Quality-Driven Stepwise Refinement of Component-Based Architectures

Lucia Kapova, Barbora Buhnova
Motivation

Abstract Models

Prediction

Measurements

Implementation

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Motivation

Abstract Models

Prediction

Low-level details!

Transactional delivery ...

Measurements

Implementation

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Motivation

Source Model

Completion

Transformation

Target Model

Sender Adaptor

Receiver Adaptor

Middleware

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Previous Work

Woodside, ICSE 2007: The future of software performance engineering
• Motivation for performance completions

Happe, WOSP 2008: A pattern-based performance completion for Message-Oriented Middleware
• Process of completion development based on measurements
• Message-oriented Middleware

Kapova, EUROMICRO 2009: Automated feature model-based generation of refinement transformations
• Automated integration of completions based on M2M Transformation
• Generation of transformation variants

Kapova, FESCA 2010: Systematic refinement of performance models for concurrent component-based systems
• Categorization and analysis of completions for concurrent component-based systems
• Identification of their interdependencies
Challenge

Source Model

Target Model

Sender Adaptor

Receiver Adaptor

Middleware

Transformation

Completion

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Completion Library

- Barrier
- ThreadPool
- Compression
- Replication
- Locking
- MOM
- Pipe & Filter
- Encryption

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Structural Conflicts: Transformation Composition

Conflict Reduction:

- Metamodel clustering
- Sequential composition

\[ T \cdot T \rightarrow T \]
### Structural Conflicts: Completion Conflicts Reduction

<table>
<thead>
<tr>
<th>Role</th>
<th>Event-based communication</th>
<th>Synchronisation</th>
<th>Concurrency</th>
<th>Message-oriented communication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PCD: Component Developer</strong> (Thread-safe Components)</td>
<td>Scoped Locking</td>
<td>Thread-specific Storage</td>
<td>Messaging Endpoints</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Strategized Locking</td>
<td>Monitor Object</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Thread-safe Interface</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Double-Checked Locking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Optimisation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rendezvous (Barrier)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>PSA: Software Architect</strong> (Component interaction)</td>
<td>Asynchronous Completion Token</td>
<td></td>
<td>Replication</td>
<td>Message Channels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Message Routing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Message Endpoints</td>
</tr>
<tr>
<td><strong>PSD: System Deployer</strong> (Infrastructure)</td>
<td>Reactor</td>
<td>Active Object</td>
<td></td>
<td>Message Bus</td>
</tr>
<tr>
<td></td>
<td>Proactor</td>
<td>Half-Sync/ Half-Async Leader-Followers</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Acceptor-Connector</td>
<td>Thread Pool</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Quality Conflicts: Lessons Learned

- Palladio Component Model (PCM) was used to implement our approach

- Three levels of Completion-based Refinement
  - Components
  - Connectors
  - Infrastructure
The Completions Library Guidelines

Annotation

Messaging Connector

Completion Order Guidelines

RMI/SOAP Connector

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Quality Conflicts: Conflicts Resolution Heuristics

- Components
  \[ Q(e) = \sum_{s_i \in S} \frac{rt(s_i)}{thp(s_i)} \]

- Connectors
  \[ Q(e) = \frac{rt(e)}{thp(link)} \]

- Infrastructure
  \[ Q(e) = \sum_{r_i \in R} ut(r_i) \]

**Suggested Candidate:** Completion Chain with minimal \( Q(e) \).
Case Study

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Case Study

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures

<table>
<thead>
<tr>
<th>[C, E, M]</th>
<th>[E, C, M]</th>
<th>[E, M, C]</th>
<th>[C, M, E]</th>
<th>[M, C, E]</th>
<th>[M, E, C]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(rt(link))</td>
<td>496 ms</td>
<td>653 ms</td>
<td>1137 ms</td>
<td>676 ms</td>
<td>1130 ms</td>
</tr>
<tr>
<td>(thp(link))</td>
<td>115 msg/s</td>
<td>72.5 msg/s</td>
<td>39 msg/s</td>
<td>70 msg/s</td>
<td>39 msg/s</td>
</tr>
<tr>
<td>(Q(e))</td>
<td>4.3</td>
<td>9.0</td>
<td>29.1</td>
<td>9.6</td>
<td>28.9</td>
</tr>
</tbody>
</table>
Case Study

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
The Completions Library Guidelines

![Diagram of RMI/SOAP Connector]

- **Annotation**: Additional processing for RMI/SOAP.
- **Completion Order Guidelines**: [C,E,M].....
- **Messaging Connector**
- **Assembly Connector**
- **Remote Call** and **Local Call**

Lucia Kapova: Quality-Driven Stepwise Refinement of Component-Based Architectures
Conclusion and Future Work

Evaluation

• Implementation Prototype: http://sdqweb.ipd.kit.edu/wiki/Chilies
• Case Study (ThreadPool, MOM, Replication, Pub-Sub Configurations, etc.).

Contributions

• Inclusion of implementation details required for absolutely correct prediction.
• Localisation, reduction and resolution of completion conflicts.
• Completion library with order guidelines.

Future Work

• Many MDD research questions arise during use of MDD:
  • Structuring, composition and maintainability of transformations
  • Rule-based composition of transformations (vertical composition)
  • How to deal with implementation details which are not so easy to decide on during design time?
  • Cyclic component interdependencies should be further researched.
  • Composition of complex subsystems for connector completions.
Thanks!
kapova@ipd.uka.de
http://sdq.ipd.uka.de