Tutorial VII. Web Service Deployment and Registry

Web services deployment
Web services registry
Web services discovery
1. Web services deployment

- If you are using TOMCAT/AXIS
  - Refactoring your package name to have a prefix such as “c408h0xx.OmniEditor”
  - `cd /u/yijun/axis/c408h0xx`
  - Copy your class files to `/u/yijun/deployed/c408h0xx`
  - Name your deployment file “deploy.wsdd”, and “undeploy.wsdd”
  - `../seawolf-deploy.sh`, `../seawolf-undeploy.sh` (for publish purposes), ask instructor if you can’t do it (permission error)
  - For test purposes: `../skywolf-deploy.sh`, `../skywolf-undeploy.sh`, or `../local.sh`
  - Go to `http://seawolf.cdf.toronto.edu:8081/axis` to verify your web service is deployed
continued

• If you are using gSOAP
  – Modify the web server example to host your own web service
  – Let the server to listen to a port number 8xxx and deploy it to CDF

nohup /u/yijun/software/gsoap-2.7/soapcpp2/samples/webserver 8xxx &
2. Web Services Repository

- **UDDI Server: web services registry**
  - A UDDI registry contains three kinds of information
    - **White pages**: Information such as the name, address, telephone number, and other contact information of a given business.
    - **Yellow pages**: Information that categorizes businesses. This is based on existing (non-electronic) standards
    - **Green pages**: Technical information about the Web services provided by a given business
  - **jUDDI**: [http://ws.apache.org/juddi](http://ws.apache.org/juddi)
    - a java implementation of UDDI specification
    - can be deployed and accessed through a UDDI browser to browse/search web services
  - jUDDI on CDF: [http://skywolf.cdf.toronto.edu:8081/juddi](http://skywolf.cdf.toronto.edu:8081/juddi)
continued

- Inquire web services on UDDI server
  - jUDDI: publisher table in database
    
    
    
    ```
    mysql -h skywolf juddi -e 'select PUBLISHER_ID from PUBLISHER'
    
    +-----+-----------------+--------+--------+-----------+-----------+-----------+-----------+-----------------+-------+-------+
    | PUB  | PUB             | LAST   | FIRST  | MIDDLE   | WORK     | MOBILE    | PAGE      | EMAIL            | ADMIN | ENABLED |
    | LISH | LER | NAME | NAME | INIT | PHONE | PHONE | R | ADDRESS |        |        |     |     |
    | ID   |    |    |     |     |      |      |   |       | E M |      |    |     |
    |      |    |    |     |     |      |      |   |       | M A |      |    |     |
    +-----+-----------------+--------+--------+-----------+-----------+-----------+-----------+-----------------+-------+-------+
    | jdoe |    |    |     |     |      |      |   |       |     |      |    |     |
    | c408h001 | Team One | NULL | NULL | NULL | NULL | NULL | NULL | NULL | john.doe@apache.org | tru   | tru   |
    | c408h002 | Team Two    | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h002@cdf.toronto.edu | tru   | tru   |
    | c408h003 | Team Three  | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h003@cdf.toronto.edu | tru   | tru   |
    | c408h004 | Team Four   | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h004@cdf.toronto.edu | tru   | tru   |
    | c408h005 | Team Five   | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h005@cdf.toronto.edu | tru   | tru   |
    | c408h006 | Team Six    | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h006@cdf.toronto.edu | tru   | tru   |
    | c408h008 | Team Eight  | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h008@cdf.toronto.edu | tru   | tru   |
    | c408h009 | Team Nine   | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h009@cdf.toronto.edu | tru   | tru   |
    | c408h010 | Team Ten    | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h010@cdf.toronto.edu | tru   | tru   |
    | c408h011 | Team Eleven | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h011@cdf.toronto.edu | tru   | tru   |
    | c408h012 | Team Twelve | NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h012@cdf.toronto.edu | tru   | tru   |
    | c408h014 | Team Fourteen| NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h014@cdf.toronto.edu | tru   | tru   |
    | c408h015 | Team Fifteen| NULL | NULL | NULL | NULL | NULL | NULL | NULL | c408h015@cdf.toronto.edu | tru   | tru   |
    +-----+-----------------+--------+--------+-----------+-----------+-----------+-----------+-----------------+-------+-------+

  ```

  - HTML-based deployed web services
    
    ```
    http://seawolf.cdf.toronto.edu:8081/axis
    ```
Deployed Web Services of CSC408H

Here is a list of published web services by the end of phase B:

- Team 01 (WSDL)
- Team 02 (to be deployed, please change file permission of /software/soap-2.7/soapcpp2/samples/mayask/calc wsdl to c408h002)
- Team 03 (WSDL)
- Team 04's home page (WSDL)
- Team 05's web service (WSDL)
- Team 06's web service (WSDL)
- Team 07's web service (WSDL)
- Team 08's web service (WSDL)
- Team 09's home page (WSDL) (to be deployed, please change file permission of c408h009/deploy.wsdd and your wsdl files to c408h009)
- Team 10's Home page (WSDL)
- Team 11's web service (WSDL)
- Team 12's web service (WSDL)
- Team 13's web service (WSDL)
- Team 14's web service (WSDL)
- Team 15's web page (WSDL)

- View the list of Web services deployed at the Tomcat/Axis server
- Call a local endpoint that lists the caller's http headers (or see its WSDL)

Validating Axis

- Validate the local installation's configuration
- Administer Axis
- SOAPMonitor

Resources

- See the course home page on how to set up your web service.
- Visit the Apache-Axis Home Page If you have problems getting Axis to work, consult the Axis Wiki and then try the Axis user mailing list
And now... Some Services

- AdminService (we'll)
  - AdminService
- OmniEdit (we'll)
  - move
  - find
  - mark
  - select
- c408h005.OmniEditor (we'll)
  - makeConnection
  - disConnect
  - sendCommand
  - getFacState
- OmniEditor-c408h012 (we'll)
  - subscribe
  - find
  - list
  - upload
  - download
  - getCurrentLocation
  - deleteChannel
  - deleteAll
- urOmni-editor (we'll)
  - Upload
  - isMyQueueEmpty
  - Download
  - InsertText
  - Delete
  - TextSelected
  - Disconnect
- OMNIEditor (we'll)
  - list
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```xml
<wsdl definitions targetNamespace="urn:src">  
  - <wsdl message name="moveResponse">  
    <wsdl part name="MoveReturn" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="selectResponse">  
    <wsdl part name="SelectReturn" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="selectRequest">  
    <wsdl part name="in0" type="xsd:string"/>
    <wsdl part name="in1" type="xsd:string"/>
    <wsdl part name="in2" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="markResponse">  
    <wsdl part name="MarkReturn" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="moveRequest">  
    <wsdl part name="in0" type="xsd:string"/>
    <wsdl part name="in1" type="xsd:string"/>
    <wsdl part name="in2" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="findRequest">  
    <wsdl part name="in0" type="xsd:string"/>
    <wsdl part name="in1" type="xsd:string"/>
    <wsdl part name="in2" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="findResponse">  
    <wsdl part name="FindReturn" type="xsd:string"/>
  </wsdl message>
  
  - <wsdl message name="markRequest">  
    <wsdl part name="MarkRequest" type="xsd:string"/>
  </wsdl message>
</wsdl definitions>
```
3. Publish and Discover Web Services

- **UDDI4J**: java implementation of UDDI client side
  - Provide API to interact with UDDI registry to:
    - Publish: How the provider of a Web service registers itself.
    - Find: How an application finds a particular Web service.
    - Bind: How an application connects to, and interacts with, a Web service after it's been found.

- **Public and private UDDI registry nodes**
  - IBM UDDI Test Registry (providing “publish” and “discover”)
  - Microsoft UDDI Test Registry (providing “publish” and “discover”)
    [http://test.uddi.microsoft.com](http://test.uddi.microsoft.com)

- **UDDI browser**: browse/search web services
4. Use Web Services

- Find/Discover the WSDLs listed in a specific web registry
- Generate the corresponding client
  - AXIS: wsdl2java
  - gSOAP: wsdl2h
- OmniEditor project:
  - Analyze the variability of hosted web services, write a generic client to use several variants of the OmniEditor, find a way to adapt to the changes
Continued (OmniEditor)

- Supporting variability
  - Compile time variability
    - Macros (compile time)
      » #define WSDLURL “c408h001.OmniEditor.wsdl”
      » make -DWSDLURL=c408h001.OmniEditor.wsdl
    - Java annotations
  - Run-time variability: generative programming
    <config>
      <service name="c408h001.OmniEditor.wsdl"/>
    </config>
    wsdIURL = getXPath("config.xml", "/config/service/@name");
  - Java properties
    » wsdIURL = System.getProperty("wsdlurl");
    » java -Dwsdlurl=... client
    » client.properties
Continued (OmniEditor)

– Pick one of the editors
  • VIM
  • Eclipse
  • Emacs
  • jEdit

– Each web services is integrated by at least one client. A system test may involve two instances of the clients