Iridescent: Tool for Semantic Web Service Annotation

Demo Presentation

Thanos G. Stavropoulos (Phd Student)
Theo Mylonides (Undergrad. Student)

Outline

- Part I - Background Study
  1. The Semantic Web
  2. Web Service Technology
  3. Semantic Web Services
     - SAWSDL
- Part II – Iridescent Usage Scenario
  A. Iridescent Presentation
  B. Smart IHU Usage Scenario

Part I

Background Study

1 The Semantic Web

- Data on the Web
  - Syntactically described (HTML)
  - Human interpretable
  - Keyword search
- Semantic Web
  - Semantically annotated data
  - Machine interpretable
  - Semantic search

1 The Semantic Web

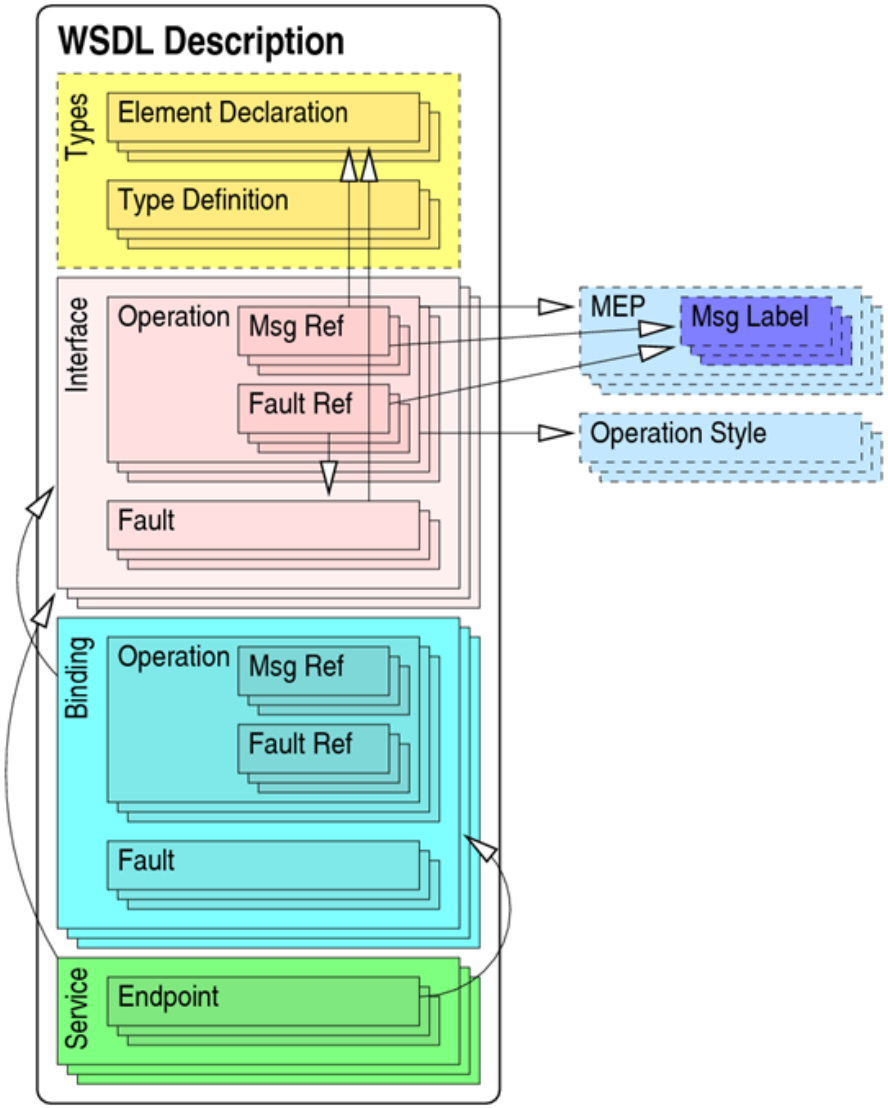
- Ontology
  - Taxonomy of interrelated concepts
- Markup-languages
  1. XML
  2. RDF
  3. OWL
- Material
  - IHU Virtual Lab
    - http://vlabs.ihu.edu.gr/index.php?id=29
  - CSD AUTH Semantic Web MSc Course
    - http://lpis.csd.auth.gr/mtpx/sw/index.htm

```
<owl:Class rdf:ID="PotableLiquid">
        <rdfs:subClassOf rdf:resource="
        #ConsumableThing" />
</owl:Class>
```

2 Web Services

- Service Computing
- Data as a Service
- W3C Standards
  - Web Service Description Language – WSDL
    - Syntactic Decriptions of IO
  - More
    - WS-* Stack (Discovery, Policy, Security)

2 Web Services WSDL



3 Semantic Web Services

- Synergy of
  - Semantic Web
  - Web Service
- Approaches
  - Top-down (Ontologies mapped to endpoints)
    - OWL-S
    - WSMO
  - Bottom-up (Endpoints annotated)
    - SAWSDL, W3C Recommendation

3 Semantic Web Services
SAWSDL

1. Annotation of WSDL in place
   - SAWSDL : modelReference

2. SchemaMappings: Transformations between
   - Ontological Structures
     - OWL, RDF
   - Web Service I/O
     - XML Schema
   - E.g.
     - XSLT ->
     - <- SPARQL

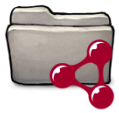
Part II

Iridescent Usage Guide
Evaluation Scenario

Iridescent Tool Overview

- SAWSDL Creator & Editor

  - Open multiple Web Service Descriptions
    - WSDL



  - Open multiple Ontologies
    - OWL,RDF
  - Semantically annotate
    - Add namespace
    - Add annotations (modelReferences)
    - Add transformations (XSLT, SPARQL)
    - Recommendations: Auto-match names

Iridescent Representation

- SAWSDL namespace import
  - sawsdl:http://www.w3.org/ns/sawsdl

- modelReference
  - Annotation

- liftingSchemaMapping
  - Transformation
  - e.g. XML to RDF/OWL (XSLT)

- loweringSchemaMapping
  - Transformation
  - e.g. RDF/OWL to XML (SPARQL)

antiRadiant Features

Aspect	Radiant	WSMO Studio	SOWER	Iridescent
Year	2007	2007	2011	2012
Documentation	✓	×	×	✓
Architecture	Eclipse 3 plugin	Eclipse 3 plugin, standalone	Web app.	standalone
WSDL files				
Local	✓	✓	✓	✓
Web	×	-	✓	✓
Multiple	✓	-	×	✓
Imports	×	×	×	✓
Find	✓	✓	?	✓
OWL files				
Local	✓	✓	✓	✓
Web	✓	-	✓	✓
Multiple	×	-	✓ same tree	✓ separately
Imports	×	-	✓	✓
Find	×	-	✓	✓
Namespace handling	✓add (outdated)	-	✓ add	✓ add/remove
Annotation	✓Drag ‘n Drop, ✓Right click	✓	✓Drag ‘n Drop	✓Drag ’n Drop, ✓Right Click, ✓Menu
Recommendation	×	×	×	✓

Usage Scenario
Smart IHU

- Live services that
  - Switch appliances On and Off
  - Read Power Consumption
  - Read Temperature, Humidity, Luminance etc.
- http://155.207.113.31:8080/aWESoME/
- Smart Building Ontology for Ambient Intelligence
  - BOnSAI http://lpis.csd.auth.gr/ontologies/bonsai/BOnSAI.owl

Scenario 1

1. Open Web Service description SmartPlugService.wsdl from local file
2. Add the SAWSDL namespace to the WSDL
3. Open BOnSAI.owl from local file
4. Search BOnSAI for class PowerConsumption

WSDL panel on the right

WSDL panel toolbar or right click on Definitions on WSDL Tree

Ontology panel on the left

Search Bar on Ontology Panel

Scenario 1 (cont'd)

5. Add ModelReference of PowerConsumption to ReadPower

6. Add ModelReference of PowerState to ReadStatus

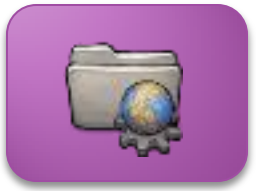
7. Observe Tree & code

Drag 'n Drop OWL Class
to WSDL node OR use button

Scenario 2

1. Open BOnSAI from URL
   http://lpis.csd.auth.gr/ontologies/bonsai/BOnSAI.owl

2. Open WSDL SensorBoardService.wsdl from URL
   http://smart.ihu.edu.gr:8080/aWESoME23G/SmartPlugService?wsdl



3. Run Recommendations on .xsd, .wsdl

Scenario 2 (cont'd)

4. Select Recommendations for .xsd
   - Temperature-> getRecentTemperatureResponse element
   - Humidity -> getRecentHumidityResponse element
   - Luminance -> getRecentLuminance element
5. Select Recommendations for .wsdl
   - Temperature-> getRecentTemperature Operation
   - Humidity -> getRecentHumidity Operation
   - Luminance -> getRecentLuminance Operation
6. Observe tree and code

Scenario 3
Schema Mappings

1. Open the PurchaseOrder Service from W3C
   http://www.w3.org/2002/ws/sawsdl/spec/examples/wsdl/PurchaseOrderService.wsdl
2. Add a lifting Schema mapping
   http://www.w3.org/2002/ws/sawsdl/spec/examples/mapping/OrderRequest2Ont.xslt
   to the OrgerRequest element

Right Click WSDL node
OR use button

Scenario 3 (cont'd)

- Observe code

```
<wsdl:description>
  ...
  <wsdl:types>
  <xsd:schema
    targetNamespace="http://www.w3.org/2002/ws/sawsdl/spec/examples/wsdl/PurchaseOrderSe

    <xsd:element name="OrderRequest">
      <xsd:complexType sawsdl:liftingSchemaMapping="http://www.w3.org/2002/ws/sawsdl/spe
        <xsd:sequence>
          <xsd:element name="firstName" type="xsd:string"/>
          <xsd:element name="lastName" type="xsd:string"/>
          <xsd:element name="item" type="item" minOccurs="1" maxOccurs="unbounded" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

Select WSDL node

Evaluation

- Ratings and Suggestions on
  - Understanding of Background
  - Functionality
  - Usability
  - Presentation

- Thank you