Easy-WSDL Toolbox
“In the beginning was the interface definition, and the interface definition language was WSDL”

Nicolas Salatgé, Phd, Research Engineer at EBM WebSourcing and WSDL enthusiast
Agenda

• Problems
• State of the art
• WSDL1.1 vs WSDL2.0
• Proposed solution: easy WSDL toolbox
  – Presentation
  – How to use easy WSDL
  – Extensions: easy SAWSDL and easy WSDL4BPEL
Problems

• SOA softwares need to handle service descriptions ➔ WSDL

• Two wsdl versions : WSDL 1.1 and WSDL 2.0

• Question
  
  What is the WSDL version that I must integrate?
  
  ... WSDL1.1, WSDL 2.0 or both

• Examples: In SOA governance software like Dragon, it is necessary to handle these two formats.
State of the Art

• **WSDL4J:**
  – Handle WSDL 1.1: read/write/create

• **Woden:**
  – Handle WSDL 2.0: read/write/create
  – Can convert WSDL1.1 to WSDL2.0 using the W3C XSL Stylesheet
    • => Problem: Some of important information in WSDL1.1 are lost
    • Example: the “name” attribute in WSDL1.1 definition does not exist in WSDL2.0 description

• **Commons problems:**
  – Must cast binding to known the used binding
  – Impossible to get the types and elements defining in the schema

• **Conclusion:**
  → **WSDL4J and Woden are not sufficient to handle all elements in service description.**
WSDL1.1 Vs. WSDL2.0

WSDL 1.1

<definitions name="...">
  <types>
  </types>
  <messages>
  </messages>
  <portTypes>
  </portTypes>
  <binding>
    <service>
      SOAP1.1
      SOAP1.2
      HTTP
      MIME
    </service>
  </binding>
</definitions>

WSDL 2.0

<description>
  <types>
  </types>
  <interface>
  </interface>
  <binding>
    <service>
      SOAP
      HTTP
      RPC
    </service>
  </binding>
</description>
Proposed solution: easy-WSDL

• Features:
  – Aggregate WSDL1.1 and WSDL 2.0 => but, have an orientation towards WSDL2.0
    • Example: When two identical concepts are two different names, we conserve the name of WSDL2.0 elements:
      – 1: definitions in WSDL1.1 and description in WSDL2.0 => Description interface in easy-WSDL
      – 2: portType in WSDL1.1 and interfaceType in WSDL2.0 => InterfaceType interface in easy-WSDL
  – Provide the most of used bindings in the two versions:
    • Note: Contrary to Wodden or WSDL4J, no cast=> the easy WSDL api provides all necessary methods (ex: getSOAP11Binding4WSDL11).
  – Allow to get elements and types defining in the schema (easy-schema)
  – Uniform API to read WSDL1.1 and WSDL2.0
  – Uniform API to write WSDL1.1 and WSDL2.0
  – Uniform API to create WSDL1.1 and WSDL2.0 (Under development)
  – Consistent conversion between WSDL1.1 to WSDL2.0 (Under development)
  – Can be extended: Easy SAWSDL and Easy WSDL4BPEL
Proposed solution: easy-WSDL
How to use easy-WSDL

• **Read a WSDL 1.1 or 2.0 (uniform API):**
  Description desc = WSDLFactory.newInstance().newWSDLReader().readWSDL(new URI("http://api.google.com/GoogleSearch.wsdl"));

• **Write a WSDL 1.1 or 2.0 (uniform API):**
  Document doc = WSDLFactory.newInstance().newWSDLWriter().getDocument(desc);

• **Create a WSDL 1.1 or 2.0 (uniform API):**
  Description desc = WSDLFactory.newInstance().newDescription();
• SAWSDL specification:
  ➔ Extends service description (WSDL 1.1 or WSDL 2.0) adding semantic annotations. These semantic annotations correspond to attributes (ex: « modelReference ») added on any element in WSDL description (endpoint, service, …etc.).

• Easy-SAWSDL extensions:
  ➔ Extends easy-WSDL API adding methods in each interface. These added methods corresponding to SA-WSDL attributes (ex: getModelReference() method)
How use easy-SAWSDL

• **Read a SA-WSDL extension in WSDL 1.1 or 2.0 (uniform API):**
  Description desc = SAWSDLFactory.newInstance().newSAWSDLReader().readSAWSDL(new URI("./src/test/resources/descriptors/purchase.wsdl"));
  => desc.getInterfaces().get(0).getModelReference();

• **Write a WSDL 1.1 or 2.0 with SA-WSDL extensions (uniform API):**
  Document doc = SAWSDLFactory.newInstance().newSAWSDLWriter().getDocument(desc);

• **Create a WSDL 1.1 or 2.0 (uniform API):**
  Description desc = SAWSDLFactory.newInstance().newSADescription();
• Easy-WSDL4BPEL extensions
  => Allow to get or set partnerLink bpel extensions in service description.
How use easy-WSDL4BPEL

• **Read a BPEL extension in WSDL 1.1 or 2.0 (uniform API):**
  Description desc = WSDL4BPELFactory.newInstance().newWSDL4BPELReader()
  .readWSDL4BPEL(new URI("./src/test/resources/descriptors/purchase.wsdl"));
  => desc.getPartnerLinkTypes().get(0).getRoles().get(0).getName();

• **Write a WSDL 1.1 or 2.0 with SA-WSDL extensions (uniform API):**
  Document doc = WSDL4BPELFactory.newInstance().newWSDL4BPELWriter()
  .getDocument(desc);

• **Create a WSDL 1.1 or 2.0 (uniform API):**
  Description desc = WSDL4BPELFactory.newInstance().newWSDL4BPELDescription();
Conclusion

• Easy-WSDL has been tested on more of 50 services
  – Assert that values returned by easy-WSDL methods (actual value) are consistent with WSDL4J methods or Woden methods (expected value)
  – Plan to pass the W3C WSDL test suite so that the project can be registered on W3C Web Site
    • http://dev.w3.org/2002/ws/desc/test-suite/Dashboard.html

• Easy-WSDL is already integrated in SOA governance software: Dragon
http://easywsdl.ow2.org