Lecture 19

Web Service Intermediaries

WS-Routing and WS-Referral
Announcements

• Suggested due date for Lab 5: November 26th (Wednesday)

• Lab 6 will be available from web site tomorrow
  – Due back December 9th

• Final exam: December 18th
  – Will put sample questions in D:\VSDev\Public\vijayk over the weekend
(Review) Global XML Web Services Architecture

[ All encoded as special headers in the SOAP message ]

- **WS-Inspection**
  - A simpler UDDI-like discovery protocol
  - Caters to scenarios where source can directly announce availability

- **WS-Security**
  - Specifies how security credentials are passed in SOAP messages, how SOAP actors should act on them …

- **BPEL4WS**
  - Encoding of “business process activities”
  - First invoke this service, then use its results to invoke other services, …

- **WS-Transaction (old), WS-Coordination (September 2003)**
  - Atomic actions involving multiple services

- **WS-Routing, WS-Referral**
(Review) Use of Web Service Intermediaries

Caching and transcoding intermediaries

Diagram:
- UDDI
- Amazon
- Google
- Interaction Form
- Proxy
- Distiller
- Cache
- MyBookService(s)
(Review) WS-Routing

- Defines a new SOAP header element named path
  - From the http://schemas.xmlsoap.org/rp namespace

```xml
<wsrp:path xmlns:wsrp="http://schemas.xmlsoap.org/rp">
  <wsrp:action /> <!-- indicates message intent -->
  <wsrp:to />  <!-- identifies ultimate receiver -->
  <wsrp:fwd>    <!-- identifies forward intermediaries -->
    <wsrp:via />  <!-- identifies an intermediary node -->
  </wsrp:fwd>
  <wsrp:rev>    <!-- identifies reverse intermediaries -->
    <wsrp:via />  <!-- identifies an intermediary node -->
  </wsrp:rev>
  <wsrp:from />  <!-- identifies sender -->
  <wsrp:id />    <!-- uniquely identifies this message -->
  <wsrp:relatesTo />  <!-- correlates message with another -->
  <wsrp:fault />  <!-- provides extra fault details -->
</wsrp:path>
```
(Review) .NET Support for WS-Routing

• Part of the Web Services Extensions (WSE) package
  – Microsoft.Web.Services assembly

Enables construction of WS-Routing aware
• Web services
  – Must be able to handle the additional headers
  – Can act upon information encoded in the headers (e.g., security tokens)

• Web service clients
  – Can add header fields (<via> elements in WS-Routing)

• Web service intermediaries
  – Can extract header fields, interpret them, modify both header/body
Building WS-Routing Aware Applications in .NET

[Code walkthrough]

- Need to use the D:\VSDev\Public\... folder
  - WSE poorly integrated with HTTP-level Windows authentication

- Creating web services
  - Adding soapExtensionTypes info to web.config
  - [WebMethod] attribute works as earlier

- Creating web clients
  - Proxy inherits from a different class
  - Access to {Request,Response}SoapContext objects

- Creating web service intermediaries
  - Endpoint, RouterBasic, RouterForward, RouterReverse, RouterAdvanced
Building Web Service Intermediaries: Recap

• Define WS-Routing aware web services
  – Add soapExtensionTypes info into web.config file
  – Define service as before using [WebService] and [WebMethod] attributes

• Define intermediary class
  – Override definitions of ProcessRequestMessage, ProcessResponseMessage
    • Can invoke other web services, process SOAP message body/header, …
  – Inform IIS about mapping between URL for intermediary and this class
    • Via an <httpHandlers> element in web.config

• Route web service request through this intermediary
  – Extract message header using proxy.RequestSoapContext
  – Add via element to the forward path
WS-Referral

"Web Services Referral Protocol (WS-Referral) is a SOAP-based stateless protocol for inserting, deleting, and querying routing entries in a SOAP router. A SOAP router is a SOAP node that exposes SOAP message relaying as a Web service, either as a standalone service or in combination with other services."

[Diagram of SOAP routing]
WS-Referral versus WS-Routing

• WS-Routing provides a mechanism for describing the actual route of a message
• WS-Referral provides a mechanism to configure how SOAP routers will build a message path
  – i.e., controls what information is contained in the WS-Routing header
  – Itself realized as special SOAP messages directed towards routers
    • Contain elements called referral statements

Main use
• To delegate responsibility of processing some aspects of a service to a third party in a manner completely transparent to the end user
Referral Statements

Five elements

- **for** Indicates the URIs for which the referral is intended

- **if** A set of conditions the recipient must understand in order to use the referral
  - Two mandatory conditions defined: *invalidates*, *ttl*
  - Extensible

- **go** A set of URIs that defines the “via” element for messages intended for the “for” URI
  - Can choose one of this set

- **desc** Additional information that can be used by the recipient
  - To help make the routing decision (e.g., which of the go URIs to use)

- **refId** Unique identifier to identify referral instance
Example of a WS-Referral Statement

For any SOAP actor name matching the SOAP actor "soap://example.org/some.doc" or SOAP actors starting with "soap://example.org/topics/icebergs", if this referral is less than 12 hours (43,200,000 milliseconds) old then go via "soap://example.com/mirror".
Example: Removing the Referral

```xml
<r:ref xmlns:r="http://schemas.xmlsoap.org/ws/2001/10/referral">
  <r:for>
    <r:prefix>soap://example.org/topics/icebergs</r:prefix>
  </r:for>
  <r:if>
    <r:invalidates>
      <r:rid>uuid:09233523-345b-4351-b623-5dsf35sgs5d6</r:rid>
    </r:invalidates>
  </r:if>
  <r:go/>
  <r:refId>uuid:09233523-345b-4351-b623-5dsf35sgs5d6</r:refId>
</r:ref>
```
Referral Messages

Transmission of referral statements through three kinds of messages

- **Register** messages: “Push” mechanism for inserting routing entries
  - Recipient explicitly accepts or rejects the update

- **Query** messages: A SOAP router can be queried for referral statements using this kind of message

- **Referrals header**: Referral statement is piggybacked onto regular SOAP messages
  - Provides a mechanism for a SOAP router to inform its client of a better route to it

- Specification lays out format for request/response messages for each of the above (including fault elements)
Support for WS-Routing in the .NET Framework

• Part of the Web Services Extension (WSE) package

• Referral statements stored in an on-disk XML file, specified to the application using the web.config file
  – Automatic look-up and registration of new referrals into this file

[Code walkthrough]
• Setting up web.config to point to XML file storing referral entries

• Statically adding referral entries to the file

• Dynamically adding referral entries
  – Via context.Referal structure