Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft





Semantic Web Service Tutorial

Katia Sycara
Massimo Paolucci

Michael Stollberg
Matthew Moran
Michal Zaremba
Mick Kerrigan
Emilia Cimpian

Stefania Galizia
Barry Norton
Liliana Cabral
John Domingue

Carnegie Mellon University

Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

PART I:
Introduction to Semantic Web Services

Michael Stollberg

Content

- The vision of the Semantic Web
- Ontologies as the basic building block
- Current Web Service Technologies
- Vision and Challenges for Semantic Web Services

The Vision

- 500 million users
- more than 3 billion pages

Static

WWW

URI, HTML, HTTP

The Vision

Serious Problems in

- information finding,
- information extracting,
- information representing,
- information interpreting and
- and information maintaining.

Static

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL

The Vision

Dynamic

Web Services
UDDI, WSDL, SOAP

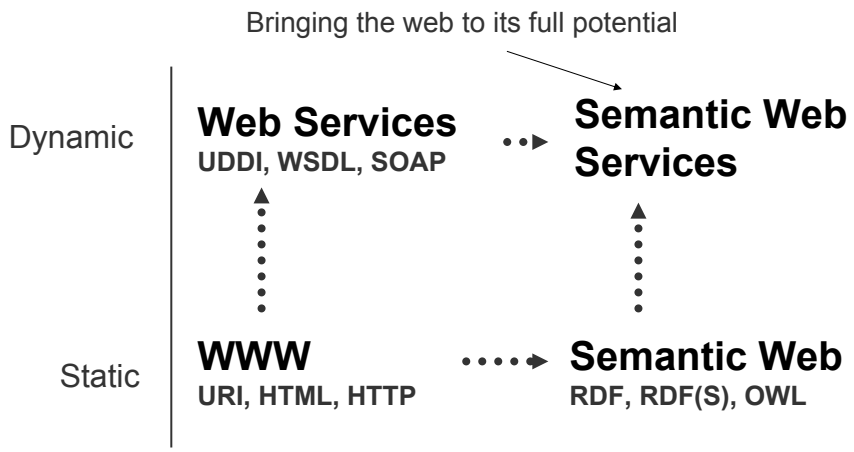
Bringing the computer back as a device for computation

Static

WWW
URI, HTML, HTTP

Semantic Web
RDF, RDF(S), OWL

The Vision

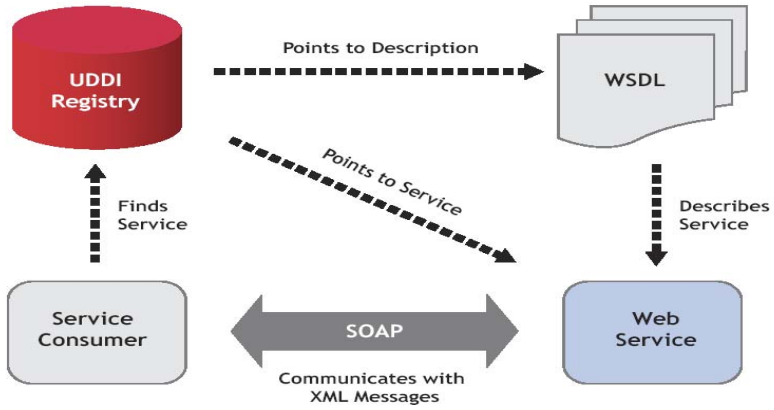


Web Services

- loosely coupled, reusable components
- encapsulate discrete functionality
- distributed
- programmatically accessible over standard internet protocols
- add new level of functionality on top of the current web

The Promise of Web Services

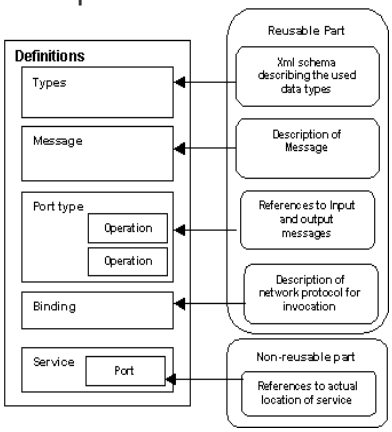
Web-based SOA as new system design paradigm



WSDL

- Web Service Description Language
- W3C effort, WSDL 2 final construction phase

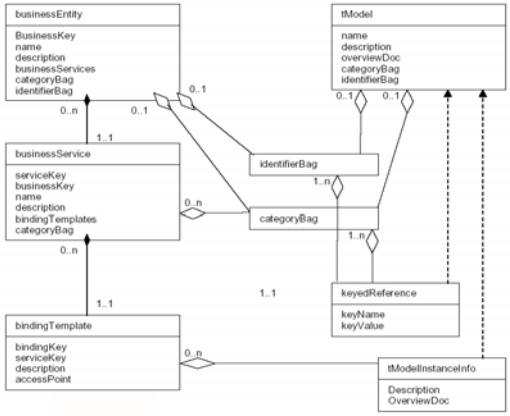
describes interface for consuming a Web Service:
- Interface: operations (in- & output)
- Access (protocol binding)
- Endpoint (location of service)



UDDI

- Universal Description, Discovery, and Integration Protocol
- OASIS driven standardization effort

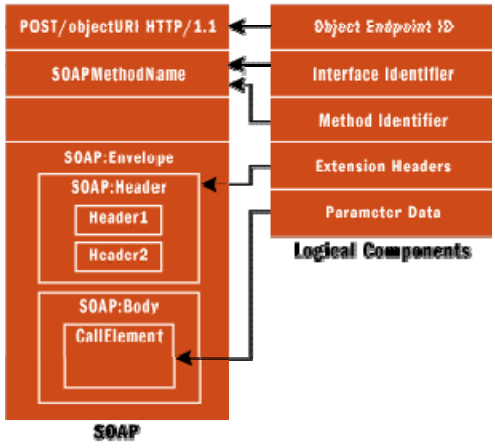
Registry for
Web Services:
- provider
- service information
- technical access



SOAP

- Simple Object Access Protocol
- W3C Recommendation

XML data transport:
- sender / receiver
- protocol binding
- communication aspects
- content



Lackings of WS Technology

- 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

=> current Web Service Technology Stack failed to realize the promise of Web Services

Semantic Web Services

Semantic Web Technology
allow machine supported data interpretation
ontologies as data model

+

Web Service Technology
automated discovery, selection, composition,
and web-based execution of services

=> Semantic Web Services as integrated solution for realizing the vision of the next generation of the Web

Semantic Web Services

- define exhaustive description frameworks for describing Web Services and related aspects
  (Web Service Description Ontologies)
- support ontologies as underlying data model to allow machine supported data interpretation
  (Semantic Web aspect)
- define semantically driven technologies for automation of the Web Service usage process
  (Web Service aspect)

Semantic Web Services: Usage Process

- Deployment — create & publish Web service description
- Discovery — determine usable services for a request
- Composition — combine services to achieve a goal
- Selection — choose most appropriate service among the available ones
- Mediation — solve mismatches (data, protocol, process) that hamper interoperation
- Execution — invoke Web services following programmatic conventions

Semantic Web Services: Execution Support

- **Monitoring** — control the execution process
- **Compensation** — provide transactional support and undo or mitigate unwanted effects
- **Replacement** — facilitate the substitution of services by equivalent ones
- **Auditing** — verify that service execution occurred in the expected way

Fachhochschule Nordwestschweiz
Hochschule für Wirtschaft

PART II:
Semantic Web Service Ontologies

Katia Sycara
Michael Stollberg

Content

- OWL-S
  - Upper Ontology
  - Service Profile
  - Process Model
  - Service Grounding
- WSMO
  - WSMO top level notions
  - Choreography and Orchestration
  - Mediation
- Differences and Commonalities

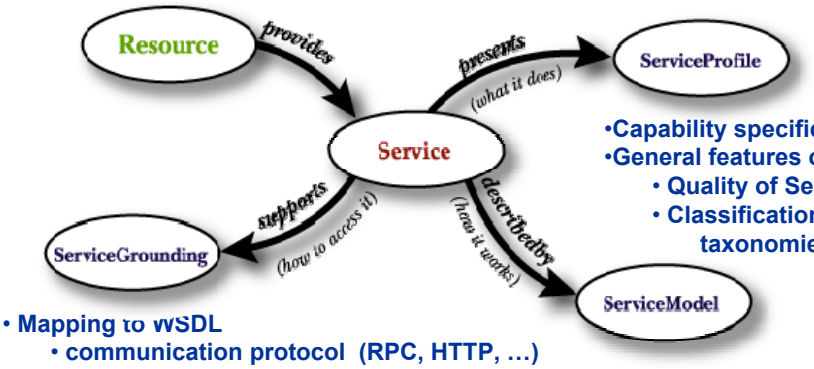
OWL-S Ontology

- 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

Complete do not compete

- OWL-S does not aim to replace the Web services standards
- rather OWL-S attempts to provide a semantic layer
  - OWL-S relies on WSDL for Web service invocation *(see Grounding)*
  - OWL-s Expands UDDI for Web service discovery *(OWL-S/UDDI mapping)*

OWL-S Upper Ontology

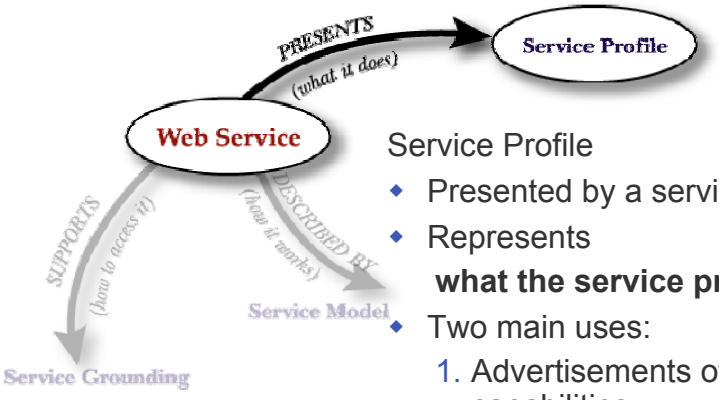


- Mapping to WSDL
  - communication protocol (RPC, HTTP, …)
  - marshalling/serialization
  - transformation to and from XSD to OWL

- 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

Service Profiles



Service Profile

- Presented by a service.
- Represents
  what the service provides
- Two main uses:
  1. Advertisements of Web Services capabilities
  2. Request of Web services with a given set of capabilities

OWL-S Profile in a Nutshell

- Describes Web service
  - What capabilities it provides:
    - What transformation the service computes
    - Type of service and products
  - General features such as
    - Agent providing the service
    - Security requirements
    - Quality guarantees of service
- Primary role: to assist discovery
  - Allows capability based search
  - Allows selection based on requirements of the requester
- Profile does not specify use/invocation

OWL-S Service Profile Capability Description

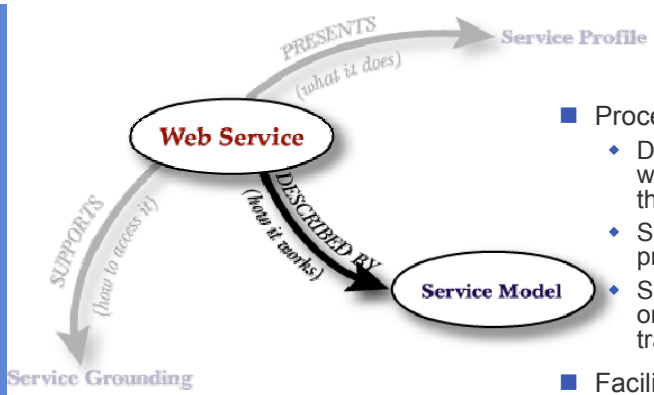
- **Preconditions**
  - Set of conditions that should hold prior to service invocation
- **Inputs**
  - Set of necessary inputs that the requester should provide to invoke the service
- **Outputs**
  - Results that the requester should expect after interaction with the service provider is completed
- **Effects**
  - Set of statements that should hold true if the service is invoked successfully.
- **Service type**
  - What kind of service is provided (eg selling vs distribution)
- **Product**
  - Product associated with the service (eg travel vs books vs auto parts)

OWL-S Service Profile Additional Properties

- **Security Parameters**
  - Specify the security capabilities of a Web service (eg support X509 Encryption)
  - Specify the security requirements of a Web service (eg a client should be able to provide X509 Encryption)
- **Quality rating**
  - What level of service quality does the Web service provide?
- **Description with standard business taxonomies**
  - How would the service be classified in standard taxonomies

This is not a closed set, new properties can be added using existing ontologies

Process Model



- Process Model
  - Describes how a service works: internal processes of the service
  - Specifies service interaction protocol
  - Specifies abstract messages: ontological type of information transmitted
- Facilitates
  - Web service invocation
  - Composition of Web services
  - Monitoring of interaction

Viewpoints of Process Model

Three viewpoints of a Web service

- **Glass Box:**
  - The Web service reveals all its internal structure
  - Which parts of the service it performs in-house, which one it subcontracts, etc
- **Black Box:**
  - The Web service model does not reveal anything about the internal working of the service
  - It just specifies what data it gathers and what data it sends back
- **Grey Box:**
  - The Web service selectively hides some parts of its Process Model, while it publicizes others

Definition of Process

- A Process represents a transformation (function). It is characterized by four parameters
  - ***Inputs***: the inputs that the process requires
  - ***Preconditions***: the conditions that are required for the process to run correctly
  - ***Outputs***: the information that results from (and is returned from) the execution of the process
  - ***Results***: a process may have different outcomes depending on some condition
    - ***Condition***: under what condition the result occurs
    - ***Constraints on Outputs***
    - ***Effects***: real world changes resulting from the execution of the process

Motivation for the Results Parameter

- Processes may terminate in exceptional states:
  - The credit company may fail to charge the credit card
  - The book may be out of stock
  - The deliver of the goods may fail
- Results support modeling of non-deterministic outcomes of Web services
  - The condition specifies when an outcome is generated
  - Each outcome is characterized by
    - a set of constraints on outputs
    - a set of effects

Example of Process

Inputs / Outputs

Precondition

Result

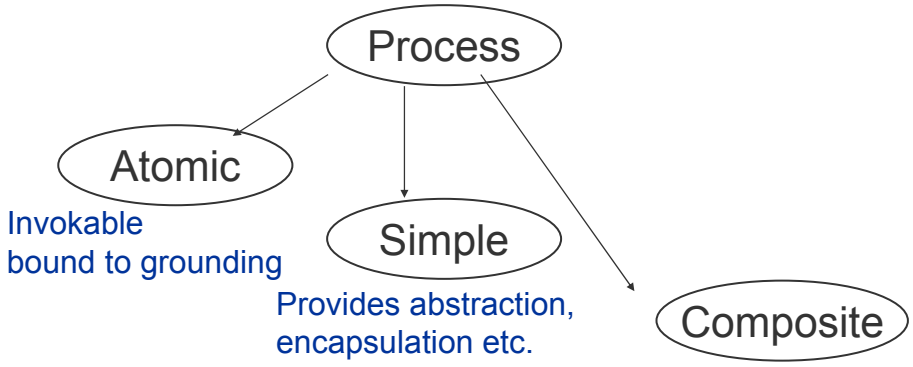
Condition

Output Constraints

Effect

```
<process:AtomicProcess rdf:ID="LogIn">
  <process:hasInput rdf:resource="#AcctName"/>
  <process:hasInput rdf:resource="#Password"/>
  <process:hasOutput rdf:resource="#Ack"/>
  <process:hasPrecondition isMember(AccName)/>
  <process:hasResult>
    <process:Result>
      <process:inCondition>
        <expr:SWRL-Condition>
          correctLoginInfo(AccName,Password)
        </expr:SWRL-Condition>
      </process:inCondition>
      <process:withOutput rdf:resource="#Ack">
        <valueType rdr:resource="#LoginAcceptMsg">
      </process:withOutput>
      <process:hasEffect>
        <expr:SWRL-Condition>
          loggedIn(AccName,Password)
        </expr:SWRL-Condition>
      </process:hasEffect>
    </process:Result>
  </process:hasResult>
</process:AtomicProcess>
```

Ontology of Processes



Invokable
bound to grounding

Provides abstraction,
encapsulation etc.

Defines a workflow
composed of process
performs

Process Model Organization

- **Process Model is described as a tree structure**
  - Composite processes are internal nodes
  - Simple and Atomic Processes are the leaves
- **Simple processes represent an abstraction**
  - Placeholders of processes that aren't specified
  - Or that may be expressed in many different ways
- **Atomic Processes correspond to the basic actions that the Web service performs**
  - Hide the details of how the process is implemented
  - Correspond to WSDL operations

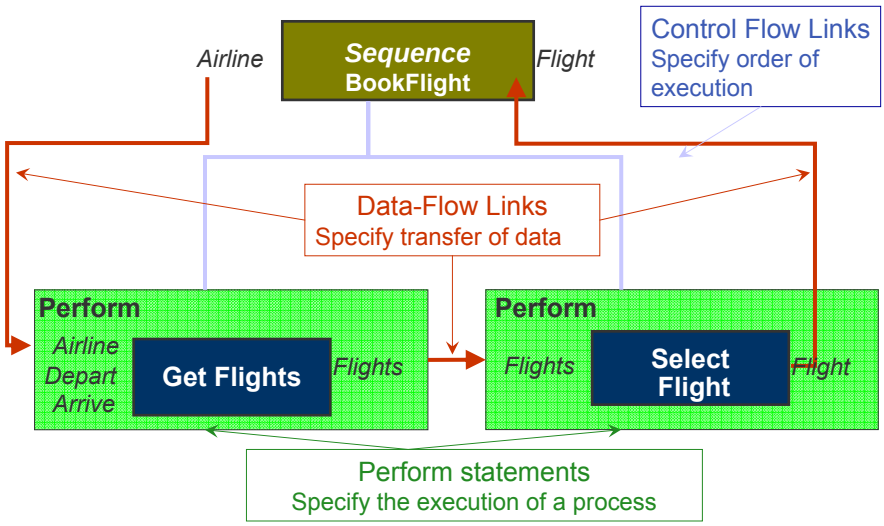
Composite Processes

- Composite Processes specify how processes work together to compute a complex function
- Composite processes define
  1. **Control Flow**

     Specify the temporal relations between the executions of the different sub-processes
  2. **Data Flow**

     Specify how the data produced by one process is transferred to another process

Example of Composite Process



Perform Construct

- *Perform* provides invocation mechanism
  - Specify context of process execution
    - input data flow
    - hooks for output data flow
- Distinction between definition and invocation of a process
  - Definition specifies the process' I/P/R
  - Perform specify when the process is invoked and with what parameters

Control Flow

- Processes can be chained to form a workflow
- OWL-S supports the following control flow constructs
  - ***Sequence/Any-Order***: represents a list of processes that are executed in sequence or arbitrary order
  - ***Conditionals***: if-then-else statements
  - ***Loops***: while and repeat-until statements
  - ***Multithreading and synchronization***: split process in multiple threads, and rendezvous (joint) points
  - ***Non-deterministic choices***: (arbitrarily) select one process of a set

Data Flow

Dataflow allows information to be transferred from process to process.

Output→Input:

The information produced by one process is transferred to another in the same control construct

Input →Input:

The information received by a composite process is transferred to the sub-processes

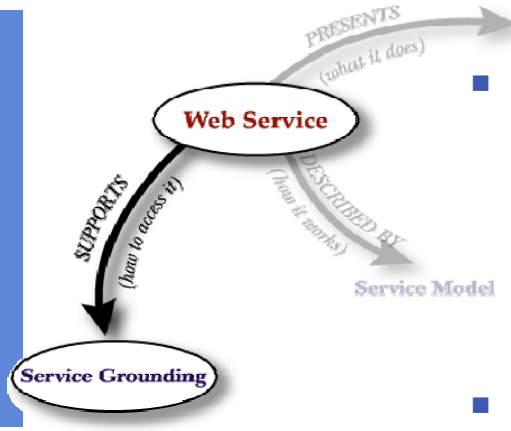
Output→Output:

The information produced by a subprocess is transferred to a super-process

Process Model

- Service Model describes
  - Set of processes that define the operations performed by the Web service
  - Control flow describing the temporal flow of processes
  - Data flow describing the transfer of information between sub-processes

Service Grounding



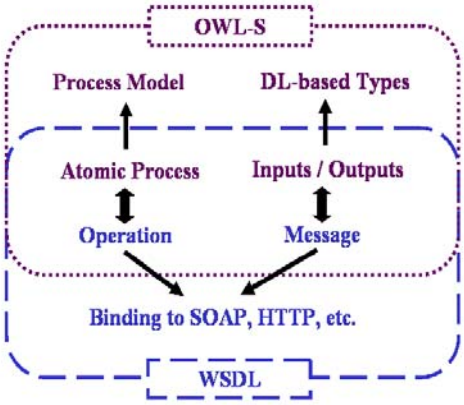
- Service Grounding
  - Provides a specification of service access information.
  - Service Model + Grounding give everything needed for using the service
  - Builds upon **WSDL** to define message structure and physical binding layer
- Specifies:
  - communication protocols, transport mechanisms, communication languages, etc.

Rationale of Service Grounding

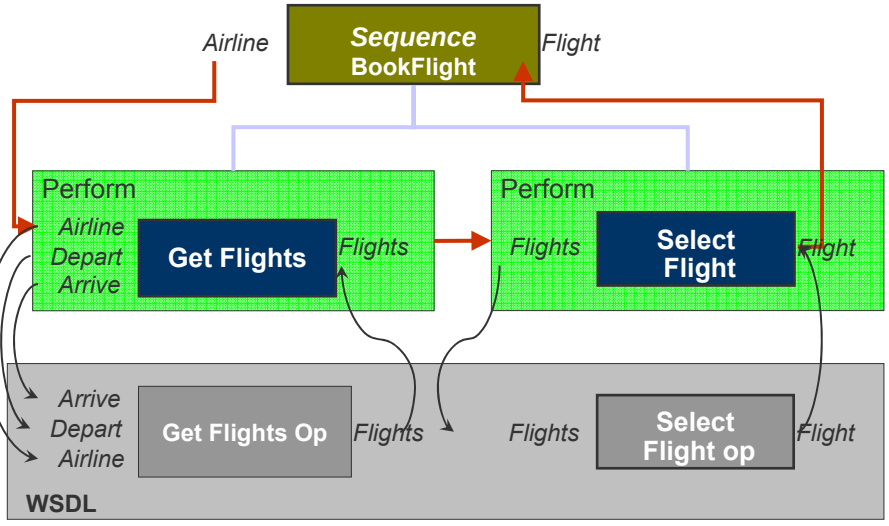
- Provides a specification of service access information.
- Service Model + Grounding give everything needed for using the service
  - Service description is for reasoning about the service
    - Decide what information to send and what to expect
  - Service Grounding is for message passing
    - Generate outgoing messages, and get incoming messages
    - Mapping XML Schemata to OWL concepts
- Builds upon **WSDL** to define message structure and physical binding layer

Mapping OWL-S / WSDL 1.1

- **Operations** correspond to Atomic Processes
- **Input/Output** messages correspond to Inputs/Outputs of processes



Example of Grounding



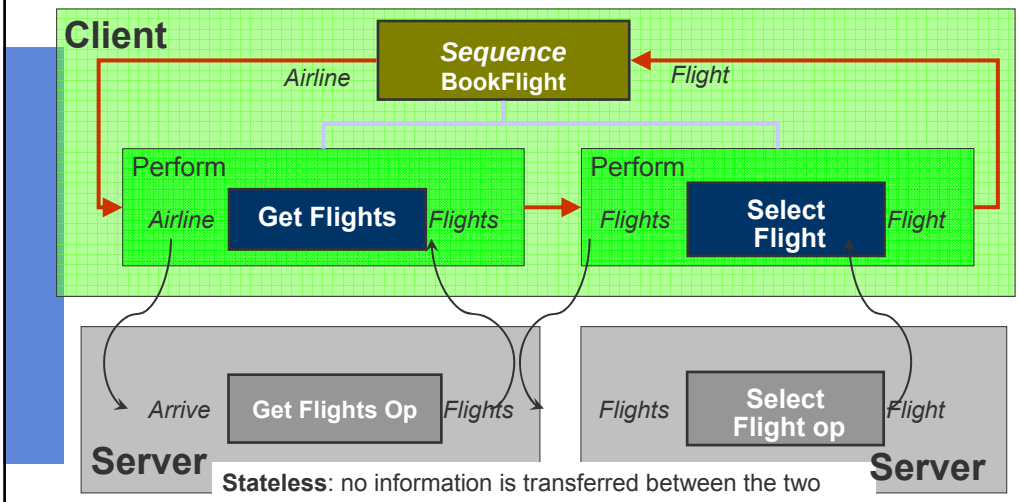
Result of using the Grounding

- Invocation mechanism for OWL-S
  - Invocation based on WSDL
  - Different types of invocation supported by WSDL can be used with OWL-S
- Clear separation between service description and invocation/implementation
  - Service description is needed to reason about the service
    - Decide how to use it
    - Decide how what information to send and what to expect
  - Service implementation may be based on SOAP and XSD types
  - The crucial point is that the information that travels on the wires and the information used in the ontologies is the same
- Allows any web service to be represented using OWL-S
  - For example: Amazon.com

Handling stateful vs stateless Web services

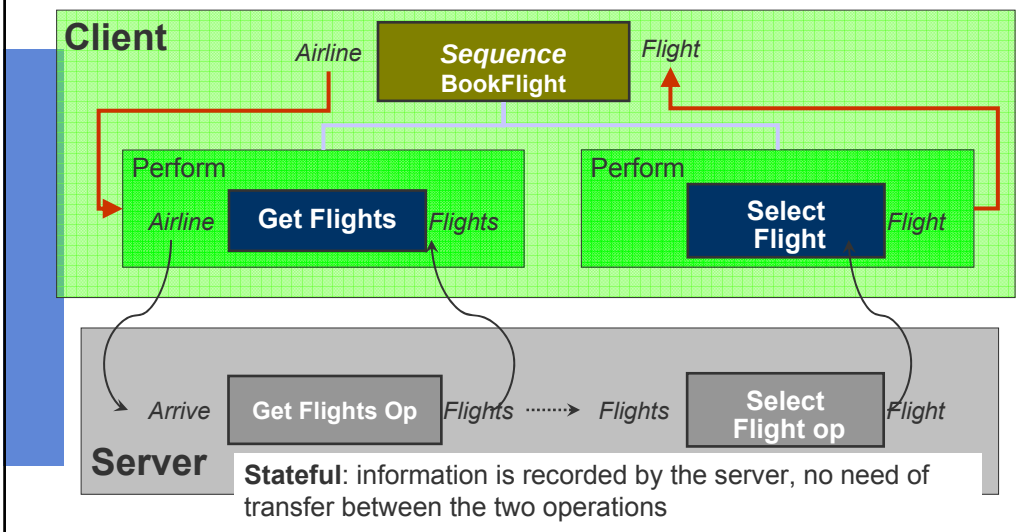
1. Stateless Web services
   - The server does not maintain the state of the computation
   - Dataflow links specify how the client communicate the state to the service
2. Stateful Web services
   - The service does maintain the state
   - No need of dataflow links since transfer of information is opaque to the client

Representing Stateful Web services



Stateless: no information is transferred between the two operations

Representing Stateless Web services



Stateful: information is recorded by the server, no need of transfer between the two operations

Conclusion OWL-S

OWL-S provides a language for the description of Web services

- Service Profile provides description of capabilities of Web Service
  - Allows capability-based discovery
- Process Model provides the description of how to use a Web service
  - Allows automatic invocation of Web service
- Service Grounding maps Atomic Processes into WSDL operations
  - Allows separation between description and implementation
  - Supports description of arbitrary Web services

Web Service Modeling Ontology WSMO

Michael Stollberg

Outline

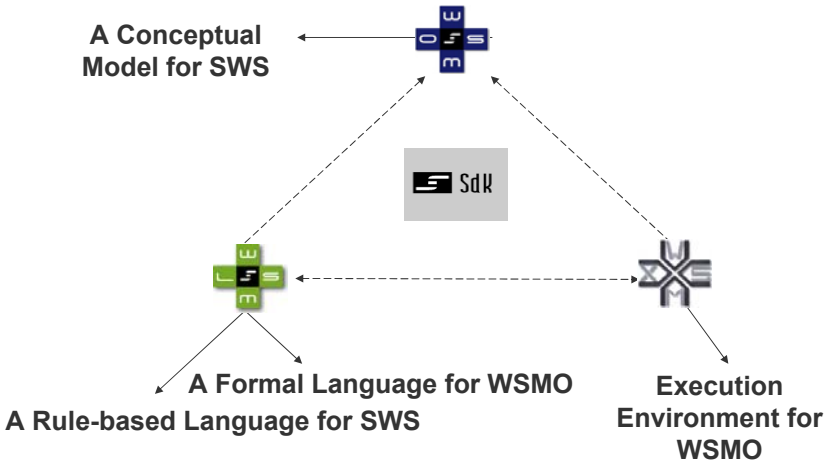
Outline

- WSMO Working Groups
- Top Level Notions
  - Ontologies
  - Web Services
  - Goals
  - Mediators

WSMO is ..

- a conceptual model for Semantic Web Services:
  - ontology of core elements for Semantic Web Services
  - a formal description language (WSML)
  - execution environment (WSMX)
- derived from and based on the Web Service Modeling Framework WSMF
- a SDK-Cluster Working Group
  - (joint European research and development initiative)

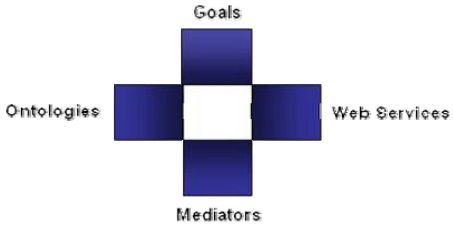
WSMO Working Groups



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 D2, version 1.2, 13 April 2005 (W3C submission)

Non-Functional Properties

Every WSMO element is described by properties that contain relevant, non-functional aspects

- Dublin Core Metadata Set:
  - complete item description
  - used for resource management
- Versioning Information
  - evolution support
- Quality of Service Information
  - availability, stability
- Other
  - Owner, financial

Non-Functional Properties List

Dublin Core Metadata
- Contributor
- Coverage
- Creator
- Description
- Format
- Identifier
- Language
- Publisher
- Relation
- Rights
- Source
- Subject
- Title
- Type

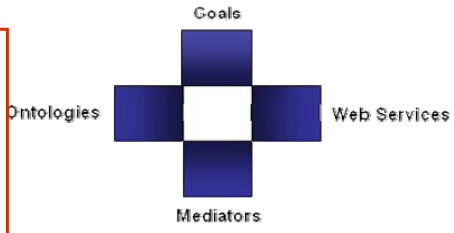
Quality of Service
- Accuracy
- NetworkRelatedQoS
- Performance
- Reliability
- Robustness
- Scalability
- Security
- Transactional
- Trust

Other
- Financial
- Owner
- TypeOfMatch
- Version

WSMO Ontologies

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 Usage & Principles

- **Ontologies are the 'data model' throughout WSMO**
  - all WSMO element descriptions rely on ontologies
  - all data interchanged in Web Service usage are ontologies
  - Semantic information processing & 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, modular approach for ontology design
  - 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)

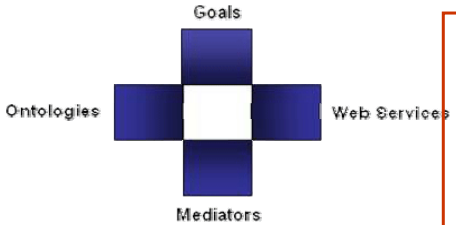
Ontology Elements:

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 Ontologies

Objectives that a client wants to achieve by using Web Services

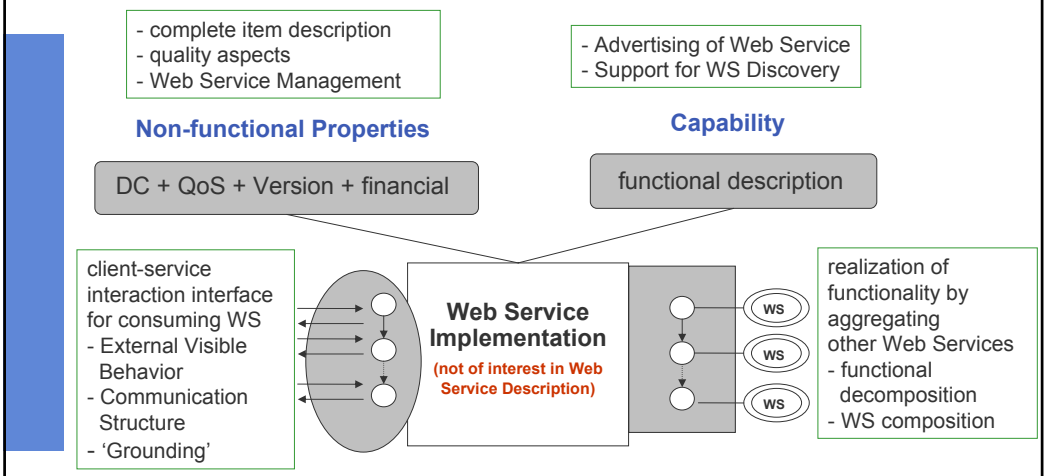
Provide the formally specified terminology of the information used by all other components



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

Connectors between components with mediation facilities for handling heterogeneities

WSMO Web Service Description



Choreography --- Service Interfaces --- **Orchestration**

Capability Specification

- **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
- **Pre-conditions**

  What a web service expects in order to be able to provide its service. They define conditions over the input.
- **Assumptions**

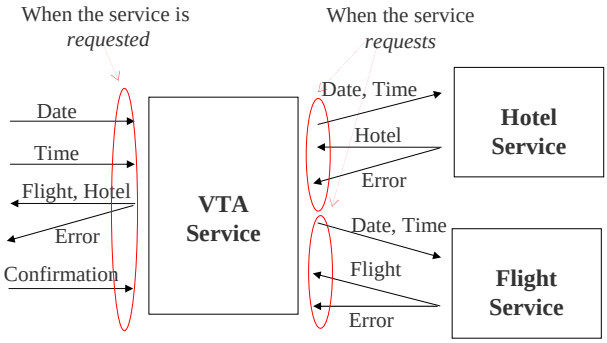
  Conditions on the state of the world that has to hold before the Web Service can be executed
- **Post-conditions**

  describes the result of the Web Service 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)

Choreography & Orchestration

- VTA example:



- **Choreography =** how to interact with the service to consume its functionality
- **Orchestration =** how service functionality is achieved by aggregating other Web Services

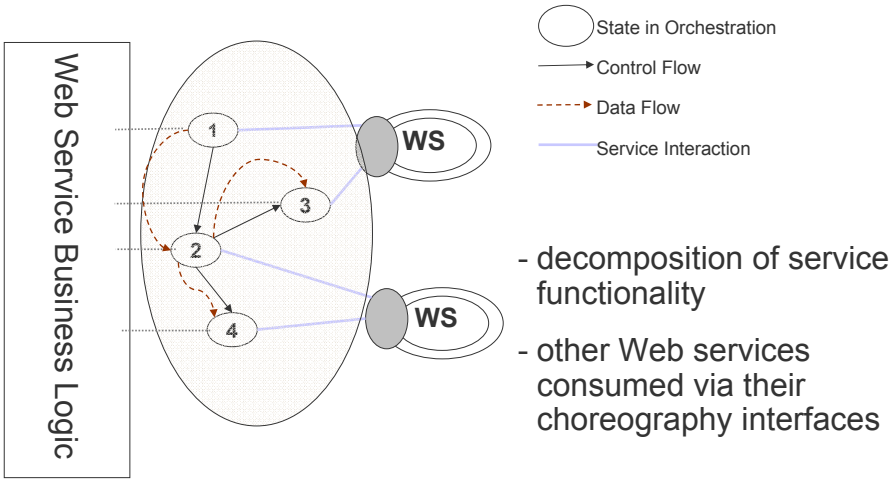
Choreography Interfaces

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)
- **Grounding**
  - executable communication technology for interaction
  - choreography related errors (e.g. input wrong, message timeout, etc.)
- **Formal Model**
  - reasoning on Web Service interfaces (service interoperability)
  - semantically enabled mediation on Web Service interfaces

Orchestration Aspects

Behavior for Interaction with aggregated Web Services



- decomposition of service functionality
- other Web services consumed via their choreography interfaces

WSMO Web Service Interfaces

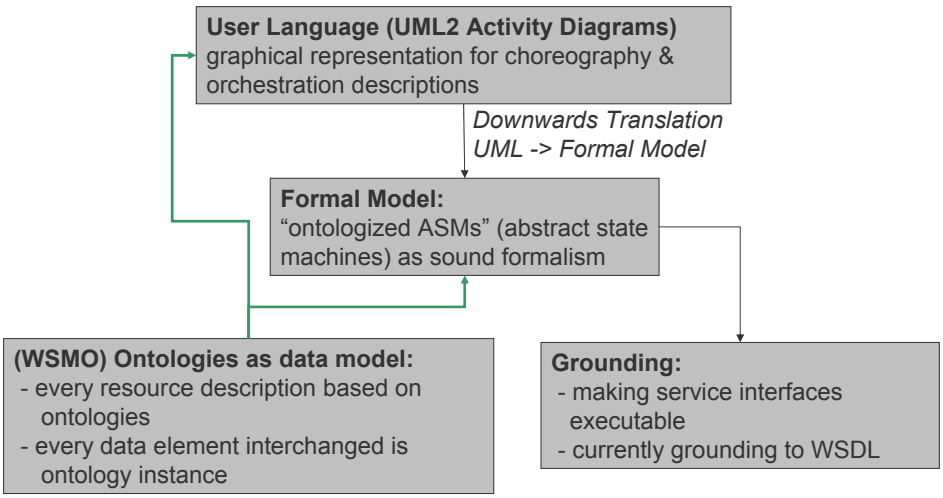
- behavior interfaces of Web services and clients for “peer-2-peer” interaction
- Choreography and Orchestration as sub-concepts of Service Interface with common description language
- service interface description aspects:
  1. represent the **dynamics** of information interchange during service consumption and interaction
  2. support **ontologies** as the underlying data model
  3. appropriate **communication technology** for information interchange
  4. sound **formal model / semantics** of service interface specifications in order to allow advanced reasoning on them

service interface = evolving ontology

Service Interface Description Approach

- Abstract State Machines (ASM) as formal framework:
  - dynamics representation: high expressiveness & low ontological commitment
  - core principles: state-based, state definition by formal algebra, guarded transitions for state changes
  - overcome the “Frame Problem”
- further characteristics:
  - not specific communication technology

Service Interface Description



Ontologized Abstract State Machines

- Vocabulary Ω:
  - ontology schema(s) used in service interface description
  - usage for information interchange: in, out, shared, controlled
- States ω(Ω):
  - a stable status in the information space
  - defined by attribute values of ontology instances
- Guarded Transition GT(ω):
  - state transition
  - general structure: ***if*** (condition) ***then*** (update)
    - condition on current state, update = changes in state transition
    - all GT(ω) whose condition is fulfilled fire in parallel

Service Interface Example

Communication Behavior of a Web Service

Ω_{in} **hasValues** {
concept A [
att1 **ofType** X
att2 **ofType** Y]
…}

Ω_{out} **hasValues** {
concept B [
att1 **ofType** W
att2 **ofType** Z]
…}

Vocabulary:
- Concept A in Ω_{in}
- Concept B in Ω_{out}

State ω_1

a **memberOf** A [
att1 **hasValue** x
att2 **hasValue** y]

received ontology instance ***a***

Guarded Transition GT(ω_1)

IF (a **memberOf** A [
att1 **hasValue** x])
THEN
(b **memberOf** B [
att2 **hasValue** m])

State ω_2

a **memberOf** A [
att1 **hasValue** x,
att2 **hasValue** y]

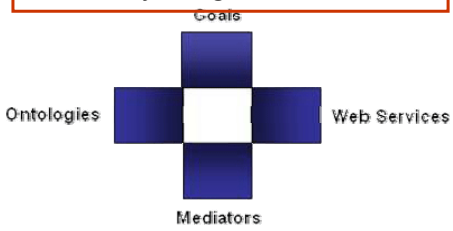
b **memberOf** B [
att2 **hasValue** m]

sent ontology instance ***b***

WSMO Goals

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)*

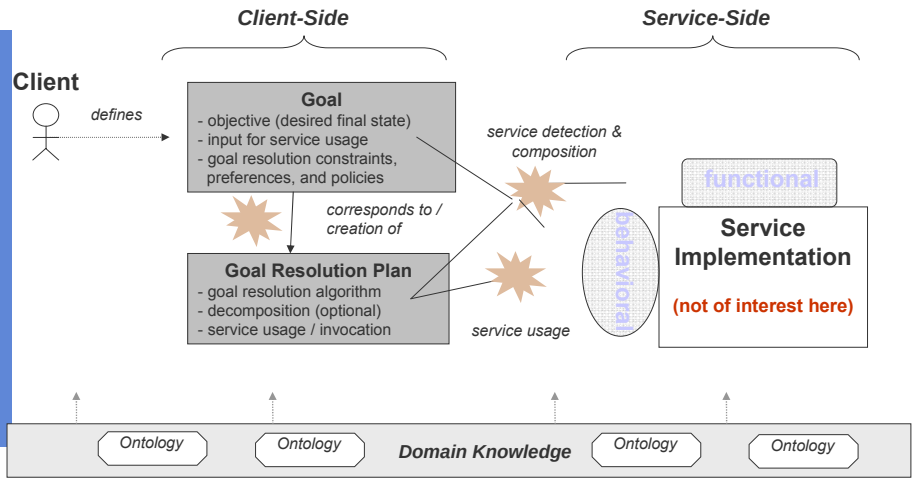
Connectors between components with mediation facilities for handling heterogeneities

Goals

Client Objective Specification along with all information needed for automated resolution

- **Goal-driven Approach**, derived from AI rational agent approach
  - ontological De-coupling of Requester and Provider
  - 'intelligent' mechanisms detect suitable services for solving the Goal
  - service re-use & knowledge-level client side support
- **Usage of Goals within Semantic Web Services**
  - A Requester (human or machine) defines a Goal to be resolved independently and on the knowledge level
  - SWS techniques / systems automatically determine Web Services to be used for resolving the Goal (discovery, composition, execution, etc.)
  - Ontological relationships & mediators used to link goals to web services
  - Goal Resolution Management is realized in implementations

Goal-driven Architecture



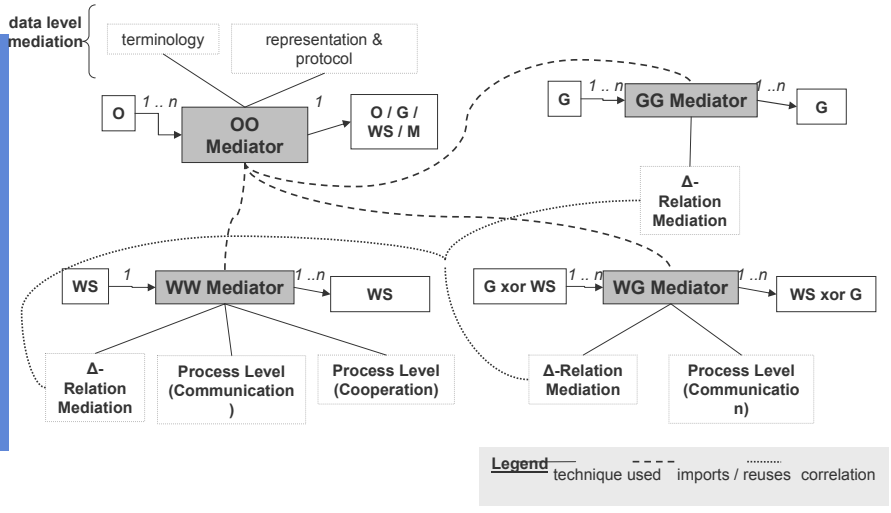
Goal Specification

- 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
- 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

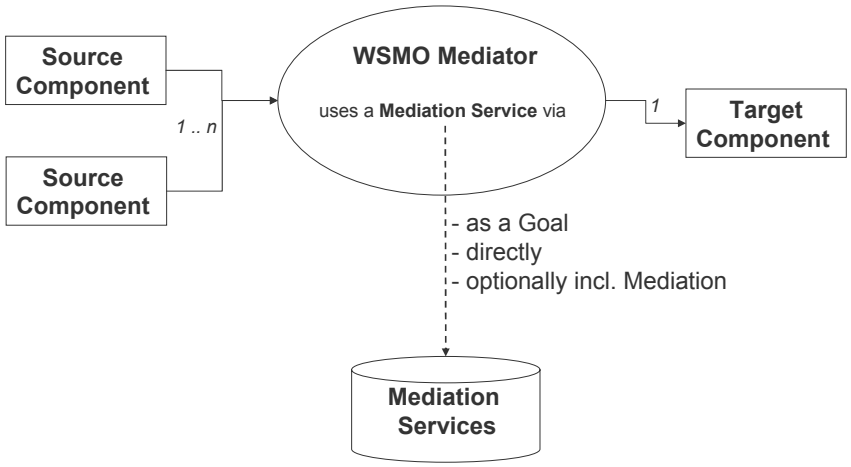
Mediation

- **Heterogeneity …**
  - Mismatches on structural / semantic / conceptual / level
  - Occur between different components that shall interoperate
  - Especially in distributed & open environments like the Internet
- **Concept of 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)
- **Levels of Mediation within Semantic Web Services** (WSMF):
  - **(1) Data Level:** mediate heterogeneous Data Sources
  - **(2) Protocol Level:** mediate heterogeneous Communication Patterns
  - **(3) Process Level:** mediate heterogeneous Business Processe

WSMO Mediators Overview

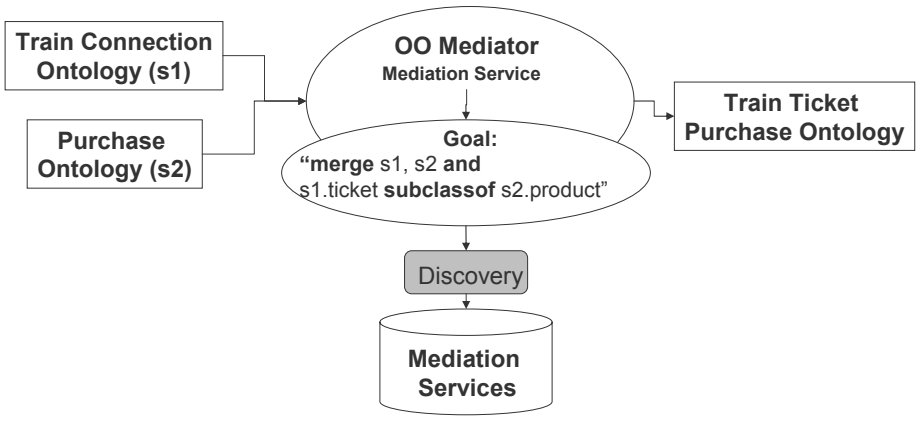


Mediator Structure



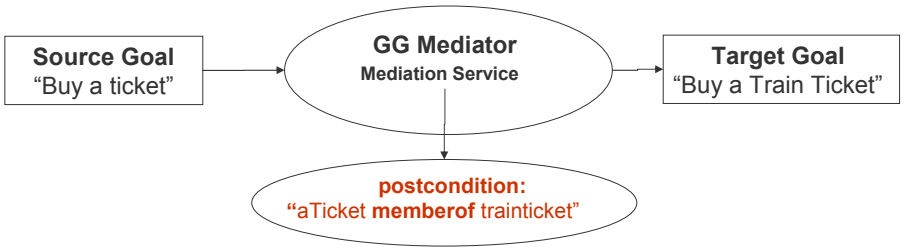
OO Mediator - Example

Merging 2 ontologies



GG Mediators

- Aim:
  - 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
- Example: Goal Refinement



WG & WW Mediators

- **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

OWL-S and WSMO

Commonalities and Differences

OWL-S and WSMO

- **OWL-S =** ontology and language to describe Web services
- **WSMO =** ontology and language for core elements of Semantic Web Service systems

Main Description Elements Correlation:

OWL-S profile	≈ WSMO capability + goal + non-functional properties
OWL-S Process Model	**≈ WSMO Service Interfaces**
OWL-S Grounding	≈ current WSMO Grounding

Perspective

- OWL-S is an ontology and a language to describe Web services
  - Strong relation to Web Services standards
    - rather than proposing another WS standard, OWL-S aims at enriching existing standards
    - OWL-S is grounded in WSDL and it has been mapped into UDDI
  - Based on the Semantic Web
    - Ontologies provide conceptual framework to describe the domain of Web services and an inference engine to reason about the domain
    - Ontologies are essential elements of interoperation between Web services
- WSMO is a conceptual model for the core elements of Semantic Web Services
  - core elements: Ontologies, Web Services, Goals, Mediators
    - language for semantic element description (WSML)
    - reference implementation (WSMX)
  - Mediation as a key element
  - Ontologies as data model
    - every resource description is based on ontologies
    - every data element interchanged is an ontology instance

OWL-S and WSMO

OWL-S profile ≈ WSMO capability +
goal +
non-functional properties

- OWL-S uses Profiles to express existing capabilities (advertisements) and desired capabilities (requests)
- WSMO separates provider (capabilities) and requester points of view (goals)

OWL-S and WSMO

OWL-S Process Model ≈ WSMO Service Interfaces

- Perspective:
  - OWL-S Process Model describes operations performed by Web Service, including consumption as well as aggregation
  - WSMO separates Choreography and Orchestration
- Formal Model:
  - OWL-S formal semantics has been developed in very different frameworks such as Situation Calculus, Petri Nets, Pi-calculus
  - WSMO service interface description model with ASM-based formal semantics
  - OWL-S Process Model is extended by SWRL / FLOWS

both approaches are not finalized yet

OWL-S and WSMO

OWL-S Grounding ≈ current WSMO Grounding

- OWL-S provides default mapping to WSDL
  - clear separation between WS description and interface implementation
  - other mappings could be used
- WSMO also defines a mapping to WSDL, but aims at an ontology-based grounding
  - avoid loss of ontological descriptions throughout service usage process
  - 'Triple-Spaced Computing' as innovative communication technology

Mediation in OWL-S and WSMO

- OWL-S does not have an explicit notion of mediator
  - Mediation is a by-product of the orchestration process
    - E.g. protocol mismatches are resolved by constructing a plan that coordinates the activity of the Web services
  - …or it results from translation axioms that are available to the Web services
    - It is not the mission of OWL-S to generate these axioms
- WSMO regards mediators as key conceptual elements
  - Different kinds of mediators:
    - OO Mediators for ensuring semantic interoperability
    - GG, WG mediators to link Goals and Web Services
    - WW Mediators to establish service interoperability
  - Reusable mediators
  - Mediation techniques under development

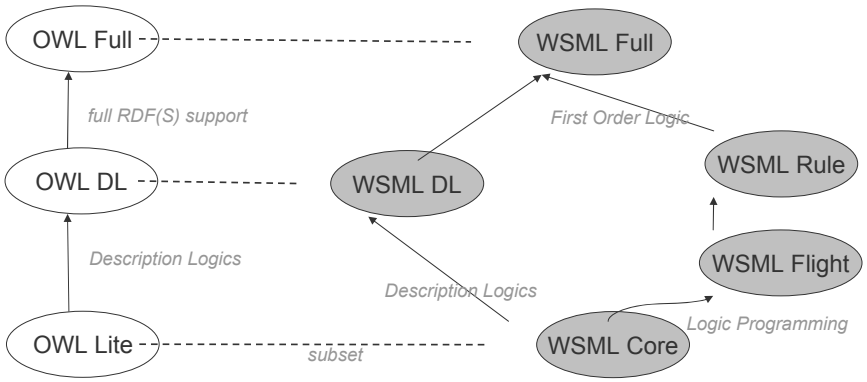
Semantic Representation

- OWL-S and WSMO adopt a similar view on the need of ontologies and explicit semantics

  but they rely on different logics

  - OWL-S is based on OWL/SWRL
    - OWL represent taxonomical knowledge
    - SWRL provides inference rules
    - FLOWS as formal model for process model
  - WSMO is based on
    - WSML a family of languages with a common basis for compatibility and extensions in the direction of Description Logics and Logic Programming
    - Ontologizes Abstract State Machines and formal model for Service Interface Descriptions

OWL vs WSML



Summary

	OWL-S	**WSMO**	***current Web Service technologies***
Discovery ***detection of suitable WS***	Profile	Goals and Web Services (capability)	*UDDI API*
Consumption & Interaction ***How to consume & aggregate***	Process Model	Service Interfaces (Choreography + Orchestration)	*BPEL4WS / WS-CDL*
Invocation ***How to invoke***	Grounding+ WSDL/SOAP	Grounding (WSDL / SOAP, ontology-based)	*WSDL / SOAP*
Mediation ***Heterogeneity handling***	-	Mediators	-

PART III: Semantic Web Service Techniques and Systems

Michael Stollberg

Contents

- The “Virtual Travel Agency Example”
  - Goal and Web service description
  - discovery
  - mediation
- SWS tools and systems
  - Web Service Execution Environment WSMX
  - OWL-S Integrated Development Environment
  - IRS

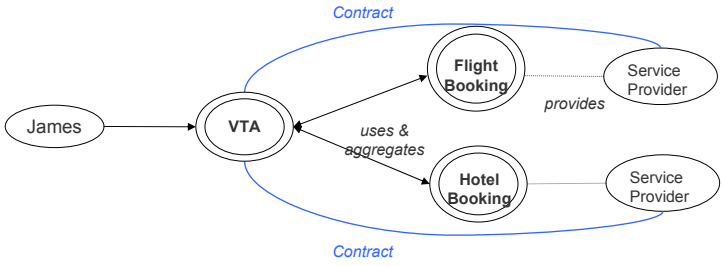
Challenges

- Web services as loosely coupled components that shall interoperate dynamically and automatically
- Techniques required for:
  - **Discovery**
    - How are Web services found and selected?
  - **Composition**
    - How to aggregate Web Services into a complex functionality?
  - **Conversation**
    - How to ensure automated interaction of Web Services?
  - **Invocation**
    - How to access and invoke Semantic Web Services?
  - **Mediation and Interoperability**
    - How are data and protocol mismatches resolved?
- Integrated systems for automated Web service usage :
  - **Editing and Management**
  - **Execution Control of Functional Components**
  - **APIs and web-based**

Virtual Travel Agency Use Case

- Michael is employed in DERI Austria and wants to book a flight and a hotel for the HICSS-39 conference
- the start-up company VTA provides tourism and business travel services based on Semantic Web Service technology

=> how does the interplay of Michael, VTA, and other Web Services look like?



Domain Ontologies

- All terminology used in resource descriptions are based on ontologies and all information interchanged should be ontology instances
- Domain Ontologies needed for this Use Case:
  - Trip Reservation Ontology,
  - Location Ontology,
  - Date and Time Ontology,
  - Purchase Ontology,

  … possibly more

Trip Reservation Ontology

- defines the terminology for trips (traveling, accomodation, holiday / business travel facilities) and reservations
- provided by community of interest (e.g. Austrian Tourism Association)
- main concepts:
  - TRIP
    - describes a trip (a journey between locations)
    - passenger, origin & destination, means of travel, etc.
  - RESERVATION
    - describes reservations for tickets, accomodation, or complete trips
    - customer, trip, price, payment
  - RESERVATION REQUEST
  - RESERVATION OFFER
  - RESERVATION CONFIRMATION
- uses other ontologies:
  - Location Ontology for origin & destination specification
  - Date and Time Ontology for departure, arrival, duration information
  - Purchase Ontology for payment related aspects

Trip Reservation Ontology

```
namespace {_"http://www.wsmo.org/ontologies/tripReservationOntology",
 dc     _"http://purl.org/dc/elements/1.1#",
 xsd    _"http://www.w3.org/2001/XMLSchema#",
 loc    _“http://www.daml.org/2003/09/factbook/factbook-ont#",
 dt     _"http://www.wsmo.org/ontologies/dateandtime.wsml#",
 po     _"http://www.wsmo.org/ontologies/purchase.wsml#"}

ontology _"http://www.wsmo.org/ontologies/tripReservationOntology#"

nonFunctionalProperties
 dc#title hasValue "Trip Reservation Ontology"
 dc#creator hasValue _”http://www.deri.org”
 dc#description hasValue “domain ontology for travel and accomodation reservation"
 dc#publisher hasValue “Austrian Toursim Association"
 version hasValue "$Revision 1.17 $"
endNonFunctionalProperties

importsOntology   {_"http://www.wsmo.org/ontologies/dateandtime.wsml",
                   _"http://www.wsmo.org/ontologies/purchase.wsml“}

usesMediator {_"http://www.wsmo.org/mediators/owl2wsml.wsml“}
```

Trip Reservation Ontology

```
concept trip
passenger impliesType po#person
origin impliesType loc#location
destination impliesType loc#location
departureDate ofType dt#dateandtime
returnDate ofType dt#dateandtime
meansOfTransport impliesType meansOfTransport
accomodation impliesType accomodation

concept reservation
 nonFunctionalProperties
  dc#description hasValue "reservations for tickets, accomodation, or complete trips“
  dc#relation hasValue reservationItemDef
 endNonFunctionalProperties
customer impliesType po#customer
reservationItem impliesType wsml#true
price impliesType po#price
payment impliesType po#payment
axiom reservationItemDef
 definedBy
  forall{?x, ?y} (?x memberOf reservation[reservationItem hasValue ?y] impliedBy
     (?y memberOf ticket) or (?y memberOf accomodation) or (?y memberOf trip) ).
```

Goal Description

- "book flight and hotel for the HICSS-39 for Michael"
- goal capability postcondition: get a trip reservation for this

```
goal _"http://www.wsmo.org/examples/goals/hicss39"
 importsOntology {_"http://www.wsmo.org/ontologies/tripReservationOntology", …}
 capability
  postcondition
   definedBy
    ?tripReservation memberOf tr#reservation[
      customer hasValue fof#michael,
      reservationItem hasValue ?tripHICSS] and
     ?tripHICSS memberOf tr#trip[
       passenger hasValue fof#michael,
       origin hasValue loc#innsbruck,
       destination hasValue loc#kauai,
       meansOfTransport hasValue ?flight,
       accomodation hasValue ?hotel] and
     ?flight[airline hasValue tr#staralliance] memberOf tr#flight and
     ?hotel[name hasValue "Grand Hyatt Kauai Resort"] memberOf tr#hotel .
```

VTA Service Description

- book tickets, hotels, amenities, etc.
- capability description (pre-state)

```
capability VTAcapability
 sharedVariables {?creditCard, ?initialBalance, ?item, ?passenger}
 precondition
  definedBy
   ?reservationRequest[
       reservationItem hasValue ?item,
       passenger hasValue ?passenger,
       payment hasValue ?creditcard,
     ] memberOf tr#reservationRequest and
    ((?item memberOf tr#trip) or (?item memberOf tr#ticket)) and
    ?creditCard[balance hasValue ?initialBalance] memberOf po#creditCard .

 assumption
  definedBy
   po#validCreditCard(?creditCard) and
   (?creditCard[type hasValue po#visa] or ?creditCard[type hasValue po#mastercard]).
```

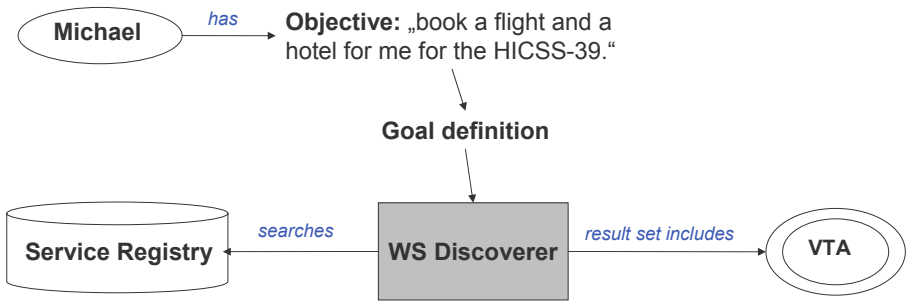
VTA Service Description

- capability description (post-state)

```
postcondition
      definedBy
        ?reservation[
            reservationItem hasValue ?item,
            customer hasValue ?passenger,
            payment hasValue ?creditcard
         ] memberOf tr#reservation .

   assumption
     definedBy
       reservationPrice(?reservation, ?tripPrice) and
        ?finalBalance= (?initialBalance - ?ticketPrice) and
        ?creditCard[po#balance  hasValue  ?finalBalance] .
```


Web Service Discovery



Semantic Web Service Discovery

Find appropriate Web Service for automatically resolving a goal as the objective of a requester

- Aims:
  - high precision discovery
  - maximal automation
  - effective discoverer architectures
- Requirements:
  - infrastructure that allows storage and retrieval of information about Web services
  - description of Web services functionality
  - description of requests or goals
  - algorithms for matching requesters for capabilities with the corresponding providers

Discovery Techniques

- different techniques available
  - trade-off: ease-of-provision <-> accuracy
  - resource descriptions & matchmaking algorithms

Ease of provision

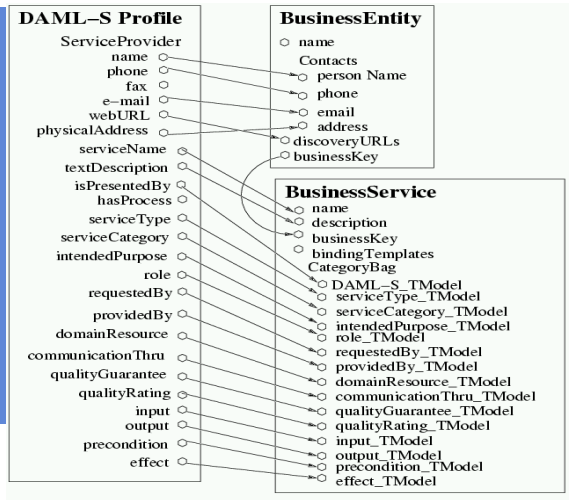
Possible Accuracy

Key Word Matching
match natural language key words in resource descriptions

Controlled Vocabulary
ontology-based key word matching

Semantic Matchmaking
… what Semantic Web Services aim at

Semantic Web Services in UDDI



- Mapping semantic resource descriptions into UDDI
- OWL-S Service Profile mapping to UDDI
- WSMO elements to UDDI mapping (for all top level elements)

⇒ ***mapping semantic descriptions to syntactic repository***

⇒ ***allows retrieval of structural information***

Controlled Vocabulary

WSMO non-functional properties

- Ontology keywords in non-functional properties
  - dc#subject contains main ontology concepts related to Web Service
  - allows pre-filtering similar to OWL-S Profile Hierarchy, but on basis on ontologies (“controlled vocabulary”)
- Example
  - a Web Service for selling train tickets in Austria

    dc#subject **hasValue** _{tc#trainticket, po#purchase, loc#austria}
  - does not precisely describe Web Service functionality

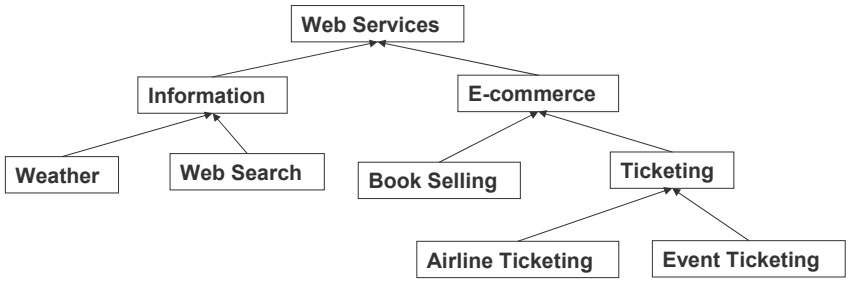
  => accuracy of discovery result meager

Lara, R., Lausen, H.; Toma, I.: (Eds): *WSMX Discovery*. WSMX Working Draft D10 v0.2, 07 March 2005.

Controlled Vocabulary

OWL-S Profile Hierarchies

- Hierarchy of Web Services
  - functional similarities (domain, in- / outputs)
  - allows pre-filtering of services on basis of categorization



http://www.daml.org/services/owl-s/1.0/ProfileHierarchy.owl

Matchmaking Notions & Intentions

Exact Match:

G, WS, O, M ⊨ ∀x. (G(x) <=> WS(x))

PlugIn Match:

G, WS, O, M ⊨ ∀x. (G(x) => WS(x))

Subsumption Match:

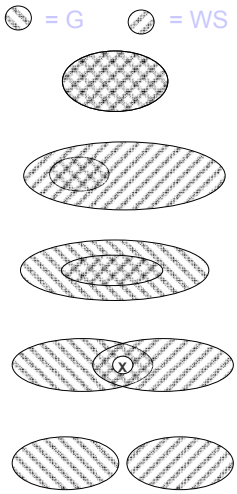
G, WS, O, M ⊨ ∀x. (G(x) <= WS(x))

Intersection Match:

G, WS, O, M ⊨ ∃x. (G(x) ∧ WS(x))

Non Match:

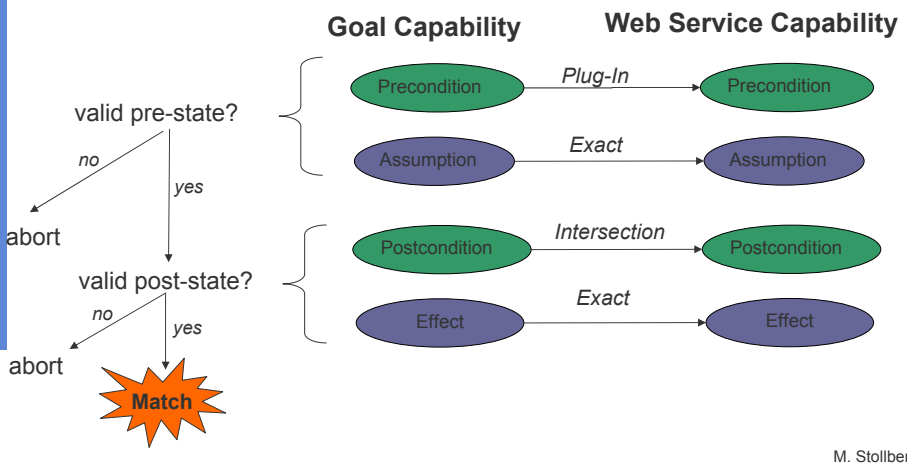
G, WS, O, M ⊨ ¬∃x. (G(x) ∧ WS(x))



Keller, U.; Lara, R.; Polleres, A. (Eds): *WSMO Web Service Discovery*. WSML Working Draft D5.1, 12 Nov 2004.

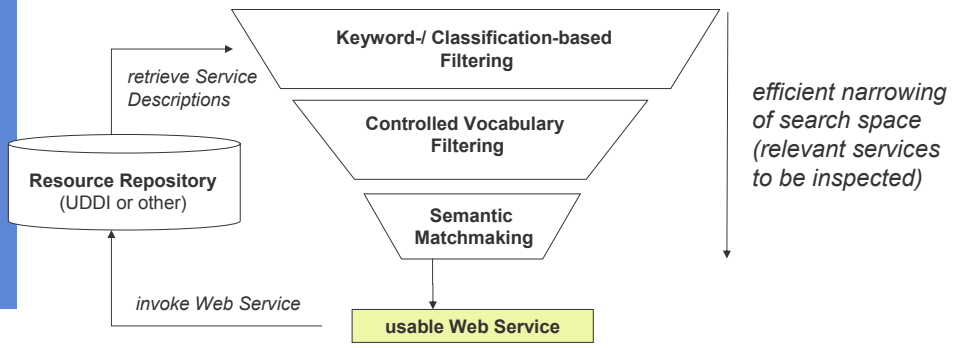
Discovery Approach

- Matchmaking Notion to be used defined for each goal capability element
- **Basic Procedure:**

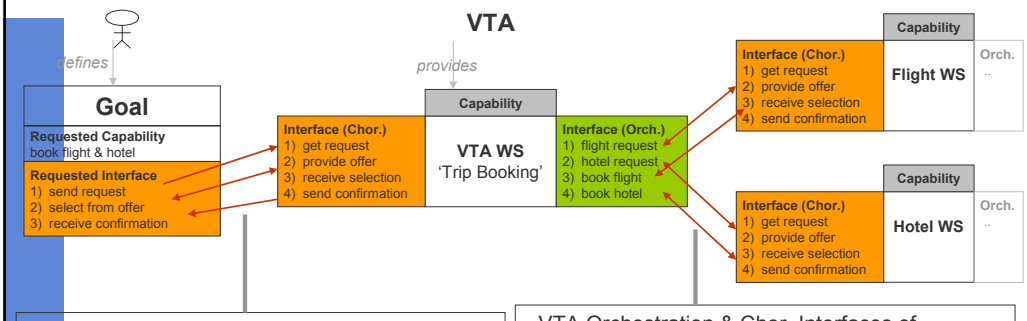


Discoverer Architecture

- Discovery as central Semantic Web Services technology
- Integrated Discoverer Architectures (under construction):



Choreography Discovery

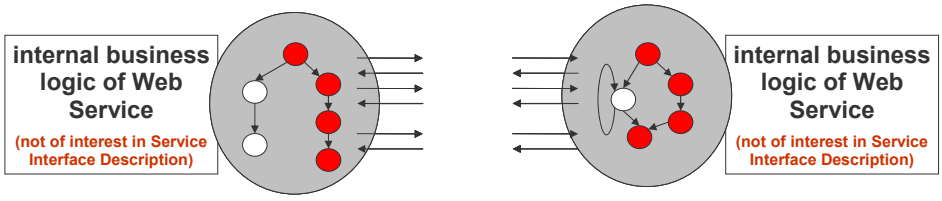


- both choreography interfaces given ("static")
- correct & complete consumption of VTA
=> existence of a valid choreography?

- VTA Orchestration & Chor. Interfaces of aggregated WS given
=> existence of a valid choreography between VTA and each aggregated WS?

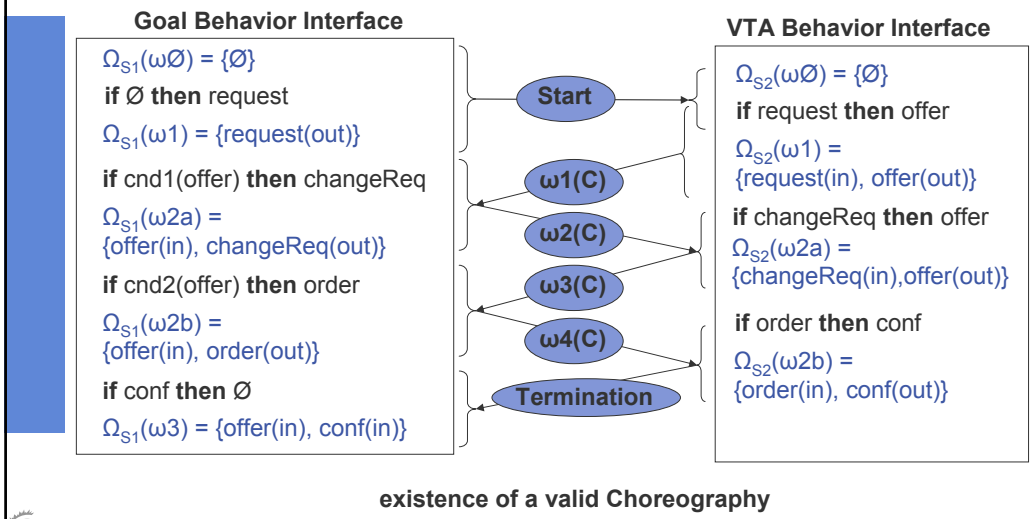
- Choreography Discovery as a central reasoning task in Service Interfaces
- 'choreographies' do not have to be described, only existence determination

Choreography Discovery



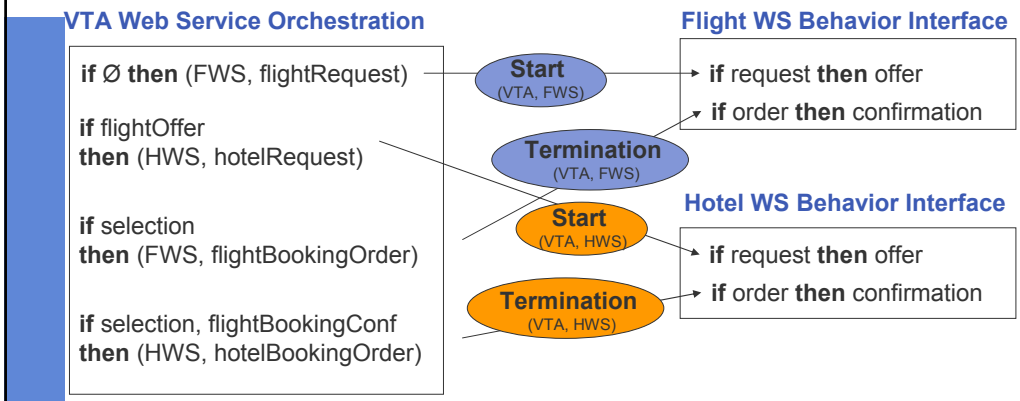
- a valid choreography exists if:
  1) **Information Compatibility**
     - compatible vocabulary
     - homogeneous ontologies
  2) **Communication Compatibility**
     - start state for interaction
     - a termination state can be reached without any additional input

Communication Compatibility Example



existence of a valid Choreography

Orchestration Validation Example



Orchestration is valid if valid choreography exists for interactions between Orchestrator and each aggregated Web Service, done by choreography discovery

Mediation

- Heterogeneity as inherent characteristic of (Semantic) Web:
  - heterogeneous terminology
  - heterogeneous languages / formalisms
  - heterogeneous communication protocols and business processes
- WSMO identifies Mediators as top level element, i.e. central aspect of Semantic Web Services
  - levels of mediation: data, protocol, processes
  - WSMO Mediator types
- Approach: declarative, generic mismatch resolution
  - classification of possible & resolvable mismatches
  - mediation definition language & mediation patterns
  - execution environment for mappings

Data Level (OO) Mediation

- Related Aspects / Techniques:
  - Ontology Integration (Mapping, Merging, Alignment)
  - Data Lifting & Lowering
  - Transformation between Languages / Formalisms
- Data Level Mismatch Classification
  - Conceptualization Mismatches
    - same domain concepts, but different conceptualization
    - different levels of abstraction
    - different ontological structure

    => *resolution only incl. human intervention*
  - Explication Mismatches
    - mismatches between:

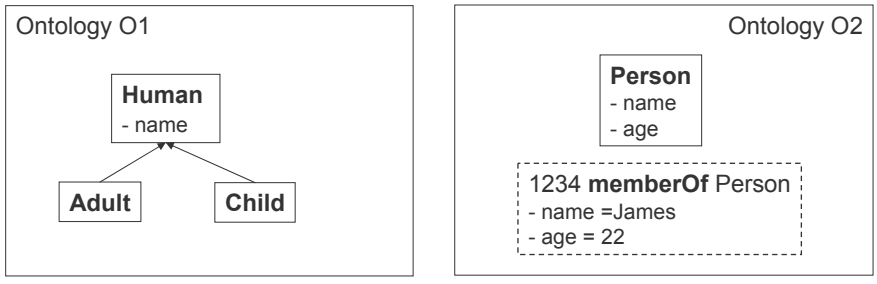
      T (Term used) **D** (definition of concepts), **C** (real world concept)

    => *automated resolution partially possible*

Ontology Mapping Language

- Language Neutral Mapping Language
  - mapping definitions on meta-layer (i.e. on generic ontological contructs)
  - independent of ontology specification langauge
  - “Grounding” to specific langauges for execution (WSML, OWL, F-Logic)
- Main Features:
  - Mapping Document (sources, mappings, mediation service)
  - direction of mapping (uni- / bidirectional)
  - mapping between Ontology Constructs:
    - classMapping, attributeMapping, relationMapping (between similar constructs)
    - classAtrributeMapping, classRelationMapping, classInstanceMapping
    - instanceMapping (explicit ontology instance transformation)
  - Conditions / logical expressions for data type mismatch handling, restriction of mapping validity, and complex mapping definitions
  - Mapping operators:
    - =, <, <=, >, >=, and, or, not
    - inverse, symmetric, transitive, reflexive
    - join, split

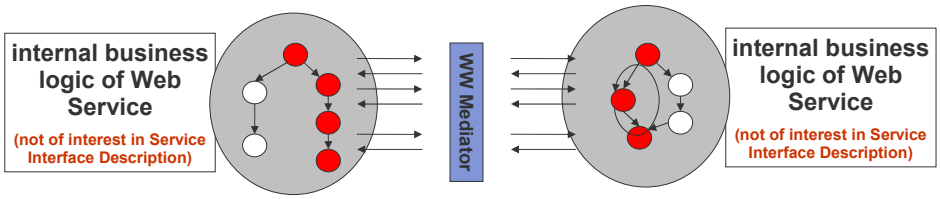
Mapping Language Example



classMapping(**unidirectional** o2:Person o1.Adult
attributeValueCondition(o2.Person.age **>=** 18))

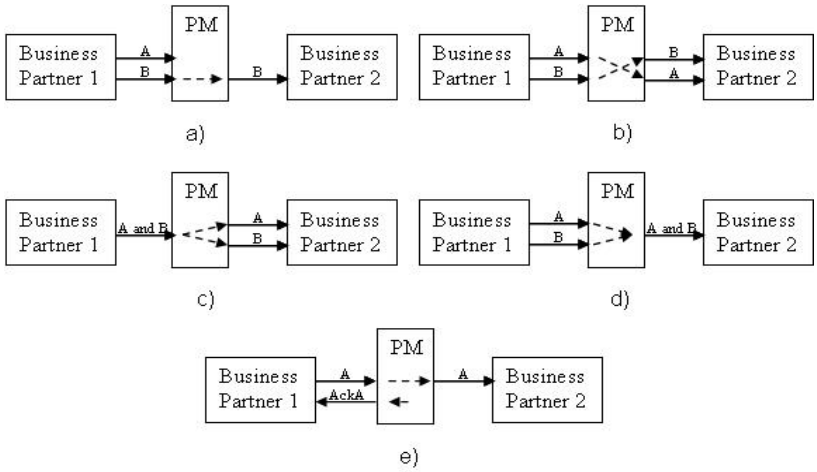
this allows to transform the instance 1234 of ontology O2 into a valid instance of ‘adult’ in ontology O1

Protocol & Process Level Mediation

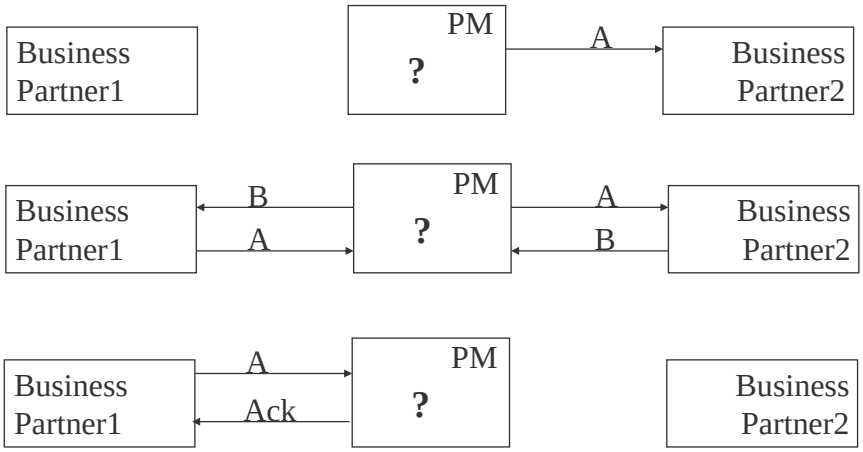


- if a choreography does not exist, then find an appropriate WW Mediator that
  - resolves possible mismatches to establish Information Compatibility (OO Mediator usage)
  - resolves process / protocol level mismatches in to establish Communication Compatibility

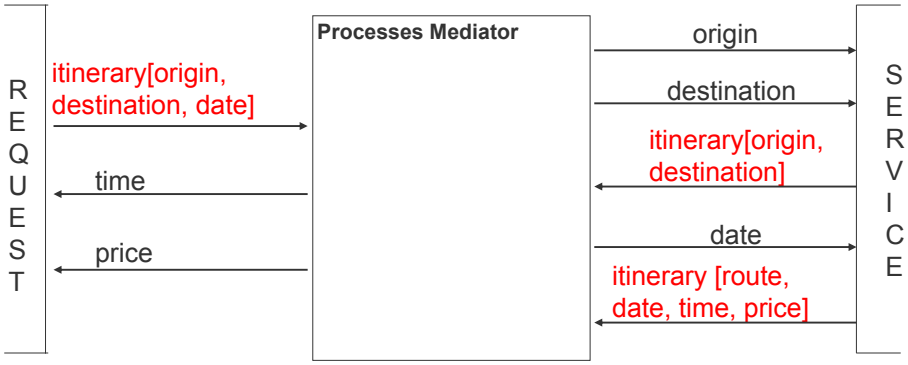
Process Mediation – Addressed Mismatches



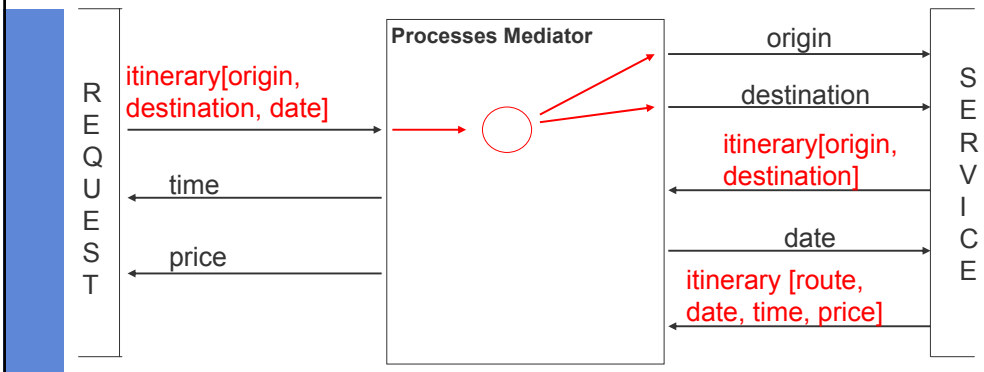
Unsolvable Mismatches



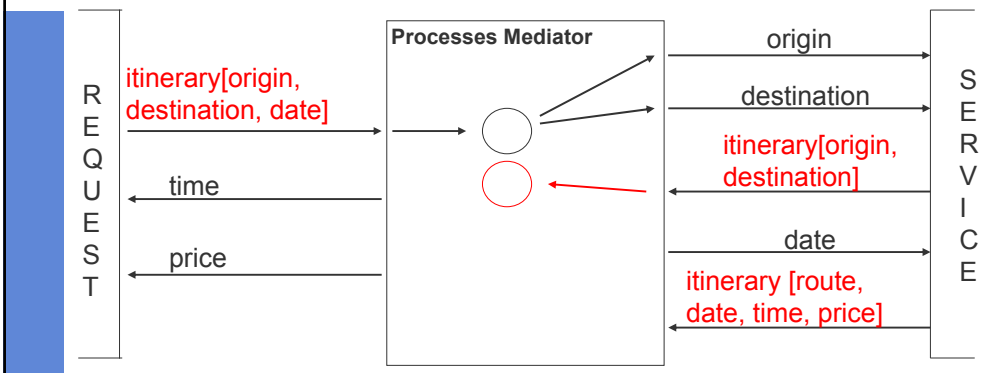
Process Mediation Example



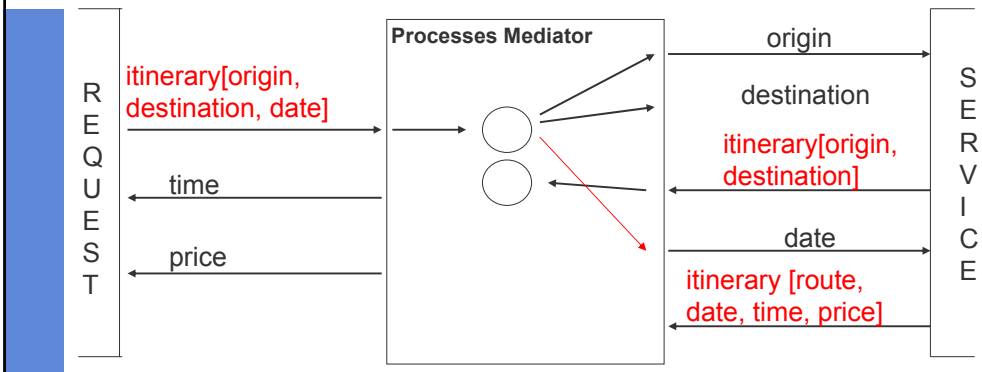
Process Mediation Example



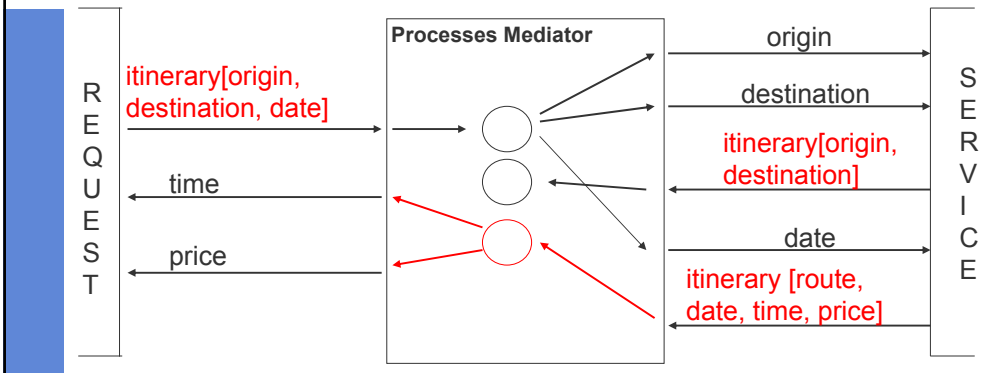
Process Mediation Example



Process Mediation Example



Process Mediation Example



Wrap-up

- The targets of the presented tutorial were to:
  - understand aims & challenges within Semantic Web Services
  - understand OWL-S and WSMO:
    - design principles & paradigms
    - ontology elements
- .. an overview of 'hot topics' within the Semantic Web and Semantic Web Services
- .. OWL-S and WSMO Tools and System Presentation
- .. do-it-yourself Hands-On Session

=> you should now be able to correctly assess emerging technologies & products for Semantic Web Services and utilize these for your future work

OWL-S and WSMO

- North-American and European initiatives with converging aims
- Offer a SWS platforms to be used by B2C and B2B applications
- Provide a backbone for advanced integration and automation of industrial and business processes
- Are the most developed SWS technologies up to now available to be used in commercial and industrial applications
- Developments towards refining and interconnecting them

Future work – OWL-S

- OWL-S is close to conclusion, but a few issues still need to be addressed
  - An exception mechanism is still missing
  - There is a need of an exec instruction for loading and executing Process Models dynamically
  - A new Grounding for WSDL 2 should be developed
- Additional issues that OWL-S does not address
  - Security and Policies are not directly expressed in OWL-S yet
  - There are no facilities for Contracting and agreement
  - There are no facilities for Web service management

Future work – OWL-S (2)

- Standardization
  - The OWL-S coalition is planning to submit a W3C note to draw attention and create momentum for W3C standardization activities on Semantic Web services
  - Members of the OWL-S coalition are already active in standardization committee such as UDDI, WSDL 2 and WS Coordination
- The Future of OWL-S
  - OWL-S is nearing its completion and it will converge in the results of the SWSI working group or future standardization activities
  - The OWL-S coalition plans to remain in existence to maintain and further develop the language if needed

Future work - WSMO

- Further develop and consolidate concepts and implementation aspects of WSMO, WSML and WSMX
  - Choreography and orchestration
  - Business process execution
  - Web services composition
  - Process and protocol mediation
- Standardization …
- WSMO & WSMX – applied in several case studies within EU funded projects
- WSMO Studio development

Beyond OWL-S and WSMO

- Although OWL-S and WSMO are the main initiatives on Semantic Web services, they are not the only activities
- Semantic Web Services Interest Group
  - Interest group founded at W3C to discuss issues related to Semantic Web Services (http://www.w3.org/2002/ws/swsig/)
- SWSI: International initiative to push toward a standardization of SWS (http://www.swsi.org)
- WSDL-S: Semantic Annotation of WSDL interfaces
- Semantic Web services are entering standardization
  - W3C working groups currently starting
  - OASIS working groups currently starting

  => eventually major influence on next generation Web technology

References OWL-S

- The main repository of papers on OWL-S is at http://www.daml.org/services/owl-s/pub-archive.html that contains many papers produced by the coalition as well as from the community at large
- The main source of information on OWL-S is the Web site http://www.daml.org/services/owl-s
- The rest of this section will report what we believe to be the most influential papers on OWL-S as well as paper referred in this tutorial

References OWL-S

- Fundamental

David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, Katia Sycara, "Bringing Semantics to Web Services: The OWL-S Approach", *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)*, July 6-9, 2004, San Diego, California, USA.

The DAML Services Coalition (alphabetically Anupriya Ankolenkar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David L. Martin, Drew McDermott, Sheila A. McIlraith, Srini Narayanan, Massimo Paolucci, Terry R. Payne and Katia Sycara), "DAML-S: Web Service Description for the Semantic Web", *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia (Italy), June, 2002.

DAML Services Coalition (alphabetically A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, H. Zeng), "DAML-S: Semantic Markup for Web Services", in *Proceedings of the International Semantic Web Working Symposium (SWWS)*, July 30-August 1, 2001.

References OWL-S

- Discovery

Lei Li and Ian Horrocks. **A software framework for matchmaking based on semantic web technology**. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003), 2003*

B. Benatallah, M. Hacid, C. Rey, F. Toumani **Towards Semantic Reasoning for Web Services Discovery**,. *In Proc. of the International Semantic Web Conference (ISWC 2003), 2003*

Daniel J. Mandell and Sheila A. McIlraith. **Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation.** In *Proceedings of the Second International Semantic Web Conference (ISWC2003),*

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara; **Importing the Semantic Web in UDDI**. In *Proceedings of Web Services, E-business and Semantic Web Workshop, 2002*

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara; "**Semantic Matching of Web Services Capabilities**." In *Proceedings of the 1st International Semantic Web Conference* (ISWC2002), 2002

References OWL-S

- Composition and Invocation

Evren Sirin, Bijan Parsia, Dan Wu, James Hendler, and Dana Nau. **HTN planning for web service composition using SHOP2**. In *Journal of Web Semantics*, To appear, 2004

Katia Sycara, Massimo Paolucci, Anupriya Ankolekar and Naveen Srinivasan, "**Automated Discovery, Interaction and Composition of Semantic Web services**," *Journal of Web Semantics*, Volume 1, Issue 1, September 2003, pp. 27-46

Massimo Paolucci, Anupriya Ankolekar, Naveen Srinivasan and Katia Sycara, "The DAML-S Virtual Machine," In *Proceedings of the Second International Semantic Web Conference (ISWC)*, 2003,

Srini Narayanan and Sheila McIlraith ``**Analysis and Simulation of Web Services"** *Computer Networks,* 42 (2003), 675-693, Elsevier Science, 2003

References OWL-S

Formal Models and Verification

Anupriya Ankolekar, Massimo Paolucci, and Katia Sycara
Spinning the OWL-S Process Model -- Toward the Verification of the OWL-S Process Models In Proceedings of Workshop on Semantic Web Services:
Preparing to Meet the World of Business Applications (ISWC 2004)

Narayanan, S. and McIlraith, S. ``**Simulation, Verification and Automated Composition of Web Services"**. *IN the Proceedings of the Eleventh International World Wide Web Conference (WWW-11)*, May, 2002

Anupriya Ankolekar, Frank Huch and Katia Sycara. "**Concurrent Semantics for the Web Services Specification Language DAML-S**." In *Proceedings of the Fifth International Conference on Coordination Models and Languages*, York, UK, April 8-11, 2002.

Anupriya Ankolekar, Frank Huch, Katia Sycara. "**Concurrent Execution Semantics for DAML-S with Subtypes**." In *The First International Semantic Web Conference (ISWC)*, 2002.

References OWL-S

Policies and Security

Ronald Ashri, Grit Denker, Darren Marvin, Mike Surridge,Terry Payne, **Semantic Web Service Interaction Protocols: An Ontological Approach,** 3rd International Semantic Web Conference (ISWC2004), Hiroshima, Japan

Lalana Kagal, Grit Denker, Tim Finin, Massimo Paolucci, Naveen Srinivasan and Katia Sycara, "**An Approach to Confidentiality and Integrity for OWL-S**", forthcoming in *Proceedings of AAAI 2004 Spring Symposium*.

Grit Denker, Lalana Kagal, Tim Finin, Massimo Paolucci, Naveen Srinivasan and Katia Sycara, "**Security For DAML Web Services: Annotation and Matchmaking**" In *Proceedings of the Second International Semantic Web Conference (ISWC 2003)*, Sandial Island, FI, USA, October 2003, pp 335-350.

References OWL-S

- Applications

Schlenoff, C., Barbera, A., Washington, R., "**Experiences in Developing an Intelligent Ground Vehicle (IGV) Ontology in Protégé**" In Proceedings of the 7th International Protege Conference, Bethesda, MD, July 6 - 8, 2004.

Aabhas V Paliwal, Nabil Adam, Christof Bornhövd, and Joachim Schaper **Semantic Discovery and Composition of Web Services for RFID Applications in Border Control** In Proceedings of Workshop on Semantic Web Services:
Preparing to Meet the World of Business Applications (ISWC 2004)

Mithun Sheshagiri, Norman Sadeh and Fabien Gandon, **Using Semantic Web Services for Context-Aware Mobile Applications**, Proceedings of MobiSys2004 Workshop on Context Awareness, Boston, June 2004

Zhexuan Song, Yannis Labrou and Ryusuke Masuoka, **"Dynamic Service Discovery and Management in Task Computing,"** pp. 310 - 318, MobiQuitous 2004, August 22-26, 2004, Boston, USA

References WSMO

- The central location where WSMO work and papers can be found is WSMO Working Group: http://www.wsmo.org
- WSMO languages – WSML Working Group: http://www.wsml.org
- WSMO implementation
  - WSMX working group : http://www.wsmx.org
  - WSMX open source can be found at: https://sourceforge.net/projects/wsmx/

References WSMO

- [WSMO Specification]: Roman, D.; Lausen, H.; Keller, U. (eds.): ***Web Service Modeling Ontology***, WSMO Working Draft D2, final version 1.2, 13 April 2005.
- [WSMO Primer]: Feier, C. (ed.): ***WSMO Primer***, WSMO Working Draft D3.1, 18 February 2005.
- [WSMO Choreography and Orchestration] Roman, D.; Scicluna, J., Feier, C. (eds.): ***Ontology-based Choreography and Orchestration of WSMO Services***, WSMO Working Draft D14, 01 March 2005.
- [WSMO Use Case] Stollberg, M.; Lausen, H.; Polleres, A.; Lara, R. (ed.): ***WSMO Use Case Modeling and Testing***, WSMO Working Drafts D3.2; D3.3.; D3.4; D3.5, 05 November 2004.
- [WSML] de Bruijn, J. (Ed.): ***The WSML Specification***, WSML Working Draft D16, 03 February 2005.

References WSMO

- [Arroyo et al. 2004] Arroyo, S., Lara, R., Gomez, J. M., Berka, D., Ding, Y. and Fensel, D: "Semantic Aspects of Web Services" in Practical Handbook of Internet Computing. Munindar P. Singh, editor. Chapman Hall and CRC Press, Baton Rouge. 2004.
- [Berners-Lee et al. 2001] Tim Berners-Lee, James Hendler, and Ora Lassila, “The Semantic Web”. Scientific American, 284(5):34-43, 2001.
- [Chen et al., 1993] Chen, W., Kifer, M., and Warren, D. S. (1993). HILOG: A foundation for higher-order logic programming. Journal of Logic Programming, 15(3):187-230.
- Domingue, J. Cabral, L., Hakimpour, F., Sell D., and Motta, E., (2004) IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services WSMO Implementation Workshop (WIW), Frankfurt, Germany, September,2004
- [Fensel, 2001] Dieter Fensel, “Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce”, Springer-Verlag, Berlin, 2001.

References WSMO

- [Gruber, 1993] Thomas R. Gruber, “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition, 5:199-220, 1993.
- [Grosof et al., 2003] Grosof, B. N., Horrocks, I., Volz, R., and Decker, S. (2003). Description logic programs: Combining logic programs with description logic. In Proc. Intl. Conf. on the World Wide Web (WWW-2003), Budapest, Hungary.
- [Kifer et al., 1995] Kifer, M., Lausen, G., and Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. JACM, 42(4):741-843.
- [Pan and Horrocks, 2004] Pan, J. Z. and Horrocks, I. (2004). OWL-E: Extending OWL with expressive datatype expressions. IMG Technical Report IMG/2004/KR-SW-01/v1.0, Victoria University of Manchester. Available from http://dl-web.man.ac.uk/Doc/IMGTR-OWL-E.pdf.
- [Stencil Group] - www.stencilgroup.com/ideas_scope_200106wsdefined.html

References Discovery

B. Benatallah, M. Hacid, C. Rey, F. Toumani **Towards Semantic Reasoning for Web Services Discovery**,. *In Proc. of the International Semantic Web Conference (ISWC 2003), 2003*

Herzog, R.; Lausen, H.; Roman, D.; Zugmann, P.: **WSMO Registry.** WSMO Working Draft D10 v0.1, 26 April 2004.

Keller, U.; Lara, R.; Polleres, A. (Eds): **WSMO Web Service Discovery**. WSML Working Draft D5.1, 12 Nov 2004.

Keller, U.; Lara, R.; Lausen, H.; Polleres, A.; Fensel, D.: **Automatic Location of Services**. In Proc. of the 2nd European Semantic Web Symposium (ESWS2005), Heraklion, Crete, 2005.

M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen and D. Fensel: **A Logical Framework for Web Service Discovery.** Proc. 1st. Intl. Workshop SWS'2004 at ISWC 2004,Hiroshima, Japan, November 8, 2004, CEUR Workshop Proceedings, ISSN 1613-0073

Lara, R., Lausen, H.; Toma, I.: (Eds): **WSMX Discovery**. WSMX Working Draft D10 v0.2, 07 March 2005.

Lei Li and Ian Horrocks. **A software framework for matchmaking based on semantic web technology**. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003), 2003.*

References Discovery

Lei Li and Ian Horrocks. **A software framework for matchmaking based on semantic web technology**. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003), 2003*

Daniel J. Mandell and Sheila A. McIlraith. **Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation.** In *Proceedings of the Second International Semantic Web Conference (ISWC2003),*

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara; **Importing the Semantic Web in UDDI**. In *Proceedings of Web Services, E-business and Semantic Web Workshop, 2002*

Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara; "**Semantic Matching of Web Services Capabilities**." In *Proceedings of the 1st International Semantic Web Conference* (ISWC2002), 2002

Preist, C.: **A Conceptual Architecture for Semantic Web Services**. In Proceedings of the 3rd International Semantic Web Conference (ISWC 2004), 2004, pp. 395 - 409.

Stollberg, M.; Keller, U.; Fensel. D.: Partner and Service Discovery for Collaboration on the Semantic Web. Proc. 3rd Intl. Conference on Web Services (ICWS 2005), Orlando, Florida, July 2005.

Acknowledgements

The development of **OWL-S** has been funded almost exclusively by the **DAML DARPA** program.

The **WSMO working groups** are funded by the European Commission under the projects **DIP**, **Knowledge Web**, **SEKT**, **SWWS**, and **ASG**; by **Science Foundation Ireland** under the **DERI-Lion** project; and by the Vienna city government under the **FIT-IT Programme** in the projects **RW²** and **TCP**.

IRS development is funded by the European Commission under the **DIP** project, and formerly **IBROW**, and by the UK EPSRC under the **AKT** project, and formerly **MIAKT**