Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes
research

Model-Based Self-Aware Performance and Resource Management using the Descartes Modeling Language

Samuel Kounev

Chair of Software Engineering
University of Würzburg

http://se.informatik.uni-wuerzburg.de/

23.02.2017

Selected References

- N. Huber, F. Brosig, S. Spinner, S. Kounev, and M. Bähr. **Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language**. *IEEE Transactions on Software Engineering (TSE)*, PP(99), 2017, IEEE Computer Society. To appear. [pdf | DOI | http]
- S. Kounev, N. Huber, F. Brosig, and X. Zhu. **A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures**. *IEEE Computer*, 49(7):53–61, July 2016, IEEE. [pdf | DOI | http]
- S. Kounev, F. Brosig, and N. Huber. **The Descartes Modeling Language**. Technical report, Department of Computer Science, University of Wuerzburg, October 2014. [http | http | .pdf]
- F. Brosig, N. Huber, and S. Kounev. **Architecture-Level Software Performance Abstractions for Online Performance Prediction**. *Elsevier Science of Computer Programming Journal (SciCo)*, Vol. 90, Part B:71-92, 2014, Elsevier. [DOI | http | .pdf]
- N. Huber, A. van Hoorn, A. Koziolek, F. Brosig, and S. Kounev. **Modeling Run-Time Adaptation at the System Architecture Level in Dynamic Service-Oriented Environments**. *Service Oriented Computing and Applications Journal (SOCA)*, 8(1):73-89, 2014, Springer-Verlag. [DOI | .pdf]
- F. Brosig, P. Meier, S. Becker, A. Koziolek, H. Koziolek, and S. Kounev. **Quantitative Evaluation of Model-Driven Performance Analysis and Simulation of Component-based Architectures**. *IEEE Transactions on Software Engineering (TSE)*, 41(2):157-175, February 2015, IEEE. [DOI | http | .pdf]
- F. Gorsler, F. Brosig, and S. Kounev. **Performance Queries for Architecture-Level Performance Models**. In *5th ACM/SPEC International Conference on Performance Engineering (ICPE 2014)*, Dublin, Ireland, 2014. ACM, New York, NY, USA. 2014. [DOI | .pdf]
- N. Herbst, N. Huber, S. Kounev and E. Amrehn. **Self-Adaptive Workload Classification and Forecasting for Proactive Resource Provisioning**. *Concurrency and Computation - Practice and Experience, John Wiley and Sons, Ltd.*, 26(12):2053-2078, 2014. [DOI | http | .pdf]
- S. Spinner, G. Casale, F. Brosig, and S. Kounev. **Evaluating Approaches to Resource Demand Estimation**. *Performance Evaluation*, 92:51 - 71, October 2015, Elsevier B.V. [DOI | http | .pdf]
- *N. Herbst, S. Kounev and R. Reussner.* ***Elasticity: What it is, and What it is Not.*** *In 10th Intl. Conference on Autonomic Computing (ICAC 2013), San Jose, CA, June 24-28, 2013.* [slides | http | .pdf]
- A. Milenkoski, M. Vieira, S. Kounev, A. Avrtizer, and B. Payne. **Evaluating Computer Intrusion Detection Systems: A Survey of Common Practices**. *ACM Computing Surveys*, 48(1):12:1-12:41, September 2015, ACM, New York, NY, USA. **5-year Impact Factor (2014): 5.949**. [http]
- A. Milenkoski, K. R. Jayaram, N. Antunes, M. Vieira, and S. Kounev. Quantifying the Attack Detection Accuracy of Intrusion Detection Systems in Virtualized Environments. In Proceedings of The 27th IEEE International Symposium on Software Reliability Engineering (ISSRE 2016), Ottawa, Canada, October 2016. IEEE, IEEE Computer Society, Washington DC, USA. October 2016.

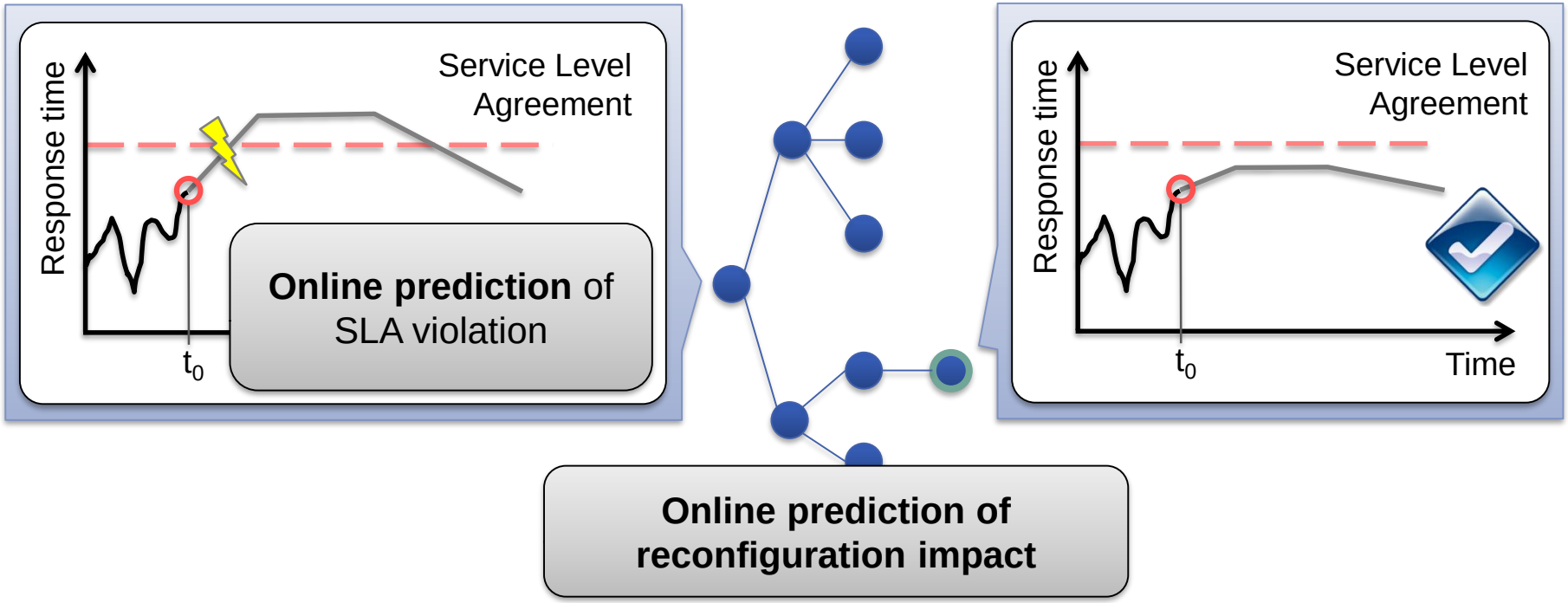
Latest Publications on DML

S. Kounev, N. Huber, F. Brosig, and X. Zhu. ***A Model-Based Approach to Designing Self-Aware IT Systems and Infrastructures***. IEEE Computer, 49(7):53–61, July 2016.

N. Huber, F. Brosig, S. Spinner, S. Kounev, and M. Bähr. ***Model-Based Self-Aware Performance and Resource Management Using the Descartes Modeling Language***. IEEE Transactions on Software Engineering (TSE), PP(99), 2017.

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING

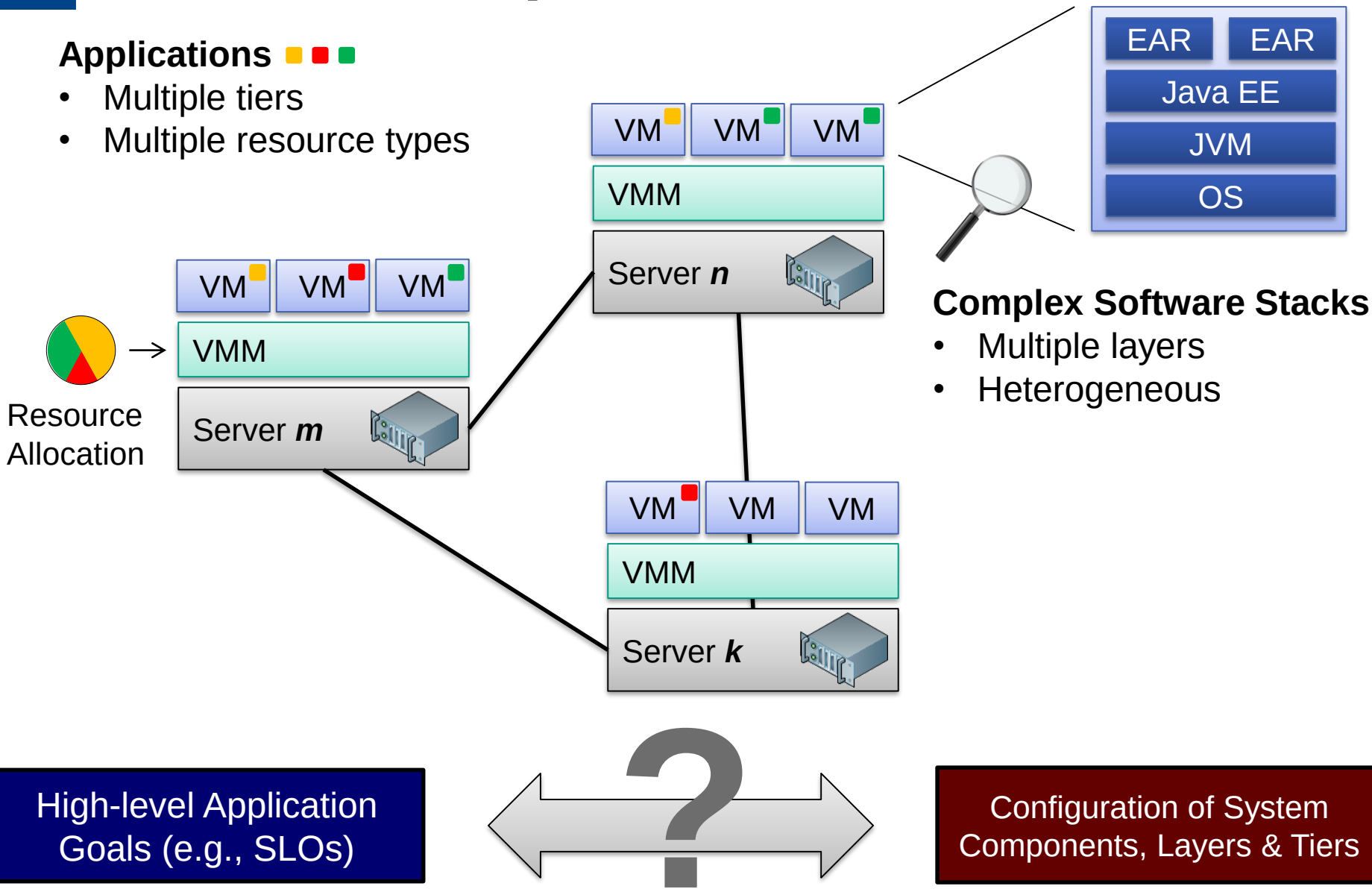
Self-Aware Performance and Resource Management: Example Scenario



→ **Example Scenario for Self-Aware Computing** (more later)

Semantic Gap Problem

Applications
- Multiple tiers
- Multiple resource types



Complex Software Stacks
- Multiple layers
- Heterogeneous

High-level Application Goals (e.g., SLOs)

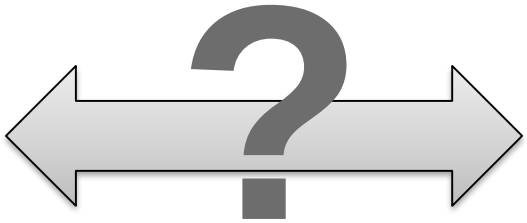
Configuration of System Components, Layers & Tiers

Semantic Gap Problem

- **Availability & Performance**
  - Services available 99.99% of the time
  - Response time of service x < 20 ms
  - Transaction throughput > 1000
  - Server utilization > 60% on average
  - „Time to recover after a failure“ < 1 min
- **Efficiency**
  - Allocate only as much resources as are actually needed
- ...

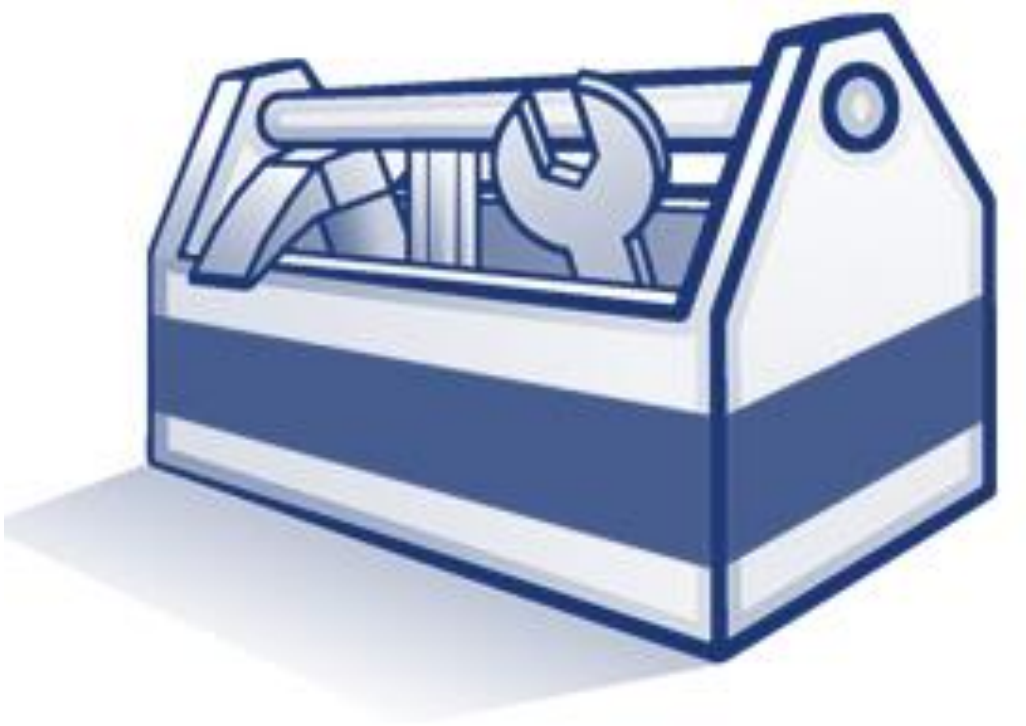
- How many vCPUs to allocate to virtual machine (VM) n?
- How much memory to allocate to VM n?
- When exactly should a reconfiguration be triggered?
- Which particular resources or services should be scaled / replicated / migrated / restarted?
- How quickly and at what granularity?

Service level objectives (SLOs)



Configuration of System Components, Layers & Tiers

Descartes Tool Chain



http://descartes.tools

Descartes Tool Chain

Descartes Modeling Language:

- DML (Descartes Modeling Language)
- DNI (Descartes Network Infrastructures Modeling)

Workload Characterization & Model Extraction:

- LIMBO Load Intensity Modeling Tool
- WCF (Workload Classification and Forecasting Tool)
- LibReDE (Library for Resource Demand Estimation)
- SPA (Storage Performance Analyzer)
- PMX (Performance Model eXtractor)

Declarative Performance Engineering:

- DQL (Descartes Query Language)

Benchmarking:

- BUNGEE Cloud Elasticity Benchmark
- hInjector Hypercall Attack Injector

Stochastic Modeling:

- QPME (Queueing Petri net Modeling Environment)

Black-Box Modeling:

- Univariate Interpolation Library

http://descartes.tools

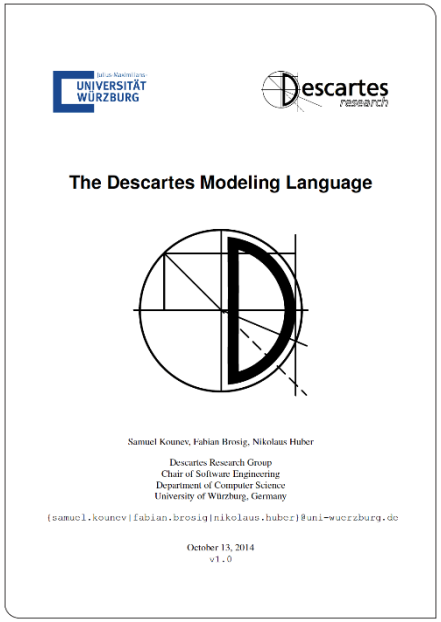
Mailing list available...

Selected Tools

- **DML** – Descartes Modeling Language (homepage, publications)
- **DML Bench** (homepage, publications)
- **DQL** – Declarative performance query language (homepage, publications)
- **LibReDE** - Library for resource demand estimation (homepage, publications)
- **LIMBO** – Load intensity modeling tool (homepage, publications)
- **WCF** – Workload classification & forecasting tool (homepage, publications)
- **BUNGEE** – Elasticity benchmarking framework (homepage, publications)
- **hInjector** – Security benchmarking tool (homepage, publications)
- Queueing Petri Net Modeling Environment (QPME)
- **Further relevant research**
  - **http://descartes-research.net/research/research_areas/**
  - **Self Aware Computing** (publications)

Descartes Modeling Language (DML)

- Architecture-level modeling language for modeling QoS and resource management related aspects of IT systems and infrastructures
  - Prediction of the impact of dynamic changes at run-time
  - Current version focused on performance including capacity, responsiveness and resource efficiency aspects



The Descartes Modeling Language

Samuel Kounev, Fabian Brosig, Nikolaus Huber

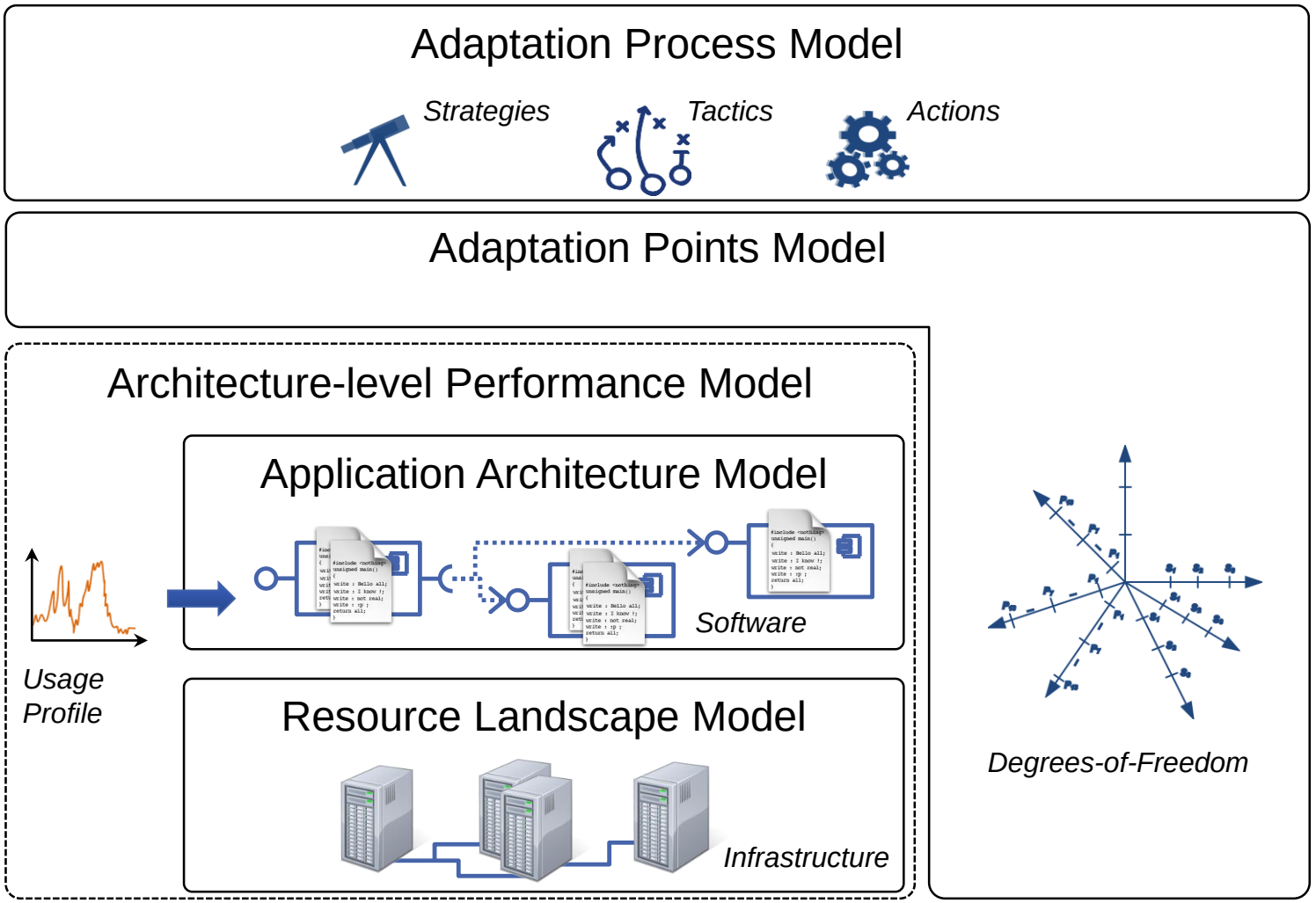
Descartes Research Group
Chair of Software Engineering
Department of Computer Science
University of Würzburg, Germany

{samuel.kounev|fabian.brosig|nikolaus.huber}@uni-wuerzburg.de

October 13, 2014
v1.0

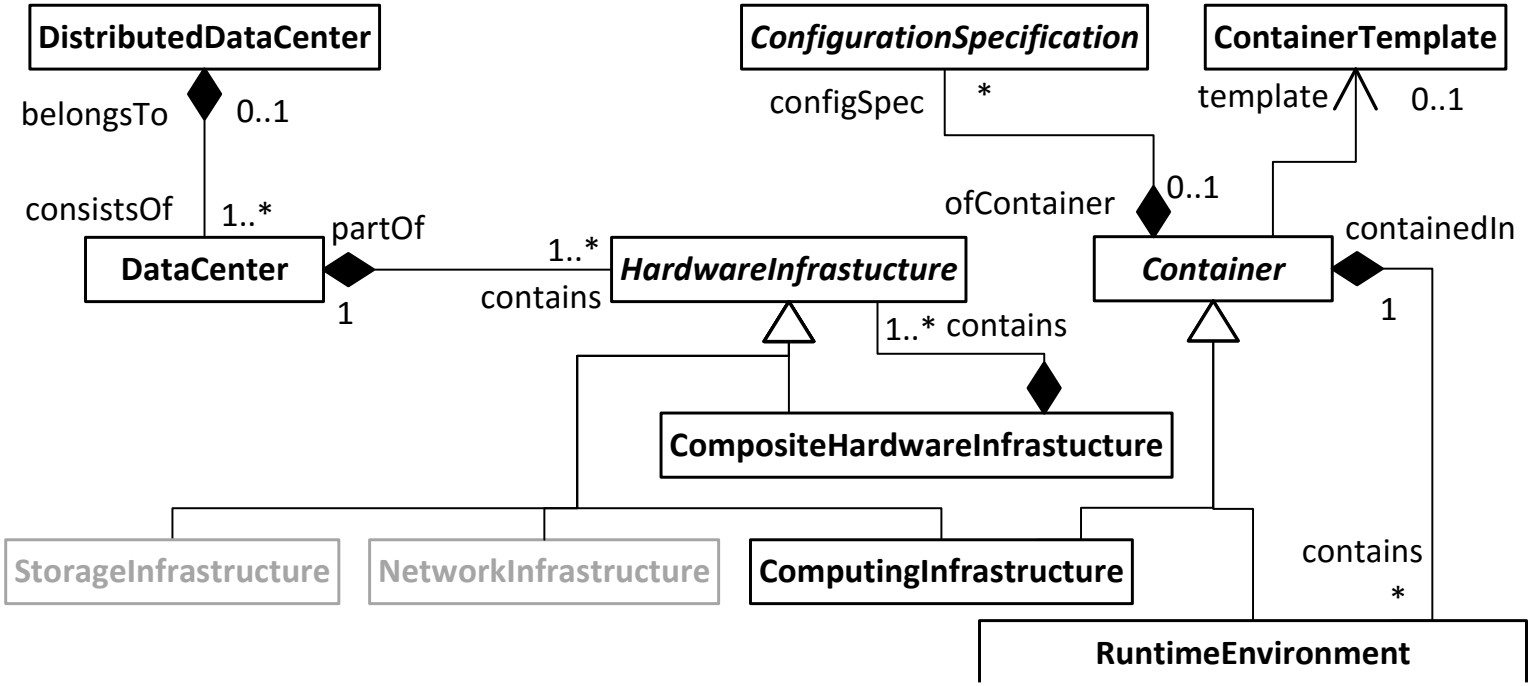
http://descartes.tools/dml

DML Sub-Models



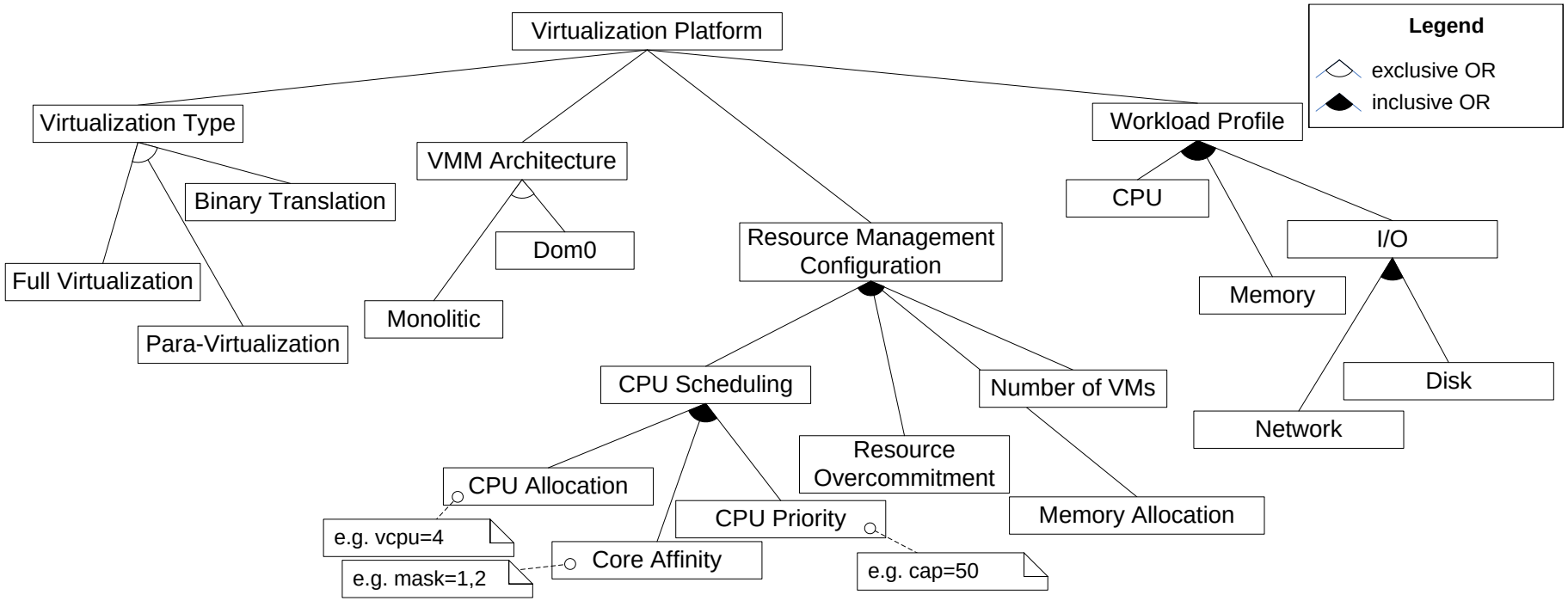
DML Implementation

- Implementation in Ecore (Eclipse Modeling Framework)
- Excerpt from meta-model: resource landscape



Example: Custom Configuration Model

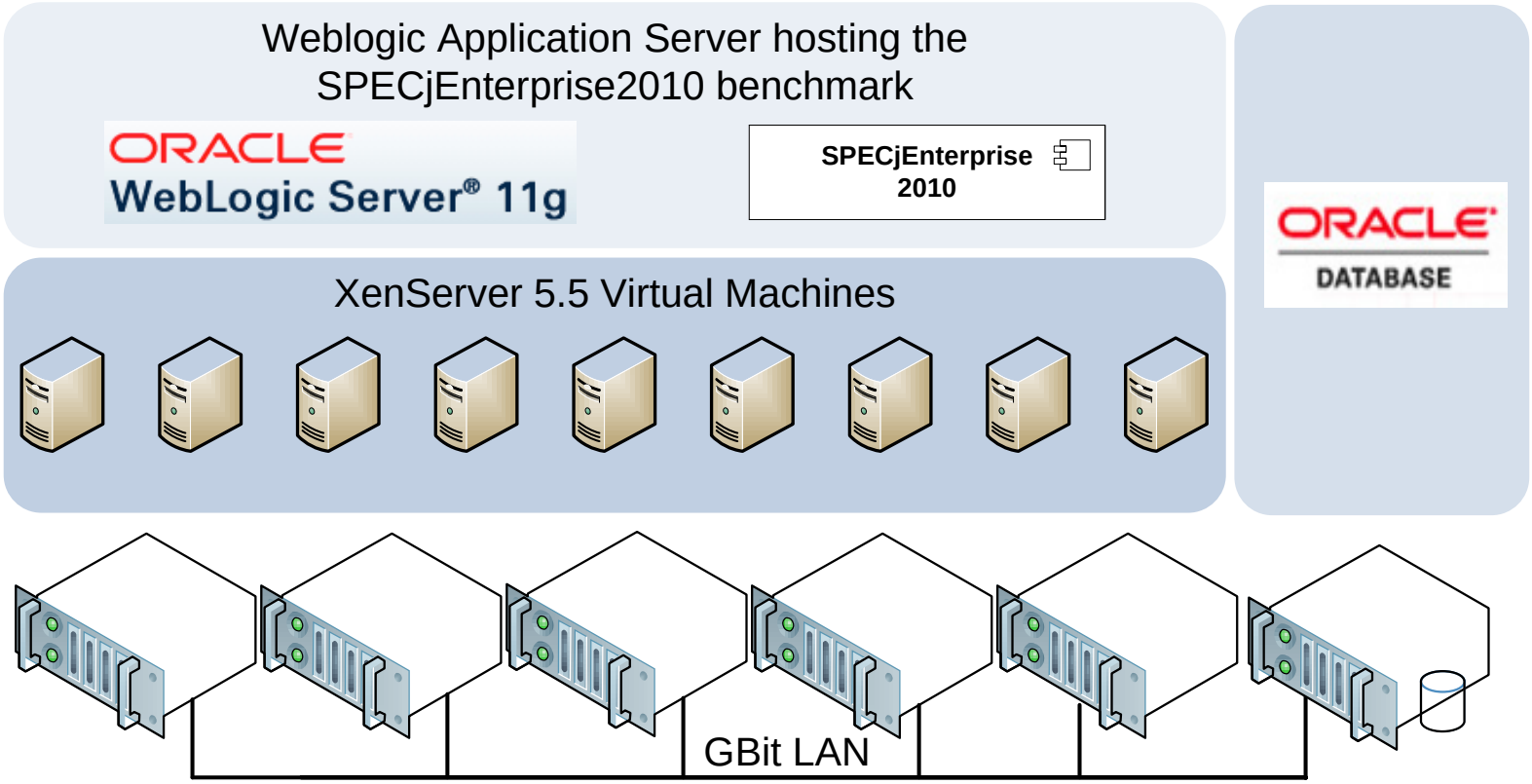
(Feature Model for the Virtualization Platform)



N. Huber, M. Quast, M. Hauck, and S. Kounev. **Evaluating and Modeling Virtualization Performance Overhead for Cloud Environments**. *International Conference on Cloud Computing and Services Science (CLOSER 2011), Noordwijkerhout, The Netherlands*, May 7-9, 2011. Best Paper Award.

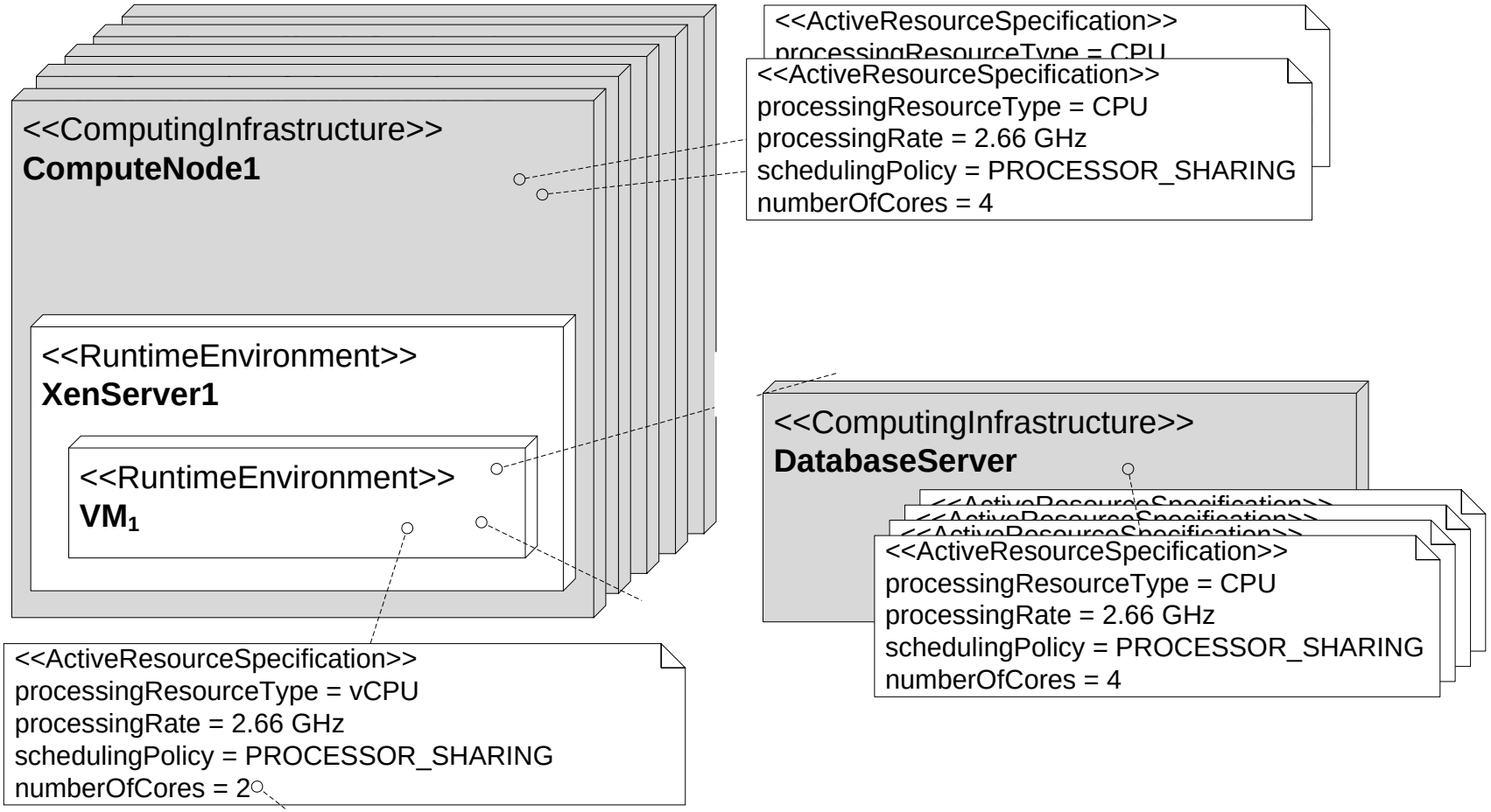
Example: WebLogic Server Cluster

(Resource Landscape)



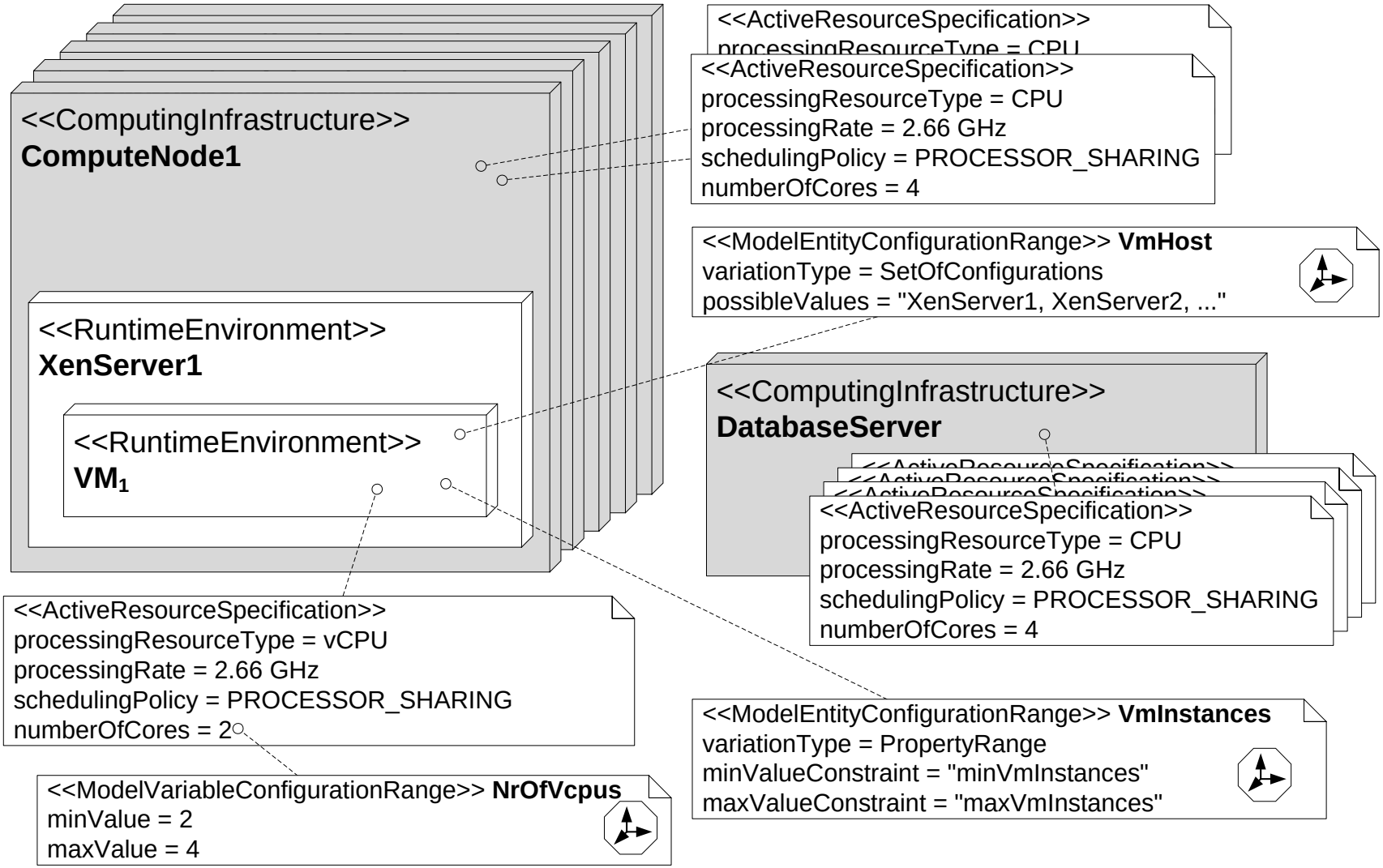
Example: WebLogic Server Cluster

(Resource Landscape Model)

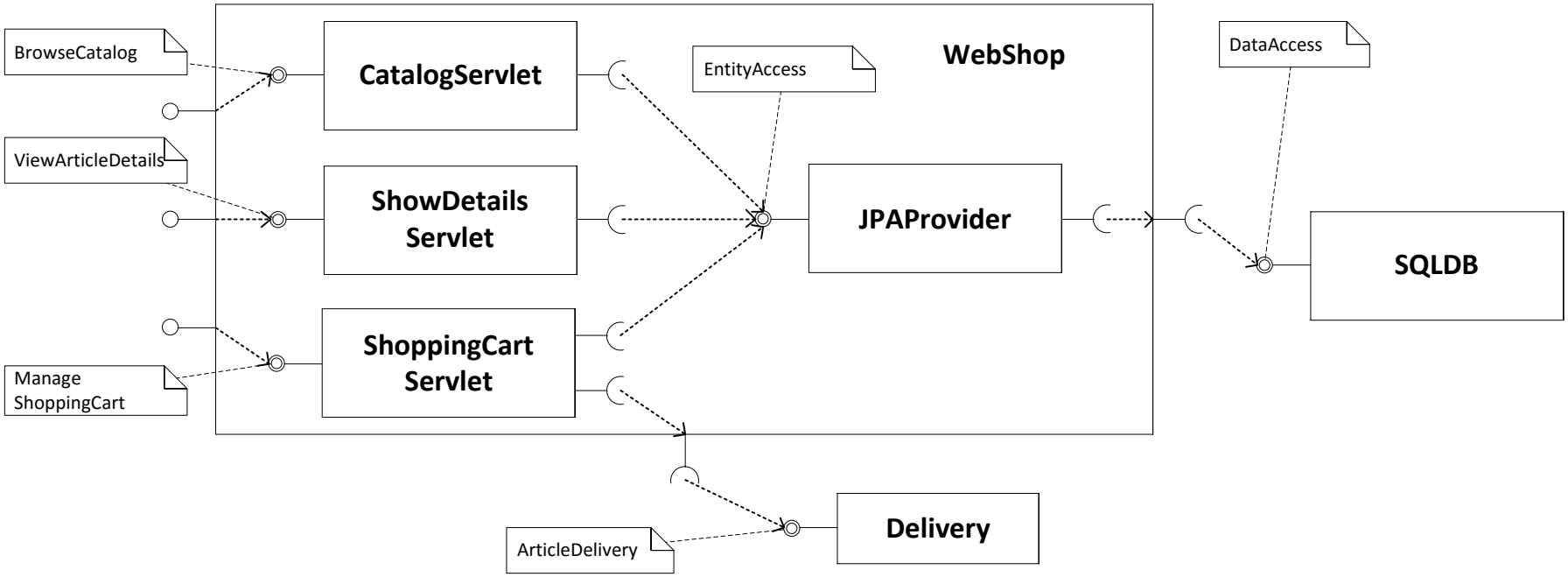


Example: WebLogic Server Cluster

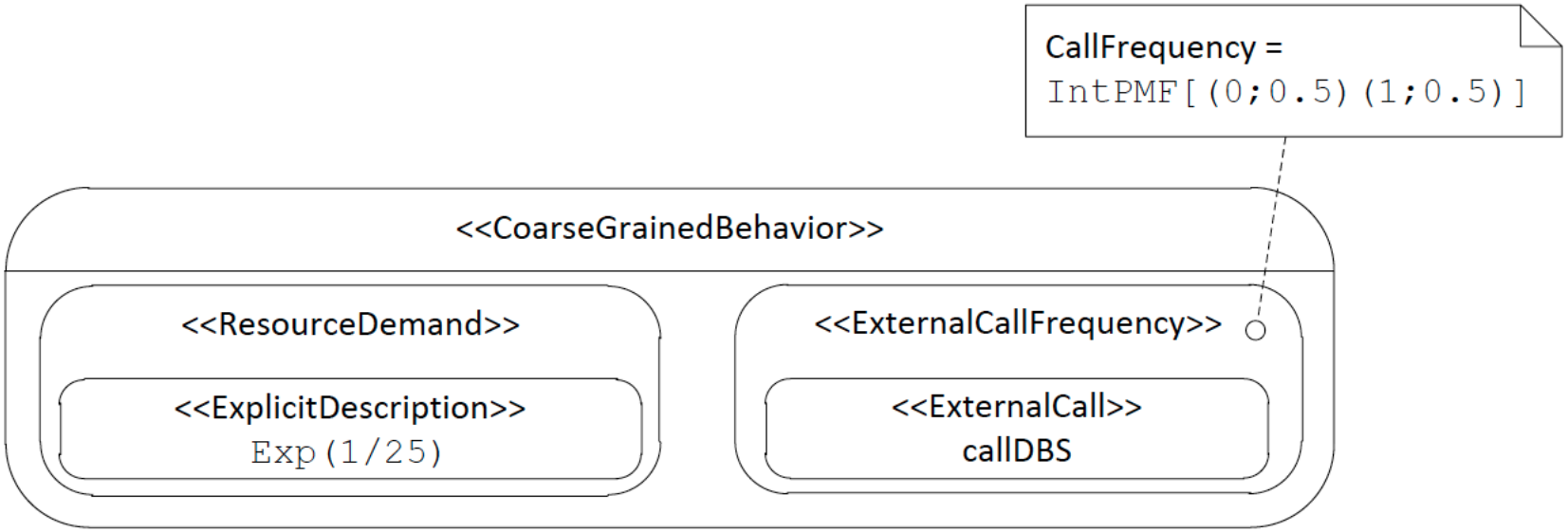
(Resource Landscape Model) + (Adaptation Points Model)



Example
(Application Architecture Model)

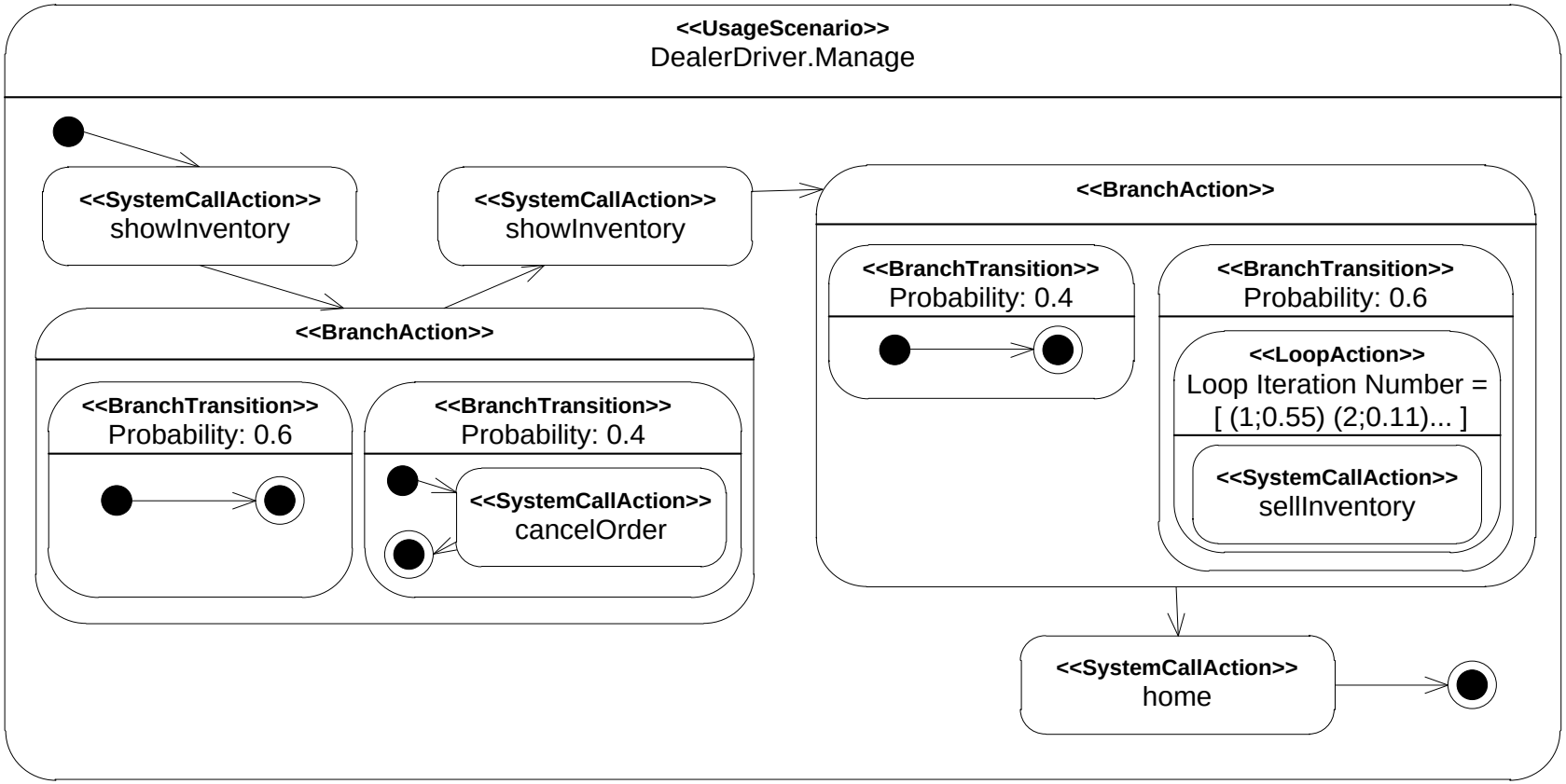


Example
(Coarse-Grained Service Behavior Model)



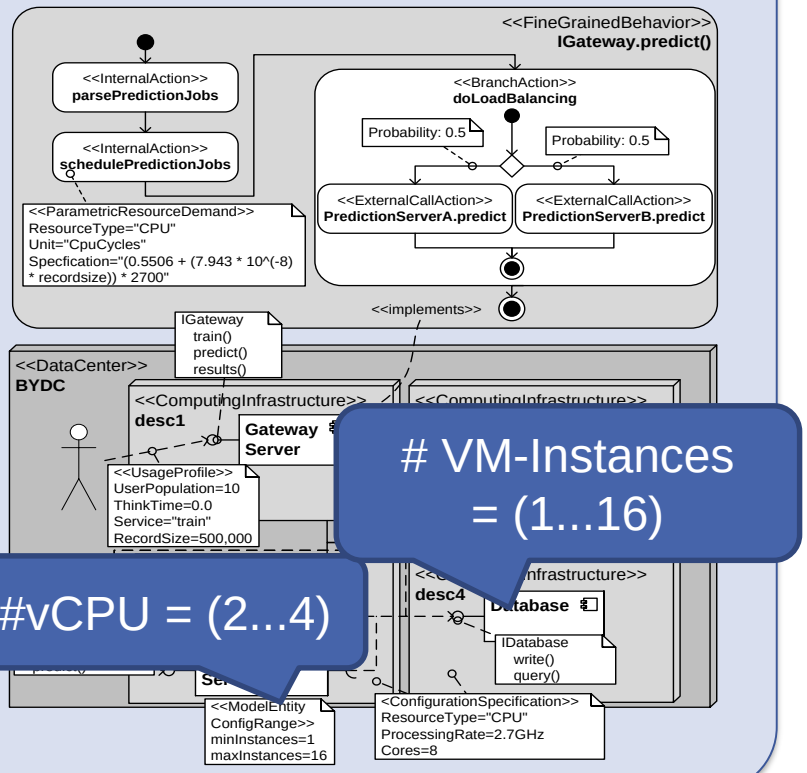
Example

(Fine-Grained Service Behavior Model)



Online Performance Prediction

Architecture-Level Performance Model



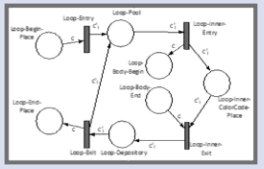
VM-Instances = (1...16)

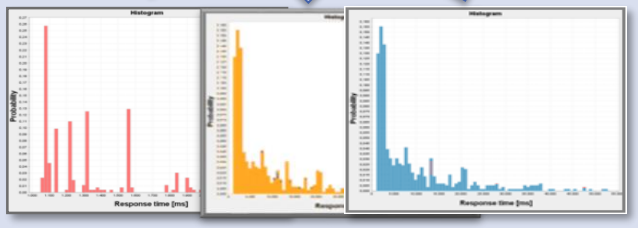
#vCPU = (2...4)

Online Performance Prediction

$$\overline{X} \leq min\left\{\frac{N}{\sum_{i=0}^{n} D_i^{sync}}, min_{1\leq i\leq n}\left\{\frac{1}{D_i}\right\}\right\}$$

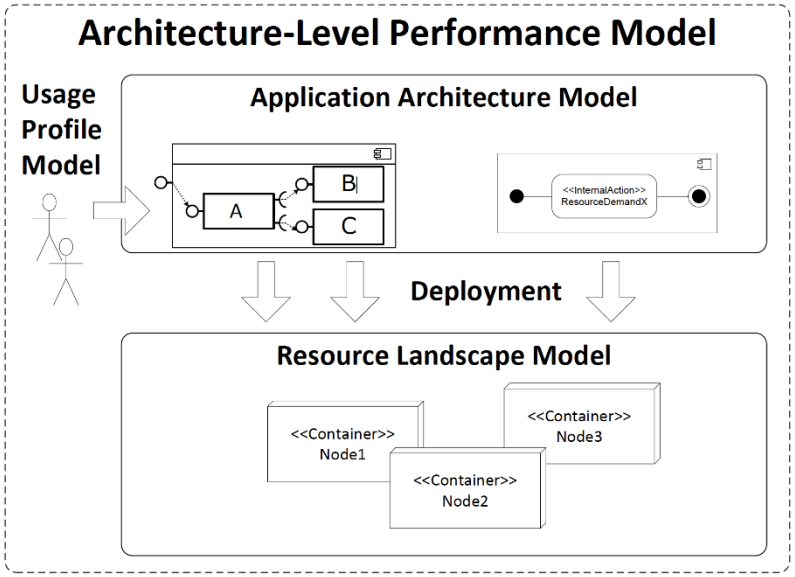
$$\overline{R} = \frac{N}{\overline{X}} \geq max\left\{\sum_{i=0}^{n} D_i^{sync}, N * max_{1\leq i\leq n}\{D_i\}\right\}$$



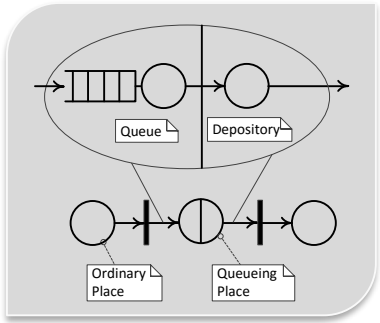


Autonomic Decision Making

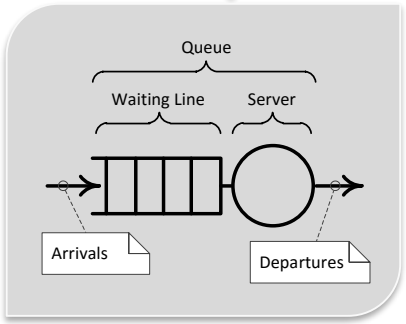
Transformations to Predictive Models



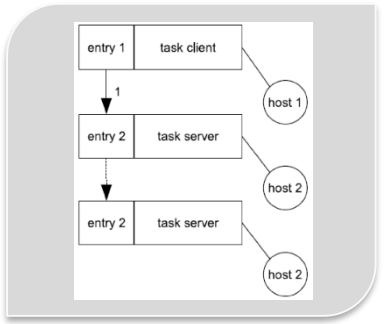
DML Instance



Queueing Petri Net

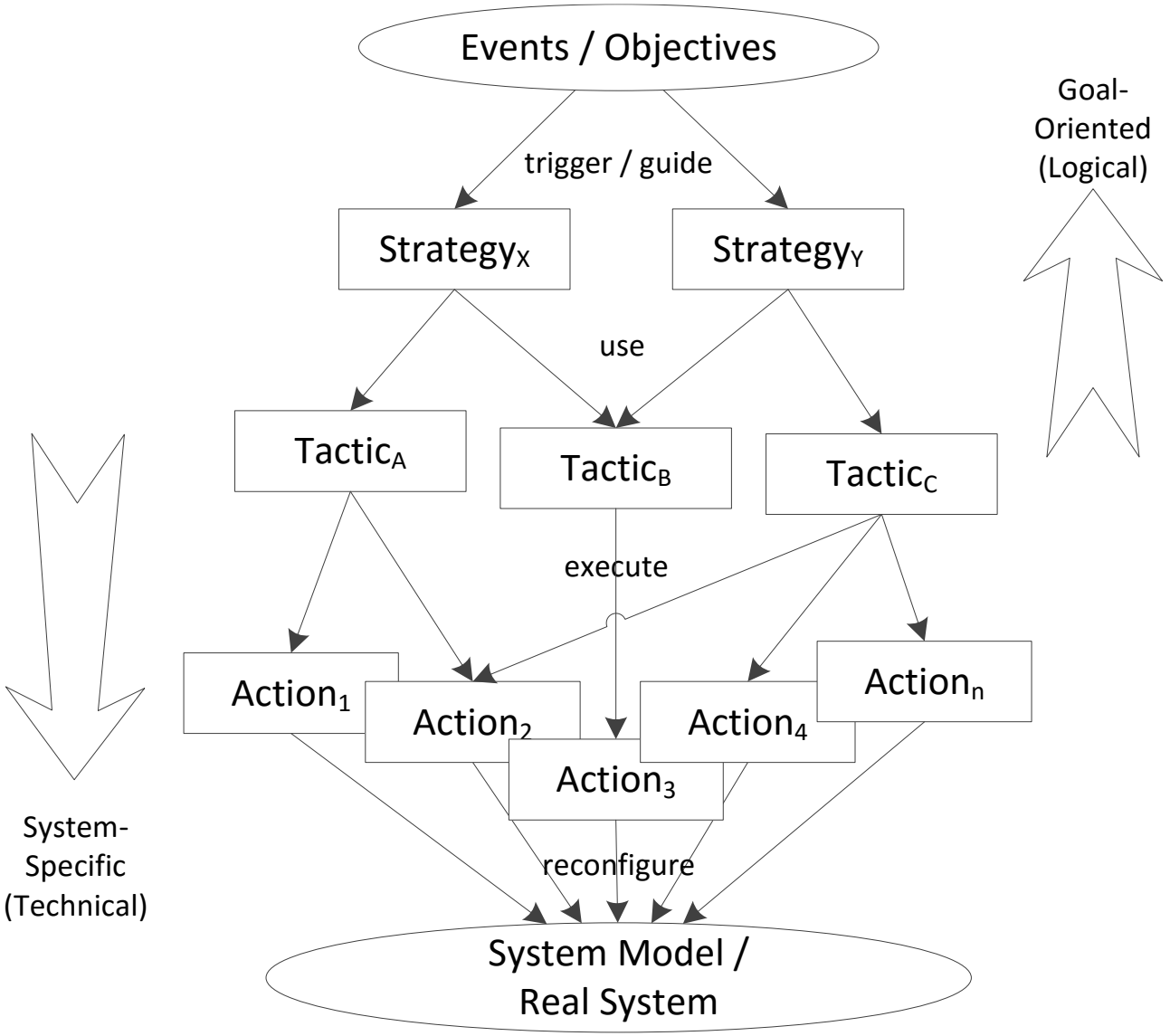


Bounds Analysis Model

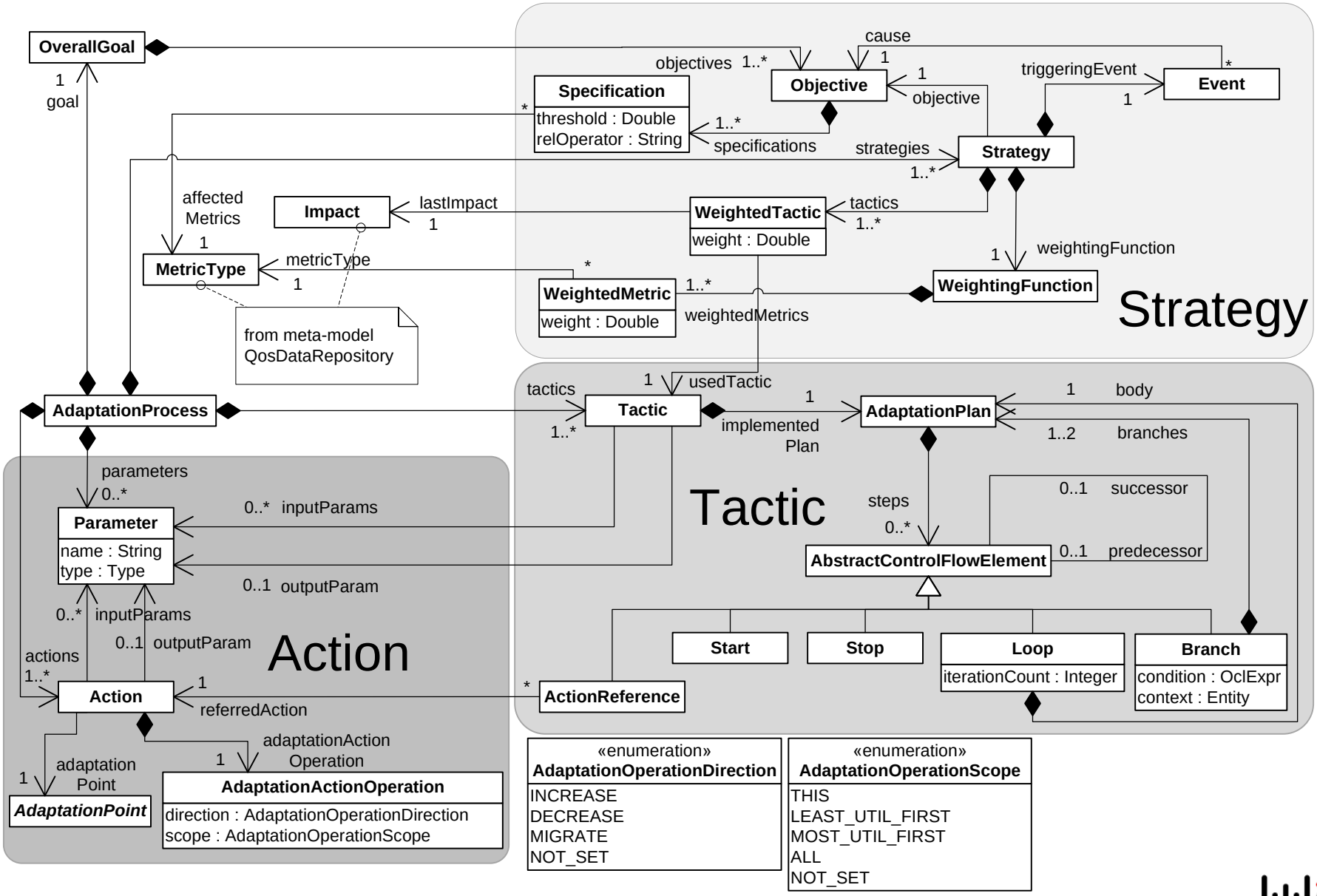


Layered Queueing Network

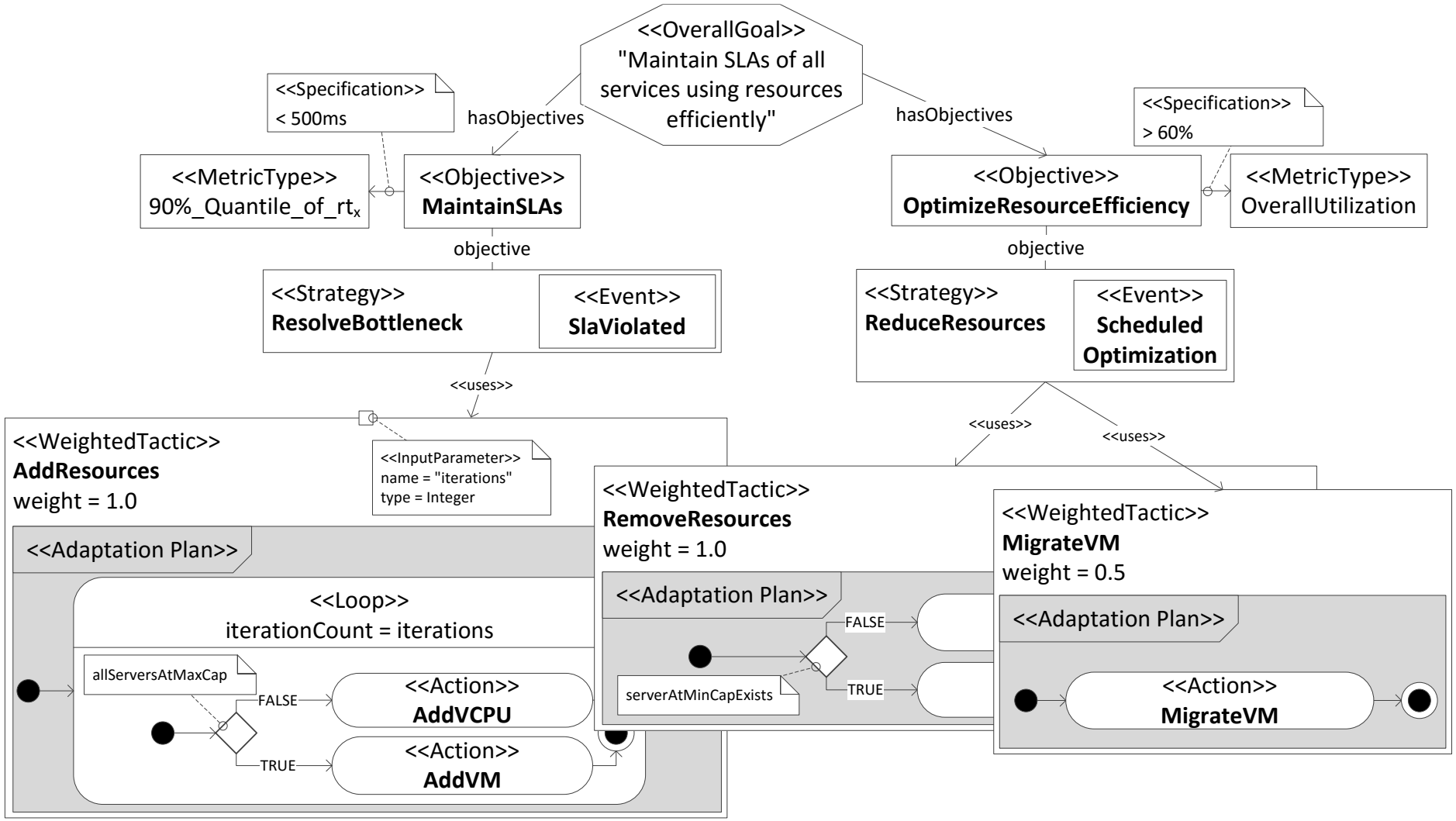
Adaptation Process Model



S/T/A Meta-Model (Strategies, Tactics and Actions)



Example: Adaptation Process Model



Descartes Tools

Descartes Modeling Language:

- DML (Descartes Modeling Language)
- DNI (Descartes Network Infrastructures Modeling)

Workload Characterization & Model Extraction:

- LIMBO Load Intensity Modeling Tool
- WCF (Workload Classification and Forecasting Tool)
- LibReDE (Library for Resource Demand Estimation)
- SPA (Storage Performance Analyzer)
- PMX (Performance Model eXtractor)

Declarative Performance Engineering:

- DQL (Descartes Query Language)

Benchmarking:

- BUNGEE Cloud Elasticity Benchmark
- hInjector Hypercall Attack Injector

Stochastic Modeling:

- QPME (Queueing Petri net Modeling Environment)

Black-Box Modeling:

- Univariate Interpolation Library

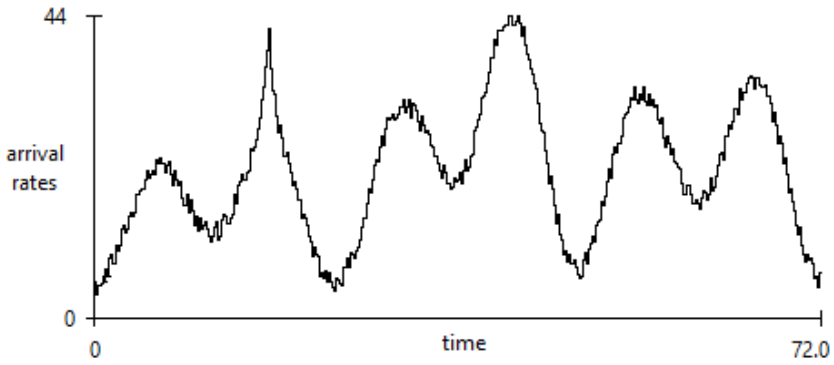
http://descartes.tools

Mailing list available...

LIMBO Tool

- Problem:
  - How to capture the load intensity variations (e.g., requests per sec) in a compact mathematical model?
  - How to forecast the load intensity (requests per sec) in future time horizons?

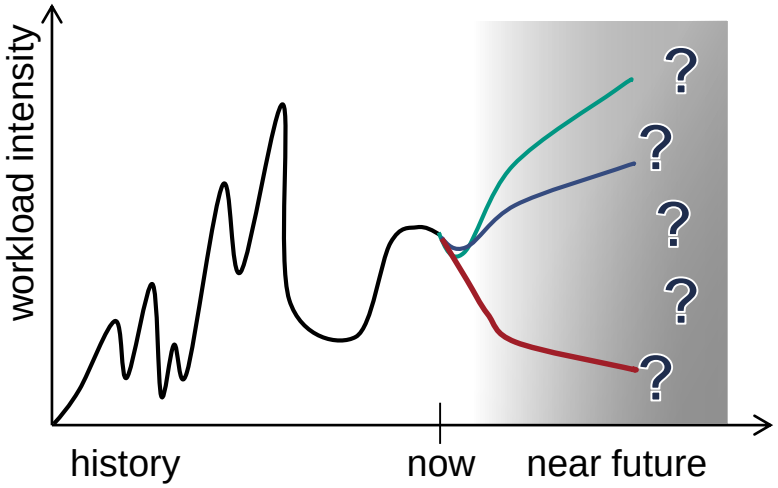
- **Load Intensity Modeling & Forecasting Tool**



http://descartes.tools/limbo

LIMBO Tool (2)

- **Workload Classification & Forecasting (WCF)**
  - Use of multiple alternative forecasting methods in parallel
  - Selection of method based on its accuracy in the past



http://descartes.tools/libmo
http://descartes.tools/wcf

IBM PHD FELLOWSHIP AWARD PROGRAMME

LibReDE Tool

- Problem: How to estimate the total service time of a given type of request/job at a given resource?

- **Library for Resource Demand Estimation**
  - Ready-to-use implementations of estimation approaches
  - Selection of a suitable approach for a given scenario

http://descartes.tools/librede

S. Spinner, G. Casale, F. Brosig, and S. Kounev. **Evaluating Approaches to Resource Demand Estimation**. *Performance Evaluation*, 92:51 - 71, October 2015, Elsevier B.V. [DOI | http | .pdf]

Summary

- **Today:** 1st Generation Cloud Computing
  - **Simple trigger/rule-based mechanisms**
  - Best effort approach
  - No performance and availability guarantees
- **Novel model-based approaches** enable **self-aware** performance and resource management
  - proactive and predictable approach

Dagstuhl-Seminar

Model-driven Algorithms and Architectures for Self-Aware Computing Systems, Jan 18-23, 2015, Dagstuhl Seminar 15041

Organizers

Jeffrey O. Kephart (IBM TJ Watson Research Center, US)
Samuel Kounev (Universität Würzburg, DE)
Marta Kwiatkowska (University of Oxford, GB)
Xiaoyun Zhu (VMware, Inc., US)

Community:
http://descartes.tools/self-aware
Dagstuhl Report:
http://drops.dagstuhl.de/opus/volltexte/2015/5038/
Seminar Page:
http://www.dagstuhl.de/15041

Definition

Self-aware Computing Systems are computing systems that:

1. ***learn models*** capturing knowledge about themselves and their environment ***on an ongoing basis*** and

2. ***reason*** using the models enabling them to ***act*** based on their knowledge and reasoning

in accordance with ***higher-level goals***, which may also be subject to change.

S. Kounev, P. Lewis, K. Bellman, N. Bencomo, J. Camara, A. Diaconescu, L. Esterle, K. Geihs, H. Giese, S. Goetz, P. Inverardi, J. Kephart and A. Zisman. **The Notion of Self-Aware Computing**. In Self-Aware Computing Systems, S. Kounev, J. O. Kephart, A. Milenkoski, and X. Zhu, editors. Springer Verlag, Berlin Heidelberg, Germany, 2017.

Extended Definition

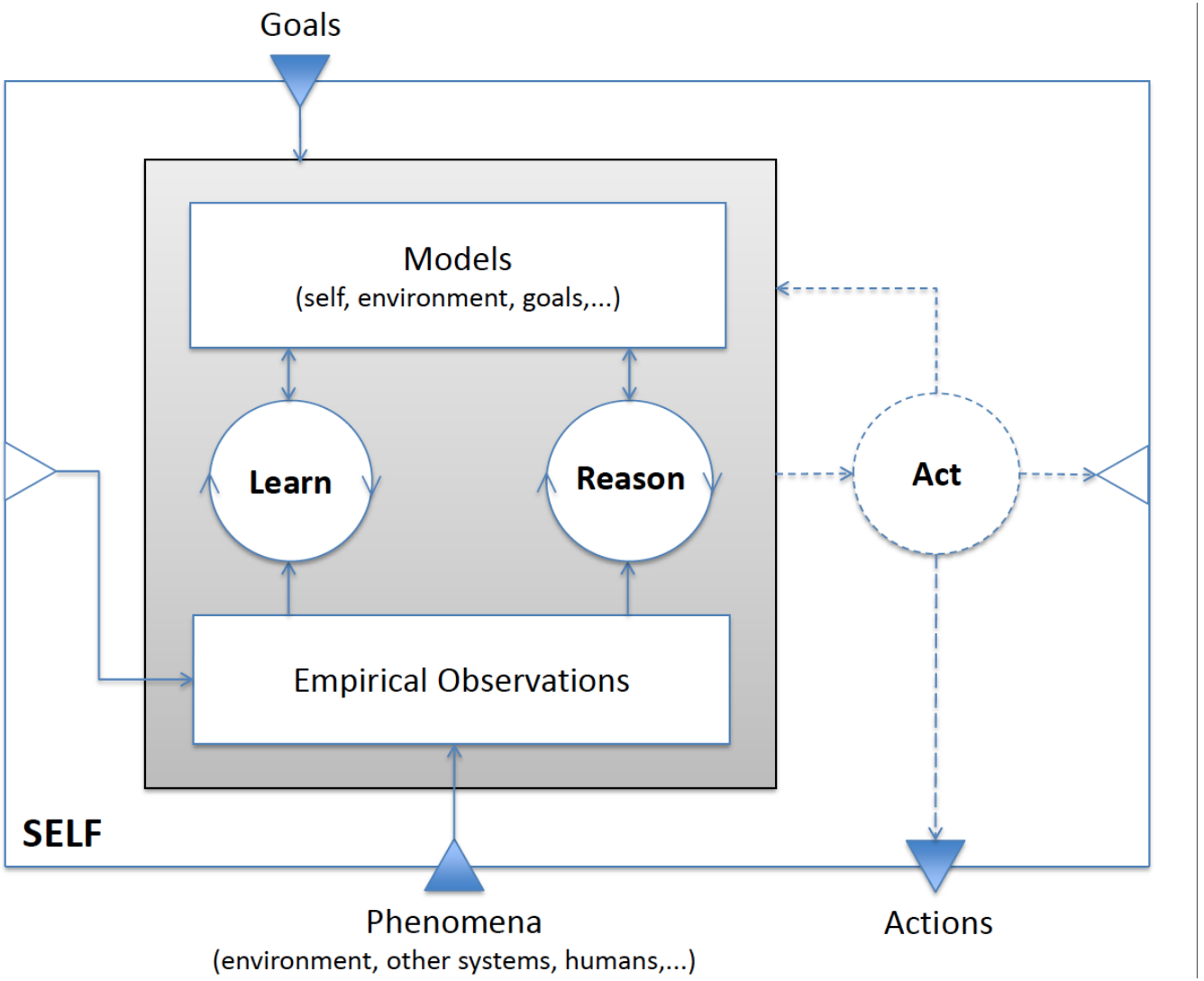
Self-aware Computing Systems are computing systems that:

1. ***learn models*** capturing ***knowledge*** about themselves and their environment (such as their structure, design, state, possible actions, and run-time behavior)
   on an ongoing basis and

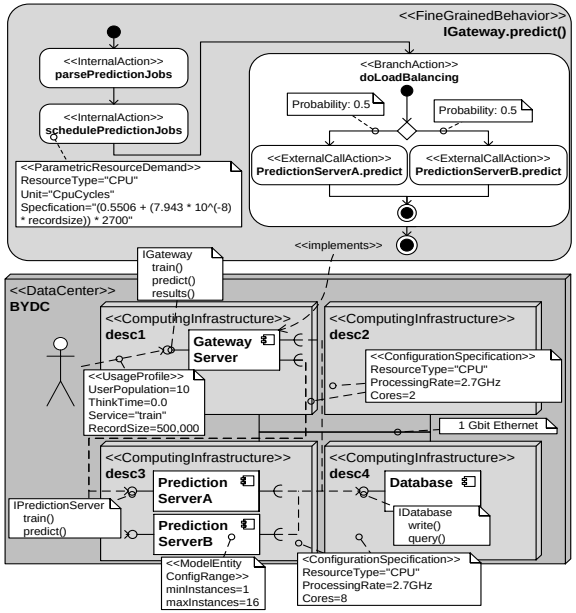
2. ***reason*** using the models (for example predict, analyze, consider, plan) enabling them to ***act*** based on their knowledge and reasoning (for example explore, explain, report, suggest, self-adapt, or impact their environment)

in accordance with ***higher-level goals***, which may also be subject to change.

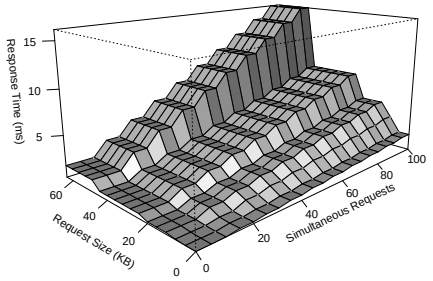
Self-Aware Learning & Reasoning Loop



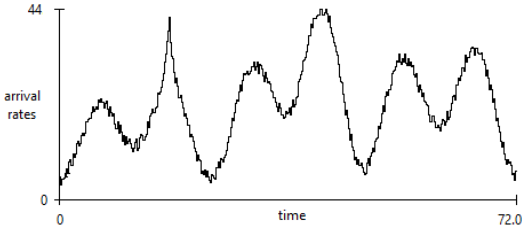
Examples of Models



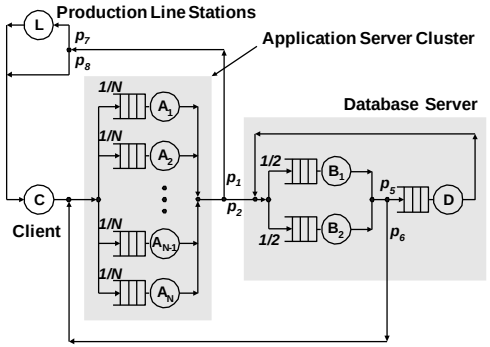
Descriptive MOF-based models



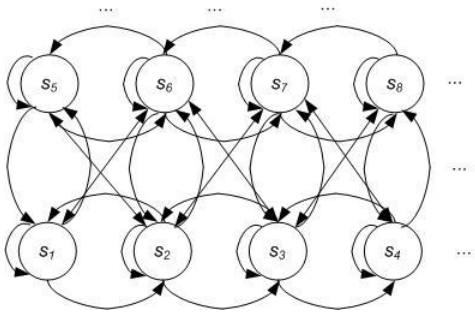
Statistical regression models



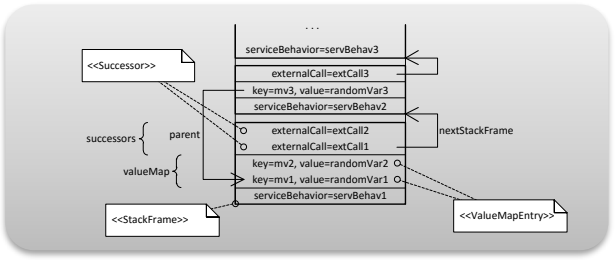
Load forecasting models



Queueing network models



Markov models



Simulation models

$$R \geq \max\left[N \times \max\{D_i\}, \sum_{i=1}^{K} D_i \right] \qquad X_0 \leq \min\left[\frac{1}{\max\{D_i\}}, \frac{N}{\sum_{i=1}^{K} D_i} \right]$$

$$\frac{N}{\max\{D_i\}[K+N-1]} \leq X_0 \leq \frac{N}{avg\{D_i\}[K+N-1]}$$

Analytical analysis models

New Book

- **„Self-Aware Computing Systems“**

  Samuel Kounev (University of Würzburg, DE)

  Jeffrey O. Kephart (IBM T.J. Watson, USA)

  Aleksandar Milenkoski (University of Würzburg, DE)

  Xiaoyun Zhu (Futurewei Technologies, Huawei, USA)

- 27 chapters, ca 700 pages, ca. 50 authors involved

S. Kounev, J. O. Kephart, A. Milenkoski, and X. Zhu. (eds.)
Self-Aware Computing Systems. Springer Verlag, Berlin Heidelberg, Germany, 2017. http://www.springer.com/de/book/9783319474724

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Descartes
research

Questions?

skounev@acm.org
http://descartes.tools