Samuel Kounov, Fabian Brosig, Nikolaus Huber

From Offline to Online Component Models for Run-Time Resource Management

Palladio Days, Karlsruhe, November 2011
Proactive Run-time Resource Management

[Graph showing Responsiveness over time with SLA violation and Online prediction for QoS prediction and reconfiguration impact analysis.]

PART 2
Online QoS prediction for reconfiguration impact analysis
Proactive Run-time Resource Management

Online reconfiguration impact prediction for trade-off analysis

Service A

VM replication/cloning

Service A

Scaling up/Improving dependability

Online prediction

Service A

Dynamic server consolidation

LiveVM migration

Service B

Service C

Online prediction

Efficiency OK

Dependability/Responsiveness OK
Proactive Run-time Resource Management

**Responsiveness**

- SLA
- SLA OK

**Cost/Energy efficiency metric**

- Optimal resource usage

**Time [mins, hours, days, weeks, months]**
Proactive Run-time Resource Management

Online QoS prediction for problem anticipation

Online QoS prediction for reconfiguration impact analysis

Online reconfiguration impact prediction for trade-off analysis

Part 1

Part 2

Part 3

Autonomic system adaptation
Coming Soon: Descartes Meta-Model

Application Level

Component A

Service A → Service A

Component B

Service A

Platform Level

Mapping

Middleware

Mapping

Virtualization Layer

Mapping

Infrastructure Level

CPU

Hardware

Storage

Network

Samuel Kounev, Fabian Brosig, Nikolaus Huber:
From offline to online component models for run-time resource management
Dynamic Model Instance Composition

Usage Scenario

Software Architecture

Middleware

Virtualization

Infrastructure

Software Design and Quality Group
Institute for Program Structures and Data Organization
Tailored Model-to-Model Transformations

- Usage Sub-model
- Soft. Arch. Sub-model
- Middleware Sub-model
- Virtualization Sub-model
- Infrastructure Sub-model

Dynamically Composed Model Instance

- Operational Analysis: Analytical Sol.
- Queueing Petri Nets: Simulation
- Full-Blown Simulation: Simulation

Part of

Transformation

17.11.2011 Samuel Kounov, Fabian Brosig, Nikolaus Huber:
From offline to online component models for run-time resource management
Online Architecture-Level Perf. Models

- Models should capture both static and dynamic aspects
- “Component” is a unit of composition *at run-time*
- Alignment with system layers as opposed to developer roles
- Dynamic model composition
  - Horizontally across components and vertically across layers
- Sub-models integrated into the system components
- Integration with monitoring data available at run-time
- Models intended for use by system as opposed to human
- Different abstraction levels & solution methods
Models for Virtualized Data Centers

- **Resource Landscape (static view)**
  - Computing resources (physical, logical)
  - Layers of resources (Hardware, Virtualization, …)

- **Variability Points (dynamic view)**
  - Configuration (#cores, #appServer, …)
  - Landscape (Deployment, Users, …)

- **Ongoing Work: Reconfiguration Language**
  - Description of the system reconfiguration
  - Example: SystemConfigA → SystemConfigB

- **PassiveResourceCapacity**
  - capacity : EBigInteger

- **PassiveResourceType**
  - passiveResourceSpecification : 0..*

- **ConfigurationSpecification**
  - parentResourceSpecification : 1
  - processingResources : 0..*

- **ActiveResourceSpecification**
  - parentResourceSpecification : 1

- **CustomConfigurationSpecification**
  - parentResourceSpecification : 0..1

- **EObject**
  - non-functionalProperties : +

- **ProcessingResourceSpecification**
  - schedulingPolicy : SchedulingPolicy
  - processingRate : EDouble

- **ProcessingResourceType**
  - activeResourceType : 1
  - nrOfParProcUnits : 1

- **NumberOfParallelProcessingUnits**
  - number : EBigInteger

- **LinkingResourceSpecification**
  - communicationLinkResourceType : 0..*
  - parentResourceSpecification : 1

- **CommunicationLinkResourceType**
  - bandwidth : EDouble

- **SchedulingPolicy**
  - «enumeration»
  - DELAY
  - FCFS
  - PROCESSOR_SHARING
  - RANDOM
  - N/A
Variability Points

- “Decorator” model
- Dynamic view of the resource landscape
Modeling the Application Level

- Service Behavior Abstractions for Different Levels of Granularity
- Deployment-Specific Resource Demands / Response Times
- Probabilistic Parameter Dependencies
Service Behavior Abstractions

- **BlackBoxBehavior**
  - No information about resources, resource demands, control flow, call frequencies,…

- **CoarseGrainedBehavior**
  - Information at component boundary level (external services, resource consumption,…)

- **FineGrainedBehavior**
  - Information about component-internals (control flow, resource demands, parametric dependencies,…)

Samuel Kounev, Fabian Brosig, Nikolaus Huber: From offline to online component models for run-time resource management
Service Behavior Abstractions

At most one behavior description per abstraction level and signature/role
Probabilistic Parameter Dependencies

- Characterize dependencies statistically

- Influencing parameters
  - Service input parameters
  - Return parameters of external service calls

- Influenced quantities
  - Loop iteration numbers (FineGrainedBehavior)
  - Branch probabilities (FineGrainedBehavior)
  - Call frequencies (CoarseGrainedBehavior)
  - Resource demands (FineGrainedBehavior, CoarseGrainedBehavior)
  - Response times (BlackBoxBehavior)
  - Input parameter of ext. service call (FineGrainedBehavior, CoarseGrainedBehavior)
Conclusions and Discussion

Vision Paper
[SCC2010]

http://www.descartes-research.net/
Service Behavior Abstractions

```
CoarseGrainedBehavior
  1
  *

  1
  1

  ExternalCall

  1
  1

  ResourceDemandSpecification

  1

  CallFrequencyDistribution

  1

  ExternalCallFrequency

  1

  BlackBoxBehavior

  1

  ResponseTimeSpecification
```
Probabilistic Parameter Dependencies

Required service: sendRequisitionToBuyer(List demands)

Provided service: scheduleWorkOrder(String assemblyId, int quantity)

b) Derived Probability Mass Function (PMF)