The Web Service Modeling Ontology (WSMO)

© Copyright 2010. Dieter Fensel and Ioan Toma

Outline

• Motivation
  • Technical solution
    – Overview and Principles
    – Annotations
    – Ontologies
    – Web Services
    – Goals
    – Mediators
    – WSML
    – WSMX
  • Illustration by a larger example
  • Extensions
  • Summary
  • References

Where are we?

<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2</td>
<td>Web Science</td>
</tr>
<tr>
<td>3</td>
<td>Service Science</td>
</tr>
<tr>
<td>4</td>
<td>Web services</td>
</tr>
<tr>
<td>5</td>
<td>Web2.0 services</td>
</tr>
<tr>
<td>6</td>
<td>Semantic Web</td>
</tr>
<tr>
<td>7</td>
<td>Web Service Modeling Ontology (WSMO)</td>
</tr>
<tr>
<td>8</td>
<td>Web Service Modeling Language (WSML)</td>
</tr>
<tr>
<td>9</td>
<td>Web Service Execution Environment (WSMX)</td>
</tr>
<tr>
<td>10</td>
<td>OWL-S and other</td>
</tr>
<tr>
<td>11</td>
<td>Light-weight Annotations</td>
</tr>
<tr>
<td>12</td>
<td>Applications</td>
</tr>
<tr>
<td>13</td>
<td>Mobile Services</td>
</tr>
</tbody>
</table>

MOTIVATION
Motivation for SWS (1)

- Current technologies allow usage of Web Services
- But:
  - Only syntactical information descriptions
  - Syntactic support for discovery, composition and execution
  - Web Service usability, usage, and integration needs to be inspected manually
  - No semantically marked up content / services
  - No support for the Semantic Web

=> Initial Web Service Technology Stack failed to realize the SOA Vision

Problem: Lack of technologies to cope with the scale envisioned for WS
Solution: Techniques for automated support for service related tasks

Motivation for SWS (2)

- Mechanized support is needed for
  - Annotating/designing services and the data they use
  - Finding and comparing service providers
  - Negotiating and contracting services
  - Composing, enacting, and monitoring services
  - Dealing with numerous and heterogeneous data formats, protocols and processes, i.e. mediation

=> Conceptual Models, Formal Languages, Execution Environments

- Existing approaches to SWS (OWL-S, SWSF, WSDL-S) do not provide a unifying solution for SWS
  => WSMO Approach

It's all about automation!

Dynamic

Web Services
UDDI, WSDL, SOAP

Semantic Web
Services

Static

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL, etc.
TECHNICAL SOLUTION

OVERVIEW AND PRINCIPLES

Web Service Modeling Ontology (WSMO) - Approach Overview

Conceptual Model for SWS

Ontology & Rule Language for the Semantic Web with built-in support for WSMO

Semantic Execution Environments and independent broker services

WSMO – Design Principles

Web Compliance

Ontology-Based

Strict Decoupling

Ontological Role Separation

Centrality of Mediation

Execution Semantics

Description versus Implementation
WSMO – Design Principles (1)

- Web Compliance
  - WSMO inherits the concept of URI (Universal Resource Identifier) for unique identification of resources as the essential design principle of the World Wide Web
  - WSMO adopts the concept of Namespaces for denoting consistent information spaces, supports XML and other W3C Web technology recommendations, as well as the decentralization of resources.

WSMO – Design Principles (2)

- Ontology-Based
  - Ontologies are used as the data model throughout WSMO
  - All resource descriptions as well as all data interchanged during service usage are based on ontologies
  - WSMO supports the ontology languages defined for the Semantic Web

WSMO – Design Principles (3)

- Strict Decoupling
  - WSMO resources are defined in isolation
  - Each resource is specified independently without regard to possible usage or interactions with other resources.

WSMO – Design Principles (4)

- Centrality of Mediation
  - Complementary design principle to strict decoupling
  - Mediation addresses the handling of heterogeneities that naturally arise in open environments
  - Heterogeneity can occur in terms of data, underlying ontology, protocol or process.
  - Mediation a first class component of the WSMO framework
• Ontological Role Separation
  – Users, or more generally clients, exist in specific contexts which will not be the same as for available Web services
  – The underlying epistemology of WSMO differentiates between the desires of users or clients and available services

• Description versus Implementation
  – WSMO differentiates between the descriptions of Semantic Web services elements (description) and executable technologies (implementation)
  – WSMO aims at providing an appropriate ontological description model, and to be compliant with existing and emerging technologies

• Execution Semantics
  – In order to verify the WSMO specification, the formal execution semantics of reference implementations like WSMX as well as other WSMO-enabled systems provide the technical realization of WSMO

• Service versus Web service
  – A Web service is a computational entity which is able (by invocation) to achieve a users goal.
  – A service in contrast is the actual value provided by this invocation
  – WSMO provides means to describe Web services that provide access (searching, buying, etc.) to services; WSMO is designed as a means to describe the former and not to replace the functionality of the latter
The Web Service Modeling Ontology (WSMO) is derived from and based on the Web Service Modeling Framework (WSMF).

WSMF provides the appropriate conceptual model for developing and describing web services and their composition.

WSMF is based on the following principle:
- Strong de-coupling of the various components that realize an e-commerce application
- Strong mediation service enabling anybody to speak with everybody in a scalable manner.

WSMF consists of four main different elements:
- *ontologies* that provide the terminology used by other elements
- *goal repositories* that define the problems that should be solved by web services
- *web services* descriptions that define various aspects of a web service
- *mediators* which bypass interoperability problems.

Top-level elements defined by WSMO:
- Objectives that a client may have when consulting a Web Service
- Semantic description of Web Services
  - Non-functional properties
  - Interfaces (usage)
- Connectors between components with mediation facilities for handling heterogeneities

(Thou shalt provide the formally specified terminology of the information used by all other components)
Dublin Core

- The Dublin Core metadata element set is a standard for cross-domain information resource description.
- An architecture and abstract model that can be used to develop application profiles to describe resources in a machine processable manner.
- 15 elements or attribute-value pairs – “simple DC”
- 55 elements or attribute-value pairs – “qualified DC”
- Elements may be displayed in any order
- Extensible
- International in scope

Annotations

- Annotations are used in the definition of WSMO elements (reuse of Dublin Core metadata elements)

Examples:
- The creator of the institute identified by [http://www.sti-innsbruck.at/](http://www.sti-innsbruck.at/) is Dieter Fensel
- The date on which the website [http://www.sti-innsbruck.at/](http://www.sti-innsbruck.at/) was created is 01.01.2006

WSMO – Ontologies

- In WSMO, Ontologies are the key to linking conceptual real-world semantics defined and agreed upon by communities of users

Examples:
- The Location Ontology ([http://www.wsmo.org/ontologies/location](http://www.wsmo.org/ontologies/location)) contains the concepts “Country” and “Address”
- The Location Ontology ([http://www.wsmo.org/ontologies/location](http://www.wsmo.org/ontologies/location)) contains the “Austria” and “Germany” instances
Ontology Specification

- Non functional properties: author, date, ID, etc.
- Imported Ontologies: importing existing ontologies where no heterogeneities arise
- Used mediators: OO Mediators (ontology import with terminology mismatch handling)

Ontology Elements:

- Concepts: set of entities that exist in the world / domain
- 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 Ontologies – Concepts

- **Concepts** constitute the basic elements of the agreed terminology for some problem domain
  - From a high-level perspective, a concept – described by a concept definition – provides attributes with names and types.
  - A concept can be a subconcept of several (possibly none) direct superconcepts as specified by the isA-relation.

  **Class**

  - concept: sub-Class wsmoElement
    - hasSuperConcept: concept
    - hasAttribute: attribute
    - hasDefinition: logicalExpression
    - multiplicity = single-valued

  **Example:**

  - The concept “Border” defines the border between two countries. It is a subclass of a more general concept “GeographicLocation”. It has two attributes countryA and countryB whose ranges are instances of concept “Country”

Logical Expressions for the Definition of Concepts

- The definition of a concept is a logical expression which can be used to define formally the semantics of the concept.
  - The logical expression defines (or restricts, respectively) the extension (i.e. the set of instances) of the concept. If C is the identifier denoting the concept then the logical expression takes one of the following forms:

    ```
    forAll ?x ( ?x memberOf C implies l-expr(?x) )
    forAll ?x ( ?x memberOf C impliedBy l-expr(?x) )
    forAll ?x ( ?x memberOf C equivalent l-expr(?x) )
    ```

    *where `l-expr(?x)` is a logical expression with precisely one free variable `?x`*

  **Example:**

  - The concept “Human” is defined as the intersection of the concepts “Primate” and “LegalAgent”

WSMO Ontologies – Relations

- **Relations** are used in order to model interdependencies between several concepts (respectively instances of these concepts)

  **Class**

  - relation: sub-Class wsmoElement
    - hasSuperRelation: relation
    - hasParameter: parameter
    - hasDefinition: logicalExpression
    - multiplicity = single-valued

  **Example:**

  - The relation “distanceInKm” has three parameters: two concepts and an integer. The relation represents the distance between two cities. It is a sub-relation of the measurement relation.
Logical Expressions for the Definition of Relations

The definition of a relation is a logical expression defining the set of instances (n-ary tuples, if n is the arity of the relation) of the relation.

- If the parameters are specified, the relation is represented by an n-ary predicate symbol with named arguments. If R is the identifier denoting the relation, then the logical expression takes one of the following forms:
  \[
  \forall v_1, \ldots, v_n \ (R[p_1 \text{ hasValue } v_1, \ldots, p_n \text{ hasValue } v_n] \implies \text{l-expr}(v_1, \ldots, v_n)) \\
  \forall v_1, \ldots, v_n \ (R[p_1 \text{ hasValue } v_1, \ldots, p_n \text{ hasValue } v_n] \equiv \text{l-expr}(v_1, \ldots, v_n))
  \]

- If the parameters are not specified, then the relation is represented by a predicate symbol where the identifier of the relation is used as the name of the predicate symbol. If R is the identifier denoting the relation, then the logical expression takes one of the following forms:
  \[
  \forall v_1, \ldots, v_n \ (R(v_1, \ldots, v_n) \implies \text{l-expr}(v_1, \ldots, v_n)) \\
  \forall v_1, \ldots, v_n \ (R(v_1, \ldots, v_n) \equiv \text{l-expr}(v_1, \ldots, v_n))
  \]

where l-expr($v_1, \ldots, v_n$) is a logical expression with precisely $v_1, \ldots, v_n$ as its free variables and $p_1, \ldots, p_n$ are the names of the parameters of the relation.

WSMO Ontologies – Instances

- Instances are either defined explicitly or by a link to an instance store, i.e., an external storage of instances and their values.
- An explicit definition of instances of concepts is as follows:
  
  Class concept sub-Class concept
  hasAttribute concept hasAttribute concept
  
  Class concept concept concept
  hasAttribute concept concept concept

Example:

- Mary is parent of the twins Paul and Susan

Example:

- The distance between Innsbruck and Munich is 234 kilometers

WSMO – the Web Service Element

- WSMO Web service descriptions consist of non-functional, functional, and the behavioral aspects of a Web service:
  
  - A Web service is a computational entity which is able (by invocation) to achieve a users goal.
  - A service in contrast is the actual value provided by this invocation.

- A WSMO Web service description consists of non-functional, functional, and behavioral aspects of a Web service.
  
  Advertisements: Advertisements
  Functional: Functional
  Non-functional: Non-functional
  Capability: Capability
  Service Interface: Service Interface
  Service: Service
  Description: Description
  IF, QoS, V: Interface, QoS, Version
  Financial: Financial
  Implementation: Implementation
  Technical: Technical
  Description: Description
WSMO – Web Service Non-Functional Properties

Non-functional properties:
- Accuracy - the error rate generated by the service
- Financial - the cost-related and charging-related properties of a service
- Network-related QoS - QoS mechanisms operating in the transport network which are independent of the service
- Performance - how fast a service request can be completed
- Reliability - the ability of a service to perform its functions (to maintain its service quality)
- Robustness - the ability of the service to function correctly in the presence of incomplete or invalid inputs.
- Scalability - the ability of the service to process more requests in a certain time interval
- Security - the ability of a service to provide authentication, authorization, confidentiality, traceability/auditability, data encryption, and non-repudiation
- Transactional - transactional properties of the service
- Trust - the trustworthiness of the service

Example:
If the client is older than 60 or younger than 10 years old the invocation price is lower than 10 euros.

WSMO – Web Service Capability

A capability defines the Web service by means of its functionality

Example:
The input for a birth registration service in Germany has to be boy or a girl with birthdate in the past and be born in Germany. The effect of the execution of the service is that after the registration the child is a German citizen.

WSMO – Web Service Interface

An interface describes how the functionality of the Web service can be achieved (i.e. how the capability of a Web service can be fulfilled) by providing a twofold view on the operational competence of the Web service:
- Choreography decomposes a capability in terms of interaction with the Web service
- Orchestration decomposes a capability in terms of functionality required from other Web services

Example:

WSMO Choreography: An Abstract State Machine Model (1)

Why ASMs-based model?
- Minimality: ASMs are based on a small assortment of modeling primitives
- Expressivity: ASMs can model arbitrary computations
- Formality: ASMs provide a formal framework to express dynamics

Basic mechanism in ASMs:
- A signature defines predicates and functions to be used in the description.
- Ground facts specify the underlying database states.
- State changes are described using transition rules, which specify how the states change by falsifying (deleting) some previously true facts and inserting (making true) some other facts.

Example:
The input for a birth registration service in Germany has to be boy or a girl with birthdate in the past and be born in Germany. The effect of the execution of the service is that after the registration the child is a German citizen.
Abstract State Machines

• Basic ASM are finite sets of conditional state transition rules of the form:
  \[ \text{if } \text{Condition} \text{ then } \text{Updates} \]
• A state is represented by a first order structure; a set with relations and functions
• Every algorithm can be rewritten as a finite number of transition rules

Abstract State Machines

• Signature is a finite collection of function names – each name comes with an indication of its arity
• Updates is a finite set of assignments of the form
  \[ f(t_1, \ldots, t_n) := t \]
• Execution can be understood as changing (or defining, if there was none) in parallel the value of the occurring functions \( f \) at the indicated arguments to the indicated value

Abstract State Machines

• A guarded rule is a transition
  \[ \text{if } \text{Condition} \text{ then } \text{Updates} \]
  where \text{Condition} is the guard under which a rule is applied
• A set of guarded updates are written usually as a list
• They are executed in parallel, so order is immaterial
• All guarded updates on the list are performed simultaneously

Abstract State Machines

• Execution of an ASM
  1. Check which rules apply
  2. Randomly select a/all rule(s)
  3. Perform update
WSMO Choreography: An Abstract State Machine Model (2)

In WSMO:
- Signatures are defined using ontologies.
- The ground facts that populate database states are instances of concepts and relations defined by the ontologies.
- State changes are described in terms of creation of new instances or changes to attribute values of existing instances.

Transition rules used in WSMO:
- \( \text{if Condition then Rules} \)
- \( \forall \text{Variables with Condition do Rules} \)
- \( \text{choose Variables with Condition do Rules} \)

A logical expression, as defined by WSML. A set of ASM rules: primitive state changes, like add, delete, or update (modify) a fact.

Examples:
- The state signature of the Amazon E-Commerce Service includes the concepts ItemSearchRequest and ItemLookupRequest with mode "in" and BrowseNodeLookupResponse, ItemContainer with mode "out".
- The ItemSearch transition rule checks for the presence of a request ItemSearchRequest and adds an instance of the corresponding ItemSearchResponse to the state (i.e. the state of the execution is changed).

WSMO Goals

- Goals are representations of an objective for which fulfillment is sought through the execution of a Web service. Goals can be descriptions of Web services that would potentially satisfy the user desires.

Example: Web Service Discovery

- Distinguish between abstract service and a specific one
  - Abstract service: a computational entity able to provide many services
  - Service: a concrete invocation of a Web service
- The task
  - Client is interested in getting a specific service
  - Identify possible service providers, which may be able to provide the requested service S for its clients
- Discovery
  - Given a goal and some Service repository determine the set of relevant service providers.

Example:
- A person named Paul has to go to goal to register his son with the German birth registration board.
Example: Web Service Discovery (cont’)

Goal: buy a travel ticket from Vienna to Berlin

Web service: sells train tickets for trips within Europe

Reasoning

Match!

WSMO MEDIATORS

Mediation
- Data Level: mediate heterogeneous Data Sources
- Protocol Level: mediate heterogeneous Communication Patterns
- Process Level: mediate heterogeneous Business Processes

Four different types of mediators in WSMO
- ggMediators: mediators that link two goals. This link represents the refinement of the source goal into the target goal or state equivalence if both goals are substitutable
- ooMediators: mediators that import ontologies and resolve possible representation mismatches between ontologies
- wgMediators: mediators that link Web services to goals, meaning that the Web service (totally or partially) fulfills the goal to which it is linked. wgMediators may explicitly state the difference between the two entities and map different vocabularies (through the use of ooMediators)
- wwMediators: mediators linking two Web services

Examples:
- The ooMediator identified by http://example.org/ooMediator translates the owl description of the iso ontology to wsml and adds the necessary statements to make them memberOf> loc:country concept of the wsmo location ontology
- The ggMediator identified by http://example.org/ggMediator links the general goal of getting a citizenship with the concrete goal of registering George
Heterogeneity may exist between exposed communication interfaces of service providers and those expected by service requesters:
- Messages in the wrong order
- Messages sent separately that are expected together
- Messages sent together that are expected separately
- Messages sent that are never expected
- Messages expected but never sent

Process Mediation required to address these heterogeneity issues and enable dynamic communication between requester and provider.

- Design-time Process Mediation
  - Input: 2 or more processes
  - Output: 1 mediator process
  - Advantage: Automatic
  - Disadvantage: Manual to semi-automatic

- Run-time Process Mediation
  - Input: 2 or more processes
  - Output: 1 mediator process
  - Advantage: No un-solvable mismatches
  - Disadvantage: Manual to semi-automatic
WEB SERVICE MODELING LANGUAGE

A set of concrete languages for the various tasks:

- Ontology / Rule Languages (static view)
  - WSML Core
  - WSML DL
  - WSML Rule
  - WSML Full
- Languages for dynamics
  - Transaction Logic over ASMs
- Mapping languages
  - for dynamics (process mediation)
  - or data (data mediation)

More in Lecture 8

WEB SERVICE MODELING EXECUTION ENVIRONMENT

WSMX – Web Service Execution Environment

- WSMX – reference implementation for WSMO/L
- Architecture and execution environment

More in Lecture 9
• The scenario is about how to identify possibly relevant services.
• With an invocation of one of the Web Services you can order a shipment by specifying, senders address, receivers address, package information and a collection interval during which the shipper will come to your premises to collect the package.
• The request contains the interval in which the shipper shall come to the requesters premises to pick up the package.
• A shipper either responds with the estimated pickup (respecting the given time constraints) or with a fault message indicating that a pick up is not possible in the requested time interval.
• If no constraints on the business hours (earliest and latest pick up time) are given one can assume 8am to 8pm. If a shipper specifies a constraint on how long in advance a shipment can be ordered, this means that the requested collection interval must end before this date. If no constraints on the length of the interval is given one can assume that a shipper requires at least an interval of 60 Minutes.

Examples of ontology elements for the shipment discovery scenario

• Ontology ShipmentOntology has annotations, dc#title (has value "Shipment Domain Ontology"), dc#contributor (has value "Maciej Zaremba, Matt Moran"), dc#date (has value 2006.10.23),
• Ontology ShipmentOntology has concepts:
  – OrderRequest has annotation dc#description whose value is "Information provided for a pickup request" and has a set of attributes:
    – from (of type ContactInfo), to (of type ContactInfo), type (of type ShipmentType), package (of type Package)
  – Package has annotation dc#description whose value is "concept of a package" and a set of attributes: quantity (of type integer), length (of type decimal), width (of type decimal), height (of type decimal), weight (of type decimal)
  – Country has annotation dc#description whose value is "concept of a country" and attributes name (of type string), continent (of type Continent)
    – …
Examples of ontology elements for the shipment discovery scenario (cont')

- Ontology ShipmentOntology has relations:
  - cityOnContinent (relation that holds between a city and the continent it belongs to) with two parameters whose types are City and Continent
  - cityInCountry (relation that holds between a city and the country it belongs to) with two parameters whose types are City and Country

- Ontology ShipmentOntology has instances:
  - Europe (member of Continent) whose name has value "Europe"
  - NY (member of City) whose name has value "New York" and country has value USA
  - Luxembourg (member of City) whose name has value "Luxembourg" and country has value Luxembourg

- Ontology ShipmentOntology has the axiom:
  - cityOnContinentDef defined by a logical formula that states that if a given city is in a certain country and that country is in a certain continent, then the given city is part of that continent

Examples of Shipping Services

- Weight:
  - Male: a package weighing 50 lbs or less
  - Female: a package weighing 50 lbs or less

- Constraints on Collection:
  - There must be at least an interval of 120 minutes for collection.
  - Collection is possible between 6am and 12pm.

- Shipment can be ordered maximum 2 business days in advance.
  - Delivery Time:
    - Package is collected 4 business days if picked up before 8pm.
    - Packages are delivered to the designated location at the end of the pickup interval.
    - Packages must be received by the end of the pickup interval.

Examples of Shipping Services (cont')

- Male:
  - Rates for Oceania like Asia (96.75/13), Rates for South America like North America (34.5/10)

- Female:
  - Rates for Oceania like Asia (60/10), Rates for South America like North America (59/12)

- Rates (flat fee/each lb):
  - Europe (50/5.75), Asia (60/8.5), North America (34/3)

- Collection can be ordered until 8pm.
  - Packages are delivered to the designated location at the end of the pickup interval.

- Shipment can be ordered maximum 2 business days in advance.
  - Delivery Time:
    - Packages are collected 4 business days if picked up before 8pm.

Examples of Shipping Services

- Male:
  - Rates for Oceania like Asia (96.75/13), Rates for South America like North America (34.5/10)

- Female:
  - Rates for Oceania like Asia (60/10), Rates for South America like North America (59/12)

- Rates (flat fee/each lb):
  - Europe (50/5.75), Asia (60/8.5), North America (34/3)

- Collection can be ordered maximum 2 business days in advance.
  - Delivery Time:
    - Packages are collected 4 business days if picked up before 8pm.
Examples of goals

Goal C3
to Smithers (Bristol)
no of packages: 1
package dimensions: (l/w/h) 10/2/3 (inch)
package weight: 20 lbs
for less than 120$

Goal D1
to Szyslak (Tunis)
no of packages: 2
package dimensions: (l/w/h) 5/3/2 (inch)
package weight: 60 lbs (each)

Goal E1
to Gumble (New York)
package dimensions: (l/w/h) 10/2/3 (inch)
package weight: 5 lbs
for less than 20$

Current Time is 7:30 am
Next day delivery

Result of Discovery Process

Goal C3
to Smithers (Bristol)
no of packages: 1
package dimensions: (l/w/h) 10/2/3 (inch)
package weight: 20 lbs
for less than 120$

Goal D1
to Szyslak (Tunis)
no of packages: 2
package dimensions: (l/w/h) 5/3/2 (inch)
package weight: 60 lbs (each)

Goal E1
to Gumble (New York)
package dimensions: (l/w/h) 10/2/3 (inch)
package weight: 5 lbs
for less than 20$

Current Time is 7:30 am
Next day delivery

⇒ Muller (includes a request for quote)
NOT: Racer (price is 176$)
NOT: Runner (price is 176$)
NOT: Walker (price is 151$)
NOT: Weasel (ships not to UK)

⇒ Muller (2 invocations, since schema does not allow to order multiple packages in one invocation)
NOT: Racer (does not ship to Tunesia)
NOT: Runner (does not ship to Tunesia)
NOT: Walker (does only ship 50lbs)
NOT: Weasel (does not ship to Tunesia)

WSMO-Lite

WSMO-Lite Ontology
layers of semantic annotations
extends

MicroWSMO

service description layer

hRESTS

EXTENSIONS
Semantics in Service Model

MicroWSMO

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

More in Lecture 11
Summary

- Semantic Web Services
  - Have the potential of improving the usability of services
  - Lots of progress in the last years
- The WSMO Approach is an active initiative in the area of SWS
  - The WSMO conceptual model consists of four core elements: Ontologies, Web Services, Goals, and Mediators
- Standardization based on the WSMO Approach is emerging
  - OASIS See TC

REFERENCES

- Mandatory reading:
  - [WSMO](http://www.wsmo.org/TR/d2/v1.3/)
- Further reading:
  - [WSMO](http://www.wsmo.org/TR/d2/v1.3/)
  - [WSMO](http://www.wsmo.org/gateway)
  - [WSMO](http://www.wsmo.org/execution)
  - [WSMO](http://www.wsmo.org/wsmx)
  - [WSMO](http://www.wsmo.org/training)
  - [WSMO](http://www.wsmo.org/products/wsmx)
  - [WSMO](http://www.oasis-open.org/homefile/semantic-svc)
- Wikipedia links:
  - [WSMO](http://en.wikipedia.org/wiki/WSMO)