Semantic Web Services: Approaches and Technologies

John Domingue
Knowledge Media Institute,
The Open University
UK
Contents

• Motivation
• Semantic Web
• Web Services
• Web Services Modelling Ontology
• Other Semantic Web Services Initiatives
  – OWL-S
  – SAWSDL
• IRS-III
Motivation
The Internet has ...
1/2 billion hosts (IP addresses)
1.17 billion users
or 17.8% of the world’s population
The Web has ...

109 million distinct web sites
29.7 billion web pages
~5 pages for every man, woman, and child on the planet
7.2 billion Web searches/month
(3.9 billion by Google) far exceed the world population
161 exabytes \((10^8 \text{ TB})\) of information was created or replicated worldwide in 2006.
That’s more than in the previous 5,000 years.
IDC estimates 6X growth by 2010 to 988 exabytes (a zetabyte) / year
New technical information doubles every 2 years.
... every 72 hours by 2010.
Achieving Web Scale

Semantic Web and Web Services
Semantic Web
SW = A Conceptual Layer over the web
SW is Heterogeneous!
θοην Δομηνυς

Β.Σχ. (Ηνων) (Χομπυτερ Σχιενχε), Υνισερ
σιτψ οφ Ωαρωιχκ.
Pη.Δ. Τη Οπεν Υνισερσιτψ.

Σινχε τη λατε 1990ο μψ ρεσεαρχη ιντερε
στο ηαπε βεεν χεντρεδ ον οντολογιεσ, τη
ε Σεμαντικξ Ωεβ ανδ Σεμαντικξ Ωεβ Σερπιχ
ες. Αν οντολογισ ις α σιαρεδ χοραλ
χεπτυαλιζειον οφ α πιεωποιτ οσερ α δ
ομαε οφ δισκοιρες. Ιν 1997 Ι δεσελοπεδ
ΩεβΩντο, αν εασψ-το-υσε οντολογψ εδιτ
ορ, ωηιχη ωας τη ηιρετ το συπποτ.
First Asian Autumn School on the Semantic Web
First Asian Autumn School on the Semantic Web
Web Services
What’s a Web Service?

- A program programmatically accessible over standard internet protocols
- Loosely coupled, reusable components
- Encapsulate discrete functionality
- Distributed
- Add new level of functionality on top of the current web
Web Services Framework

UDDI Registry

Points to Description

WSDL

Points to Service

Describes Service

Service Consumer

Communicates with XML Messages

SOAP

Web Service
Welcome to Amazon Web Services

Amazon Web Services provides developers with direct access to Amazon's robust technology platform. Build on Amazon's suite of web services to enable and enhance your applications. We innovate for you, so that you can innovate for your customers. Browse developer innovations in our Solutions Catalog to see the possibilities.

What's New?

Announcing "AWS Start-Up Challenge": Win $100,000 Plus an Investment Offer From

Amazon (September 12, 2007)

AWS has just launched the AWS Start-Up Challenge, a contest for entrepreneurs and software developers that will award the winner $50,000 in cash, $50,000 in AWS credits, an investment offer from Amazon.com, and more. What are you waiting for? Submit your idea now.

Amazon Flexible Payments Service (Amazon FPS) - Limited Beta (August 2, 2007)

Amazon Web Services has opened a limited beta of Amazon Flexible Payments Service (Amazon FPS). Amazon FPS is the first payments service designed from the ground up specifically for developers. The set of web services APIs allows the movement of money between any two entities, humans or computers. It is built on top of Amazon's reliable and scalable payment infrastructure. Learn more about this new service.

Announcing the New Version of Alexa Web Search Service (June 6, 2007)

Alexa Web Search service, offered by Amazon Web Services, launched a new version of the Alexa Web Search service that returns up to 1 million search results from a single query, allows complex queries...
Amazon Elastic Compute Cloud (Amazon EC2) - Limited Beta

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Just as Amazon Simple Storage Service (Amazon S3) enables storage in the cloud, Amazon EC2 enables "compute" in the cloud. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

Amazon EC2 Functionality

Amazon EC2 presents a true virtual computing environment, allowing you to use web service interfaces to request machines for use, load them with your custom application environment, manage your network's access permissions, and run your image using as many or few systems as you desire.

To use Amazon EC2, you simply:

- Create an Amazon Machine Image (AMI) containing your applications, libraries, data and associated configuration settings. Or use our pre-configured, templated images to get up and running immediately.
- Upload the AMI into Amazon S3. Amazon EC2 provides tools that make storing the AMI simple. Amazon S3 provides a safe, reliable and fast repository to store your images.
- Use Amazon EC2 web service to launch...
First Asian Autumn School on the Semantic Web
Problems with Web Services Today

• Descriptions are syntactic
• All tasks associated with web services application development have to be carried out by humans:
  – discovery, composition and invocation
• Problems of scalability
First Asian Autumn School on the Semantic Web

SWS Vision

- **Dynamic**
  - Web Services (UDDI, WSDL, SOAP)
  - Web (URI, HTML, HTTP)
  - Semantic Web Services
  - Semantic Web (RDF, OWL)

- **Static**
  - Syntax
  - Semantics
Semantic Web Services (is)

- Semantic Web Technology
  - Machine readable data
  - Ontological basis

Applied to

- Web Services Technology
  - Reusable computational resources

To automate all aspects of application development through reuse
Semantic Web Service Broker

First Asian Autumn School on the Semantic Web
Web Service Modelling Ontology (WSMO)
WSMO Design Principles

• Web Compliance
• Ontology-Based
• Strict Decoupling
• Centrality of Mediation
• Ontological Role Separation
• Description versus Implementation Execution Semantics
• Service versus Web service
WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities
Non-Functional Properties

• Every WSMO element can be described by properties that contain relevant, non-functional aspects.

• Sample information sets are:
  – Dublin Core Metadata Set:
    • For resource management
  – Versioning Information
    • For evolution support
  – Quality of Service Information
    • For availability, stability
  – Other

First Asian Autumn School on the Semantic Web
Non-Functional Properties List

<table>
<thead>
<tr>
<th>Dublin Core Metadata</th>
<th>Quality of Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contributor</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Coverage</td>
<td>NetworkRelatedQoS</td>
</tr>
<tr>
<td>Creator</td>
<td>Performance</td>
</tr>
<tr>
<td>Description</td>
<td>Reliability</td>
</tr>
<tr>
<td>Format</td>
<td>Robustness</td>
</tr>
<tr>
<td>Identifier</td>
<td>Scalability</td>
</tr>
<tr>
<td>Language</td>
<td>Security</td>
</tr>
<tr>
<td>Publisher</td>
<td>Transactional</td>
</tr>
<tr>
<td>Relation</td>
<td>Trust</td>
</tr>
<tr>
<td>Rights</td>
<td></td>
</tr>
<tr>
<td>Source</td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
</tbody>
</table>
WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components

Semantic description of Web Services:
- **Capability** (functional)
- **Interfaces** (usage)

Connectors between components with mediation facilities for handling heterogeneities
Ontology Description and Usage

• Ontologies are used as the ‘data model’ throughout WSMO
  – WSMO is defined in terms of itself
  – All data-types used in Web Service interfaces are ontology concepts
  – Discovery, mediation and composition are based on ontology reasoning

• WSMO Ontology Language WSML
  – Conceptual syntax for describing WSMO elements
  – Logical language for axiomatic expressions (WSML Layering)
WSMO Ontology Design

• Modularization
  – import / re-using ontologies
• De-Coupling
  – heterogeneity handled by OO Mediators
Ontology Specification

- **Non functional properties** (see before)
- **Imported Ontologies**
  - importing existing ontologies where no heterogeneities arise
- **Used mediators**
  - OO Mediators (ontology import with terminology mismatch handling)
- **Ontology Elements:**
  - **Concepts**: set of concepts that belong to the ontology, incl.
  - **Attributes**: set of attributes that belong to a concept
  - **Relations**: define interrelations between several concepts
  - **Functions**: special type of relation (unary range = return value)
  - **Instances**: set of instances that belong to the represented ontology
  - **Axioms**: axiomatic expressions in ontology (logical statement)
WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities
Goals

- Ontological De-coupling of Requester and Provider
- Derived from task / problem solving methods/domain model
- Structure and reuse of requests
  - Search
  - Diagnose
  - Classify
  - Personalise
  - Book a holiday
- Requests may in principle not be satisfiable
- Ontological relationships & mediators used to link goals to Web services
Goal Specification (1/2)

- Non functional properties
- Imported Ontologies
- Used mediators
  - OO Mediators: importing ontologies with heterogeneity resolution
  - GG Mediator:
    - Goal definition by reusing an already existing goal
    - allows definition of Goal Ontologies
Goal Specification (2/2)

- **Requested Capability**
  - describes service functionality expected to resolve the objective
  - defined as capability description from the requester perspective

- **Requested Interface**
  - describes communication behaviour supported by the requester for consuming a Web Service (Choreography)
  - Restrictions / preferences on orchestrations of acceptable Web Services
WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Provide the formally specified terminology of the information used by all other components

Connectors between components with mediation facilities for handling heterogeneities

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*
WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

DC + QoS + Version + financial

Capability

functional description

- Advertising of Web Service
- Support for WS Discovery

client-service interaction interface for consuming WS
- External Visible Behavior
- Communication Structure
- ‘Grounding’

Web Service Implementation
(not of interest in Web Service Description)

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

Choreography --- Service Interfaces --- Orchestration

First Asian Autumn School on the Semantic Web
WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

DC + QoS + Version + financial

Capability

functional description

Web Service Implementation
(not of interest in Web Service Description)

- Advertising of Web Service
- Support for WS Discovery

Choreography --- Service Interfaces --- Orchestration

- client-service interaction interface for consuming WS
  - External Visible Behavior
  - Communication Structure
  - ‘Grounding’

realization of functionality by aggregating other Web Services
  - functional decomposition
  - WS composition
WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

DC + QoS + Version + financial

Capability

functional description

- Advertising of Web Service
- Support for WS Discovery

Web Service Implementation

(not of interest in Web Service Description)

Choreography --- Service Interfaces --- Orchestration

client-service interaction interface for consuming WS
- External Visible Behavior
- Communication Structure
- ‘Grounding’

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

First Asian Autumn School on the Semantic Web
Capability Specification (1/2)

- Non functional properties
- Imported Ontologies
- Used mediators
  - OO Mediator: importing ontologies with mismatch resolution
  - WG Mediator: link to a Goal wherefore service is not usable a priori
Capability Specification (2/2)

- **Pre-conditions**
  - What a web service expects in order to be able to provide its service
  - Define conditions over the input.

- **Assumptions**
  - Conditions on the state of the world that has to hold before the Web service can be executed

- **Post-conditions**
  - Describes the result of the WS in relation to the input, and conditions on it

- **Effects**
  - Conditions on the state of the world that hold after execution of the Web service (i.e. changes in the state of the world)
WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

- Advertising of Web Service
- Support for WS Discovery

Capability

DC + QoS + Version + financial

functional description

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

Web Service Implementation
(not of interest in Web Service Description)

client-service interaction interface for consuming WS
- External Visible Behavior
- Communication Structure
- ‘Grounding’

Choreography --- Service Interfaces --- Orchestration

First Asian Autumn School on the Semantic Web
Choreography & Orchestration

VTA example:

When the service is requested

Date
Time
Flight, Hotel
Error
Confirmation

When the service requests

Date, Time
Hotel
Error
Confirmation

Confirmation

VTA Service

Hotel Service

Flight Service
Choreography Aspects (1/2)

• **Interface for consuming Web Service**
  – External Visible Behavior
    • those aspects of the workflow of a Web Service where Interaction is required
    • described by workflow constructs: sequence, split, loop, parallel
  – Communication Structure
    • messages sent and received
    • their order (communicative behavior for service consumption)
    • choreography related errors (e.g. input wrong, message timeout, etc.)
Choreography Aspects (2/2)

• *Interface for consuming Web Service*
  – Grounding
    • concrete communication technology for interaction
  – Formal Model
    • reasoning on Web Service interfaces (service interoperability)
    • allow mediation support on Web Service interfaces
WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

- Advertising of Web Service
- Support for WS Discovery

Capability

DC + QoS + Version + financial

Web Service Implementation

(not of interest in Web Service Description)

- client-service interaction interface for consuming WS
  - External Visible Behavior
  - Communication Structure
  - ‘Grounding’

Choreography --- Service Interfaces --- Orchestration

realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

First Asian Autumn School on the Semantic Web
Orchestration Aspects

Control Structure for aggregation of other Web Services

- decomposition of service functionality
- all service interaction via choreographies
Orchestration Aspects

• Service interfaces are concerned with service consumption and interaction
• Choreography and Orchestration as sub-concepts of Service Interface
Common requirements for service interface description

- Represent the dynamics of information interchange during service consumption and interaction
- Support ontologies as the underlying data model
- Appropriate communication technology for information interchange
- Sound formal model / semantics of service interface specifications in order to allow operations on them.
Orchestration Definition

process (control + data flow) of goals
Runtime Orchestration

process (control + data flow) between “states”
+ communication behavior of orchestrating Web Service
WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

Semantic description of Web Services:
- **Capability** *(functional)*
- **Interfaces** *(usage)*

Connectors between components with mediation facilities for handling heterogeneities

Provide the formally specified terminology of the information used by all other components.
Mediation (Wiederhold, 94)

- Mediators as components that resolve mismatches
- Declarative Approach
- Semantic description of resources
- ‘Intelligent’ mechanisms that resolve mismatches independent of content
- Mediation cannot be fully automated (integration decision)
Mediation

- For 1$ on programming, $5 - $9 on integration
- Mismatches on structural / semantic / conceptual / level
- Assume (nearly) always necessary
- Description of role
Levels of Mediation within Semantic Web Services

- **Data Level**
  - mediate heterogeneous Data Sources
- **Functional Level**
  - mediate mismatches between Web Service/Goal and Web Service/Goals functionalities
- **Process/Protocol Level**
  - mediate heterogeneous Business Processes/Communication Patterns

- **Layers of Mediators**
  - Specification Layer – WSMO Mediators
  - Implementation Layer – Levels of Mediation
WSMO Mediators Overview

First Asian Autumn School on the Semantic Web
Mediator Structure

First Asian Autumn School on the Semantic Web

WSMO Mediator uses a Mediation Service via
- as a Goal
- directly
- optionally incl. Mediation

Mediation Services

Specification layer

Implementation layer

Source Component

1..n

Source Component

Target Component

Source Component

OO Mediator - Example

Merging 2 ontologies

OO Mediator
Mediation Service

Goal:
“merge s1, s2 and s1.ticket subclassof s2.product”

Train Connection
Ontology (s1)

Purchase
Ontology (s2)

Train Ticket
Purchase Ontology

Discovery

Mediation Services

First Asian Autumn School on the Semantic Web
GG Mediators

• Support specification of Goals by re-using existing Goals
• Allow definition of **Goal Ontologies** (collection of pre-defined Goals)
• Terminology mismatches handled by OO Mediators
GG Mediator Example

Source Goal
“Buy a ticket”

GG Mediator
Mediation Service

Target Goal
“Buy a Train Ticket”

postcondition:
“aTicket memberof trainticket”
WG Mediators

- Link a Web Service to a Goal and resolve occurring mismatches
- Match Web Service and Goals that do not match a priori
- Handle terminology mismatches between Web Services and Goals
  - broader range of Goals solvable by a Web Service
WW Mediators

- Enable interoperability of heterogeneous Web Services
  - support automated collaboration between Web Services
- **OO Mediators** for terminology import with data level mediation
- Protocol Mediation for establishing valid multi-party collaborations
- Process Mediation for making Business Processes interoperable
WW Mediator Example
Data Level Mediation (1/2)

• **Scope**
  – Solving terminological mismatches

• **Related Aspects / Techniques:**
  – Ontology Integration (Mapping, Merging, Alignment)
  – Data Lifting & Lowering
  – Transformation between Languages / Formalisms
Data Level Mediation (2/2)

• Terminology Mismatches Classification
  – Conceptualization Mismatches
    • same domain concepts, but different conceptualization
    • different levels of abstraction
    • different ontological structure
    • => resolution only includes human intervention
  – Explication Mismatches
    • mismatches between:
      – T (Term used), D (definition of concepts), C (real world concept)
    • => automated resolution partially possible
Functional Level Mediation (1/2)

• **Scope**
  – Solving functional mismatches between goals and/or ws

• **Related Aspects/Techniques**
  – Discovery
  – Semantic Matchmaking

• **Matchmaking Mismatches**
Functional Level Mediation (2/2)

= G/WS  = G/WS

Exact Match  PlugIn Match  Subsumption Match  Intersection Match  No Match
Process Level Mediation (1/2)

• Scope
  – Resolves communication mismatches and establish behavior compatibility

• Related Aspects/Techniques
  – Data and control flow composition

• Process Mismatches
  – Signature terminology mismatches (need for data level mediation)
  – Communication/behavior mismatches
Process Level Mediation (2/2)
Other Semantic Web Service Initiatives
OWL-S
OWL-S Ontology

- OWL-S is an OWL ontology to describe Web services
- OWL-S leverages on OWL to
  - Support capability based discovery of Web services
  - Support automatic composition of Web Services
  - Support automatic invocation of Web services
  - OWL-S provides a semantic layer over Web services standards
    - OWL-S relies on WSDL for Web service invocation
    - OWL-s Expands UDDI for Web service discovery
First Asian Autumn School on the Semantic Web

OWL-S Upper Ontology

- Mapping to WSDL
  - communication protocol (RPC, HTTP, ...)
  - marshalling/serialization
  - transformation to and from XSD to OWL

- Capability specification
- General features of the Service
  - Quality of Service
  - Classification in Service taxonomies

- Control flow of the service
  - Black/Grey/Glass Box view
  - Protocol Specification
  - Abstract Messages
WSMO OWL-S Comparison

• Historical
  – OWL-S planning (agents)
  – WSMO knowledge modelling and B2B integration

• Representation
  – OWL-S based on OWL
  – WSMO on WSML family

• WSMO explicit conceptualisation of user context
• WSMO explicit conceptualisation of mediation
• WSMO Interfaces ⇒ process model
  – WSMO provides choreography + orchestration while OWL-S provides only orchestration
  – WSMO service interface description model with ASM-based formal semantics
  – OWL-S formal semantics has been developed in very different frameworks such as Situation Calculus, Petri Nets, Pi-calculus
  – OWL-S Process Model is extended by SWRL / FLOWS
• OWL-S Grounding ⇒ current WSMO Grounding
Semantic Annotations for WSDL (SAWSDL)

W3C Candidate Recommendation
SAWSDL Scope

Annotated using modelReference

Annotated using modelReference and schemaMapping

No SAWSDL annotations defined for these WSDL components

First Asian Autumn School on the Semantic Web

Picture from: http://www.w3.org/TR/wsd120-primer/
First Asian Autumn School on the Semantic Web
IRS-III
Design Principles

• Ontology-Based
• Ontological Role Separation
• Brokering role
• Capability Based Invocation
• Single representation language
• Ease of Use
• Seamless publishing of services
• Inspectable
• Interoperable with SWS Frameworks and Platforms
• Executable semantic descriptions
Features of IRS-III (1/3)

• Based on Soap messaging standard
• Provides Java API for client applications
• Provides built-in brokering and service discovery support
• Provides capability-centred service invocation
Features of IRS-III (2/3)

- Publishing support for variety of platforms
  - Java, Lisp, Web Applications, Java Web Services
- Enables publication of ‘standard code’
  - Provides clever wrappers
  - One-click publishing of web services
Features of IRS-III (3/3)

• Integrated with standard Web Services world
  – Semantic web service to IRS
  – ‘Ordinary’ web service
IRS-III Overall Architecture
First Asian Autumn School on the Semantic Web

IRS-III Server

HTTP Server

SOAP Handler

Invoker - Choreography Interpreter - Mediation Handler - Orchestration Interpreter

OCML Reasoner

SWS Library

IRS-III Server
IRS-III Demo Context
European Travel Scenario

First Asian Autumn School on the Semantic Web
European Travel Demo
IRS-III Demo
Summary (1/2)

• Semantic Web Services
  – Potential to cope with Web scale
  – Applies SW to automate application development through reuse of Web services

• WSMO
  – Ontology describing Web services
  – Goals, Mediators, Web Services
  – Choreography and Orchestration
Summary (2/2)

- **OWL-S**
  - SWS initiative based on OWL
- **SAWSDL**
  - W3C recommendation
  - Embeds semantics into WSDL files
- **IRS-III**
  - SWS broker
  - WSMO compliant
Relevant URLs

- **WSMO**
  - [http://www.wsmo.org/](http://www.wsmo.org/)

- **IRS-III**

- **OWL-S**
  - [http://www.daml.org/services/owl-s/](http://www.daml.org/services/owl-s/)

- **SAWSDL**
Acknowledgements

- Mike Brodie
- Michael Stollberg
- Enrico Motta
- Frank van Harmelen
- WSMO Working Group
- DIP project
Thanks
First Asian Autumn School on the Semantic Web