

## Semantic Web Services

### Lightweight SWS: WSMO-Lite, MicroWSMO

Lecture X – 4<sup>th</sup> June 2009  
Dieter Fensel

slides from Jacek Kopecný



### Where are we?

| #  | Date                   | Title                                |
|----|------------------------|--------------------------------------|
| 1  | 5 <sup>th</sup> March  | Introduction                         |
| 2  | 12 <sup>th</sup> March | Web Science                          |
| 3  | 19 <sup>th</sup> March | Service Science                      |
| 4  | 26 <sup>th</sup> March | Web Services (WSDL, SOAP, UDDI, XML) |
| 5  | 2 <sup>nd</sup> April  | Web 2.0 and RESTful services         |
| 6  | 23 <sup>rd</sup> April | WSMO                                 |
| 7  | 30 <sup>th</sup> April | WSML                                 |
| 8  | 7 <sup>th</sup> May    | WSMX                                 |
| 9  | 14 <sup>th</sup> May   | OWL-S and others                     |
| 10 | 28 <sup>th</sup> May   | <b>WSMO-Lite, MicroWSMO</b>          |
| 11 | 4 <sup>th</sup> June   | SWS Use Cases                        |
| 12 | 18 <sup>th</sup> June  | seekda: the business point of view   |
| 13 | 25 <sup>th</sup> June  | Mobile services                      |
| 14 | 2 <sup>nd</sup> July   | Exam Preparation                     |



### Overview

- Motivation
- Technical solutions
  - WSMO-Lite Service semantics
  - SAWSDL, WSMO-Lite
  - hRESTS, MicroWSMO

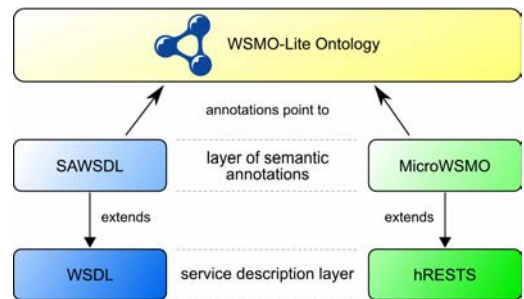
## Motivation

## Motivation



- SAWSDL
  - New W3C standard
  - Building on SAWSDL, URIs
  - Assuming RDF, OWL
- Lightweight approach
  - WSMO modularized
  - Key pieces of service semantics
  - Simple semantic representation

## Technical Solutions



## WSMO-Lite Ontology: Service Semantics

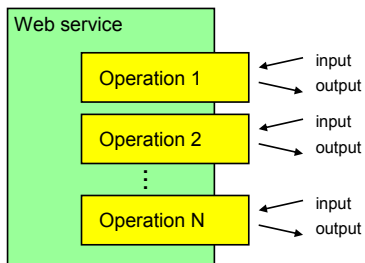


## Overview



- WSDL-based service model
- SAWSDL-based annotations
- Four types of semantics
  - Expressed in RDF(S)
  - Attached to the service model

## WSDL Simplified



## Service Model in RDFS



```
Service          a rdfs:Class .
  hasOperation    a rdf:Property .
Operation        a rdfs:Class .
  hasInputMessage a rdf:Property .
  hasOutputMessage a rdf:Property .
  hasInputFault   a rdf:Property .
  hasOutputFault  a rdf:Property .
Message          a rdfs:Class .
```

## SAWSDL Properties



- On elements of the service model.
- modelReference
  - Pointers to semantics
- liftingSchemaMapping
  - Mapping from WS on-the-wire messages to KR (RDF)
- loweringSchemaMapping
  - Mapping from KR to WS messages

## SAWSDL Lifting/Lowering



## Service Semantics



For automation of:

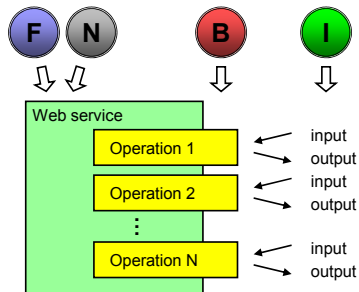
- Discovery
- Ranking, selection
- Composition
- Invocation
- etc.

## Types of Service Semantics



- Functional
  - What the service does
- Behavioral
  - How the client talks to the service
- Information model
  - For handling data
  - Incl. lifting/lowering
- Nonfunctional
  - Policies, QoS, price, location etc.

## Semantics in Service Model



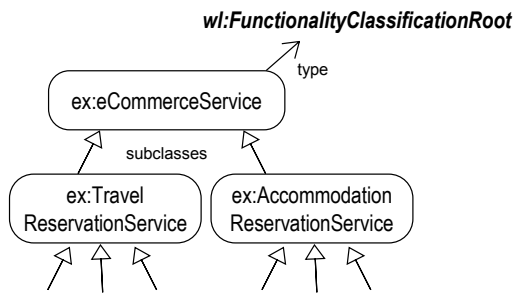
## Functional Semantics



- For service discovery, composition
- *Category*
  - Functionality categorization
  - E.g. eCl@ss, UDDI
  - Or tagging, folksonomies
- *Capability*
  - Precondition, Effect
  - Using WSML rule languages



## Category Example



## Category Example (RDF)

```
ex:eCommerceService
  rdf:type wl:FunctionalityClassificationRoot .
ex:TravelReservationService
  rdfs:subClassOf ex:eCommerceService .
ex:AccommodationReservationService
  rdfs:subClassOf ex:eCommerceService .
...
```

## Capability Example

```
ex:RomaHotelReservationPrecondition
  rdf:type wl:Precondition ;
  rdf:value ""
    ?request
      [ numberOfGuests hasValue ?guests
        and city hasValue ?city ]
      memberOf ReservationData
  and ?guests <= 10
  and ?city = 'Roma'
  ""^^wsmi:AxiomLiteral .
```

## Nonfunctional Semantics


- For ranking and selection
- Not constrained, any ontologies
- Example:



```
ex:PriceSpecification
  rdfs:subClassOf wl:NonFunctionalParameter .
ex:ReservationFee
  rdf:type ex:PriceSpecification ;
  rdf:value "15"^^ex:euroAmount .
```


## Behavioral Semantics



- For invocation, composition, process mediation 
- Functionalities on operations
  - Capabilities, categories
- Client selects operation to invoke next
  - Instead of being strictly guided by an explicit process
- Example functional category for operations:  
WebArch interaction safety

## Information Semantics



- For invocation, composition, data mediation 
- Not constrained, any ontologies
- Refer to course *Semantic Web* (WS)

## Service Semantics Summary



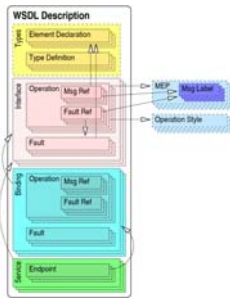
wl:Ontology  
wl:FunctionalityClassificationRoot  
wl:Precondition  
wl:Effect > together form a *Capability*  
wl:NonFunctionalParameter

Different types of semantics on one component:  
– E.g. functionality and nonfunc. property on a service

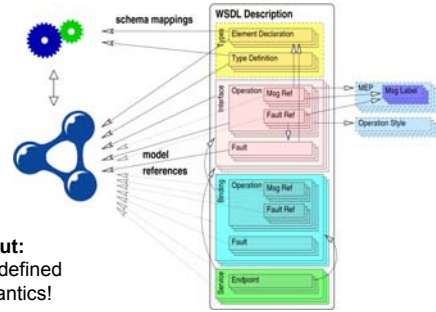
## WSMO-Lite in SAWSDL



## WSDL

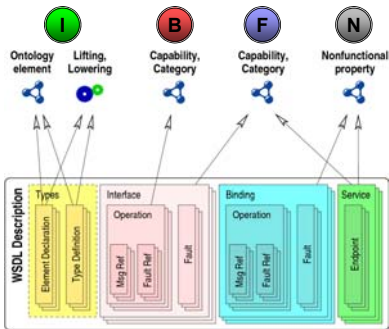


## SAWSDL



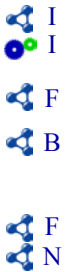
But:  
no predefined semantics!

## WSMO-Lite in SAWSDL



## WSMO-Lite Example

```
<wsdl:description>
  <wsdl:types> <xs:schema>
    <xs:element name="ReservationRequest"
      sawsdl:modelReference="&ex;Reservation"
      sawsdl:loweringSchemaMapping="&ex;ResMapping.xsparql" ... />
  </xs:schema> </wsdl:types>
  <wsdl:interface name="HotelReservations"
    sawsdl:modelReference="
      &ex;AccommodationReservationService">
    <wsdl:operation name="searchForRooms"
      sawsdl:modelReference="&wsdlx;SafeInteraction">
      ...
    </wsdl:operation>
  </wsdl:interface>
  <wsdl:service name="RomaHotels" interface="HotelReservations"
    sawsdl:modelReference="&ex;RomaHotelReservationPrecondition
      &ex;ReservationFee" ... />
</wsdl:description>
```

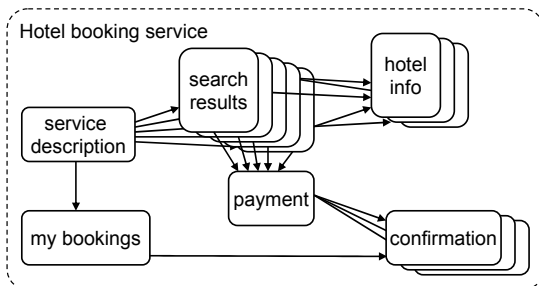


# hRESTS & MicroWSMO

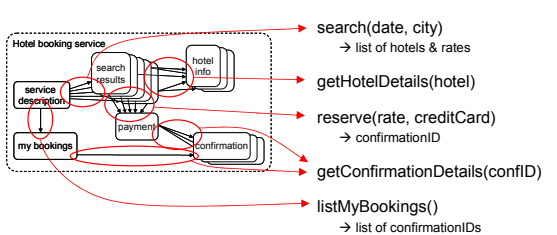
## Recap: RESTful WS

- A RESTful Web service is:
  - A set of Web resources
  - Interlinked
  - Data-centric, not functionality-centric
  - Machine-oriented

## RESTful Web Services



## Hypermedia → Operations



nouns vs. verbs



## hRESTS



- "There's usually an HTML page"
  - There's no WSDL for Web apps
  - APIs described mostly in **text**
- Identifying machine-readable parts
  - Service, its operations
  - Resource address, HTTP method
  - Input/output data format
- **hRESTS** microformat

## Example Description



Description of the ACME Hotels service:

The operation `getHotelDetails` is invoked using the method GET at `http://example.com/h/{id}`, with the ID of the particular hotel replacing the parameter `id`. It returns the hotel details in an `ex:hotelInformation` document.

## Example HTML



```
<p>Description of the
ACME Hotels service:</p>
<p>
The operation <code>getHotelDetails</code> is
invoked using the method <span>GET</span>
at <code>http://example.com/h/{id}</code>,
with the ID of the particular hotel replacing
the parameter <code>id</code>.
It returns the hotel details in an
<code>ex:hotelInformation</code> document.
</p>
```

## Example hRESTS



```
<div class="service" id="svc">
<p>Description of the
<span class="label">ACME Hotels</span> service:</p>
<div class="operation" id="op1">
<p>
The operation <code class="label">getHotelDetails</code> is
invoked using the method <span class="method">GET</span>
at <code class="address">http://example.com/h/{id}</code>,
with <span class="input">the ID of the particular hotel replacing
the parameter <code>id</code>.</span>
It returns <span class="output">the hotel details in an
<code>ex:hotelInformation</code> document.</span>
</p></div>
</div>
```

## Example w/o hRESTS



```
<p>Description of the
    ACME Hotels    service:</p>

<p>
The operation
    getHotelDetails    is
invoked using the method
    GET
at
    http://example.com/h/{id}
with
    the ID of the particular hotel replacing
the parameter <code>id</code>.
It returns
    the hotel details in an
<code>ex:hotelInformation</code> document.
</p>
```

## hRESTS



- HTML for RESTful Service Description
- Introduces the service model structure
  - service (+ label)
  - operations (+ address, method)
  - input, output
- Can also be in RDFa
- Basis for extensions:
  - MicroWSMO adds semantic annotations

## MicroWSMO



- Extends hRESTS
  - model for model references
  - lifting, lowering
- Applies WSMO-Lite semantics

## MicroWSMO



```
<div class="service" id="svc">
<p><span class="label">ACME Hotels</span> is a
  <a rel="model" href=".../ecommerce/hotelReservation">
    hotel reservation</a> service.</p> ...
<div class="operation" id="op1"><p> ...
  <span class="input">A particular hotel ID replaces the param
    <a rel="model" href=".../onto.owl#Hotel"><code>id</code></a>
    (<a rel="lowering" href=".../hotelID.xslt">lowering</a>).
  </span> ...
</p></div>
</div>
```

## Semantics Implied in Web



- Hypermedia → behavioral semantics
  - Links become available through interaction
- Uniform interface → functional semantics
  - GET, PUT, DELETE have known effects
  - GET is *safe*, PUT and DELETE idempotent
- Self-description → information model
  - Operation output data can specify what it is
    - GRDDL, other semantic annotations

## Conclusions



## Conclusions



- Building on **SAWSDL**
  - Both for **WS-\*** and **RESTful** services
- **Bottom-up** annotations
  - Based on tech already known by WS people
- **Modular** descriptions
  - Only use what you need

## Further Developments



- Adding WSMO-Lite and MicroWSMO support to WSMX
- Tool support for service annotation
- Standardization of WSMO-Lite

## References



- SAWSDL: <http://w3.org/TR/sawSDL>
- WSMO-Lite: <http://cms-wg.sti2.org/TR/d11>
- hRESTS & MicroWSMO:  
<http://cms-wg.sti2.org/TR/d12>
- RDF: <http://w3.org/TR/rdf-primer>
- RDFS: <http://w3.org/TR/rdf-schema>
- REST: <http://en.wikipedia.org/wiki/REST>  
– Includes RESTful Web services
- eCI@ss: <http://www.eclass-online.com>

## Next Lecture



| #  | Date                   | Title                                |
|----|------------------------|--------------------------------------|
| 1  | 5 <sup>th</sup> March  | Introduction                         |
| 2  | 12 <sup>th</sup> March | Web Science                          |
| 3  | 19 <sup>th</sup> March | Service Science                      |
| 4  | 26 <sup>th</sup> March | Web Services (WSDL, SOAP, UDDI, XML) |
| 5  | 2 <sup>nd</sup> April  | Web 2.0 and RESTful services         |
| 6  | 23 <sup>rd</sup> April | WSMO                                 |
| 7  | 30 <sup>th</sup> April | WSML                                 |
| 8  | 7 <sup>th</sup> May    | WSMX                                 |
| 9  | 14 <sup>th</sup> May   | OWL-S and others                     |
| 10 | 28 <sup>th</sup> May   | WSMO-Lite, MicroWSMO                 |
| 11 | 4 <sup>th</sup> June   | <b>SWS Use Cases</b>                 |
| 12 | 18 <sup>th</sup> June  | seekda: the business point of view   |
| 13 | 25 <sup>th</sup> June  | Mobile services                      |
| 14 | 2 <sup>nd</sup> July   | Exam Preparation                     |



## Questions?

