

Semantic Web Technologies

Semantic Web Services: Approaches

Jos de Bruijn
 jos.debruijn@deri.org

Digital Enterprise Research Institute (DERI)
 University of Innsbruck, Austria

June 6, 2006

Outline

- OWL-S
 - Service Profile
 - Service Model
 - Service Grounding

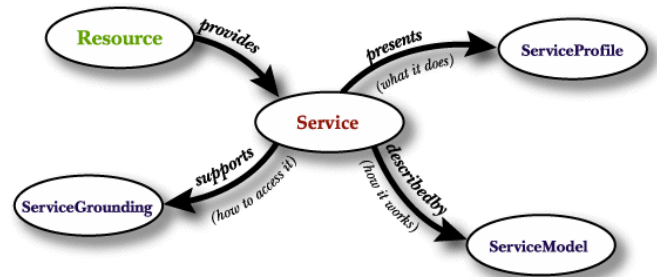
- WSMO
 - Ontologies
 - Web Services
 - Goals
 - Mediators

WSDL-S

OWL-S: OWL-Services

- ▶ OWL Ontology for modeling Services
- ▶ Functionality of the service
- ▶ Non-functional properties
- ▶ Implementation (process) of the service
- ▶ Grounding: how to invoke service
 - ▶ Connection with WSDL

OWL-S Upper Ontology of Services

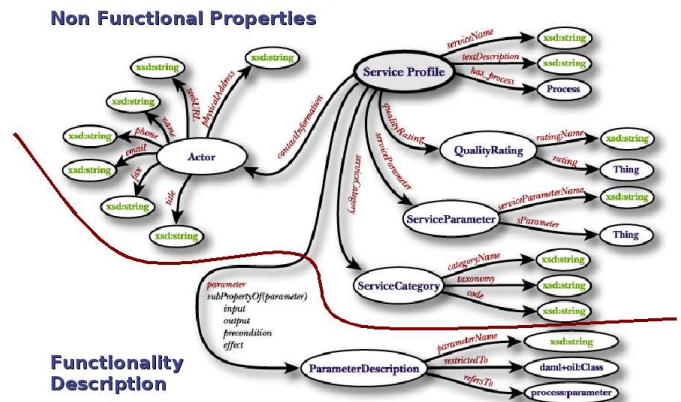


Based on slides by David Martin

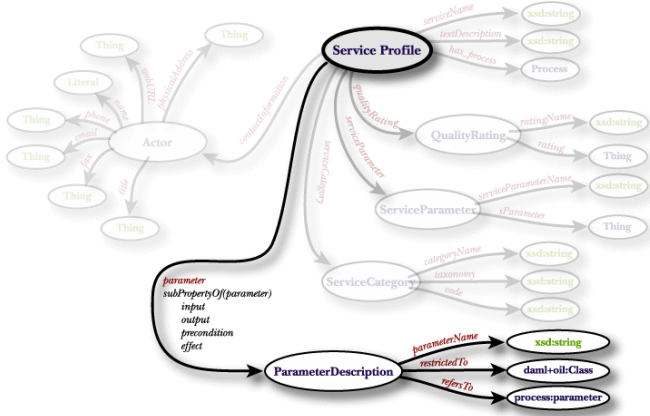
OWL-S Service Profile

- ▶ High-level description of service
 - ▶ Functional Description
 - ▶ Non-functional Description
- ▶ Used for
 - ▶ Advertising
 - ▶ Discovery
 - ▶ Selection

Service Profile



Functional Description



384/425

Functional Description

- ▶ What the service **provides**
- ▶ in terms of **IOPEs**
- ▶ **Inputs**
 - ▶ Typed input “variables”
 - ▶ Typing using OWL classes
- ▶ **Outputs**
 - ▶ Typed output “variables”
 - ▶ Typing using OWL classes
- ▶ **Preconditions**
 - ▶ Conditions over the input
 - ▶ Choice between different FOL-based formalisms
- ▶ **Effects**
 - ▶ Also **postconditions**
 - ▶ Conditions over the output
 - ▶ Effects of the service
 - ▶ Choice between different FOL-based formalisms

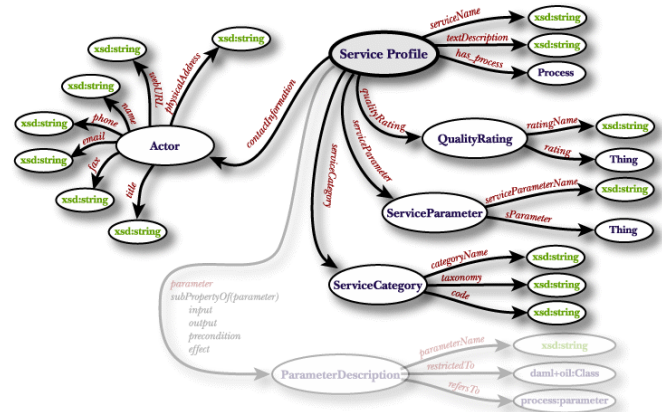
385/425

Functional Description (cont'd)

- ▶ DL-based Discovery
 - ▶ Matching inputs, outputs
 - ▶ DL-based reasoning
- ▶ Using preconditions, effects
 - ▶ Precondition, effect are closed formulas
 - ▶ Impossible to quantify over both
 - ▶ $\forall x(\text{precondition}(x) \Rightarrow \text{effect}(x))$
 - ▶ Not possible to relate input and output
 - ▶ Consider simple case of addition: $x + y = z$, with x, y inputs, z output

386/425

Non-Functional Description



387/425

Non-Functional Description (cont'd)



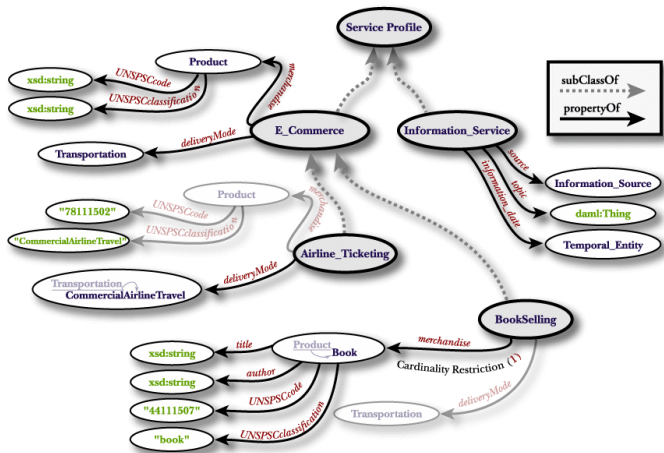
388/425

Usage of Profiles

- ▶ Class-hierarchical yellow pages
 - ▶ Implicit capability characterization
 - ▶ Arrangement of attributes on class hierarchy
 - ▶ Can use multiple inheritance
 - ▶ Relies primarily on non-functional properties
- ▶ “Summary” of Functionality of service
 - ▶ IOPE
 - ▶ Advertising
 - ▶ Discovery

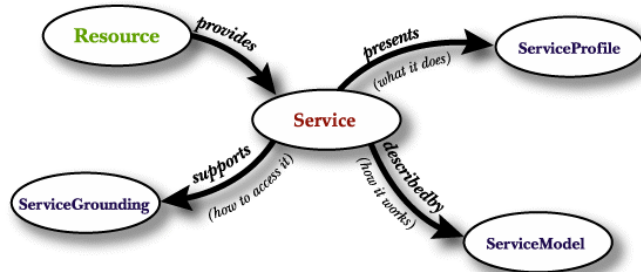
389/425

Profile Hierarchy



390/425

OWL-S Upper Ontology of Services



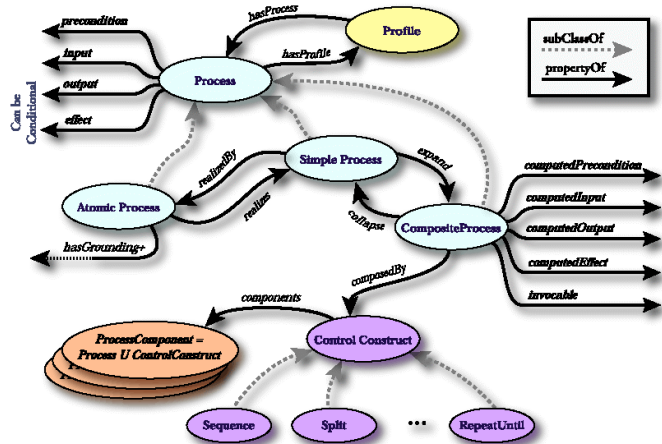
392/425

Service/Process Model

- ▶ Process
 - ▶ Description of service provider's behavior
 - ▶ Tells service user how and when to interact (read/write messages)
- ▶ & Process control
 - ▶ Ontology of process state; supports status queries
 - ▶ (stub at present)
 - ▶ Monitoring
- ▶ Used for:
 - ▶ Service **invocation, planning/composition, interoperation, monitoring**
- ▶ All processes have
 - ▶ Inputs, outputs, preconditions and effects
- ▶ Composite processes
 - ▶ Control flow
 - ▶ Data flow

393/425

Process Model

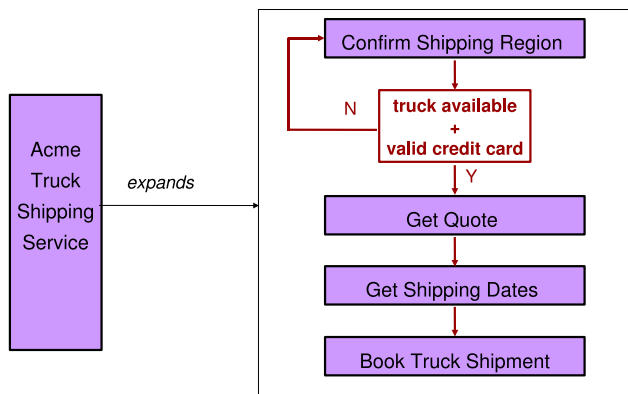


394/425

Process Example

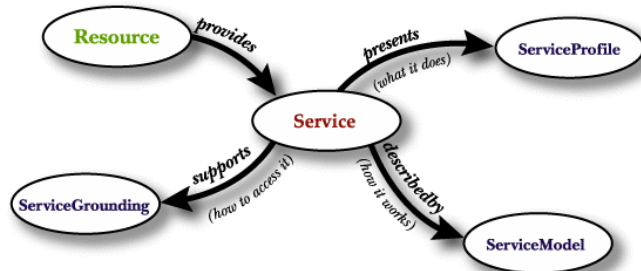
AcmeTruckShpng

ExpandedAcmeTruckShpng



395/425

OWL-S Upper Ontology of Services



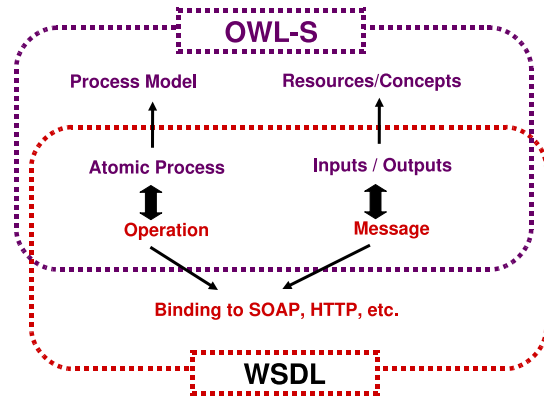
397/425

OWL-S Grounding

- ▶ Implementation-specific
 - ▶ Message format
 - ▶ Transport mechanisms
 - ▶ Protocols
 - ▶ Serializations of types (OWL Classes)
- ▶ Service Model describes interaction
- ▶ Grounding describes how to interact
- ▶ Grounding builds upon WSDL

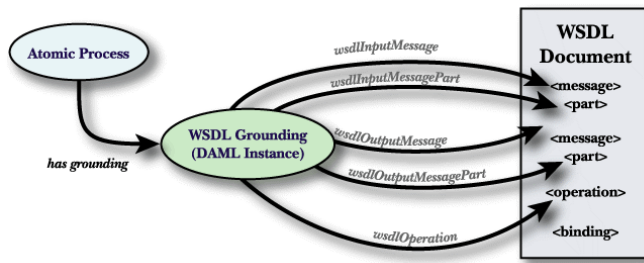
398/425

OWL-S/WSDL Grounding



399/425

OWL-S/WSDL Grounding (cont'd)



400/425

Review: OWL-S Upper Ontology of Services



401/425

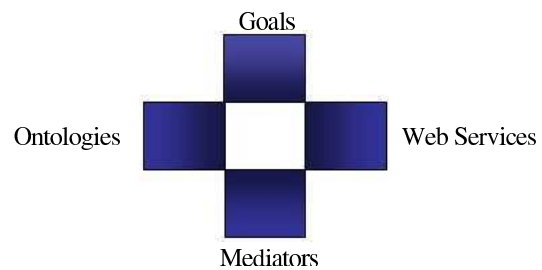
The Web Service Modeling Ontology WSMO

Introduction

- ▶ An ontology for Semantic Web Services
- ▶ Provides conceptual model for SWS
- ▶ Based on the Web Service Modeling Framework WSMF
- ▶ Principles of WSMO:
 - ▶ Ontology-based descriptions
 - ▶ Strict decoupling of components
 - ▶ Strong mediation between components
 - ▶ Interface vs. Implementation

403/425

The Web Service Modeling Ontology WSMO



404/425

The Web Service Modeling Ontology WSMO

Ontologies

- ▶ Provide terminology for:
 - ▶ Data exchanged between service requesters and providers
 - ▶ Description of other WSMO elements
- ▶ Ontologies consist of:
 - ▶ Concepts
 - ▶ Attributes
 - ▶ Relations
 - ▶ Functions
 - ▶ Instances
 - ▶ Axioms

406/425

The Web Service Modeling Ontology WSMO

Web Service descriptions

- ▶ Functionality offered by the Web Service
- ▶ Functional description, in the form of a **capability**:
 - ▶ Assumptions
 - ▶ Cannot be checked
 - ▶ Usually indicate dependency on real world
 - ▶ Preconditions
 - ▶ Conditions over the input
 - ▶ Effects
 - ▶ Changes in the real world as a result of execution of the Web Service
 - ▶ Postconditions
 - ▶ Relation between the input and the output

408/425

The Web Service Modeling Ontology WSMO

Web Service descriptions (cont'd)

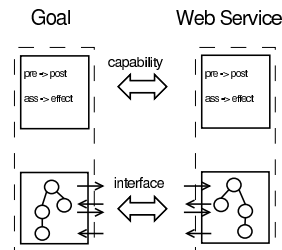
- ▶ Behavioral description, in the form of an **interface**:
 - ▶ Choreography
 - ▶ How to interact with the service
 - ▶ Orchestration
 - ▶ Use of external Web Service to realize the functionality
 - ▶ Both choreography and orchestration are decompositions of the capability

409/425

The Web Service Modeling Ontology WSMO

Goals

- ▶ Functionality requested from the Web Service
- ▶ Description symmetric to Web Service description:
 - ▶ Capability
 - ▶ Interface



411/425

The Web Service Modeling Ontology WSMO

Mediators

- ▶ Connect heterogeneous components
- ▶ Resolve heterogeneity in different levels
 - ▶ Data - differences in data representation
 - ▶ Protocol - differences in interaction styles
 - ▶ Process - differences in business processes

413/425

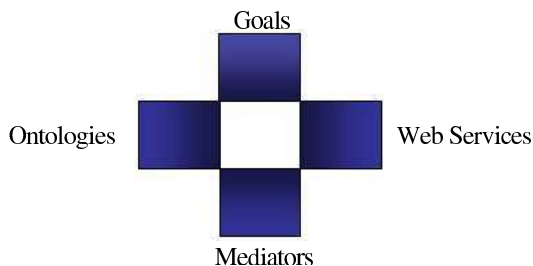
The Web Service Modeling Ontology WSMO

Types of Mediators

- ▶ OO Mediators
 - ▶ Connect ontologies to any other component (including mediators)
 - ▶ Resolve mismatches conflicts between ontologies
- ▶ WW Mediators
 - ▶ Link Web Services to services they depend on
 - ▶ Resolve representation differences through OO Mediators
- ▶ WG Mediators
 - ▶ Link Goals and Web Services
 - ▶ Resolve differences in data, protocol and process between requester and provider
- ▶ GG Mediators
 - ▶ Connect generic and refined Goals

414/425

Review of WSMO



415/425

Idea of WSDL-S

- ▶ OWL-S, WSMO comprehensive approaches for WS description
 - ▶ top-down approach
- ▶ WSDL-S extends existing standards
 - ▶ WSDL known standard
 - ▶ provide minimal extension
- ▶ WSDL-S agnostic about language/model for WS description
 - ▶ allow to use OWL-S, WSMO, ...
 - ▶ OWL, OCL,
- ▶ choice of language/model **outside** of WSDL-S
- ▶ WSDL-S takes as standardization of Semantic Web Services

417/425

WSDL-S Extension of WSDL (I)

- ▶ **Interface**
 - ▶ **modelReference**: reference to semantic model
 - ▶ **category**: placing WS in taxonomy
- ▶ **Operation**
 - ▶ **modelReference**: reference to semantic model
 - ▶ **precondition**
 - ▶ **name**
 - ▶ either **modelReference**: definition of precondition in semantic model, or
 - ▶ **expression**: expression defining the precondition in chosen representation language
 - ▶ **effect**
 - ▶ **name**
 - ▶ either **modelReference**: definition of effect in semantic model, or
 - ▶ **expression**: expression defining the effect in chosen representation language

418/425

WSDL-S Extension of WSDL (II)

- ▶ XML Schema elements
 - ▶ **modelReference** at elements in XML Schema
 - ▶ **schemaMapping**: defines translation of XML to semantic representation language

419/425

WSDL-S: Interface

```
<interface name="PurchaseOrder">
  <!--Category is added as an extensible element of
  an interface-->
  <wssem:category name="Electronics"
    taxonomyURI="http://www.naics.com/"
    taxonomyCode="443112" />
</interface>
```

420/425

WSDL-S: Operation

```
<operation name="processPurchaseOrder"
  pattern="wsdl:in-out"
  wssem:modelReference="Rosetta:RequestPurchaseOrder" >
  <input messageLabel="processPurchaseOrderRequest"
    element="tns:processPurchaseOrderRequest"/>
  <output messageLabel="processPurchaseOrderResponse"
    element="processPurchaseOrderResponse"/>
  <!--Precondition and effect are added as extensible
  elements on an operation-->
  <wssem:precondition name="ExistingAcctPrecond"
    wssem:modelReference="POontology#AccountExists"/>
  <wssem:effect name="ItemReservedEffect"
    wssem:modelReference="POontology#ItemReserved"/>
</operation>
```

421/425

WSDL-S: XML Schema mapping

```
<complexType name="POAddress"
  wssm:schemaMapping="http://.../POAddress.xsl"
  wssm:modelReference="POOntology#Address" >
  <all>
    <element name="recipientInstName" type="string" />
    <element name="streetAddr1" type="string" />
    <element name="streetAdd2" type="string" />
    <element name="city" type="string" />
    <element name="state" type="string" />
    <element name="zipCode" type="string" />
    <element name="country" type="string" />
  </all>
</complexType>
```

422/425

WSDL-S: XML Schema mapping (cont'd)

```
<?xml version='1.0' ?>
<xsl:transform version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<xsl:template match="/">
  <Address rdf:ID="Address1">
    <has_Receiver rdf:datatype="xs:string">
      <xsl:value-of select="POAddress/reipientInstName"/>
    <has_StreetAddress rdf:datatype="xs:string">
      <xsl:value-of select="concat(...)" />
    </has_StreetAddress >
    . . . .
  </has_Receiver>
</Address>
</xsl:template>
```

423/425

Summary

OWL-S

- Service Profile
- Service Model
- Service Grounding

WSMO

- Ontologies
- Web Services
- Goals
- Mediators

WSDL-S

424/425

Further reading

- ▶ OWL-S specification: <http://www.daml.org/services/owl-s/1.1/>
- ▶ WSMO specification: <http://www.wsmo.org/TR/d2/>
- ▶ WSDL-S specification: <http://www.w3.org/Submission/WSDL-S/>
- ▶ D. Roman, U. Keller, H. Lausen, R. Lara, J. de Bruijn, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web service modeling ontology. *Applied Ontology*, 1(1):77-106, 2005.
- ▶ J. de Bruijn, D. Fensel, U. Keller, and R. Lara. Using the web service modelling ontology to enable Semantic eBusiness. *Communications of the ACM*, special issue on the semantic e-business vision, 48(12):43-47, December 2005.

425/425