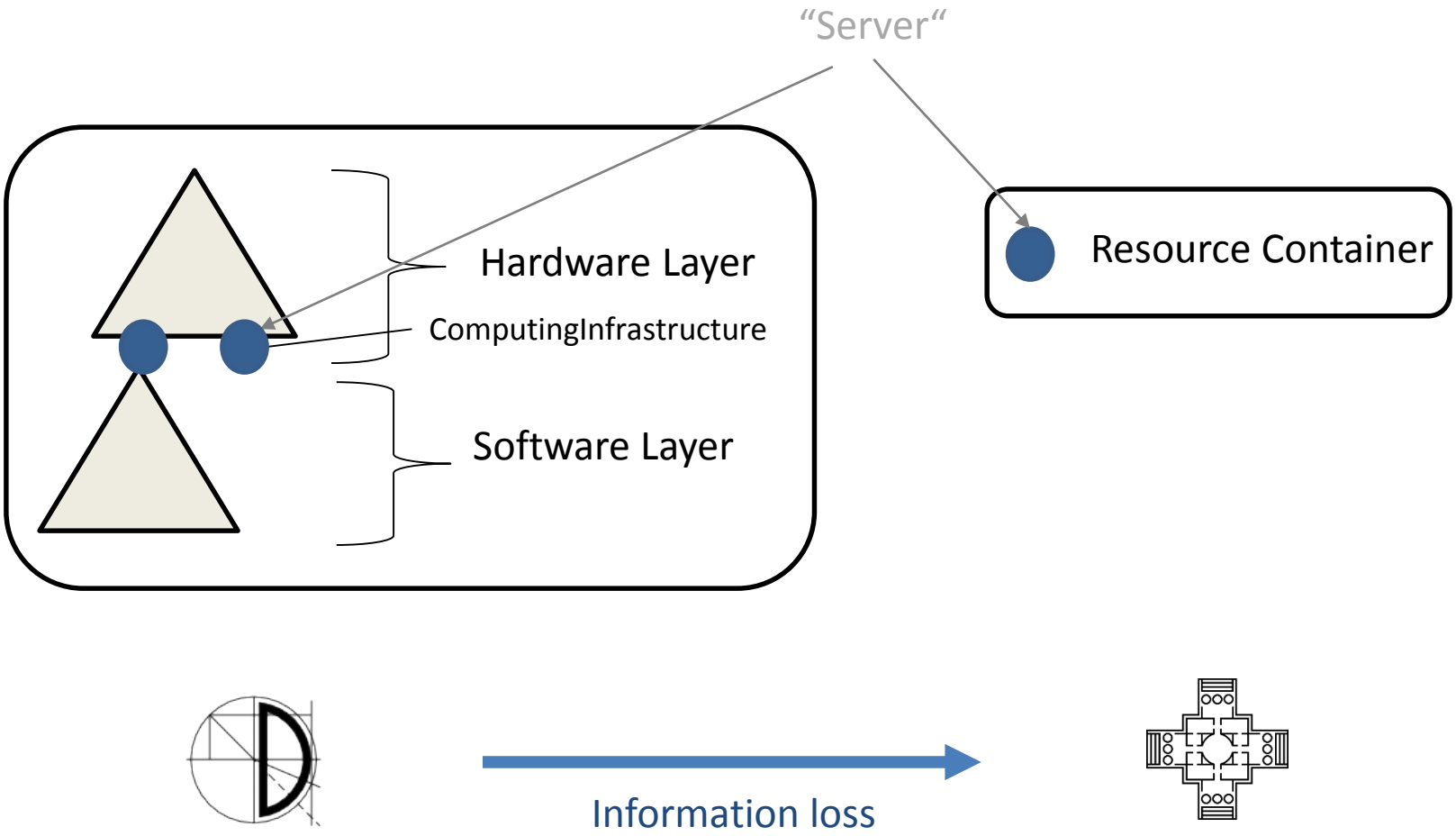


Specification of Processing Resources



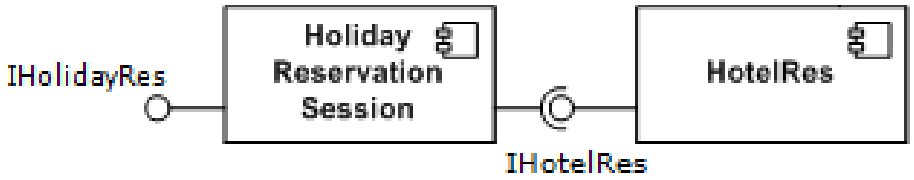


Processing Resources



Service Behavior

Example:

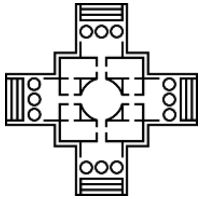


DML Abstraction Layers

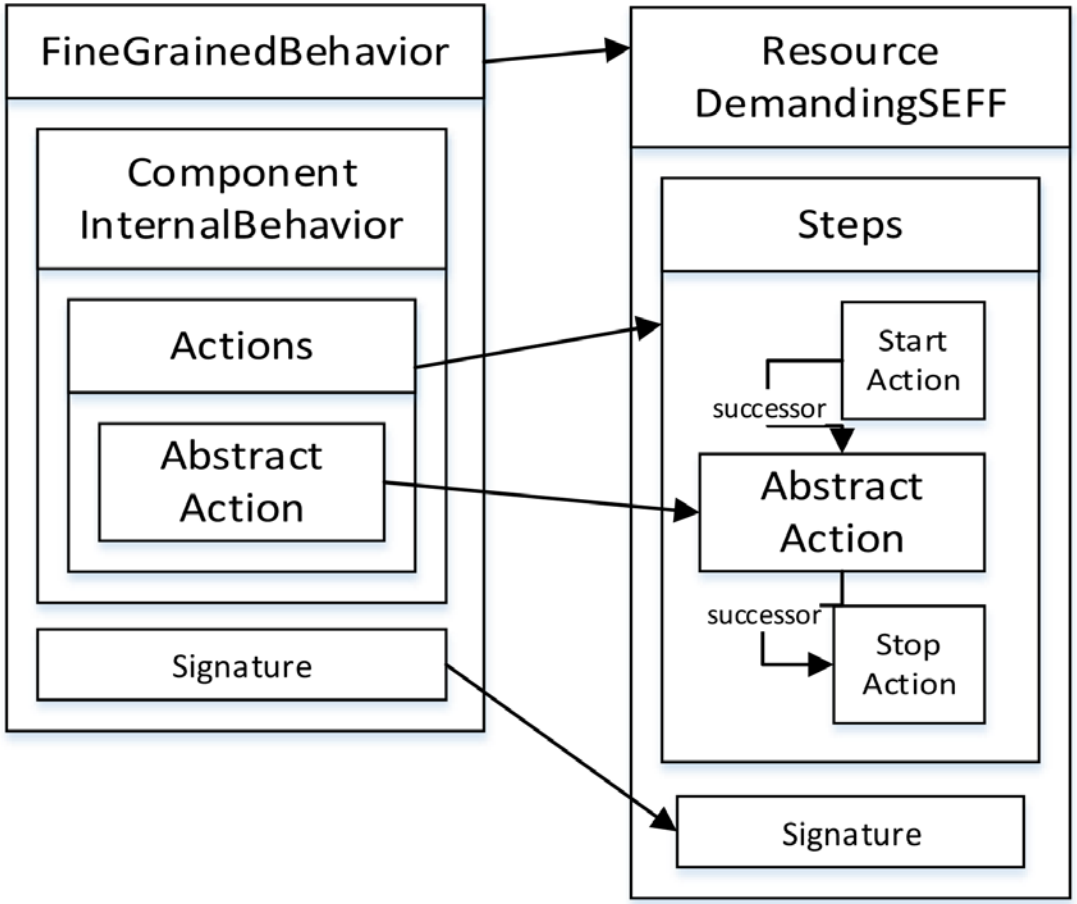
- Fine-grained behavior
- Coarse-grained behavior
- Blackbox behavior

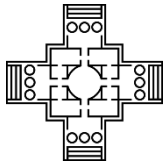
PCM Abstraction Layer

- Fine-grained behavior/SEFF

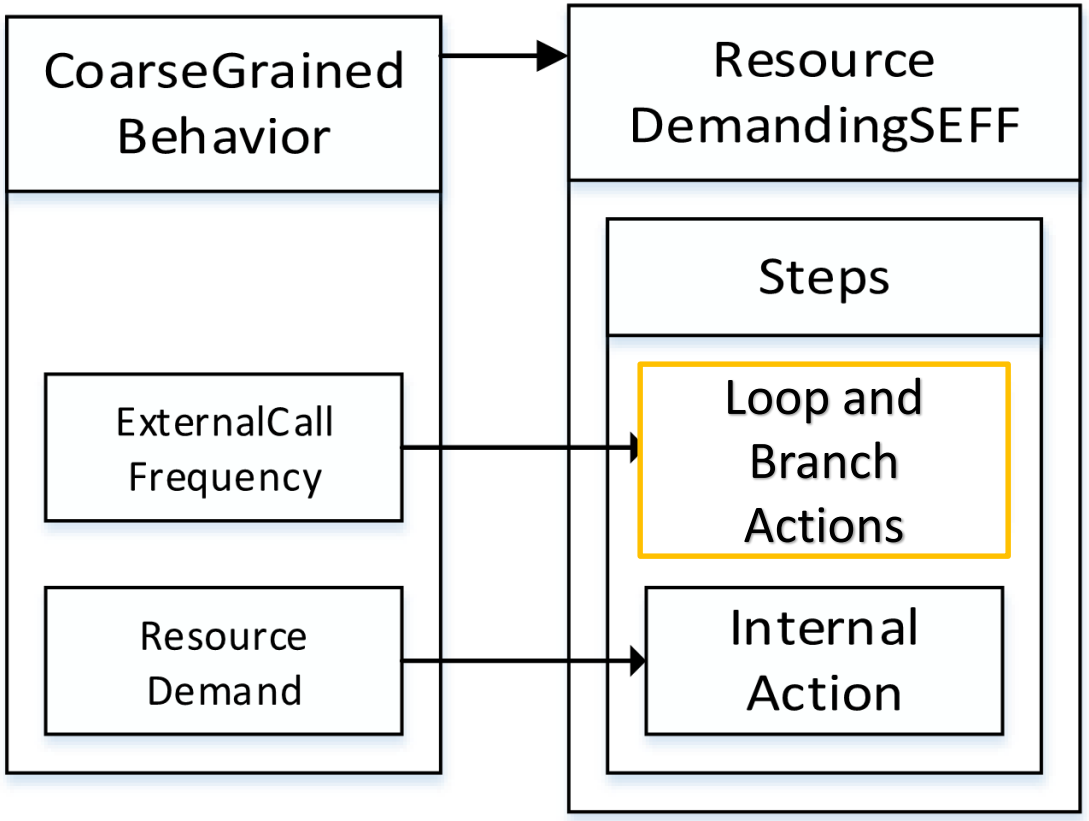


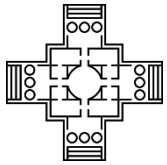
Fine-Grained Behavior Mapping



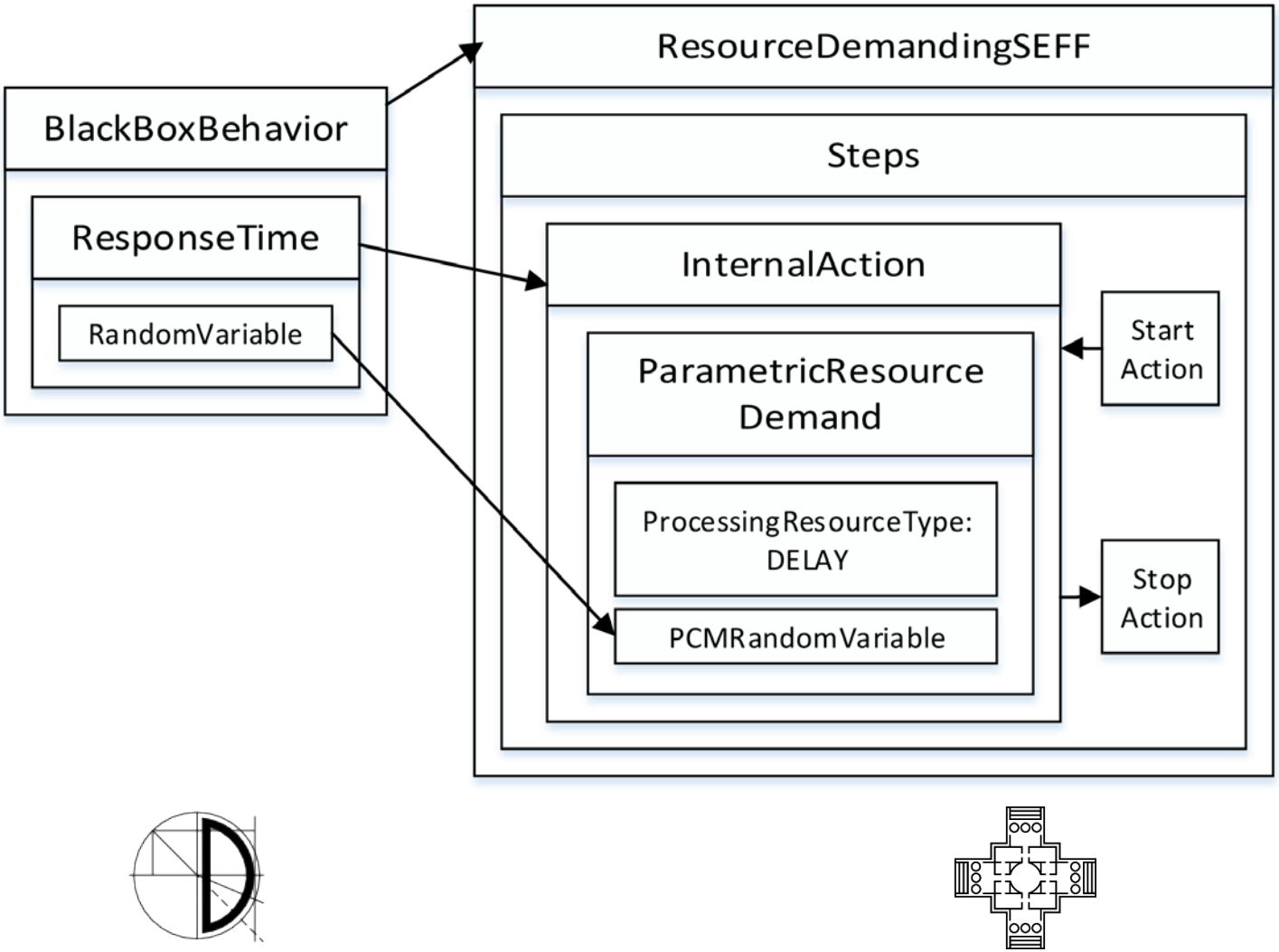


Coarse-Grained Behavior Mapping

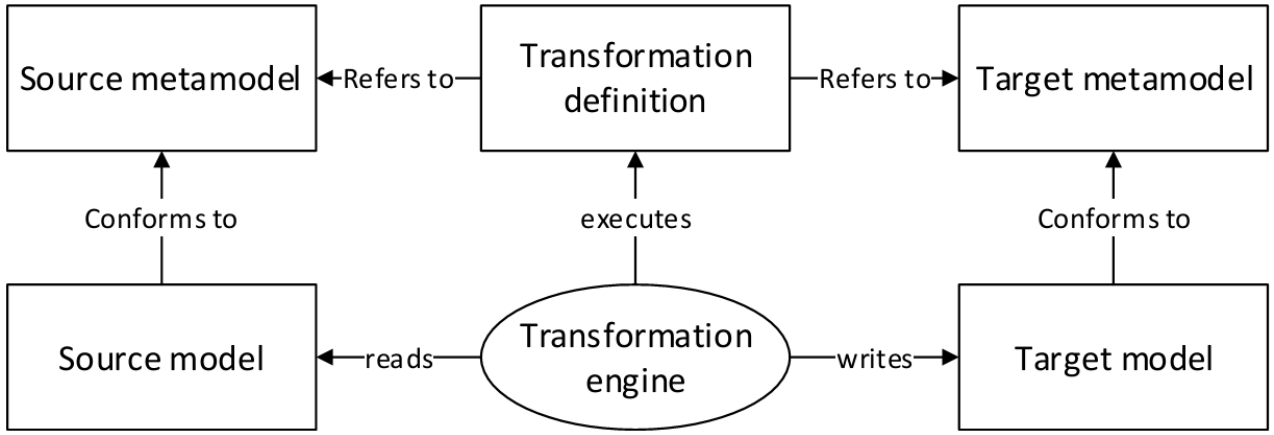




Blackbox Behavior Mapping



Technical Transformation Alternatives



Direct-Manipulation	Relational/ Declarative	Operational/ Imperative	Hybrid
• Multi-purpose programming language • e.g. JAVA	• „Constraint solving“ • e.g. QVT-R	• e.g. Xtend	• Declarative and imperative • e.g. ATL, ETL

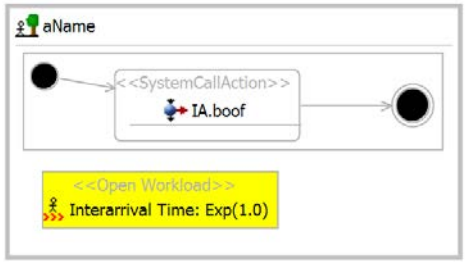
Evaluation

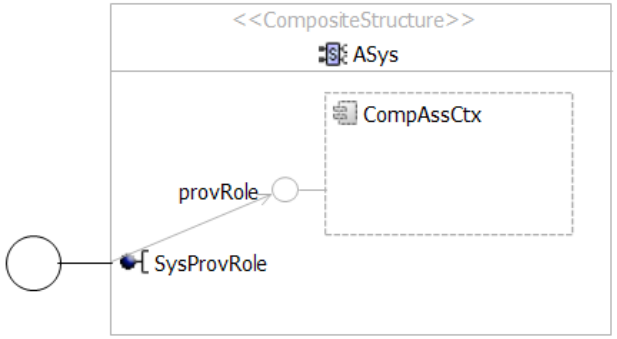
26 Unit Tests

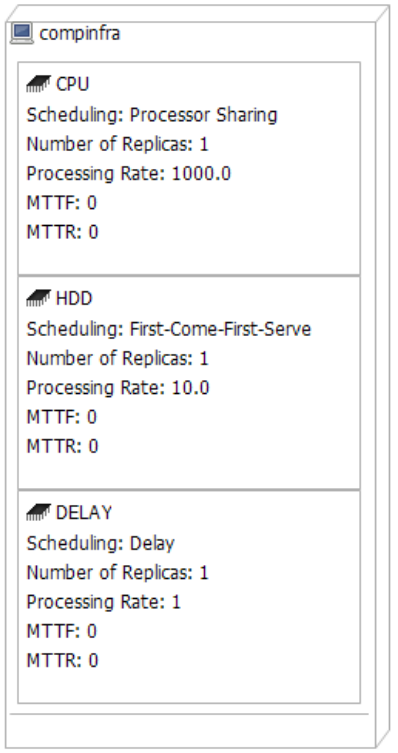
Components	Behavior	Interface	Resource Landscape
BasicComponent	BlackBoxBehavior	Interface	LinkingResource
ComposedStructure	FineGrainedBehavior	Signature	ComputingInfrastructure
AssemblyConnector	ReleaseAction	Parameter	
ProvidingDelegation Connector	AquireAction	PrimitiveDataType	
RequiringDelegation Connector	ExternalCallAction	ComposedDataType	
AssemblyContext	InternalCallAction		

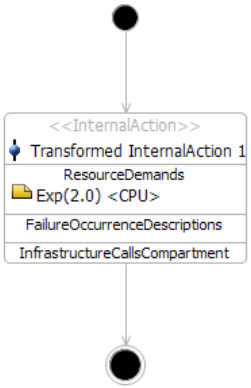
StoEx
BoolPMF, DoublePMF, EnumPMF, BoxedPDF, variableExpression, Term, Multiplication

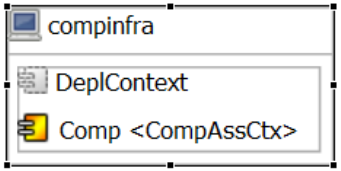
Case-study

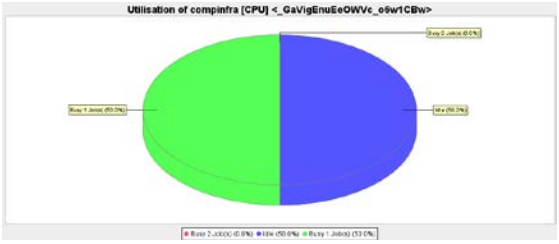

aName
<<SystemCallAction>>
IA.boof
<<Open Workload>>
Interarrival Time: Exp(1.0)


<<CompositeStructure>>
ASys
CompAssCtx
provRole
SysProvRole


compinfra
CPU
Scheduling: Processor Sharing
Number of Replicas: 1
Processing Rate: 1000.0
MTTF: 0
MTTR: 0
HDD
Scheduling: First-Come-First-Serve
Number of Replicas: 1
Processing Rate: 10.0
MTTF: 0
MTTR: 0
DELAY
Scheduling: Delay
Number of Replicas: 1
Processing Rate: 1
MTTF: 0
MTTR: 0


<<InternalAction>>
Transformed InternalAction 1
ResourceDemands
Exp(2.0) <CPU>
FailureOccurrenceDescriptions
InfrastructureCallsCompartment


compinfra
DeplContext
Comp <CompAssCtx>


Utilisation of compinfra [CPU]

Summary

Problem	• Tooling has to be developed for each formalism • Comparison challenging and time consuming • Manual transformation error-prone
Idea	• Model transformation of DML to PCM
Benefit	• Reuse of existing PCM tooling for DML • Improved understanding of differences and similarities • Simplified change of formalism
Action	• Implement a model-to-model transformation

THANK YOU

https://se3.informatik.uni-wuerzburg.de/descartes/dml2pcm

Literature

F. Brosig, N. Huber, and S. Kounev, "Architecture-level software performance abstractions for online performance prediction," *Science of Computer Programming*, vol. 90, pp. 71–92, Sep 2014. [Online]. Available: http://dx.doi.org/10.1016/j.scico.2013.06.004

S. Becker, H. Koziolek, and R. Reussner, "The palladio component model for model-driven performance prediction," *Journal of Systems and Software*, vol. 82, no. 1, pp. 3–22, Jan 2009. [Online]. Available: http://dx.doi.org/10.1016/j.jss.2008.03.066

F. Brosig, "Architecture-level software performance models for online performance prediction," Ph.D. dissertation, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, 2014. [Online]. Available: http://nbn-resolving.org/urn:nbn:de:swb:90-435372

N. Huber, F. Brosig, and S. Kounev, "Modeling dynamic virtualized resource landscapes," in *Proceedings of the 8th International ACM SIGSOFT Conference on Quality of Software Architectures*, ser. QoSA '12. New York, NY, USA: ACM, 2012, pp. 81–90. [Online]. Available: http://doi.acm.org/10.1145/2304696.2304711

N. R. Herbst, N. Huber, S. Kounev, and E. Amrehn, "Self-adaptive workload classification and forecasting for proactive resource provisioning," *Concurrency and Computation: Practice and Experience*, vol. 26, no. 12, pp. 2053–2078, Mar 2014. [Online]. Available: http://dx.doi.org/10.1002/cpe.3224

S. Kounev, F. Brosig, and N. Huber, "The Descartes Modeling Language," Department of Computer Science, University of Wuerzburg, Tech. Rep., October 2014. [Online]. Available: http://www.descartes-research.net/dml/

H. Koziolek, "Parameter dependencies for reusable performance specifications of software components," Ph.D. dissertation, University of Oldenburg, Germany, March 2008.

L. Schubert, "An evaluation of model transformation languages for uml quality engineering," Master's thesis, Masterarbeit im Studiengang Angewandte Informatik am Institut für Informatik, ZFI-MSC-2010-01, ISSN 1612-6793, Zentrum für Informatik, Georg-August-Universität Göttingen, 2010., Mar. 2010.