Torben Weis

QCCS

Quality Controlled Component based Systems
QCCS
European IST Project
http://www.qccs.org

IRISA

TU-Berlin

SEMA

KD Software
QoS (Quality of Service)

> Examples for QoS
  > Bandwidth
  > Latency
  > Security
  > Scalability (concurrent invocations)

> QoS is a non-functional property of a system

> Our Goal:
  Modeling of QoS-aware components
QoS & Components

> Disadvantages of pure objects
  > Connected by type (in the model)
  > Connections at runtime are hardly trackable
  > Therefore, contract management is difficult

> Advantages of components
  > Components reside on a higher level of abstraction
  > Coupled with connectors (in the model)
  > Reflection allows tracking of connections
  > Contract management is easier
QoS & Adaption

> Modern applications must *adapt* to a changing environment
  
  > Bandwidth
  
  > (Un)secure network connections
  
  > Device constraints (Screen size, CPU, …)

> Adaption becomes an important feature and crosscuts the entire design

> Adaption should be addressed at the modeling level
UML2 Components – Black Box View

Connector

Port
UML 2 Components – White Box View

Delegation
Modeling QoS-Categories

- **Availability**
  - Direction: 
  - Type: float [Percent]
  - Unit: 

- **Persistence**
  - Direction: 
  - Type: Period [Amount]
  - Unit: 

---

### <<enumeration>>

- **Period**
  - Call
  - Minute
  - Hour
... putting things together ...
Modeling Contract Dependencies

- **CPerformance**
  - isMandatory = false
  - Performance
  - responseTime: int [Sec]

- **CPrecision**
  - isMandatory = false
  - Precisions
  - digits: int [Amount]

- **CPersistence**
  - isMandatory = false
  - Persistence

- **P1_Offline**
  - isMandatory = false

- **P2_Offline**
  - isMandatory = false

- **MyComponent**

- **Data**
  - **MyComponent**
  - **ISql**

- Dependency from **Port** to **MyComponent**

- Dependency from **MyComponent** to **Data**

- Dependency from **MyComponent** to **P2_Offline**

- Dependency from **P1_Offline** to **MyComponent**

- Dependency from **P2_Offline** to **MyComponent**

- Dependency from **MyComponent** to **Persistence**

- Dependency from **Persistence** to **ISql**

- Dependency from **MyComponent** to **Data**
Invocation-Protocols / 1
Invocation-Protocols / 2

Client

: IGraphics

begin()

loop [lines]
drawLine()

end()
Entity Components

- **Entity Component**
  - `Customer`
    - `<<component>>`
    - `<<key>>` +`CustomerID`: int
    - `+Name`: string
  - `Address`
    - `+Street`: string
    - `+House`: string
    - `0..*`
  - `City`
    - `<<key>>` +`PostalCode`: int
    - `+Name`: string
  - `Company`
    - `+Credit`: float
  - `Private`
    - `+Birthday`: string

- **Home-Component**
  - `Customer`

- **Association**
  - `Bookstore`
    - `<<component>>`
    - `StoreItem`
      - `+Amount`: int
    - `OrderItem`
      - `+Date`: string
      - `+Amount`: int
      - `+CustomerID`: int
      - `<<key>>` +`OrderID`: int

- **Key**
Doing the MDA thing …

QoS Mechanisms

Mechanism Interface

Contract Interface

Business Logic

PIM

PSM
Aspects in UML / 1

Component realization

Aspect invocation

Contract realization

Parameters

New class
Aspects in UML / 2

Woven component

New association

New generalizations
Thank you.

Questions?

Torben Weis
weis@ivs.tu-berlin.de
FG Intelligente Netze und Management verteilter Systeme
TU Berlin
www.ivs.tu-berlin.de