Palladio

Prediction of Performance Properties

Heiko Koziolek (koziolek@ipd.uka.de)
Klaus Krogmann (krogmann@ipd.uka.de)
Ralf Reussner (reussner@ipd.uka.de)

Chair Software Design and Quality
Institute for Program Structures and Data Organization (IPD)
Faculty of Informatics, Universität Karlsruhe (TH)
Performance Prediction

Performance model of a component-based software architecture

Performance data
- Execution time
- Throughput
- Resource utilisation
Palladio: The Approach

A Component Model

Multiple Analysis Methods

A Development Process
Agenda

- Research Group
- Overview: The Role Concept
- The Palladio Component Model
- Analysis Methods & Transformations
- Excerpts from CoCoME Models & Prediction Results
- Conclusion
Our Research Group

People & Topics
The Role Concept

Overview

&

Developer Roles tied to the Palladio Component Model
Development Roles

Component Developers

Software Architect

System Deployer

Domain Expert

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion
Models and Analyses

Palladio Component Model

- Stochastic Regular Expr.
- SPA with Scheduling
- Queueing Network
- Performance Prototype
- Java Code Skeletons

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion

Palladio Component Model

A Component Model for early Design Time Performance Predictions
Named after the Italian renaissance architect Andrea Palladio (1508–1580)

- Context-independent component specifications: Parameterised for re-use
- Split into sub-models:
  - Domain specific modelling languages
  - Specific for developer roles
Influences on Component Performance

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion
Sub-Models

- Repository model
  - Components and Interfaces
  - Service Effect Specification (SEFF)
- System model
  - Component Assembly
- Resource environment model
  - Resource types model
- Allocation model
- Usage model
Execution Time of $a()$ 

Service Effect Specification (SEFF)
System

Considered within a PCM instance for analyses and simulations.

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion

Klaus Krogmann - Palladio: Prediction of Performance Properties 21/08/2007 15
Meta-Model

- **Syntax**
  - Concrete syntax: Similar to UML 2 diagrams
  - Abstract syntax:
    - PCM is defined in the ECORE meta-meta-model
    - Different concepts than UML 2 meta-model

- **Semantics**
  - Static semantics: OCL constraints
  - Dynamic semantics: Technical report
Analysis Methods & Transformations

Simulation of Queuing Networks & Stochastic Process Algebras
Analysis Methods

- Queuing Network
  - Simulation solution
  - Support of concurrency, scheduling strategies

- Stochastic Regular Expression
  - Analytical solution
  - No concurrency, but faster than simulation

- Stochastic Process Algebra
  - Hybrid solution (analysis + simulation)
  - High-level support of concurrency

- All Support
  - Parameterisation (usage, assembly, allocation, implementation)
  - Arbitrary distribution functions
Transformations

- Used by analysis methods
- Model-2-model / model-2-text transformations use openArchitectureWare (oAW)

- Output
  - POJOs
  - EJB-System

- Supports
  - “QoS-Prototype”
  - Code skeletons
CoCoME

Palladio Models
Due to the size of CoCoME we will present only excerpts from our CoCoME models

- Repository
- SEFF
- Resource Environment
- Allocation
- Usage Model

- Prediction results
Repository Model

TradingSystem.Inventory.Application
- SEFF <queryEnterpriseById>
- SEFF <getMeanTimeToDelivery>

TradingSystem.Inventory.Application.Store
- SEFF <bookSale>
- SEFF <getStore>
- SEFF <getProductsWithLowStock>
- SEFF <getAllProducts>
- SEFF <getProductsWithOptionalStockItem>
- SEFF <orderProducts>
- SEFF <getOrder>
- SEFF <rollInReceivedOrder>
- SEFF <calculatePrice>
- SEFF <markProductsUnavailableInStock>
- SEFF <flushDatabase>
- SEFF <getProductWithStockItem>

<<Provides>>
<<Requires>>

EnterpriseQueryIf
- org.cocome.tradingsystem.inventory.data.enterprise...
- int getMeanTimeToDelivery (org.cocome.tradingsystem...)

PersistenceIf
- org.cocome.tradingsystem.inventory.data.persistence...

StoreIf
- org.cocome.tradingsystem.inventory.application.st...
- org.cocome.tradingsystem.inventory.application.st...
- org.cocome.tradingsystem.inventory.application.st...
- org.cocome.tradingsystem.inventory.application.st...
- org.cocome.tradingsystem.inventory.application.st...
- org.cocome.tradingsystem.inventory.application.st...
- org.cocome.tradingsystem.inventory.application.st...
- void rollInReceivedOrder (org.cocome.tradingsystem...)
- void markProductsUnavailableInStock (org.cocome.tradingsystem...)
- void flushDatabase()

ProductDispatcherIf
- org.cocome.tradingsystem.inventory.application.st...
SEFF for bookSale()
SEFF <bookSale>

<<ExternalCallAction>>
PersistenseIf.getPersistenceContext
InputVariableUsage

<<InternalAction>>
persistence context setup
ResourceDemand
19 ms <CPU>

<<LoopAction>>
ProductWithStockItemIterator
sellTO.NUMBER_OF_ELEMENTS

<<ExternalCallAction>>
StoreQueryIf.queryStockItemById
InputVariableUsage
OutputVariableUsage

<<InternalAction>>
catchBlock:
ResourceDemand
20 ms <CPU>

<<ExternalCallAction>>
PersistenseIf.getPersistenceContext
InputVariableUsage
Resource Environment Model

Klaus Krogmann - Palladio: Prediction of Performance Properties

Roles, Analysis Methods -> CoCoME -> Conclusion

21/08/2007

«ResourceContainer»

StoreClient

«ResourceContainer»

EnterpriseClient

«ResourceContainer»

StoreServer

«ResourceContainer»

EnterpriseServer

«LinkingResource»

«ProcessingResourceSpecification»
ProcessingRate: 1.0
Units: CPU-Units

«CommunicationLinkResourceSpecification»
Throughput: 10.0
Units: MBit
Latency: 0
Usage Model

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion

Klaus Krogmann - Palladio: Prediction of Performance Properties  21/08/2007  31
Prediction Results (1)

- **Response time**
- bookSale()
- 20 Stores
- Open workload
- Minimum: 1,050 ms
- Maximum: 2,400 ms

![Histogram showing response time distribution]

- **Busy**
- **Idle**

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion

Klaus Krogmann - Palladio: Prediction of Performance Properties 21/08/2007 32
Prediction Results (3)

- Waiting time
- Enterprise Server (CPU)
- 20 Stores
- Open workload
- 30% of the requests are handled within 100 ms

Possible Service Level Agreements
X% of requests have a waiting time of:
- 90% < 1000 ms
- 30% < 100 ms

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion
Klaus Krogmann - Palladio: Prediction of Performance Properties 21/08/2007 33
- Response time
- bookSale()
- 1 Store
- Open workload
- Minimum: 1,050 ms
- Maximum: 1,825 ms

Histogram

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion

Klaus Krogmann - Palladio: Prediction of Performance Properties 21/08/2007 34
Prediction Results (2)

- Response time
- bookSale() 
- 50 Stores 
- Open workload 
- Minimum: 1,000 ms 
- Maximum: 52,000 ms 
- CPU queue is overfull

Roles ▶ Component Model ▶ Analysis Methods ▶ CoCoME ▶ Conclusion
Utilisation

Enterprise Server (CPU)

50 Stores

Open workload

Up to 46 concurrent jobs

Idle: 1.8%
Results

- Simulation used
- All results base on the specification of the non-functional properties from the CoCoME chapter: Static time consumptions
- The results show that the enterprise server cannot handle 200 concurrently accessing store servers (as specified)
Conclusion
Limitations

▪ Current Palladio approach
  – No persistent component state
  – No dynamic architectures
  – Only one-to-one connectors

▪ Modelling CoCoME
  – Embedded part of the system (“POS”) was left out
  – Exceptions not modelled
  – POS and database were considered system-external
Lessons learned

- Support of “sub-systems” apart from composite components would have been useful
- One-to-many connections and replication (store servers) should be supported by the PCM
- CoCoME: good debugging and testing system
Future Work

- Automation of model reconstruction for given source code
- “High-level-modelling”: Concurrency, synchronisation
- Dynamic Architectures
- Resource Model

- Measure an adapted implementation of CoCoME
  → compare to the prediction results
More Information

http://sdqweb.ipd.uka.de/wiki/CoCoME-PCM

- CoCoME-Models
- Further prediction results
- Tools downloads: Modelling and Prediction
- Documentation of the Palladio Component Model
- Development Process in details
A Component Model
- Context independent specification
- Sub-models reduce complexity
- Arbitrary distribution functions

Multiple Analysis Methods
- Queuing network based simulation
- Stochastic process algebra

A Development Process
- Applicable with the PCM
- Explicit support of component ideas
References

- See the CoCoME book chapter