



Semantic Web Services

John Domingue and David Martin

Acknowledgements



- Mary Rowlatt
- Leticia Gutierrez
- Michael Stollberg
- Liliana Cabral
- Vlad Tanasescu
- Alessio Gugliotta
- WSMO Working Group
- DIP project
- OWL-S Coalition
- Sheila McIlraith
- Terry Payne
- Task Computing project
- Ryusuke Masuoka



Web Services

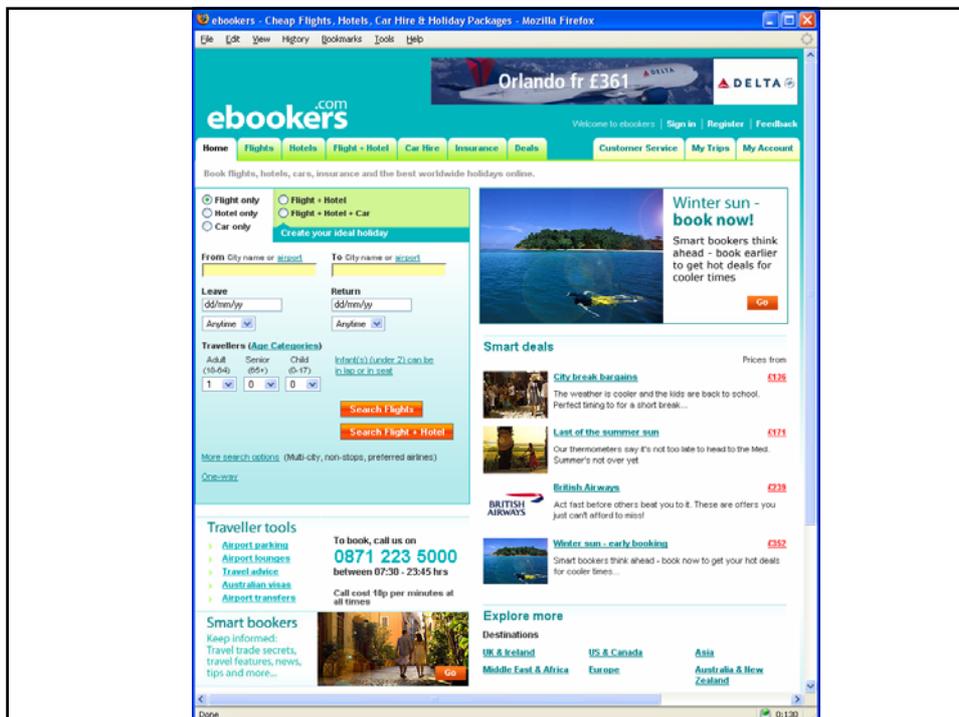
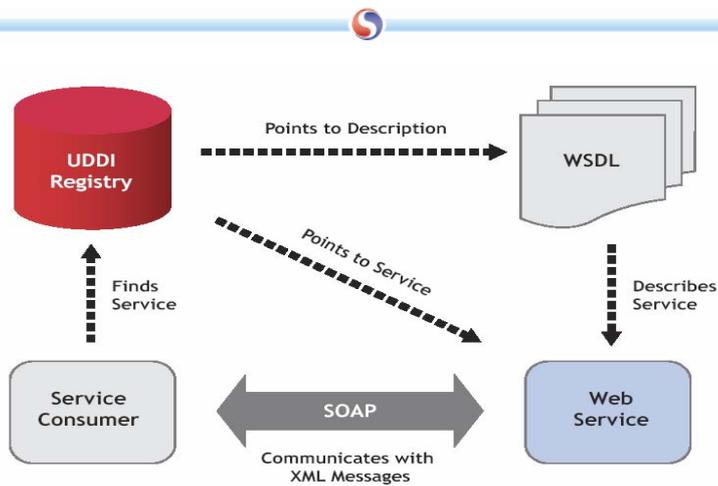
John Domingue

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



Amazon.com: Homepage: Amazon Web Services - Mozilla Firefox

amazon.com | Make Money | See all 41 Product Categories | Your Account | Cart | Your Lists | Help

Program Overview | Marketplace | Associates | Advantage | **Web Services** | Paid Placements | On-demand Publishing

Search Amazon.com

amazon web services™

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!

Learn About Amazon Web Services

- [AWS Home](#)
- [Why Use AWS?](#)
- [What's New in AWS?](#)
- [Upcoming Events](#)
- [Success Stories](#)
- [Solutions Catalog](#)
- [Create an Account](#)
- [Contact Us](#)
- [FAQs](#)

Browse Web Services

- [Amazon E-Commerce Service](#)
- [Amazon Elastic Compute Cloud \(Beta\)](#)
- [Amazon Flexible Payments Service](#)
- [Amazon Mechanical Turk \(Beta\)](#)
- [Amazon Simple Storage Service](#)
- [Amazon Simple Queue Service](#)
- [Alexa Web Services](#)
- [Browse All Web Services](#)

Developer Connection

Done 0:130

Sign-up for AWS

- Create scalable and reliable apps
- Build new solutions and make money
- Join an innovative developer community
- [Sign-up today](#)

ATTENTION: Start-Ups

Are you a hot start-up or VC? Join AWS for the Start-Up Project, a half day seminar in [Silicon Valley](#), [San Francisco](#), [New York](#), and [Boston/Cambridge](#).

AWS Blog



Learn more about other exciting and innovative developments around Amazon Web Services from the eyes and words of Amazon Evangelist, Jeff Barr.

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.com: Amazon EC2, Amazon Elastic Compute Cloud, Virtual Grid Computing: Amazon Web Services - M...

amazon.com | Make Money | See all 41 Product Categories | Your Account | Cart | Your Lists | Help

Program Overview | Marketplace | Associates | Advantage | **Web Services** | Paid Placements | On-demand Publishing

Search Amazon.com

amazon web services™

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 requisition 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 configure

Learn About Amazon Web Services

- [AWS Home](#)
- [Why Use AWS?](#)
- [What's New in AWS?](#)
- [Upcoming Events](#)
- [Success Stories](#)
- [Solutions Catalog](#)
- [Create an account](#)
- [Contact Us](#)
- [FAQs](#)

Browse Web Services

- [Amazon E-Commerce Service](#)
- [Amazon Elastic Compute Cloud \(Beta\)](#)
- [Amazon Flexible Payments Service \(Beta\)](#)
- [Amazon Mechanical Turk \(Beta\)](#)
- [Amazon Simple Storage Service](#)
- [Amazon Simple Queue Service](#)
- [Alexa Web Services](#)
- [Browse All Web Services](#)

Developer Connection

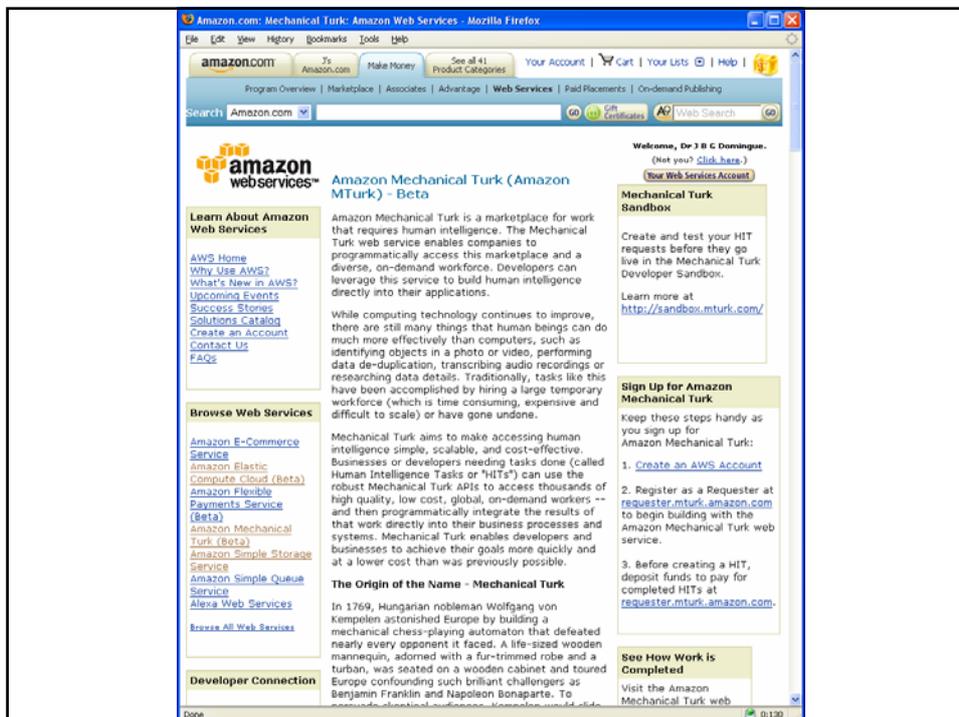
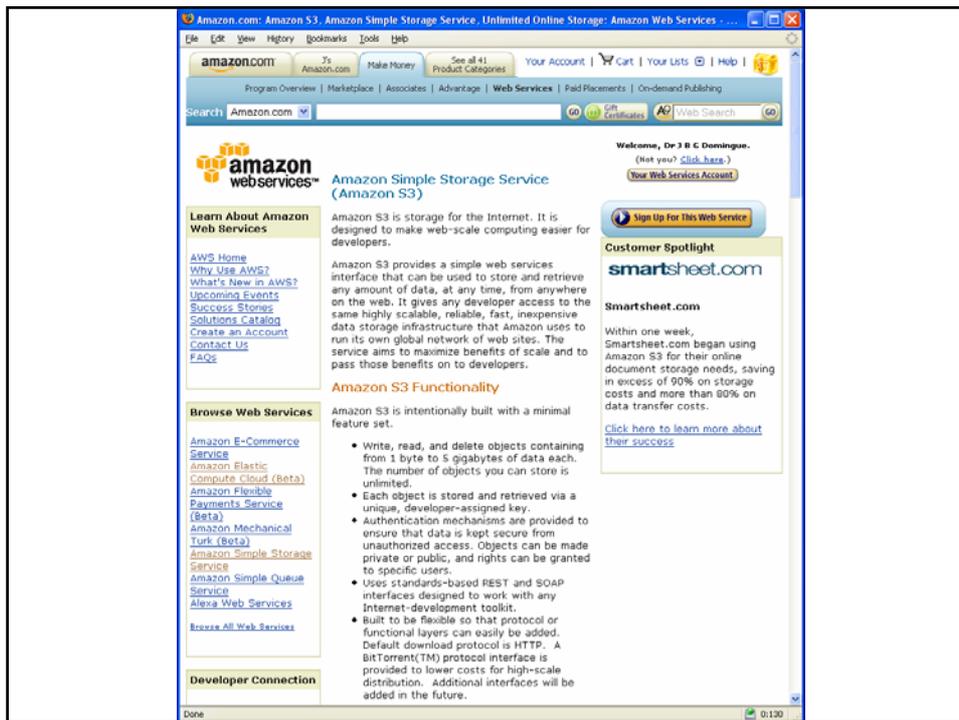
Done 0:130

Sign Up For This Web Service

Customer Spotlight

Jamglue.com

Jamglue, a music mixing site, uses Amazon EC2 to power their backend audio processing. Jamglue built and launched their service using Amazon S3 and Amazon EC2 and has saved considerable time and money. [Click here for the entire story](#)

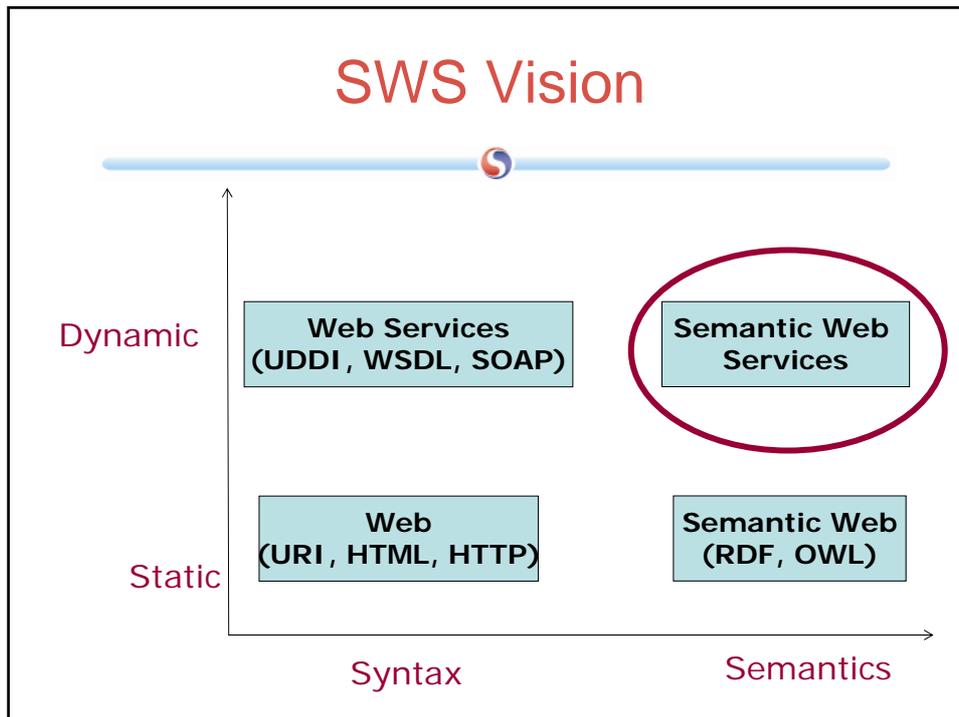


Problems with Web Services Today

- 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

Semantic Web Services

John Domingue



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



OWL-S

David Martin

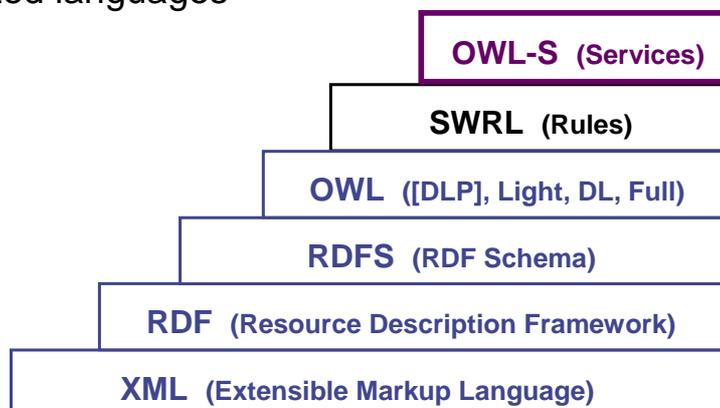
What is OWL-S?

- Ontology Web Language for Services
- An OWL ontology/language for (formally) describing properties and capabilities of Web services
- An approach that draws on many sources
 - Description logic
 - AI planning
 - Workflow
 - Formal process modeling
 - Agents
 - Web services

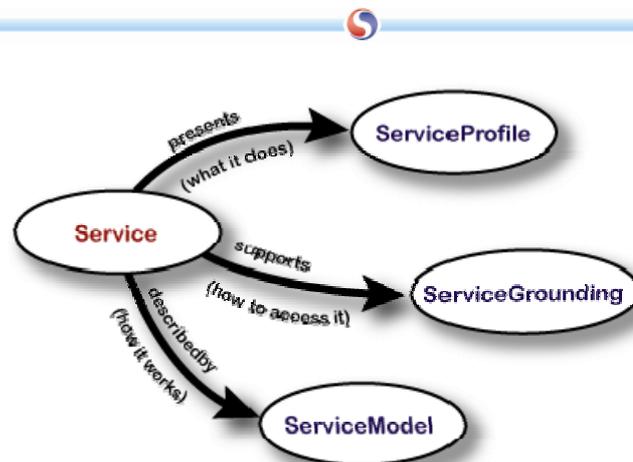
<http://www.daml.org/services/owl-s>

Layered Approach to Language Development

OWL-S: an ontology expressed in OWL and related languages



Upper Ontology of Services



*Ontology images compliments of Terry Payne,
University of Southampton*

Service Profile: "What does it do?"

High-level characterization/summary of a service

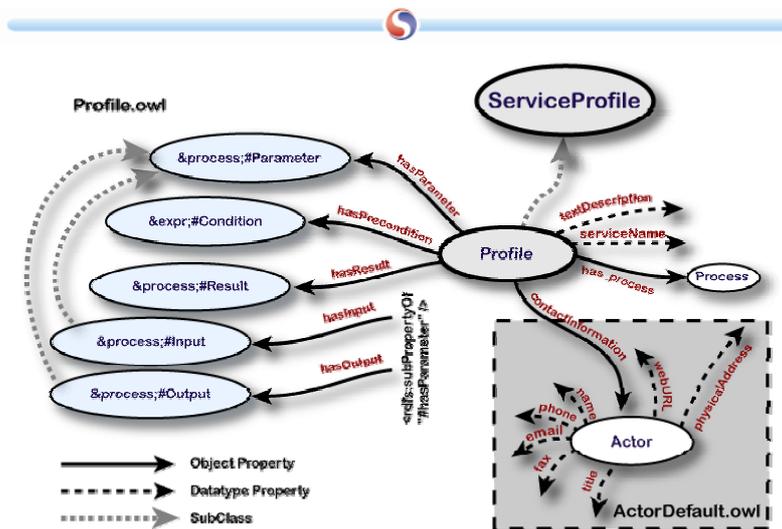
Used for

- Populating service registries
 - A service can have many profiles
- Automated service discovery
- Service selection (matchmaking)

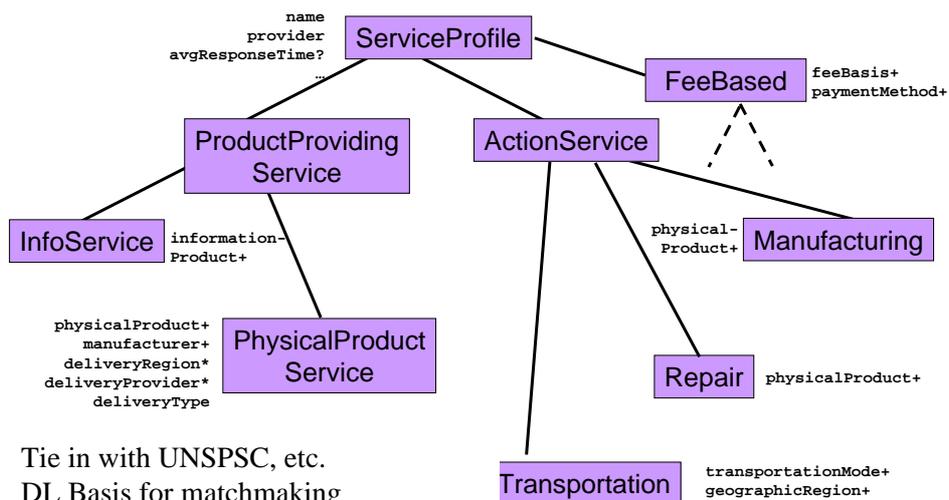
One can derive:

- Service advertisements
- Service requests

Service Profile (partial)



Class Hierarchies of Services



Tie in with UNSPSC, etc.
DL Basis for matchmaking
Multiple profiles; multiple hierarchies

Service Profile: Styles of use



- Class hierarchical yellow pages
 - Implicit capability characterization
 - Arrangement of attributes on class hierarchy
 - Can use multiple inheritance
 - Relies primarily on “non-functional” properties
- Process summaries for planning purposes
 - More explicit
 - Inputs, outputs, preconditions, effects
 - Less reliance on formal hierarchical organization
 - Summarizes process model specs
 - Relies primarily on functional description

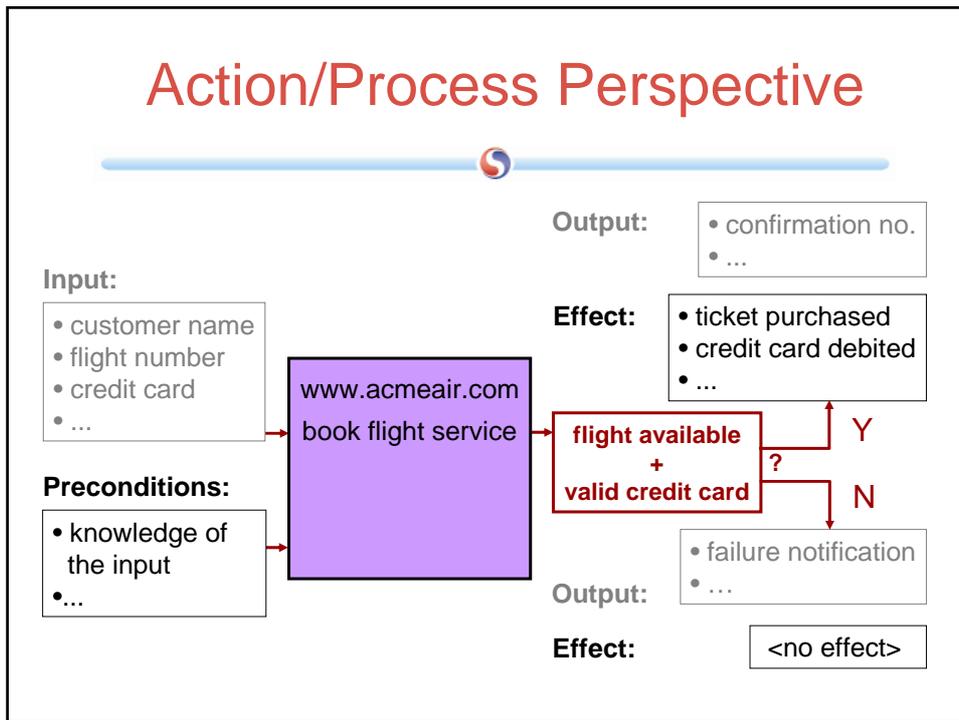
Process Model: “How does it work?”



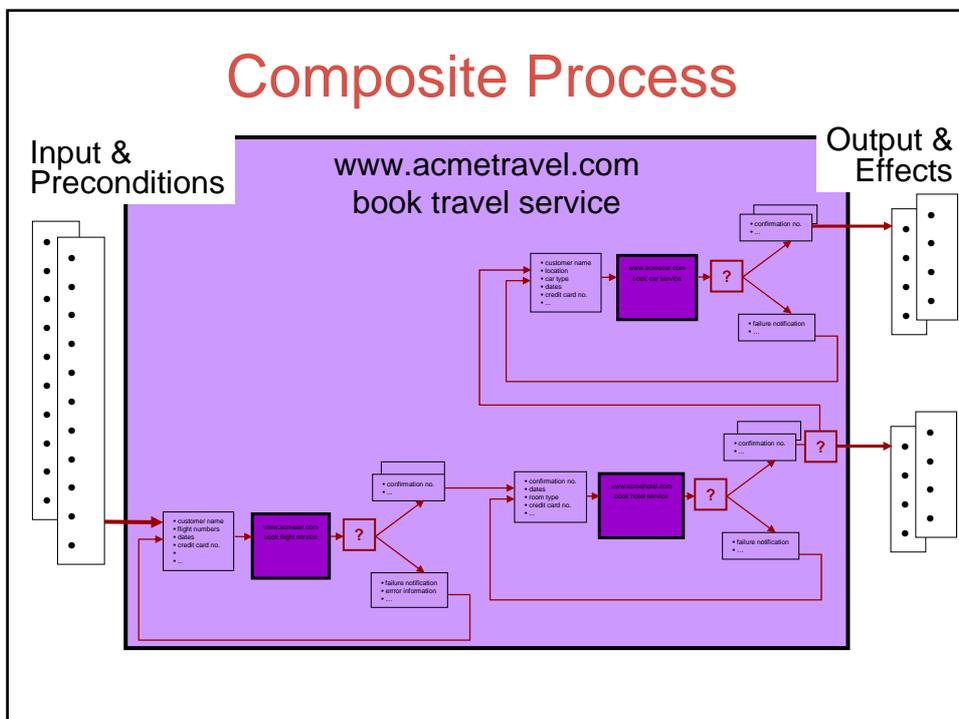
Process

- Potentially interpretable description of service provider’s behavior
- Tells service user how and when to interact (read/write messages)
- Used for:
 - Service **invocation**, **planning/composition**, **interoperation**, **monitoring**
- All processes have
 - Inputs, outputs, preconditions and effects
- Composite processes
 - Control flow
 - Data flow
- OWL standard serializations; presentation syntax

Action/Process Perspective



Composite Process

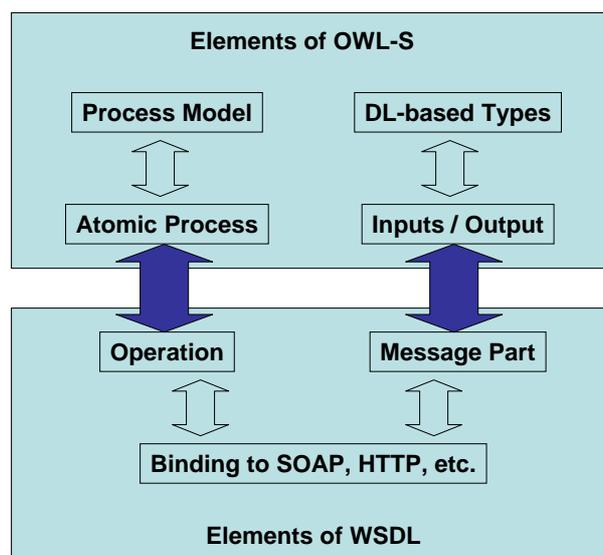


Service Grounding: “How to access it”



- Implementation specific
- Message formatting, transport mechanisms, protocols, serializations of types
- Service Model + Grounding give everything needed for using the service
- Builds upon WSDL

OWL-S / WSDL Grounding (pre-SAWSDL)



OWL-S: Summary & Status



- Describes “what it does”, “how it works”, “how to access it”
 - Profile, Process, Grounding subontologies
- Ties in naturally with WSDL, UDDI
- Additional semantics supports
 - Automation of various Web service tasks
 - Varied applications (later slides)
- W3C member submission
 - <http://www.w3.org/Submission/2004/07/>
 - Corresponds to 1.1 release on daml.org
- 1.2 release completed
- Publications, tools, examples
 - See <http://www.daml.org/services/owl-s/>
 - See <http://www.semwebcentral.org>

Applications Using OWL-S



- Many examples, including
 - **Task Computing**
 - Software Interoperability
 - e-Science
 - Geospatial Data / Query Integration
 - Autonomous Vehicles

Application: Task Computing

Technology to enable easy orchestration
of devices and e-services,
and support users in executing complex tasks

Fujitsu Laboratories of America, Inc.
MINDLab of the University of Maryland

<http://www.taskcomputing.org>

*Thanks to
Ryusuke Masuoka
for use of this material*

Goals of Task Computing

- Minimize/facilitate user interaction
 - User interface: mouse clicks and voice
- Focus on **What** (task) instead of **How** (means)
- No preprogramming of devices for tasks

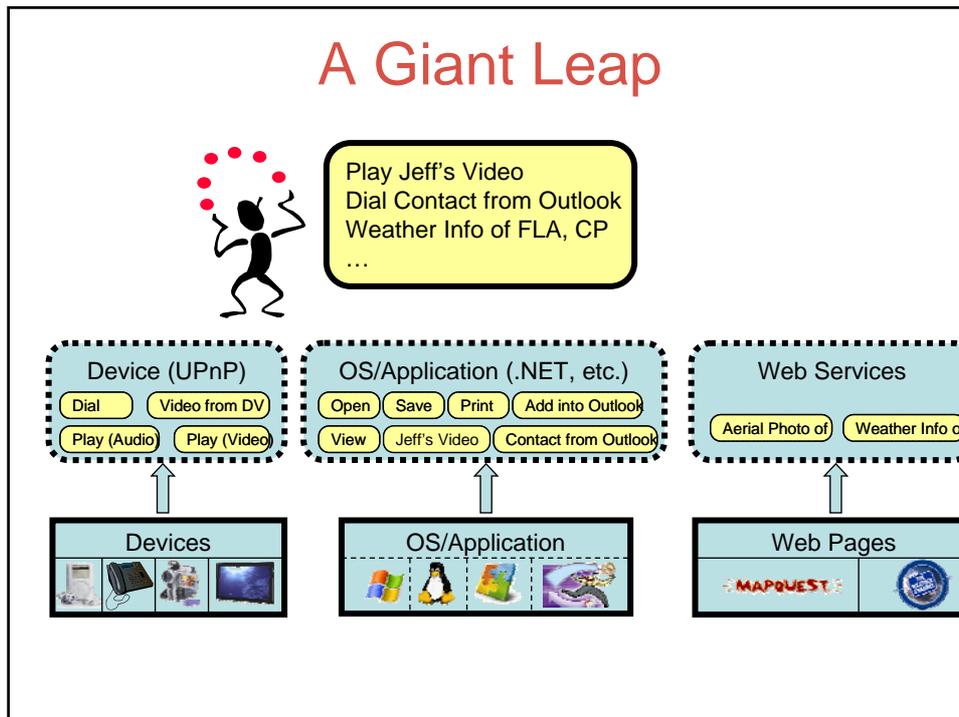
User wants to do "Tasks"



Filling the gap

"Services" offered means
Web services, UPnP, etc.

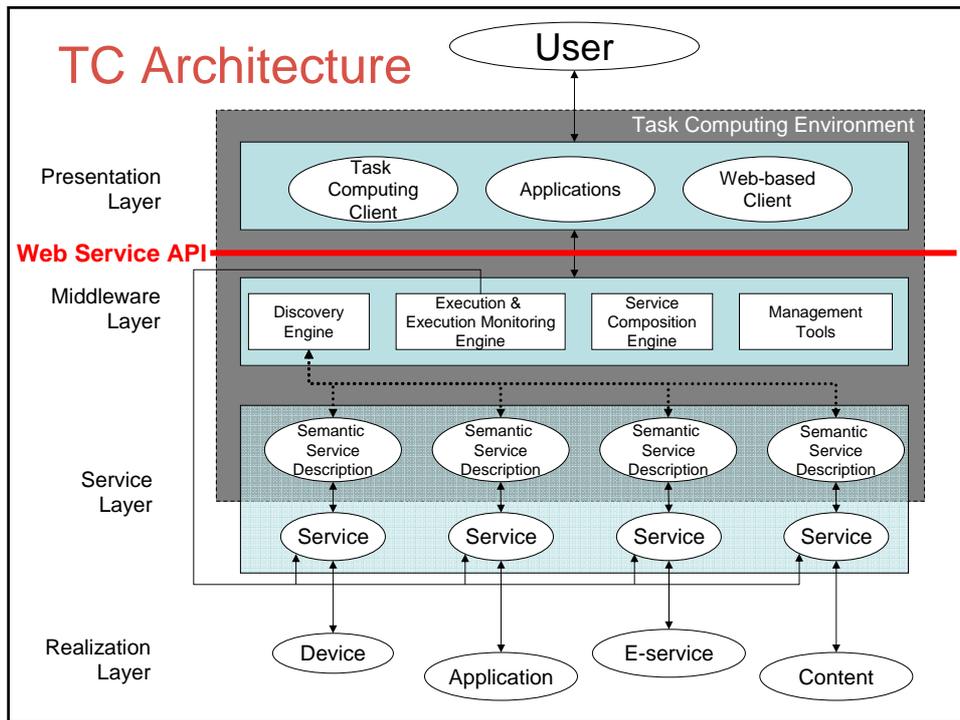
A Giant Leap



Fujitsu's Task Computing



- Approach:
 - Integration of Semantic Web and Web Service computing
 - Abstraction of functionality as services: Describe functionality of device or services in OWL-S
 - Use of UPnP for Semantic Service Discovery Mechanism (SSDM) and for service invocation
- More application examples:
 - Display presentation file from mobile computer on the projector in a room you visit for the first time, without connecting a VGA cable
 - Display pictures from a mobile phone on a TV in any room and print it on an available foto printer, without moving memory cards around
- Precondition: Devices are network-ready and equipped with lightweight web server





Web Service Modelling Ontology (WSMO)

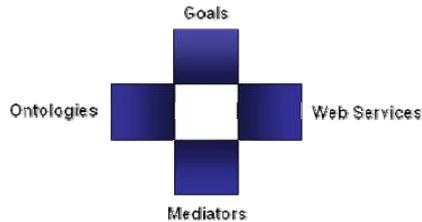
John Domingue

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*)

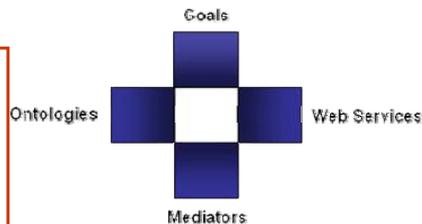
Connectors between components with mediation facilities for handling heterogeneities

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

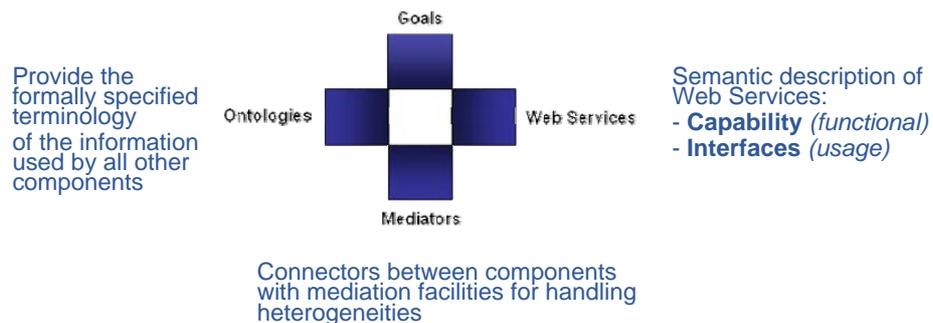


Semantic description of Web Services:
- **Capability** (*functional*)
- **Interfaces** (*usage*)

Connectors between components with mediation facilities for handling heterogeneities

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

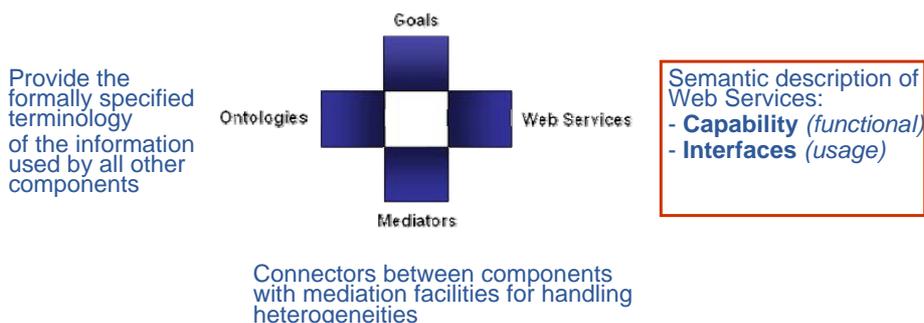


Goals

- 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

WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services



WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

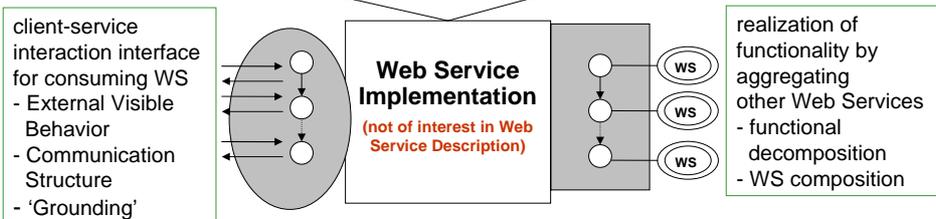
Non-functional Properties

DC + QoS + Version + financial

- Advertising of Web Service
- Support for WS Discovery

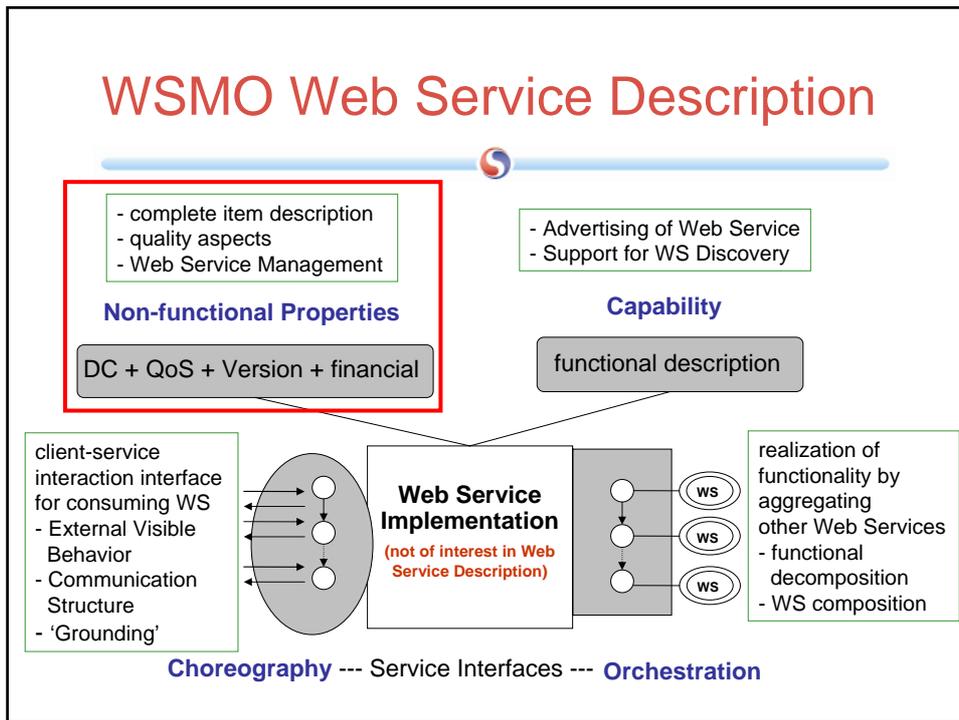
Capability

functional description

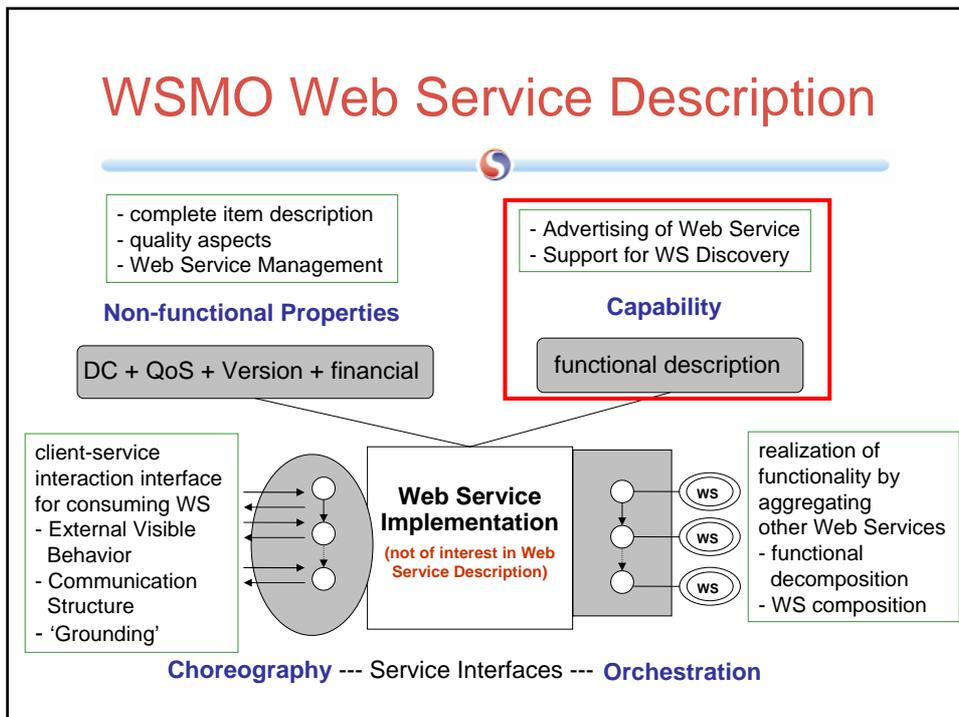


Choreography --- Service Interfaces --- Orchestration

WSMO Web Service Description



WSMO Web Service Description



WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

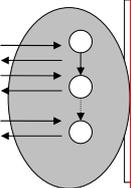
DC + QoS + Version + financial

- Advertising of Web Service
- Support for WS Discovery

Capability

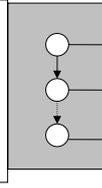
functional description

- 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

WSMO Web Service Description

- complete item description
- quality aspects
- Web Service Management

Non-functional Properties

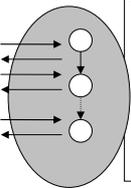
DC + QoS + Version + financial

- Advertising of Web Service
- Support for WS Discovery

Capability

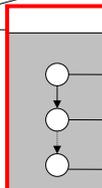
functional description

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



Web Service Implementation

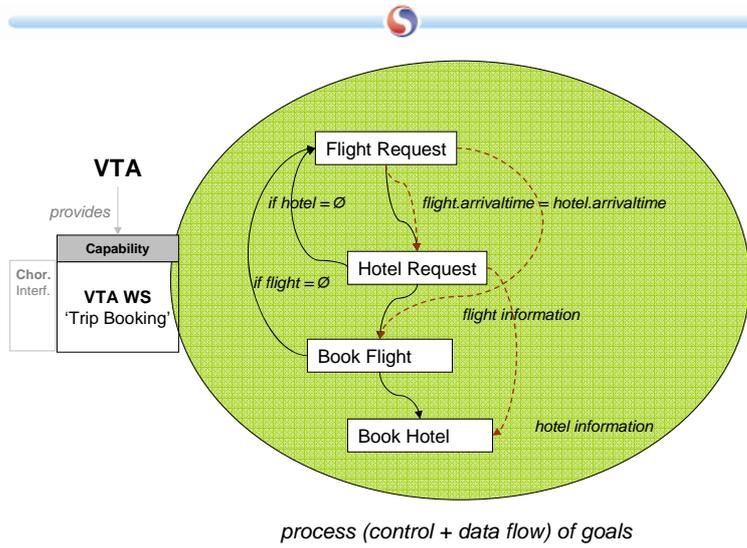
(not of interest in Web Service Description)



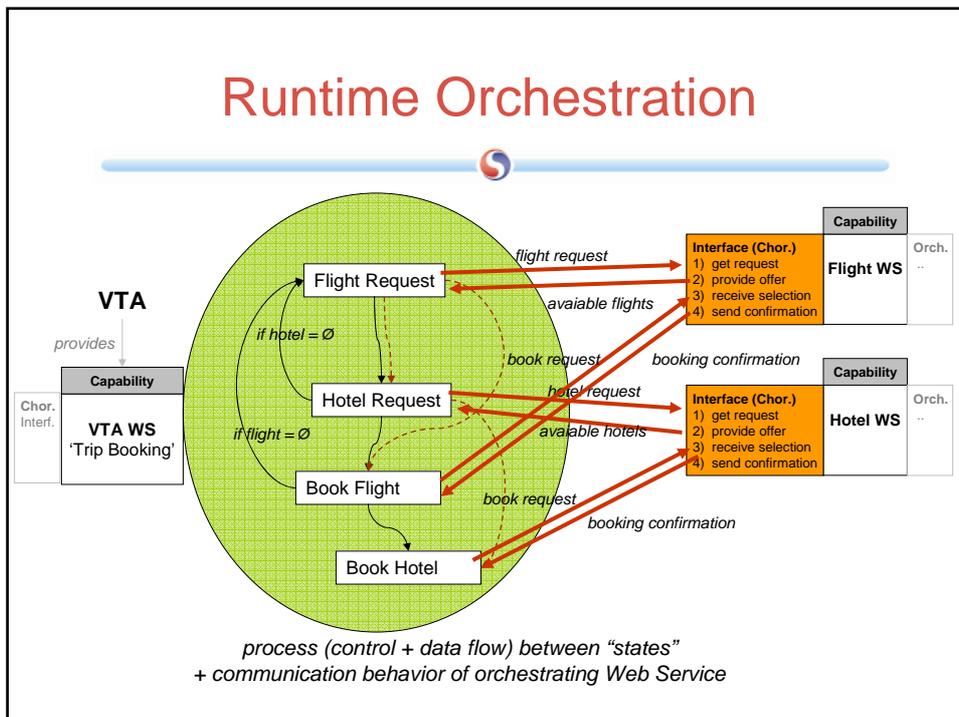
- realization of functionality by aggregating other Web Services
- functional decomposition
- WS composition

Choreography --- Service Interfaces --- Orchestration

Orchestration Definition



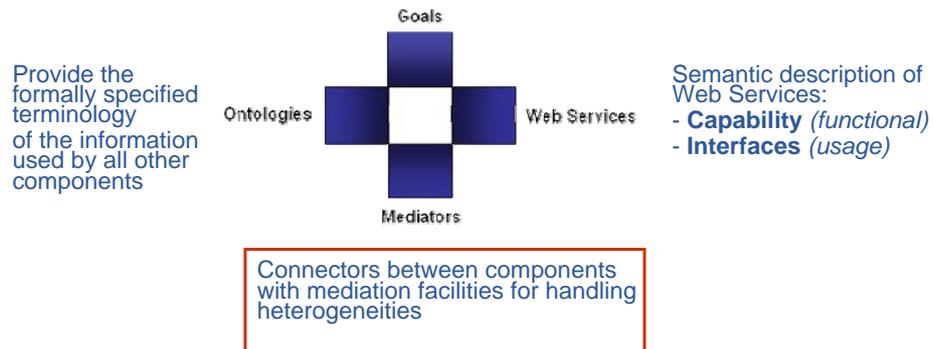
Runtime Orchestration



WSMO Top Level Notions



Objectives that a client wants to achieve by using Web Services

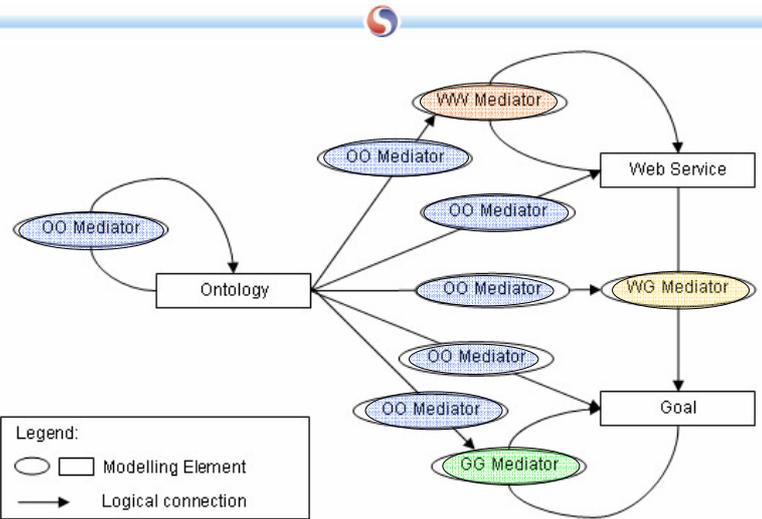


Mediation

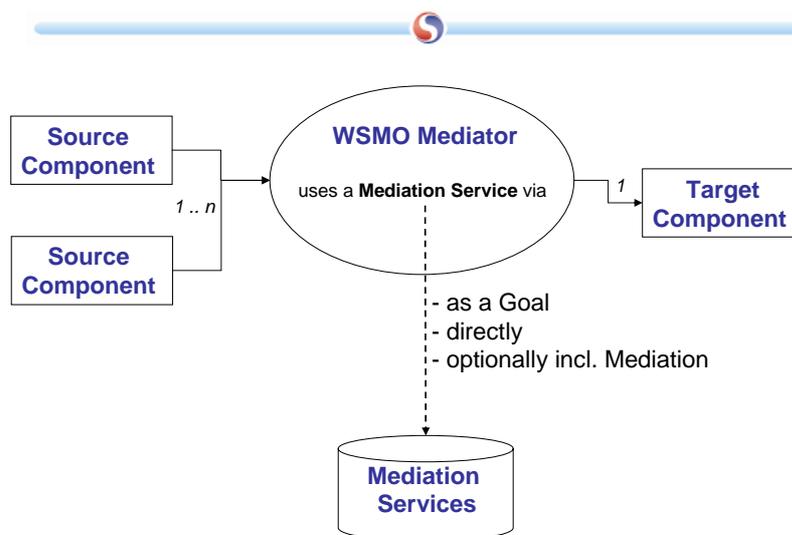


- For 1\$ on programming, \$5 - \$9 on integration © IBM, Nelson Mattos
- Mismatches on structural / semantic / conceptual / level
- Assume (nearly) always necessary
- Description of role

WSMO Mediators Overview



Mediator Structure



WSMO based Application using IRS-III

John Domingue



Data, Information and Process Integration
with Semantic Web Services



BBC CATEGORIES TV RADIO COMMUNICATE WHERE I LIVE INDEX SEARCH

BBC NEWS

You are in: UK

News Front Friday, 31 January, 2003, 10:27 GMT

Page World

UK

England N Ireland Scotland Wales Politics Business Entertainment Science/Nature Technology Health Education

----- Talking Point -----

Country Profiles In Depth

----- Adam Harley, who pulled on to

'My 20-hour battle through the snow'



Motorists have been stuck in lengthy jams

While motorists across Britain have been struggling in to work along icy roads, few have suffered as much as those stuck on the M11 in Cambridgeshire.

WATCH/LISTEN REAL MEDIA

ON THIS STORY

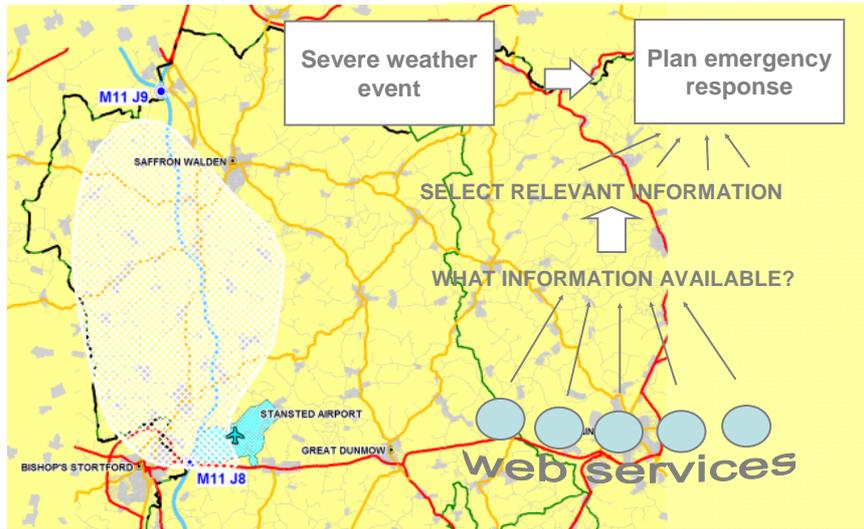
- The BBC's Samantha Simmonds "Hundreds of passengers are still stranded at Heathrow and Gatwick airports"
- Jonathan Smith, E. Midlands Electricity "We've 30,000 customers without electricity"
- Rebecca Rees, AA "People have spent the whole night in their cars"

TALKING POINT

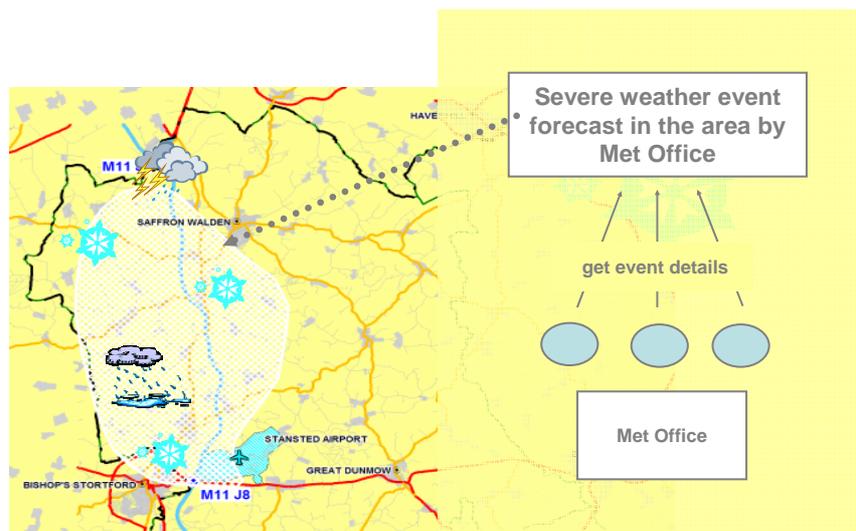
Snow storm

Your 

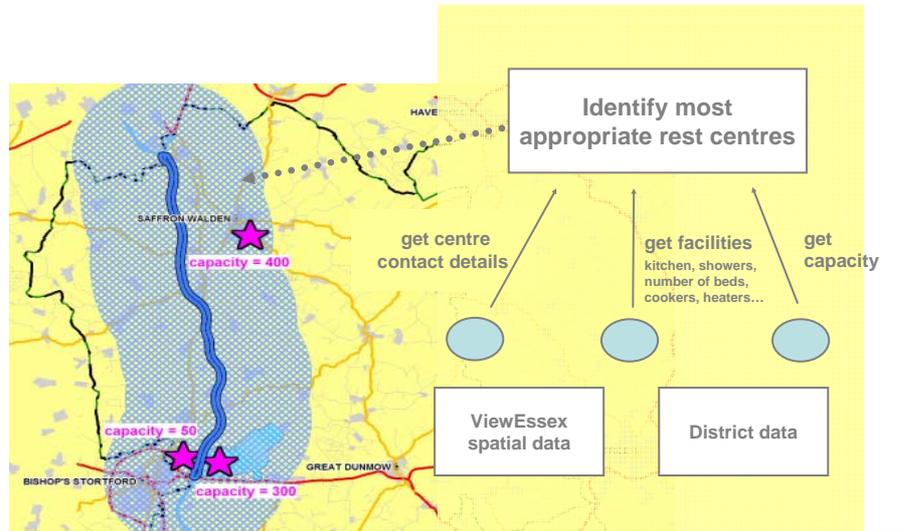
Emergency planning scenario



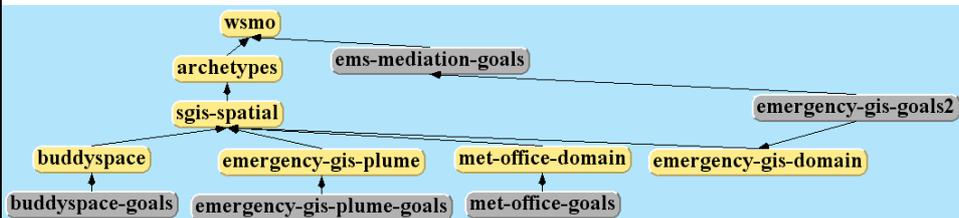
Emergency planning scenario



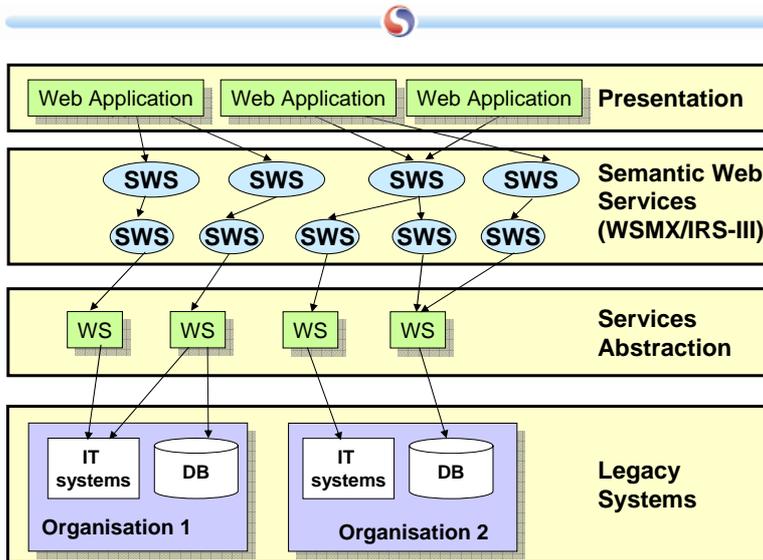
Emergency planning scenario



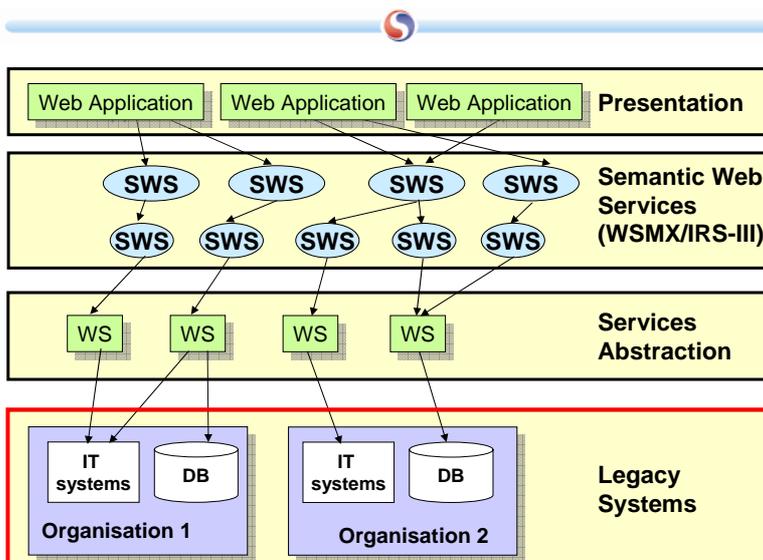
eMerges Ontologies



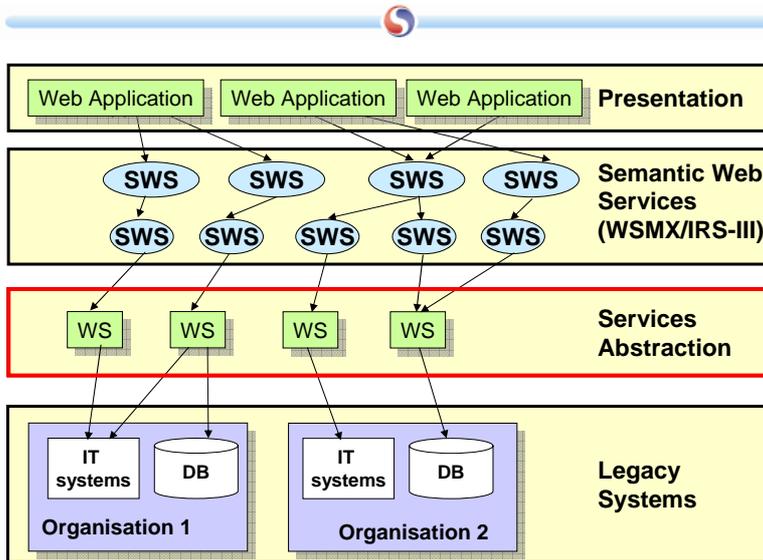
Generic Application Structure



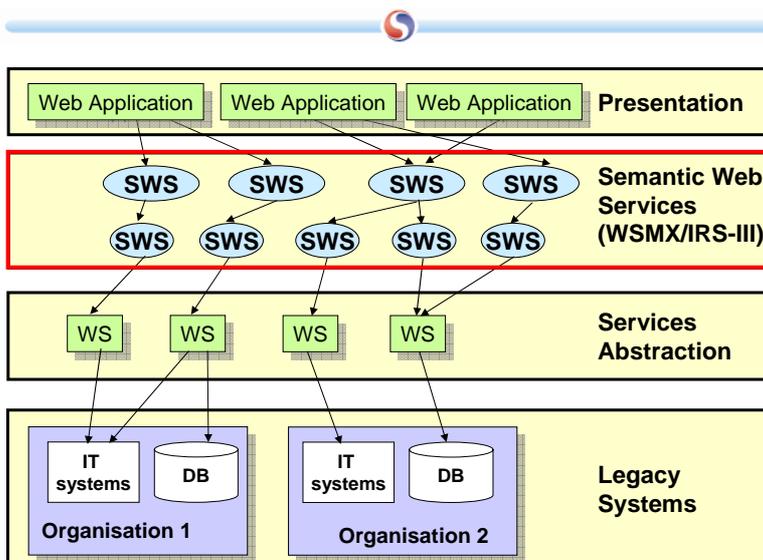
Generic Application Structure



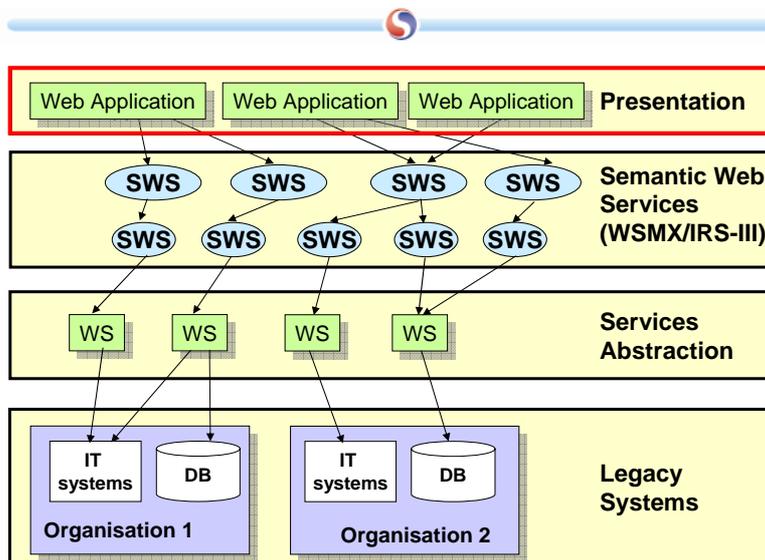
Generic Application Structure



Generic Application Structure

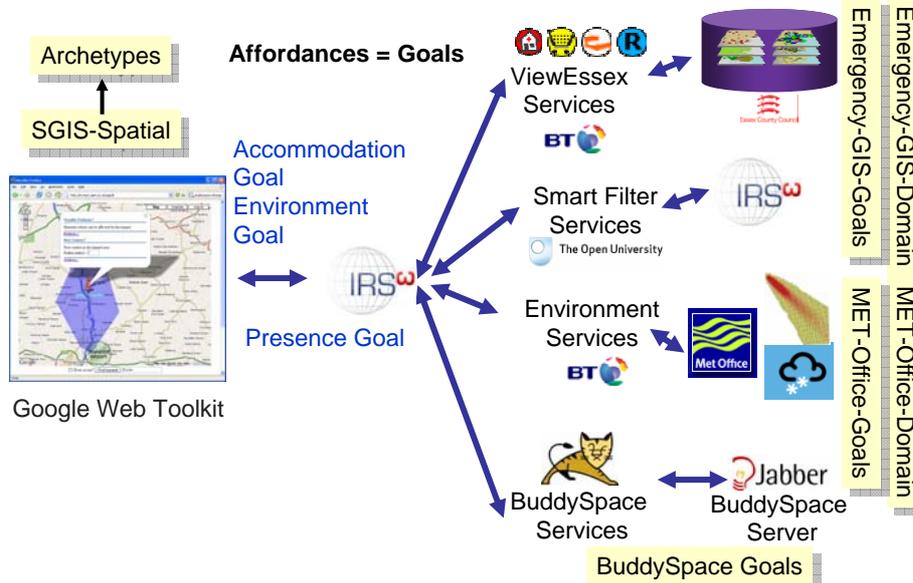


Generic Application Structure



Video of Emergency Planning (GIS) Prototype V1

EMerges Prototype Architecture



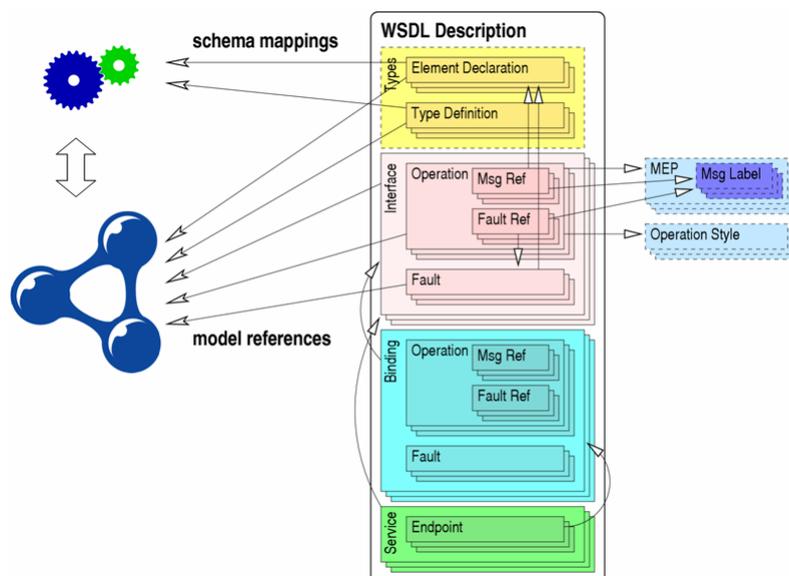
SAWSDL

David Martin

SAWSDL

- **S**emantic **A**nnotations for **WSDL** and XML Schema
- W3C Recommendation, August, 2007
- Largely based on WSDL-S
 - Some SAWSDL ideas also appeared earlier in OWL-S
- A simple, incremental approach
 - Builds naturally on the WSDL-centric view of Web services

Technical Overview of SAWSDL (1)



Technical Overview of SAWSDL (2)

- 3 extensibility elements
 - *modelReference*
 - *liftingSchemaMapping*
 - *loweringSchemaMapping*
- Can be used in both WSDL and XML Schema documents
- Values are lists of URIs
- No Preconditions and Effects

```
<wsdl:description>
  <wsdl:types>
    <xs:schema elementFormDefault="qualified">
      <xs:element name="OrderRequest"
        sawsdl:modelReference="..."
        sawsdl:liftingSchemaMapping="..."
        sawsdl:loweringSchemaMapping="...">
        ...
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:interface name="Order"
    sawsdl:modelReference="...">
    <wsdl:operation name="order"
      pattern="..."
      sawsdl:modelReference="...">
      <wsdl:input element="OrderRequest" />
      <wsdl:output element="OrderResponse" />
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>
```

modelReference

- May be used with every element within WSDL
- “However, SAWSDL defines its meaning only for
 - wsdl:interface
 - wsdl:operation
 - wsdl:fault
 - xs:element
 - xs:complexType
 - xs:simpleType
 - xs:attribute.”

```
<wsdl:description>
  <wsdl:types>
    <xs:schema elementFormDefault="qualified">
      <xs:element name="OrderRequest"
        sawsdl:modelReference=
          "http://ontology/po#OrderRequest">
        </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:interface name="Order"
    sawsdl:modelReference=
      "http://.../products/electronics">
    <wsdl:operation name="order"
      pattern="..."
      sawsdl:modelReference=
        "http://ontology/po#RequestPurchaseOrder">
      <wsdl:input element="OrderRequest" />
      <wsdl:output element="OrderResponse" />
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>
```

Schema Mapping Attributes

- *liftingSchemaMapping*
 - lift data from XML to a semantic model
- *loweringSchemaMapping*
 - lower data from a semantic model to XML
- Can map to XSLT script

```
<wsdl:description>
  <wsdl:types>
    <xs:schema elementFormDefault="qualified">
      <xs:element name="OrderRequest"
        sawsdl:liftingSchemaMapping=
          "http://.../mapping/Response2Ont.xslt"
        sawsdl:loweringSchemaMapping=
          "http://.../mapping/Ont2Request.xml">
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:interface name="Order"
    sawsdl:modelReference="...">
    <wsdl:operation name="order"
      pattern="..."
      sawsdl:modelReference="...">
      <wsdl:input element="OrderRequest" />
      <wsdl:output element="OrderResponse" />
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>
```

SAWSDL References

- The Standard
 - <http://www.w3.org/TR/sawSDL/>
- Implementation Report
 - <http://www.w3.org/2002/ws/sawSDL/CR/>
- Initial Specs for Use with OWL-S
 - “Bringing Semantic Annotations to Web Services: OWL-S from the SAWSDL Perspective”
 - ISWC 2007
 - “Grounding OWL-S in SAWSDL”
 - ICSOC 2007
- Initial Specs for Use with WSMO
 - WSMO Grounding
 - <http://www.wsmo.org/TR/d24/d24.2/v0.1/20070427/>
 - WSMO Lite
 - http://wsmo.org/TR/d11/v0.2/20070622/d11v02_20070622.pdf

Conclusion



- The service paradigm is becoming an important and *integral* part of the Web (including *intranets*)
- SWS aims to provide an expressive, comprehensive framework for handling activities on the Web
 - Enabling greater automation of *discovery, selection, invocation, composition, monitoring*, and other service management tasks
 - Should enable use of agents on the Web
 - Simplicity and widespread adoption of WS building blocks are enablers
- Many tools & applications exist today; mostly prototype
- Many challenges remain
 - SWS is a vigorous research area
- Strong interest and many paths to adoption also exist
 - E.g., the standards path
- Stay tuned – it will be interesting to see how far (and how fast) the service/process/agent-oriented Web will evolve!

Relevant URLs



- DIP
 - <http://dip.semanticweb.org/>
- IRS-III
 - <http://kmi.open.ac.uk/projects/irs/>
- OWL-S
 - See earlier slide
- SAWSDL
 - See earlier slide
- WSMO Working Group
 - <http://www.wsmo.org/>