SOAP Processing Model

Sang Shin
Michèle Garoche
www.javapassion.com

“Learning is fun!”

Agenda

- SOAP Processing Model
- “role” Attribute
- “mustUnderstand” Attribute
- “relay” Attribute
- Using intermediary Nodes

SOAP Processing Model

SOAP Processing Model

- Describes the (logical) actions taken by a SOAP node on receiving a SOAP message
- Common processing
  - Analyze SOAP-specific elements (elements with SOAP "env" namespace)
  - Check that the SOAP message is syntactically correct

SOAP Processing Model

- Intermediate nodes can be added to the message path without forcing changes to the messages
  - Message path can be changed
- Messages can be changed without forcing nodes to change their processing model

SOAP Header Block Attributes

- “role” attribute
- “mustUnderstand” attribute
- “relay” attribute

“role” Attribute in SOAP 1.2 (same as “actor” Attribute in SOAP 1.1)

“role” Attribute (env:role=...)

- Identifies the role played by the intended target of that header block
- Custom roles and standard roles
- If absent, the SOAP message is targeted for ultimateReceiver
- Please note that <env:Body> element does not have a “env:role” attribute
  - <env:Body> is always for ultimateReceiver

Standardized Roles

- none
  - "http://www.w3.org/2003/05/soap-envelope/role/none"
  - SOAP nodes MUST NOT act in this role.
- next
  - "http://www.w3.org/2003/05/soap-envelope/role/next"
  - Each SOAP intermediary and the ultimate SOAP receiver MUST act in this role.
- ultimateReceiver
  - "http://www.w3.org/2003/05/soap-envelope/role/ultimateReceiver"

Example7a: SOAP message showing a variety of header blocks

```xml
<?xml version="1.0" ?>
 <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
   <p:oneBlock xmlns:p="http://example.com"
        env:role="http://example.com/Log">
   ...
   </p:oneBlock>
   <q:anotherBlock xmlns:q="http://example.com"
        env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
   ...
   </q:anotherBlock>
   <r:aThirdBlock xmlns:r="http://example.com">
   ...
   </r:aThirdBlock>
  </env:Header>
  <env:Body >
   ...
  </env:Body>
 </env:Envelope>
```

“mustUnderstand” Attribute

“mustUnderstand” Attribute

- If set to "true", the targeted SOAP node must process the header block (with the semantics described in the header's specification), or generate a fault
- If set to “false” or not present, the targeted SOAP node may process the header block

When to Use “mustUnderstand” Attribute?

- To ensure SOAP nodes do not ignore mandatory header blocks which are important to overall purpose of the application
  - Example: Making sure all SOAP receivers who play the role of “logger” must perform logging

Processing Header Blocks Might include

- Removing the header (default behavior)
- Reinserting the header with the same or altered value
- Inserting a new header

Example7b: SOAP message with “mustUnderstand” attribute

```
<?xml version="1.0" ?>
 <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
   <p:oneBlock xmlns:p="http://example.com"
       env:role="http://example.com/Log"
       env:mustUnderstand="true">
   ...
   </p:oneBlock>
   <q:anotherBlock xmlns:q="http://example.com"
    env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
   ...
   </q:anotherBlock>
   <r:aThirdBlock xmlns:r="http://example.com">
   ...
   </r:aThirdBlock>
  </env:Header>
  <env:Body >
   ...
  </env:Body>
 </env:Envelope>
```

“relay” Attribute (new from SOAP 1.2)

“relay” Attribute (env:relay=...)

- Newly introduced in SOAP 1.2
- Indicates whether a header block targeted at a SOAP intermediary must be relayed or not if it is not processed
  - Default behavior for a header block targeted at a role played by a SOAP intermediary is that it must be removed before the message is relayed
  - You don't want to unknowingly propagate header blocks

When to Use **relay=”true”** Attribute?

- Application designer would like to introduce a new feature, manifested through a SOAP header block, targeted at any capable intermediary which might be encountered in the SOAP message path
- Make it available to those intermediaries that "understood" it, but ignored and relayed onwards by those that did not
  - Use it with env:mustUnderstand=”false” or env:mustUnderstand attribute not present
  - Use it with env:role=”../next”

Example7c: SOAP message with “relay” attribute

```
<?xml version="1.0" ?>
 <env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <p:oneBlock xmlns:p="http://example.com"
          env:role="http://example.com/Log"
              env:mustUnderstand="true">
    ...
    </p:oneBlock>
    <q:anotherBlock xmlns:q="http://example.com"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
      env:relay="true">
    ...
    </q:anotherBlock>
    <r:aThirdBlock xmlns:r="http://example.com">
    ...
    </r:aThirdBlock>
  </env:Header>
  <env:Body >
  ...
  </env:Body>
 </env:Envelope>
```

Using SOAP Intermediaries

Scope of SOAP Specification

- SOAP spec. does not specify how a message path is determined and followed
  - Other specifications will deal with it
- SOAP spec. does describe, though, how a SOAP node should behave if it receives a SOAP message for which it is not the ultimate receiver
- Two types of intermediaries
  - Forwarding intermediaries
  - Active intermediaries

Forwarding Intermediary

- Forwards the SOAP message to another SOAP node
- Things that forwarding intermediary does
  - Remove all processed SOAP header blocks
  - Remove all non-relay'able SOAP header blocks that were targeted at the forwarding node but ignored during processing
  - Retain all relay'able SOAP header blocks that were targeted at the forwarding node but ignored during processing.

Active Intermediary

- Undertake additional processing that can modify the outbound SOAP message in ways not described in the inbound SOAP header blocks
- Example processing
  - security services, annotation services, content manipulation services

Active Intermediary

- Processing done by active SOAP intermediaries should be detectable by SOAP nodes in the downstream
  - inserting a header block

Example16: SOAP message after an active intermediary has inserted a mandatory header

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
 <env:Header>
  <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
     env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
        env:mustUnderstand="true">
   <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</reference>
   <m:dateAndTime>2001-11-29T13:20:00.000-05:00</m:dateAndTime>
  </m:reservation>
  <n:passenger xmlns:n="http://mycompany.example.com/employees"
     env:role="http://www.w3.org/2003/05/soap-envelope/role/next"
       env:mustUnderstand="true">
   <n:name>Åke Jógvan Øyvind</n:name>
  </n:passenger>
  <z:travelPolicy
    xmlns:z="http://mycompany.example.com/policies"
      env:mustUnderstand="true">
   <z:class>economy</z:class>
   <z:fareBasis>non-refundable<z:fareBasis>
   <z:exceptions>none</z:exceptions>
  </z:travelPolicy>
 </env:Header>
```

Thank you!

Sang Shin
Michèle Garoche
www.javapassion.com
“Learning is fun!”