Model-based Analysis of executable Business Processes for Web Services

Hendrik Voigt

Database- and Information Systems Group
Paderborn University

5.4.2004
Overview (1)

BPEL
Implementation

UML
Abstraction

CSP
Input for Model Checker

Model-based Analysis of executable Business Processes for Web Services
Overview (2)

Model-based Analysis of executable Business Processes for Web Services

Syntactic Analysis

Semantic Analysis

Model of a Business Process

Code of a Business Process

Results

Transformation of Model

Modell as Input

Transformation
BPEL

- XML-language
- Relationship with WSDL
- Peer-to-peer interaction
- Local viewpoint
BPEL-Example

<receive name="receiveOrder"
  partnerLink="ns:buyerLink"
  portType="ns:onlineShopPT"
  operation="ns:receiveOrder"
  variable="order"
  createInstance ="yes"/>

• Who is involved in the process?
• What is done during the process?
• In which order is it done?
UML

- Use Case diagrams
- Class diagrams
- Component diagrams
- Sequence diagrams
- Activity diagrams
Model-based Analysis of executable Business Processes for Web Services

UML

• **Interaction protocol and business process**
  - Stereotypes receive, reply and invoke
  - Port Type and Partner Links
  - Control-flow and activities for interaction

• **Model understandable for business experts**
CSP

• Process algebra based on ...
  – Processes
  – Events
  – Channels
  – Operators

• Tool support (FDR2)

• Consistency Workbench
Proofable Properties

- **Syntactic Analysis**
  - UML diagrams

- **Semantic Analysis**
  - Deadlock
  - Livelock
  - Message sequences
Adopted Techniques

• Transformation based on Pair Grammars
• Pair Grammar and Graph Grammars
• Production rules
• Functionality
### Example Pair Grammar

#### Left production rule of pair grammar

<table>
<thead>
<tr>
<th>Act:=</th>
<th><code>&lt;sequence&gt;</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>ActivityA1</code></td>
</tr>
<tr>
<td></td>
<td><code>ActivityA2</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;/sequence&gt;</code></td>
</tr>
</tbody>
</table>

#### Right production rule of pair grammar

```
<table>
<thead>
<tr>
<th>Act</th>
<th>ActivityA1</th>
<th>ActivityA2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>ActivityA1</td>
<td>ActivityA2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>ActivityA1</td>
<td>ActivityA2</td>
</tr>
</tbody>
</table>
```

Hendrik Voigt
Similarities

- Transformation between text and graph based languages
- Consistency of the different representations must be ensured
- Maybe graph grammars suitable for refactoring
Questions & Remarks