

Quality-Aware Design of Software Systems

Barbora Bührenová

Faculty of Informatics, Masaryk University
Brno, Czech Republic

LASARIS SEMINAR

April 11, 2013



① Introduction

Motivation

Focus of the Talk

② Overview of Existing Techniques

Foundations

Industrial Techniques

Research Techniques

③ Palladio Approach

Performance Analysis

Reliability Analysis

Combined Techniques

④ Conclusion

Summary

Challenges



① Introduction

Motivation

Focus of the Talk

② Overview of Existing Techniques

Foundations

Industrial Techniques

Research Techniques

③ Palladio Approach

Performance Analysis

Reliability Analysis

Combined Techniques

④ Conclusion

Summary

Challenges



Motivation

Large-scale software systems with complex architecture

- support of critical business processes in enterprise inf. systems
- quality = customer trust & satisfaction = money

Different ways of understanding the quality

- not only system correctness!

Other quality attributes

- performance
- reliability
- security
- energy consumption
- maintainability
- ... and many others



Focus of the talk

Focus

- Information systems with complex architectures
- Quality in terms of performance and reliability

Goal

- Formal techniques assisting software architects in the development of high-quality systems



① Introduction

Motivation

Focus of the Talk

② Overview of Existing Techniques

Foundations

Industrial Techniques

Research Techniques

③ Palladio Approach

Performance Analysis

Reliability Analysis

Combined Techniques

④ Conclusion

Summary

Challenges



Performance

Performance reflects the ability of a software system to fulfil the requirements on fast **response time** and high **throughput** of the system while minimizing the **usage of computational resources**.

Performance attributes

- response time
- throughput
- resource utilization



Reliability

Reliability is the probability that a software system will perform the required functionality according to the design restrictions without **faults** and **failures** in a given time span.

Reliability attributes

- probability of failure on demand
- mean time to failure



Performance vs. reliability

Differences

- Conflicting objectives
- Tuning techniques
- Prediction questions

Similarities

- Quantitative quality attributes
- Both influenced by very similar architectural elements
- Architectural models and prediction techniques



Industrial techniques for performance/reliability assessment

After implementation (measurement-based)

- profiling and measurement of an implemented and deployed system
- **pro** – low effort (no additional model needed)
- **cons** – too late to revert initial design decisions

Before implementation (prototype-based)

- implement a prototype and measure its characteristics when deployed on the target platform
- **pro** – supports early decisions
- **cons** – very expensive, time consuming, hardware can be hardly changed, imprecise (many measurements needed for statistical validity)



Industrial techniques for performance/reliability tuning

After implementation

- faster/more reliable hardware (execution environment in general)
- redundancy (reliability), component derating (reliability)
- multi-threading (performance)
- code and architecture refactoring

During implementation

- fine-tuning of micro-level issues (performance)
- optimizing compilers (performance)
- error detection (reliability), fault tolerance (reliability)

Donald Knuth: "We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil" [1974]



Goals of ongoing research

Develop techniques with the following properties

- integrate both quality assessment and tuning
- design-time techniques (model-based)
- integrated into the development process
- easy evaluation of different configurations (changing/updating both software and hardware)
- automated quality assessment
- model-based prototype generation
- combination of formal models with UML

Additionally

- cost-effective (comparing to industrial techniques)
- time-effective (scalability of formal analysis)



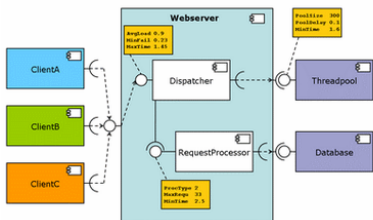
Quality engineering techniques

Focus

- Information systems with complex architectures

Implications

- Complex systems → formal methods may fail due to system size
- Defined architecture → compositional reasoning



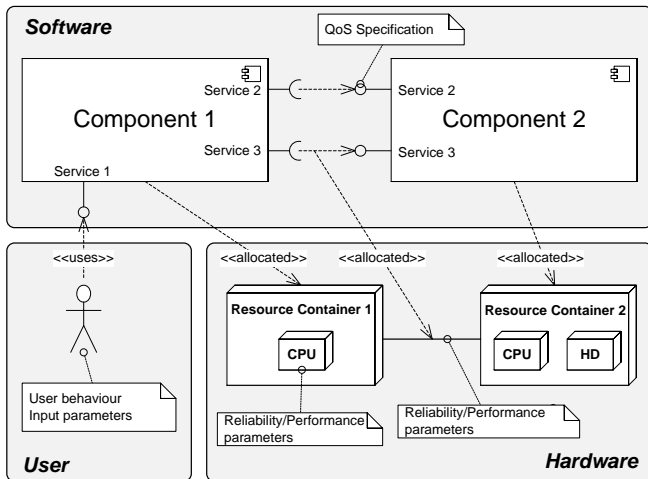
Techniques for inf. systems with complex architectures

Architecture-driven analysis

- defined in a modular way
- each architectural entity seen as independent
- each element assigned with a (certified) quality information
 - i.e. **software component** → service: QoS as response time or probability of failure-free operation
 - i.e. **hardware component** → CPU: processing rate, mean time to failure/repair
- parameterized specification needed (due to independence) → easy element reuse and update



Techniques for inf. systems with complex architectures



Techniques for inf. systems with complex architectures

The techniques support architecture design in:

- **prediction** of the expected values of performance and reliability attributes
- evaluation of alternative **design decisions**
- **sensitivity analysis** (as an effect of parameterization
 - identification of crucial components (both software and hardware)
 - relaxing uncertainties (in input parameters, system usage)
- suggestions for **design improvement** (architecture optimization)
- **trade-off analyses** (performance and reliability as conflicting objectives)



① Introduction

Motivation

Focus of the Talk

② Overview of Existing Techniques

Foundations

Industrial Techniques

Research Techniques

③ Palladio Approach

Performance Analysis

Reliability Analysis

Combined Techniques

④ Conclusion

Summary

Challenges



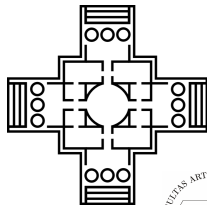
Palladio – Software Architecture Simulation

Palladio – Software Architecture Simulation

- developed since 2003 as a research project of the Uni of Oldenburg
- currently developed by Karlsruhe Institute of Technology (KIT), FZI Research Center for Information Technology, and University of Paderborn
- Website <http://www.palladio-simulator.com/>

Palladio means

- Science and Research
- Consulting for Industry
- Free tooling



Palladio – Software Architecture Simulation



Palladio

The Quality Software People.

HOME | NEWS | SUPPORT | CONTACT | IMPRINT

UK DE

Analysis of the high-scale IT setup at Ericsson

[more...](#)



Analysis
Tools
Consulting
Science
About Palladio

You are here: Home

Monday, 28 March 2011

Free Tools



We offer free tools for analysing the quality of your software: Performance, reliability, maintainability.
[Tool downloads & documentation](#)

Consulting for Industry



We provide professional consulting and developer training to enhance the quality of your software.
[Portfolio & industry success stories](#)

Science and Research



We are leading members of the research community in model-based software quality prediction.
[Research projects & cooperations](#)



Palladio – Scenarios

Scenarios

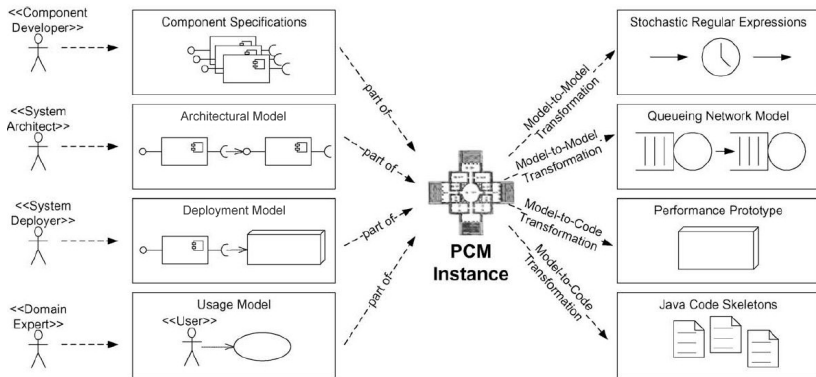
- Sizing
- Scalability
- Load Balancing
- Configuration
- Optimization
- Design Alternatives

Analysis Dimensions

- Performance
- Reliability
- Maintainability
- Costs



Palladio – Analysis overview



Performance analysis with Palladio

Performance is

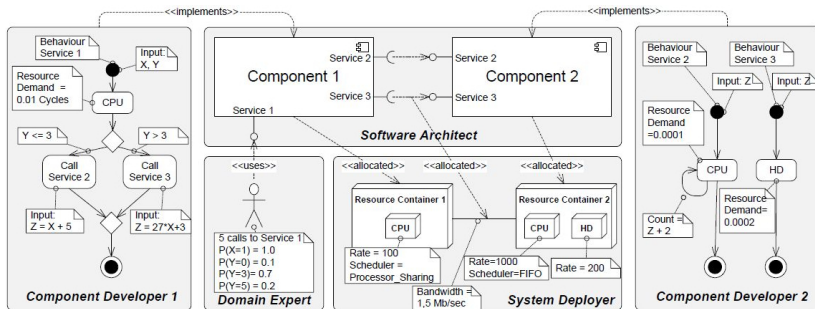
- response time
- throughput
- resource utilization

Influencing factors

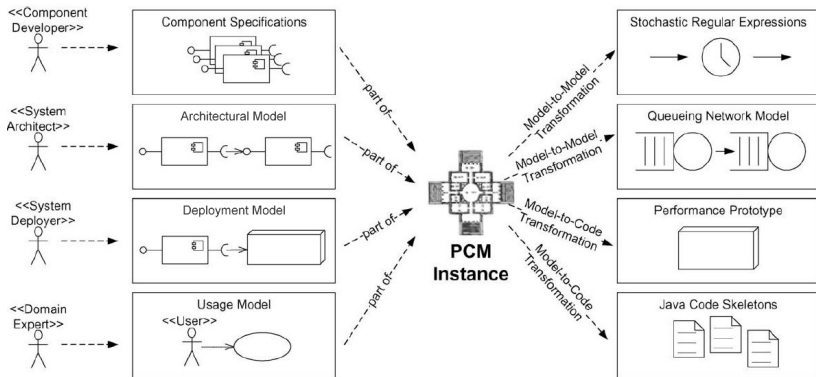
- Required functionality (services)
- Execution environment
- Usage profile



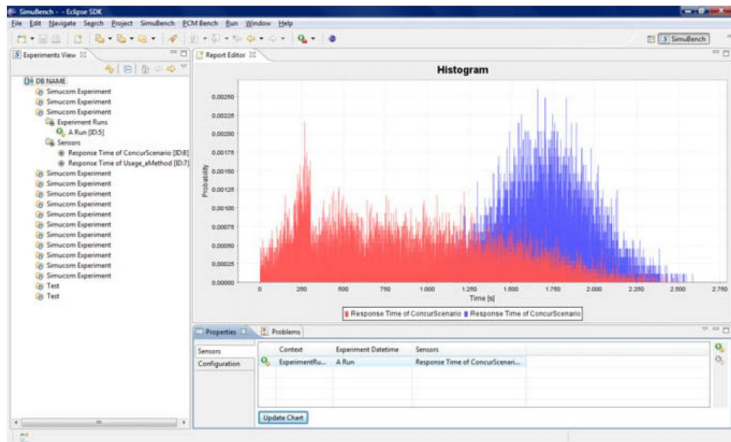
Performance modelling in Palladio



Performance assessment in Palladio



Performance assessment results



Reliability analysis

Reliability is

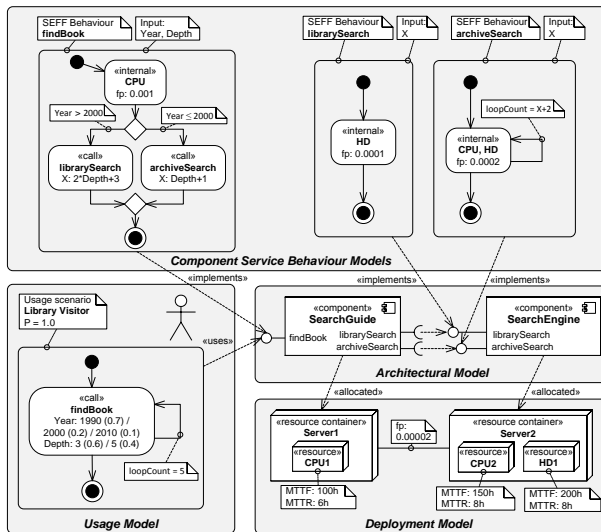
- software reliability (validity and correctness) – e.g. probability of failure on demand
- hardware reliability (availability) – e.g. mean time to failure
- system reliability (usage implied software failure influenced by employed hardware)

Influencing factors

- Required functionality (services)
- Execution environment
- Usage profile



Reliability modelling in Palladio



Reliability assessment in Palladio

Methods of evaluation

- Formal analysis (Markov model)
- Simulation (generated Java code)
- Sensitivity analysis

Types of reported results

- Predicted reliability values
- Evaluation of alternative design decisions
- Identification of reliability-critical architectural elements



Reliability assessment in Palladio

The screenshot displays the Palladio IDE interface for a Java project named 'Management Information System'. The main workspace shows a 'Service Effect Specification' diagram with components like 'SchedulerServer', 'ApplicationServer', and 'DBServer'. A 'Component Allocation' panel on the right lists allocated components such as 'Allocation_WebServer', 'Allocation_Scheduler', 'Allocation_ReportingEngine', 'Allocation_CacheInfo', and 'Allocation_Database'. A 'Simulation Monitor' window at the bottom left shows a 'Stop' button and simulation status. A 'Reliability Prediction' window at the bottom right displays the following output:

```

PCM Solver Console: Analysis Tool Output:
Controller - Path "" is not an existing Directory. Skipping sensitivity analysis
Solve - Solved parametric dependencies: 98 ms
Solve - Number of states in Markov Chain: 6372
Solve - Number of transitions in Markov Chain: 6870
Solve - Finished Markov transformation: 1650 ms
Solve - Number of performed transformation runs: 32
Solve - Probability of Success: 0,9960837669884645
  
```

Four callout boxes highlight key features: 'Service Effect Specification' (top left), 'Component Allocation' (top right), 'Simulation Monitor' (bottom left), and 'Reliability Prediction' (bottom right).

Architecture optimization and trade-off analysis

Architecture optimization

- PreOpteryx sub-project within Palladio
- automated design improvement via architecture changes
- based on optimization algorithms (evolutionary-based in PreOpteryx)
- multi-objective optimization (performance, reliability, cost)
- degrees of freedom (HW/SW parameters, allocation, replication, ...)

Trade-off analysis

- reliability, performance and cost as conflicting objectives
- selection of the design alternative with the highest overall value



1 Introduction

Motivation

Focus of the Talk

2 Overview of Existing Techniques

Foundations

Industrial Techniques

Research Techniques

3 Palladio Approach

Performance Analysis

Reliability Analysis

Combined Techniques

4 Conclusion

Summary

Challenges



Summary

Overview of Existing Techniques

- Foundations
- Industrial Techniques
- Research Techniques

Palladio Approach

- Performance Analysis
- Reliability Analysis
- Combined Techniques



Challenges of design-time quality assessment

Performance

- high dependence on low-level details (platform dependent, e.g. scheduling strategies)

Reliability

- accuracy of the input data (failure probabilities and hardware availability)

Both

- knowledge gap between software engineers/architects and quality experts
- minimization of the modelling effort



Thank you

Thank you for your attention!
Any questions?

