Compositors for WSDL 2.0

Roberto Chinnici, Sun Microsystems

Glen Daniels, Sonic Software

Amelia Lewis, TIBCO Software

Adi Sakala, IONA

Umit Yalçinalp, Oracle Corporation

WSD f2f Boston Jan 2004

Requirements

- Allow composition of properties and features
- Provide uniform semantics for optional and required features and properties via composition
- Introduce minimal changes to the specification

Compositor

- Is the unit of composition
- Identified by a specific URI
    - all
    - choice
    - one-or-more
    - zero-or-more
- Is extensible
    - Possible to add new compositors (URI)
    - Allows extensions, thus composition of extensions
- Supports recursive composition
    - a compositor may contain other compositors
- May occur where feature/properties are allowed
    - Interface
    - Operation
    - Binding
    - Binding Operation

Pseudo-Syntax

```
<compositor uri="...">
   <documentation />?
   [feature/> | <property/> | <compositor/> |
    <wsoap:module/> |
    <extensibility-element/>]+
</compositor>
```

Supports/Requires Semantics

- Currently, provided by `required` attribute on Feature, Property and wsoap:module
- With proposal, enclosing compositor provides the semantics
  - All: all members are required
  - Choice: exactly one member is required
  - One-or-more: at least one member is required
  - Zero-or-more: none of the members are required, but supported

Example

4 features: f1 and f2 (required), f3 and f4 (optional)

Today (without compositors):

```
<binding>
   …
   <feature uri="f1" required="true"/>
   <feature uri="f2" required="true"/>
   <feature uri="f3" required="false"/>
   <feature uri="f4" required="false"/>
</binding>
```

Example (cont)

Tomorrow (with compositors):

```
<binding>
    <compositor uri="all">
        <feature uri="f1"/>
        <feature uri="f2"/>
        <compositor uri="zero-or-more">
            <feature uri="f3"/>
             <feature uri="f4"/>
        </compositor>
     </compositor>
</binding>
```

Feature Enhancements

```
<feature
     uri="xs:QName">
     <documentation />?
  [  <compositor/> |   <extensibility-element/>]*
</feature>
```

- Allows a Feature to be configured
- Composition defines specific configuration

Example: choice

```
<binding…>
  <operation .../>
  <operation .../>
  <compositor uri="http://www.w3.org/wsdl20/choice">
    <feature uri="http://example.com/authentication/username-
    token"/>
    <feature uri="http://example.com/authentication/X.509-
    token"/>
  </compositor>
</interface>
```

Example: Composition of SOAP Modules using all

```
<binding>…
    <compositor uri="http://www.w3.org/wsdl20/all">
        <wsoap:module  uri="…/OperationNameModule"/>
        <wsoap:module uri= "…"/>
        ….
     </compositor>
</binding>
```

Example: Composition of Features with all

```
<compositor uri="http://www.w3.org/wsdl20/all">
        <feature uri="http://example.com/security/dsig"/>
        <feature uri="http://example.com/security/non-repudiation/country/us">
        <feature uri="http://example.com/security/non-repudiation/receipt-ack">
 </compositor>
```

Feature Configuration using one-or-more

```
<binding>
<compositor uri="…">
  <feature uri="http://example.com/reliability">
      <compositor uri="http://www.w3.org/wsdl20/one-or-more">
        <property uri="http://example.com/reliability/guaranteed-
    delivery"><value>true</value>
        </property>
        <property uri="http://example.com/reliability/message-ordering">
          <value>total-ordering</value>
        </property>
        <property uri="http://example.com/reliability/duplicate-removal>
          <value>true</value>
        </property>
      </compositor>
  </feature>…<feature ../>…
</compositor>
</binding>
```

Composition with zero-or-more

```
<interface name="example4">
   <operation .../>
   <operation .../>
  <compositor uri="http://www.w3.org/wsdl20/zero-or-more">
    <feature uri="http://example.com/ack">
      <compositor uri="http://www.w3.org/wsdl20/all">
      <property uri="http://example.com/ack/QoS/within-24-hours">
        <value>true</value>
      </property>
      </compositor>
    </feature>
    <feature uri=..../>
  </compositor>
</interface>
```

Conclusion

- Composition of features and properties are extremely useful
- Composition defines what is required or optional
- 4 most common composition techniques are defined
- Introduces minimal changes to the specification