Aspect-oriented workflow patterns for web service composition

Mathieu Braem
System and Software Engineering Lab (SSEL)
Vrije Universiteit Brussel
Outline

- Web service compositions
- Plug and play compositions using the SCE
- Crosscutting concerns in compositions
- AOP with Padus
- Directions for Padus
Outline

- Web service compositions
- Plug and play compositions using the SCE
- Crosscutting concerns in compositions
- AOP with Padus
- Directions for Padus

Keywords

- AOP
- Workflows
- Webservice composition
Web service composition

- Web services expose existing software as external services, through standardized protocols
- Compositions add value by providing more advanced services
- Specialized languages to express these compositions, e.g. WS-BPEL
Service creation environment

- Create web service compositions on a higher level of abstraction
- Plug-and-play composition of building blocks in a visual editor
- Developer is guided in creating valid compositions
SCE Screenshot

Palette
Canvas
Outline

Properties and verification report

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Allows looking up Alcatel employees by e-mail or telephone number</td>
</tr>
<tr>
<td>Name</td>
<td>Directory</td>
</tr>
</tbody>
</table>
Crosscutting concerns

BPEL process

- Authentication
- Billing
- Activity
- Billing
- Authentication

Aspects (containing advice)

- Authentication
- Billing

Pointcuts (sets of joinpoints)
Padus aspects

<aspect ...>
  <using>
    <namespace ... />
    <partnerLink ... />
    <variable ... />
  </using>
  <before joinpoint="Jp"
    pointcut="invoking(Jp, 'MyService', 'MyPortType', Op)">
    <bpws:sequence>
      <bpws:assign>...</bpws:assign>
      <bpws:invoke ... />
    </bpws:sequence>
  </before>
</aspect>
<aspect ...
  <using>
    <namespace ... />
    <partnerLink ... />
    <variable ... />
  </using>
  <before joinpoint="Jp"
    pointcut="invoking(Jp, ‘MyService’, ‘MyPortType’, Op)">
    <bpws:sequence>
      <bpws:assign>...</bpws:assign>
      <bpws:invoke ... />
    </bpws:sequence>
  </before>
</aspect>
<aspect ...>
<using>
  <namespace .../>
  <partnerLink .../>
  <variable .../>
</using>

<before joinpoint="Jp" pointcut="invoking(Jp, 'MyService', 'MyPortType', Op)"
</before>

</aspect>
<aspect ...
    <using>
        <namespace ... />
        <partnerLink ... />
        <variable ... />
    </using>
    <before joinpoint="Jp"
        pointcut="invoking(Jp, 'MyService', 'MyPortType', Op)">
        <bpws:sequence>
            <bpws:assign>...
            <bpws:invoke/>
        </bpws:sequence>
    </before>
</aspect>
Padus key concepts

- Pointcuts are logic queries
- Static weaver
  Weaving = modifying logic representation of BPEL process
- Separate connector to apply aspects to processes and specify interaction resolution
Future directions

‣ Workflow specific advice types
‣ Protocol-based pointcuts
‣ Describe AO approach for workflow patterns
Requirements for workflow languages, not specific to one language

- Describe control flow constructs

- Range from simple “sequence” pattern to complex patterns for synchronization
Looking for collaboration on these topics

- AOP for workflow languages, web services
- web service composition, WS-BPEL
Looking for collaboration on these topics

- AOP for workflow languages, web services
- web service composition, WS-BPEL
Looking for collaboration on these topics

- AOP for workflow languages, web services
- Web service composition, WS-BPEL

WP1 Programming languages
WP2 Modelling languages
Looking for collaboration on these topics:

- AOP for workflow languages, web services (WP2/1)
- Web service composition, WS-BPEL (WP2/1)

WP1 Programming languages
WP2 Modelling languages
Thank you.