Encapsulation of Structural Adaptation by Composite Components

Steffen Göbel

Dresden University of Technology, Germany

WOSS’04 Newport Beach, CA
10/31 – 11/01/2004
Motivation: Components with different QoS profiles: COMQUAD

Idea: Mapping QoS profiles to configurations

Generalization: Model for adaptable components
Modeling reconfiguration with UML
MDA approach for the development

Conclusions and Outlook
Motivation: COMQUAD

- COMponent with QUantitative properties and ADaptivity
- Description of component nets at level of specifications
- Selection of implementations based on required QoS
- Adaptation by changing QoS profiles

Problem: How can we develop a component with multiple QoS profiles?
Idea: QoS Profile = Configuration

- QoS profile is mapped to internal configuration of a composite component
- Encapsulation of adaptation logic
Adaptable Composite Component =

Subcomponents + Reconfiguration + Parameter Mapping

Component parameters are mapped to different configurations

Parameters can be changed from inside or outside of the composite component
Component Model

Management interface
Parameter interface

<<controls>>
Glue Code
Adaptation operators
Aspect operators

Adaptation Manager
Repository
Adaptation Specification
Context Model

Active configuration

Steffen Göbel
Possible reconfiguration operations:
- Adding components and connections
- Removing components and connections
- Replacing components (special case)
- Adding Postprocessing to a VideoDecoder Component
Configuration and Variations

- **Configuration** = complete graph of a composite component
  - Represent operating modes
  - Can be tested by unit tests
  - Can be mapped to *enumeration parameter*
  - Example: QoS profiles

- **Variations** = changes of a configuration
  - Can be applied limited or unlimited times
  - Every variation has an inverse variation
  - Can be mapped to graph grammars
  - Can be mapped to *integer parameter*
MDA Approach

PIM
- Subcomponents
- Reconfiguration Models
  - Model of adaptable Component

PSM
- EJB
- COMQUAD
- ...

Runtime
- EJB
- COMQUAD
- ...

Steffen Göbel
Conclusions and Outlook

- Encapsulation of structural adaptation by composite components
  → Easier for developers
  - Parameter mapping
  - Self-adaptable components
  - Graphical specification of reconfigurations
  - MDA approach for development

- Challenges
  - Adaptation of stateful components
  - Synchronization
  - Tool support
Run-time Adaptation of Stateful Components

- Problem: State must be transferred from old to new component instance

```
old
State
Transformer
State
Replacement
Transformer
new
State
```

- or

```
Transformer
Generic State
Transformer
Ontology
```
- Actually composite components to avoid dependencies during adaptation
- Problem: Adaptation must be executed on different computers at the same time

Change of encryption

Adaptation Synchronization necessary!
Adaptation by coupling of parameters

Encryption parameter

<<Coupling>>

Change of encryption
Development Methodology for Adaptation

**Design Time**
- Platform independent
- Graphical Description

**Development Time**
- Direct changes of source code possible
- Many adaptation possibilities can be planned for deployment and Runtime

**Deployment Time**
- Source code not available, only changes or extensions of binary code possible

**Runtime**
- Remove, replace or reorder
- Change of component parameters

Adaptation mechanisms
- many
- few