



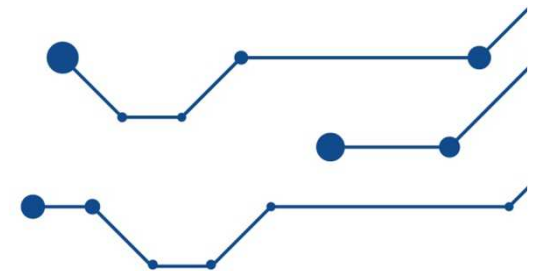
Dublin, 2014-03-22

# Using Performance Models to Support Load Testing in a Large SOA Environment

Industrial Track

Christian Vögele

fortiss GmbH  
An-Institut Technische Universität München



# Agenda

1. Introduction
2. Motivation
3. Project Context
4. Using Performance Models to Support Load Testing
  - Palladio Component Models
  - Generating Palladio Component Models
  - Selecting Usage Scenarios
  - Transforming Performance Models
  - Predicting Service Workloads
  - Evaluation
5. Conclusion & Future Work

# Agenda

1. **Introduction**
2. Motivation
3. Project Context
4. Using Performance Models to Support Load Testing
  - Palladio Component Models
  - Generating Palladio Component Models
  - Selecting Usage Scenarios
  - Transforming Performance Models
  - Predicting Service Workloads
  - Evaluation
5. Conclusion & Future Work

# Introduction

## fortiss Performance Management Group

fortiss

- An-Institut Technische Universität München
- Application-oriented research institute
- Industry collaboration to improve the applicability of research results in practice



- Performance analysis and prediction
- Performance Management Work
- Focus on complex enterprise applications

# Agenda

1. Introduction
- 2. Motivation**
3. Project Context
4. Using Performance Models to Support Load Testing
  - Palladio Component Models
  - Generating Palladio Component Models
  - Selecting Usage Scenarios
  - Transforming Performance Models
  - Predicting Service Workloads
  - Evaluation
5. Conclusion & Future Work

# Motivation

**Complication:** *Challenges for load tests in large service-oriented architectures*

- Service workloads are hard to predict by service consumers
- Immature services are not yet in production and need to be scaled before load tests start
- Complex test data requirements due to a large variety of service providers and their heterogeneous data sets

**Resolution:** *We propose the use of performance models to support SOA load tests to ...*

- derive test scenarios
- predict workloads for service providers (services operations and their call frequency)
- derive service operations involved in a test

**Contribution:** *This information helps to...*

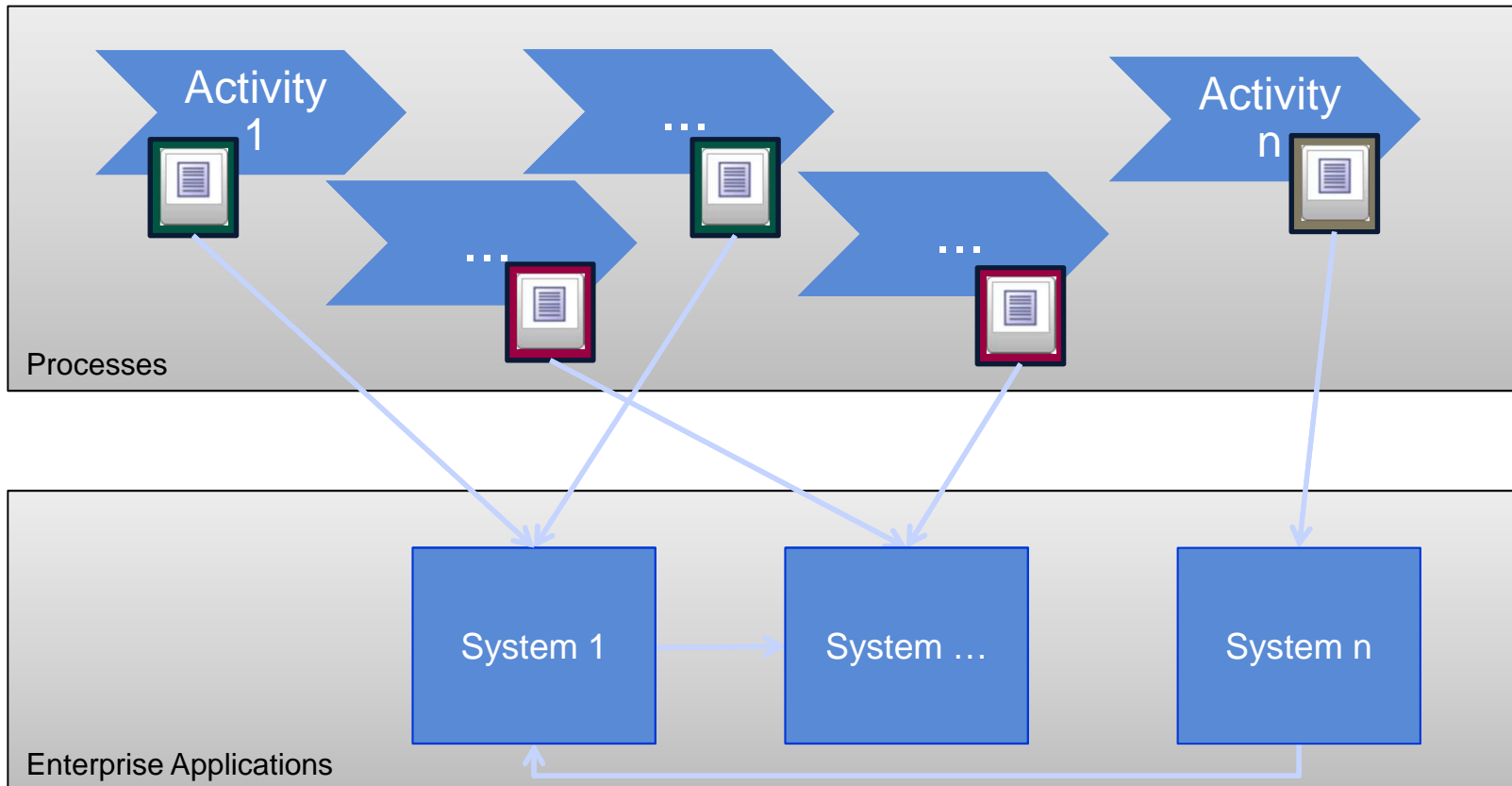
- select appropriate test scenarios which match test goals
- supports the capacity planning of service providers
- derive test data requirements due to information about service operations involved in a test

# Agenda

1. Introduction
2. Motivation
- 3. Project Context**
4. Using Performance Models to Support Load Testing
  - Palladio Component Models
  - Generating Palladio Component Models
  - Selecting Usage Scenarios
  - Transforming Performance Models
  - Predicting Service Workloads
  - Evaluation
5. Conclusion & Future Work

# Project Context

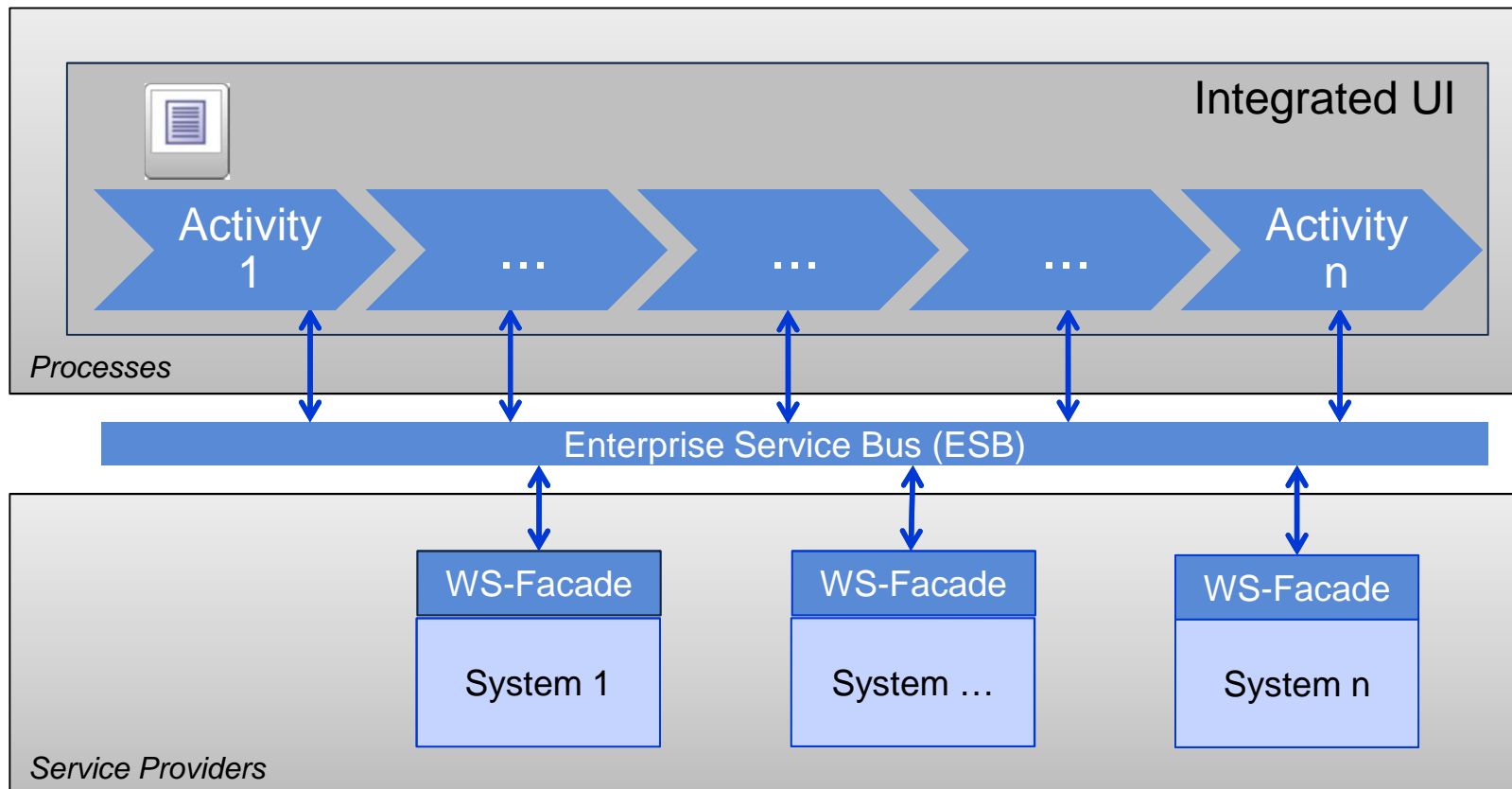
## Current IT Landscape





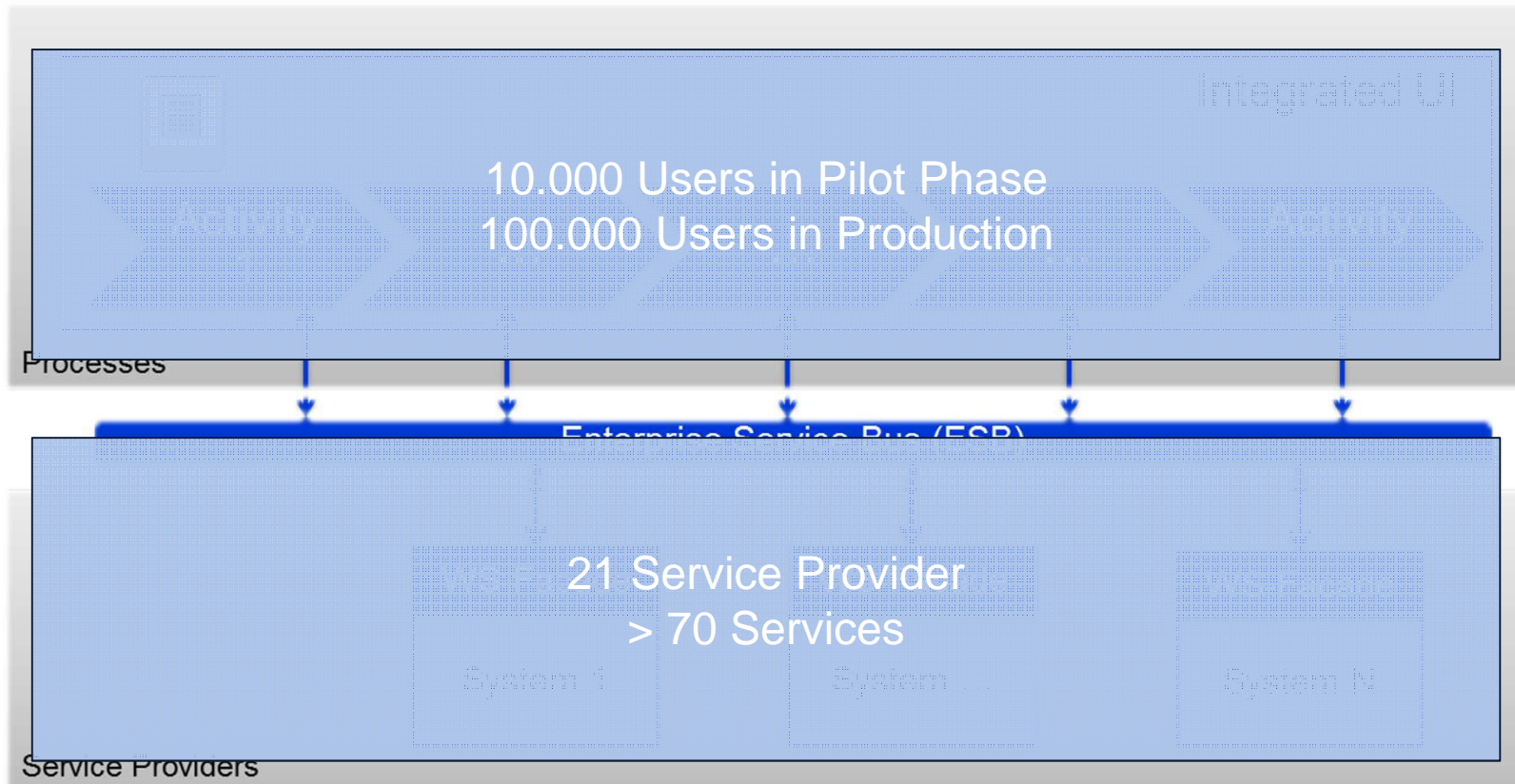
# Project Context

## Target IT Landscape (Service-oriented Architecture)



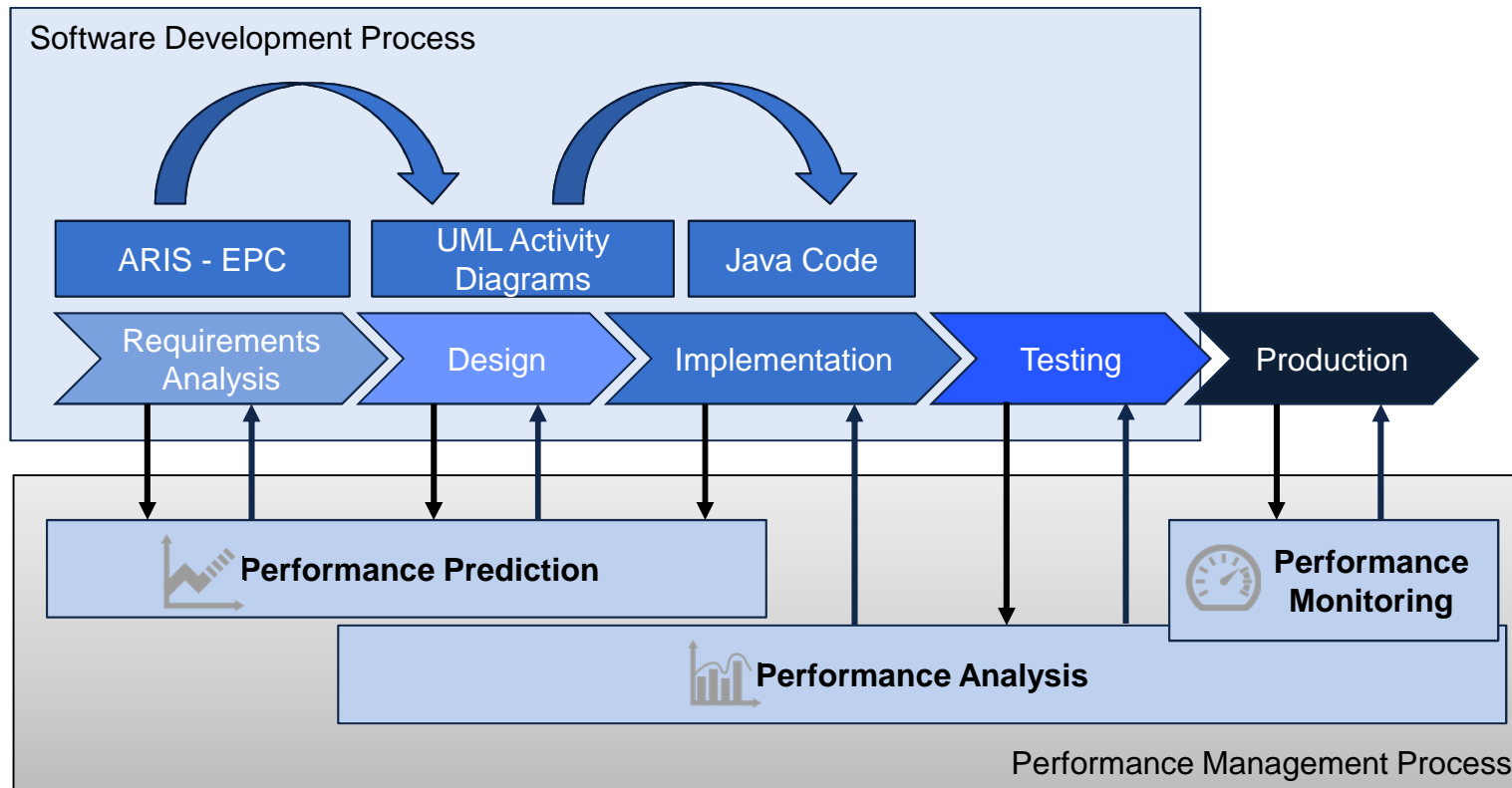
# Project Context

## Target IT Landscape (Service-oriented Architecture)



# Project Context

## Performance Management Process

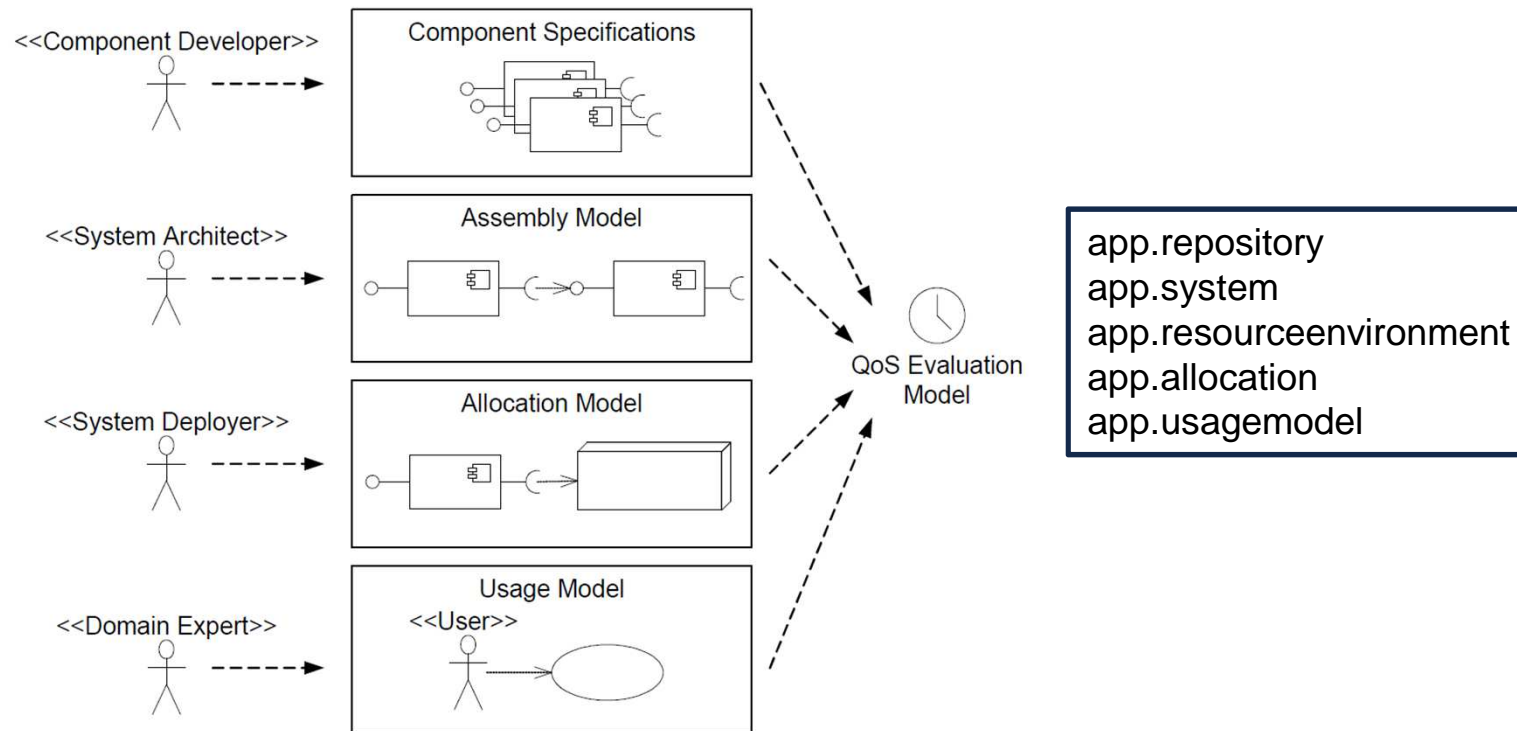


# Agenda

1. Introduction
2. Motivation
3. Project Context
- 4. Using Performance Models to Support Load Testing**
  - Palladio Component Models
  - Generating Palladio Component Models
  - Selecting Usage Scenarios
  - Transforming Performance Models
  - Predicting Service Workloads
  - Evaluation
5. Conclusion & Future Work

# Using Performance Models to Support Load Testing

## Palladio Component Models

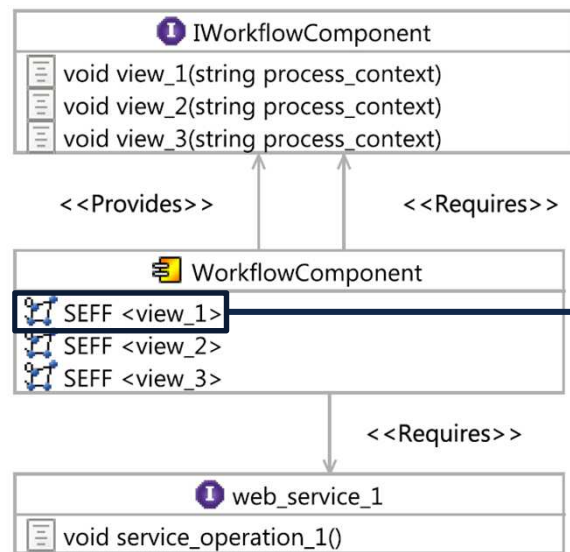


Becker et al. (2009)

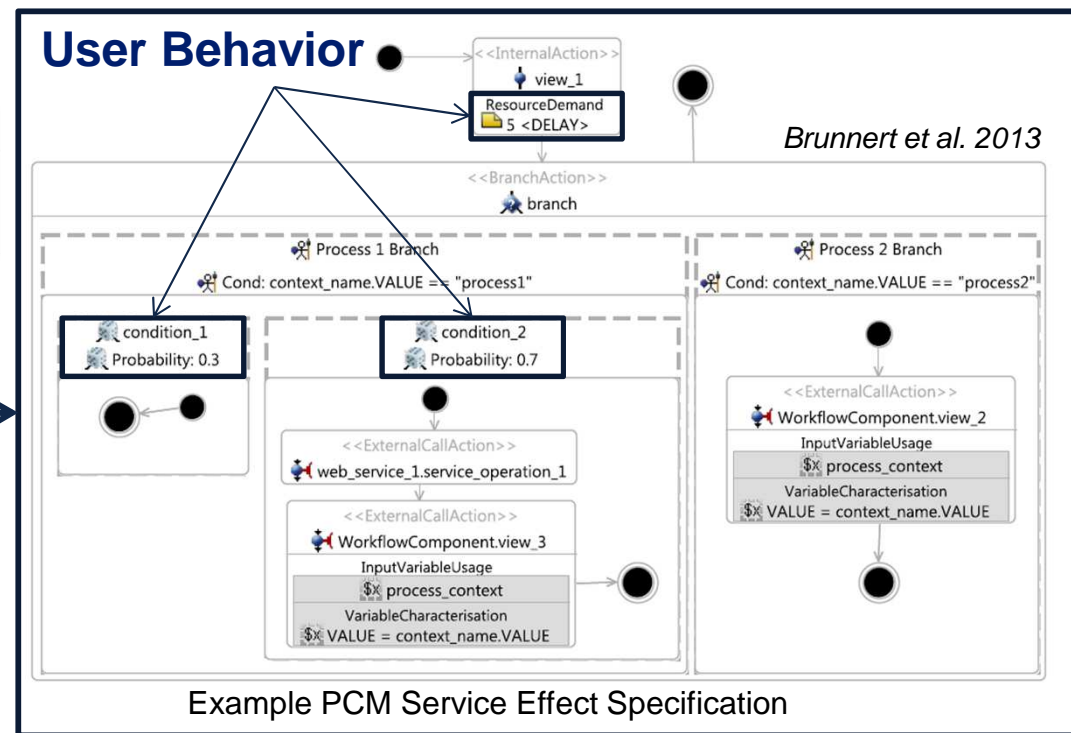
# Using Performance Models to Support Load Testing

## Generating Palladio Component Models

- Repository, usage and system models are generated based on UML-based business processes models
- Resource environment and allocation models are generated based on static information



Example PCM Repository Model

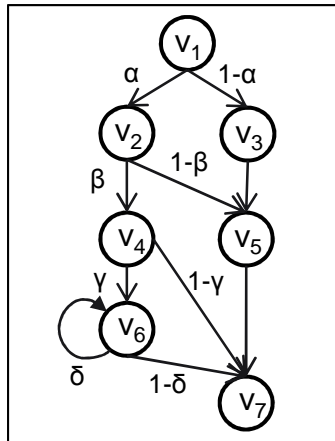


Example PCM Service Effect Specification

# Using Performance Models to Support Load Testing

## Selecting Usage Scenarios

User Behavior Graph in derived PCM Models



**Between the view transitions the service calls are modeled!**

Extracted Usage Scenarios

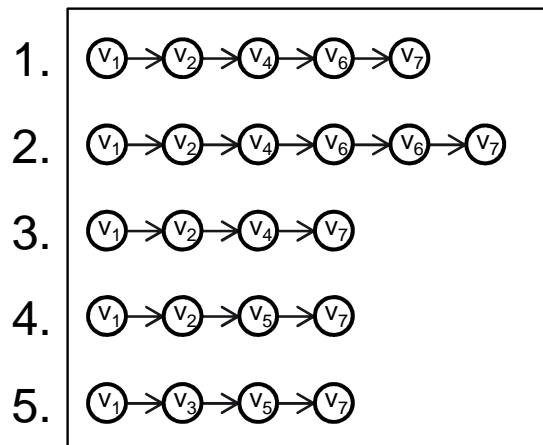
1.  $V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_6 \rightarrow V_7$
2.  $V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_6 \rightarrow V_6 \rightarrow V_7$
3.  $V_1 \rightarrow V_2 \rightarrow V_4 \rightarrow V_7$
4.  $V_1 \rightarrow V_2 \rightarrow V_5 \rightarrow V_7$
5.  $V_1 \rightarrow V_3 \rightarrow V_5 \rightarrow V_7$

- Recursive depth-first search
- Multiply probabilities per usage scenario
  - i.e. usage scenario 1:  $p_1 = \alpha * \beta * \gamma * (1-\delta)$
- Define thresholds for
  - minimum probability
  - minimum/maximum user actions
- Test experts can select usage scenarios which match their performance goals

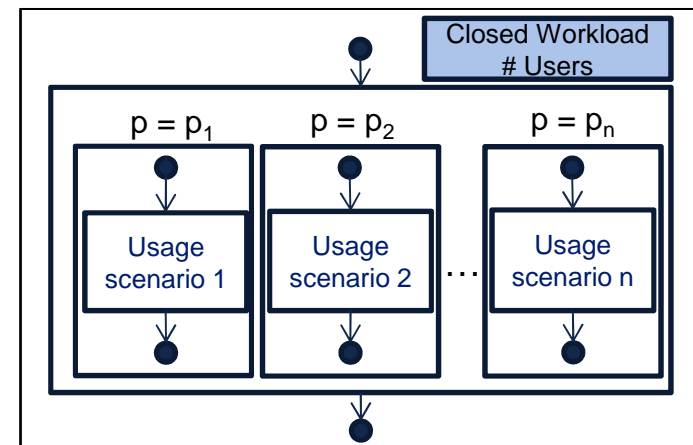
# Using Performance Models to Support Load Testing

## Transforming Performance Models

### Extracted Usage Scenarios



### PCM Usage Model



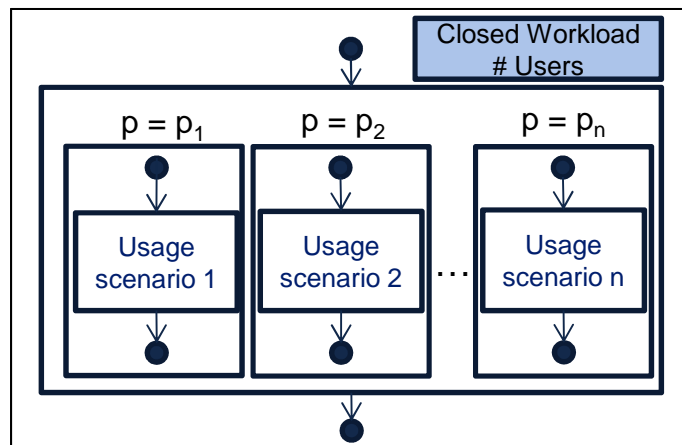
- We need to represent these usage scenarios independently from each other
- PCM models are therefore transformed to allow the simulation of different usage scenario combinations



# Using Performance Models to Support Load Testing

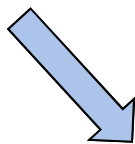
## Predicting Service Workloads

### PCM Usage Model

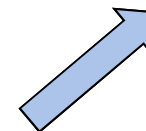


### Workload Predictions

- Service operations involved in a test
- Number of service operation invocations over time
- Throughput per usage scenario



- Using transformed models as input for a simulation engine
- The number of simulated users can be varied to assess the impact of different user counts

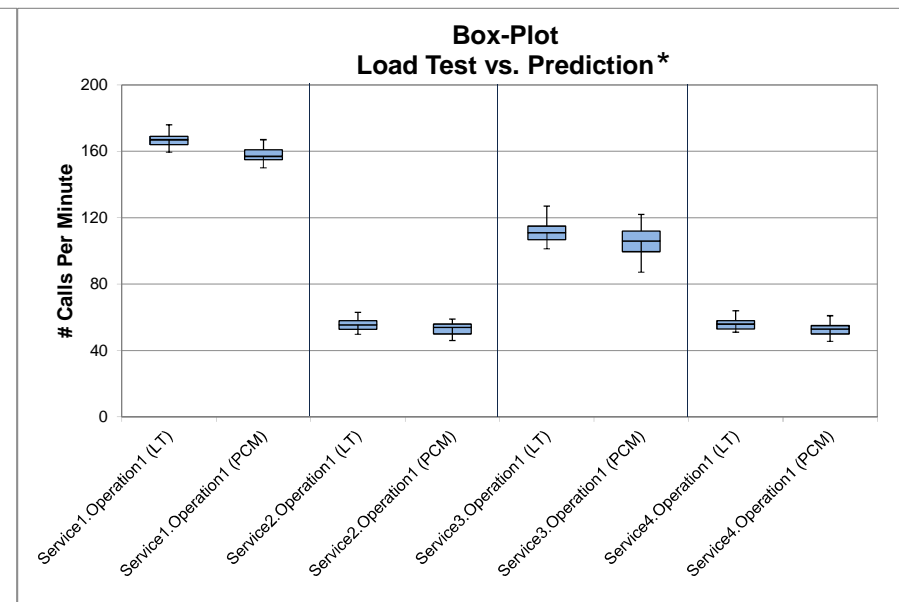
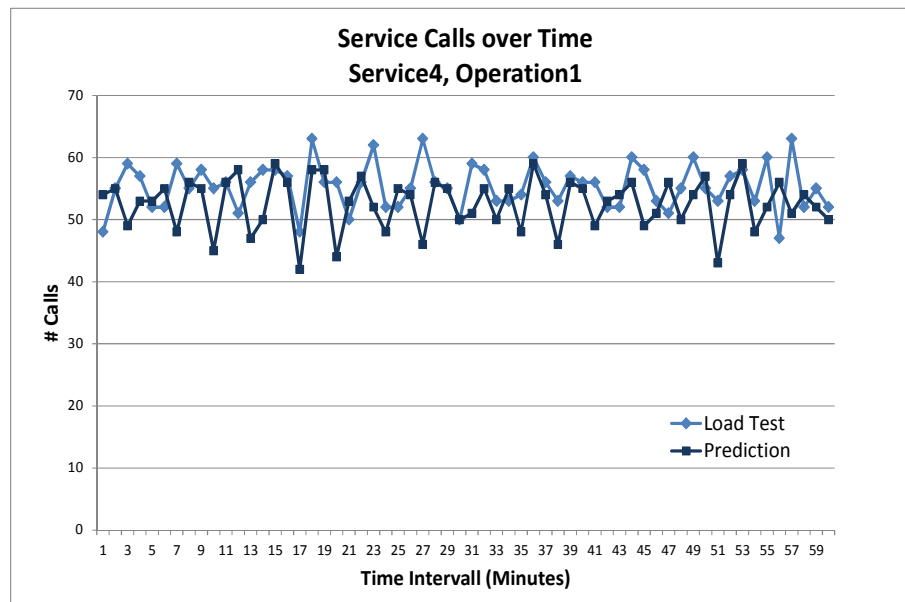


# Using Performance Models to Support Load Testing Evaluation

| Load Test                                                                             |             |         | Prediction   |             |         |                       |
|---------------------------------------------------------------------------------------|-------------|---------|--------------|-------------|---------|-----------------------|
| Web Service*                                                                          | Operation * | # Calls | Web Service* | Operation * | # Calls | Prediction Error in % |
| Service1                                                                              | Operation1  | 9.998   | Service1     | Operation1  | 9.455   | 5,43%                 |
| Service1                                                                              | Operation2  | 9.986   | Service1     | Operation2  | 9.454   | 5,33%                 |
| Service1                                                                              | Operation3  | 9.998   | Service1     | Operation3  | 9.455   | 5,43%                 |
| Service1                                                                              | Operation4  | 3.341   | Service1     | Operation4  | 3.152   | 5,66%                 |
| Service2                                                                              | Operation1  | 3.324   | Service2     | Operation1  | 3.154   | 5,11%                 |
| Service3                                                                              | Operation1  | 6.644   | Service3     | Operation1  | 6.310   | 5,03%                 |
| Service4                                                                              | Operation1  | 3.323   | Service4     | Operation1  | 3.153   | 5,12%                 |
| Service4                                                                              | Operation2  | 3.323   | Service4     | Operation2  | 3.154   | 5,09%                 |
| Service4                                                                              | Operation3  | 3.321   | Service4     | Operation3  | 3.154   | 5,03%                 |
| Service4                                                                              | Operation4  | 3.323   | Service4     | Operation4  | 3.155   | 5,06%                 |
| Service4                                                                              | Operation5  | 3.323   | Service4     | Operation5  | 3.155   | 5,06%                 |
| <b><i>One usage scenario: Load Test Duration 1 Hour, 75 Users Closed Workload</i></b> |             |         |              |             |         |                       |

\* Names of services and operations are anonymized

# Using Performance Models to Support Load Testing Evaluation



\* The first operation of each service

# Agenda

1. Introduction
2. Motivation
3. Project Context
4. Using Performance Models to Support Load Testing
  - Palladio Component Models
  - Generating Palladio Component Models
  - Selecting Usage Scenarios
  - Transforming Performance Models
  - Predicting Service Workloads
  - Evaluation
- 5. Conclusion & Future Work**

# Conclusion & Future Work

## Conclusion

- Predictions based on performance models can simplify the load test planning and execution in a SOA project
- The applicability of the approach depends on the availability of software models depicting the control flow of enterprise applications

## Future Work

- Extended evaluation, i.e. more than one usage scenario
- Machine learning to prioritize usage scenarios based on different test goals
  - i.e. probability of execution, resource utilization, resource coverage
- Automatic load test script generation for selected usage scenarios
- Capacity planning using performance models enhanced with resource demand information (Brunnert et al. 2013)

# Points to discuss

## **Request for feedback**

- Are there other ways how this work could be evaluated?

## **A thought-provoking statement or discussion question about the area**

- Could this approach be interesting to support load testing of other systems (other than SOA) as well?

Thanks for your attention!

**Questions?**

## Contact //

Christian Vögele, Andreas Brunnert, Alexandru Danciu, Daniel Tertilt  
fortiss GmbH

An-Institut Technische Universität München

Guerickestraße 25 · 80805 München · Germany

{voegele,brunnert,danciu,tertilt}@fortiss.org, [www.fortiss.org](http://www.fortiss.org)

Helmut Krcmar

Technische Universität München

Chair for Information Systems

Boltzmannstr. 3 · 85748 Garching · Germany

[krcmar@in.tum.de](mailto:krcmar@in.tum.de), [www.winfobase.de](http://www.winfobase.de)



# Related Work

- S. Becker, H. Koziolok, and R. Reussner. The palladio component model for model-driven performance prediction. *Journal of Systems and Software*, 82(1):3 – 22, 2009.
- A. Brunnert, A. Danciu, C. Vögele, D. Tertilt, and H. Krcmar. Integrating the palladio-bench into the software development process of a soa project. In *Proceedings of the Symposium on Software Performance: Joint Kieker/Palladio Days*, pages 30–38, 2013.
- A. Brunnert, C. Vögele, and H. Krcmar. Automatic performance model generation for java enterprise edition (ee) applications. In M. S. Balsamo, W. J. Knottenbelt, and A. Marin, editors, *Computer Performance Engineering*, volume 8168 of *Lecture Notes in Computer Science*, pages 74–88. Springer Berlin Heidelberg, 2013.
- G. Canfora and M. Di Penta. Service-oriented architectures testing: A survey. In *Software Engineering*, pages 78–105. Springer, 2009.
- Y. Liu, I. Gorton, and L. Zhu. Performance prediction of service-oriented applications based on an enterprise service bus. In *International Computer Software and Applications Conference (COMPSAC)*, pages 327–334, Beijing, China, 2007.



Dublin, 2014-03-22

# Using Performance Models to Support Load Testing in a Large SOA Environment

Industrial Track

Christian Vögele

fortiss GmbH  
An-Institut Technische Universität München

