

## Semantic Web Services

# Exercise sheet 11

## Light-weight Annotations

### Exercise 1 (SA-WSDL) (5 points)

In the context of the VTA scenario, introduced in Exercise Sheet 4, let's consider an imaginary hotel company, called *The Blue Hotel* that allows its clients to check for rooms availability using the *BlueHotelService*.

- Create a simple ontology in WSMML that can be used to annotate the BlueHotelService
- Extend the WSDL description of the BlueHotelService in Listing 1 by including SA-WSDL annotations. Use concepts from the ontology you created before.

Listing 1: WSDL description of BlueHotelService

```
<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/ns/wSDL"
  targetNamespace="http://www.bluehotel.com/wSDL/BlueHotelService"
  xmlns:tns="http://www.bluehotel.com/wSDL/BlueHotelService"
  xmlns:bhns="http://www.bluehotel.com/schemas/BlueHotelService"
  xmlns:soap="http://www.w3.org/ns/wSDL/soap"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wSDLx="http://www.w3.org/ns/wSDL-extensions">

  <documentation>
    This document describes the Blue Hotel Web service.
  </documentation>

  <types>
    <xs:schema
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://www.bluehotel.com/schemas/BlueHotelService"
      xmlns="http://www.bluehotel.com/schemas/BlueHotelService">

      <xs:element name="checkAvailability" type="t:CheckAvailability"/>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="checkInDate" type="xs:date"/>
          <xs:element name="checkOutDate" type="xs:date"/>
          <xs:element name="roomType" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
  </types>
</description>
```

```
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="checkAvailabilityResponse" type="tCheckAvailabilityResponse"/>
<xs:complexType>
  <xs:sequence>
    <xs:element name="roomType" type="xs:string"/>
    <xs:element name="rateType" type="xs:string"/>
    <xs:element name="rate" type="xs:double"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="invalidDataError" type="xs:string"/>
</xs:schema>
</types>
<interface name = "BlueServiceInterface" >
  <fault name = "invalidDataFault" element = "bhns:invalidDataError"/>

  <operation name="opCheckAvailability"
    pattern="http://www.w3.org/ns/wSDL/in-out"
    style="http://www.w3.org/ns/wSDL/style/iri"
    wsdl:safe = "true">
    <input messageLabel="In"
      element="bhns:checkAvailability" />
    <output messageLabel="Out"
      element="bhns:checkAvailabilityResponse" />
    <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
  </operation>
</interface>

<binding name="BlueServiceSOAPBinding"
  interface="tns:BlueServiceInterface"
  type="http://www.w3.org/ns/wSDL/soap"
  wssoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP"/>

  <fault ref="tns:invalidDataFault"
    wssoap:code="soap:Sender"/>

  <operation ref="tns:opCheckAvailability"
    wssoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>
</binding>

<service name="BlueService" interface="tns:BlueServiceInterface">
  <endpoint name="reservationEndpoint"
    binding="tns:BlueServiceSOAPBinding"
    address = "http://www.bluehotel.com/BlueService"/>
</service>
</description>
```

## Exercise 2 (WSMO-Lite) (5 points)

Transform the annotated WSDL obtained in Exercise 1 into the WSMO-Lite RDFS form. The WSMO-Lite representation should include models for the capability of the service and at least one non-functional property (e.g. price of the service).

### **Exercise 3 (hRESTS) (5 points)**

Consider Flickr RESTful service<sup>1</sup>. Using hRESTS annotate the HTML pages describing the Upload Photos and Replacing Photos operations

### **Exercise 4 (MicroWSMO) (5 points)**

Enhance the obtained hRESTS description from Exercise 3 with MicroWSMO pointers.

---

<sup>1</sup> <http://www.flickr.com/services/api/>