Semantic Web Processes

Jorge Cardoso¹, Amit Sheth²,³
¹University of Madeira
²LSDIS Lab, Computer Science, University of Georgia
³Semagix, Inc

September 22 to 25, 2003
Thuringia, Germany

Our Focus (1)

- Web services and their composition into Web Processes promise to power eCommerce and eServices
- Supporting Web Processes on multi-enterprise and Web scale require addressing heterogeneity/Integration, availability, dynamic change and performance challenges
- Semantics is seen as the key enabler to address these challenges; Semantic Web Processes build upon Web Services and Semantic Web technologies
- This tutorial is about adding semantics to Web Services, and exploiting them in Web Process Lifecycle (Specification, Discovery, Composition, Execution)
  - Functional perspective takes form of process composition involving Web Service Discovery, addressing semantic heterogeneity handling
  - Operational perspective takes form of the research on QoS Specification for Web Services and Processes.

Our Focus (2)

Web Process Composition
Web Process QoS
Web Service Annotation
Web Service Discovery
Web Service QoS

The Basics

What are Web Services, Web Processes, and Semantics?

Web Services: Definition

"Web services are a new breed of Web application. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. … Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service."

IBM web service tutorial

Services Aspect of Web Services

- Modular: Components are reusable and it is possible to compose them into larger components.
  - Available: Services must be exposed outside of the particular paradigm or system they are available in. Business services can be completely decentralized and distributed over the Internet. The dynamic enterprise and dynamic value chains become achievable and may be even mandatory.
- Described: Services have a machine-readable description that can be used to identify the interface of the service.
- Implementation-independent: The service interface is independent of the ultimate implementation.
- Published: Service descriptions are made available in a repository where users can find the service and use the description to access the service.
Why Web Services?

Enterprise Java Beans
Microsoft DCOM
Sun ONC/RPC (Open Network Computing)

Web Services
UDI
WSDL
Jini
RMI (Remote Method Invocation)
Sun ONC/RPC (Open Network Computing)

Why Web Services?

IP, UDP, TCP

<table>
<thead>
<tr>
<th>Feature</th>
<th>CORBA</th>
<th>Web Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Model</td>
<td>Object Model</td>
<td>SOAP Message exchange model</td>
</tr>
<tr>
<td>Client Server Coupling</td>
<td>Tight Coupling</td>
<td>Loose Coupling</td>
</tr>
<tr>
<td>Parameter Passing</td>
<td>Pass by reference/value</td>
<td>Pass by value only</td>
</tr>
<tr>
<td>Type Checking</td>
<td>Static + Runtime type checking only</td>
<td>RunTime type checking only</td>
</tr>
<tr>
<td>State</td>
<td>Stateless, Unrelated (Web Services)</td>
<td>Related (Web Processes)</td>
</tr>
<tr>
<td>Firewall Traversal</td>
<td>Work in Progress</td>
<td>Uses HTTP port 80</td>
</tr>
<tr>
<td>Service Discovery</td>
<td>CORBA-embedding Service</td>
<td>UDDI</td>
</tr>
<tr>
<td>Communication Mode</td>
<td>1-way, 2-way sync</td>
<td>2-way sync (Web Services)</td>
</tr>
</tbody>
</table>

What are Web Processes ? (1)?

- **Web Processes** are next generation workflow technology to facilitate the interaction of organizations with markets, competitors, suppliers, customers etc. supporting enterprise-level and core business activities
  - encompass the ideas of both intra and inter organizational workflow.
  - created from the composition of Web services
- When all the tasks involved in a Web process are semantically described, we may call such process as **Semantic Web Processes**

What are Web Processes ? (2)?

- Web processes describe how Web services are connected to create reliable and dependable business solutions
- Web processes allow businesses to describe sophisticated processes that can both consume and provide Web services
- The role of Web processes within the enterprise is to simplify the integration of business and application processes across technological and corporate domains

Web Process
An Example

- Graphical example of a web process
Web Processes Composition

Architectures for Web Processes*

Stages of architectural evolution
- Process Portal
  - One stop for e-services, p2p interactions between buyer and sellers
  - E-Gov, industry automation, Life Science
- Process Vortex
  - Interactions between buyer and seller through a third party marketmaker, predefined processes, shared ontology
- Dynamically Trading Processes

* From Sheth, Aalst, Arpinar, "Processes driving the Networked Economy" 1999

Globalization of Processes

BIG Challenges

- Heterogeneity and Autonomy
  - Syntactic, semantic and pragmatic
  - Complex rules/regulations related to B2B and e-commerce interactions
  - Solution: Machine processable descriptions
- Dynamic nature of business interactions
  - Needs: Efficient Discovery, Composition, etc.
- Scalability (Enterprises → Web)
  - Needs: Automated service discovery/selection and composition

Proposition: Semantics is the most important enabler to address these challenges

What are Semantics and Ontologies?

- An ontology includes a vocabulary of terms, and some specification of their meaning.
- The goal is to create an agreed-upon vocabulary and semantic structure for exchanging information about that domain.

Roadmap

Annotation of Web Services
Web Process Composition
Semantic Web
Web Processes Quality of Service
Web Service Discovery
Semantics for Web Processes

- **Data/Information Semantics**
  - **What:** Formal definition of data in input and output messages of a web service
  - **Why:** for discovery and interoperability
  - **How:** by annotating input/output data of web services using ontologies

- **Functional/Operational Semantics**
  - Formally representing capabilities of web service for discovery and composition of Web Services by annotating operations of Web Services as well as provide preconditions and effects; Annotating TPA/SLA (future work)

- **Execution Semantics**
  - Formally representing the execution or flow of a services in a process or operations in a service for analysis (verification), validation (simulation) and execution (exception handling) of the process models using State Machines, Petri nets, activity diagrams etc.

- **QoS Semantics**
  - Formally describing operational metrics of a web service/process to select the most suitable service to carry out an activity in a process using QoS model [Cardoso and Sheth, 2002] for web services
Semantics for Web Process Life-Cycle

Web Processes Architecture

Web Process Architecture

Web Service Semantic Annotation

Web Services

WSDL

- WSDL defines services as collections of network endpoints or ports. A port is defined by associating a network address with a binding; a collection of ports define a service.
- SOAP is a message layout specification that defines a uniform way of passing XML-encoded data. It also defines a way to bind to HTTP as the underlying communication protocol. SOAP is basically a technology to allow for "RPC over the web".
- XML was designed to describe data and to focus on what data is.
To enhance the discovery, composition, and orchestration of Web services, it is necessary to increase the description of their interfaces.

One solution is to annotate WSDL interfaces with semantic metadata based on relevant ontologies.

An ontology is a specification of a representational vocabulary for a shared domain of discourse.

**Semantic Annotation of Web Services**

**How to Annotate?**

- Map Web service’s input & output data as well as functional description using relevant data and function/operation ontologies, respectively

- **How?**
  - Borrow from schema matching
  - Semantic disambiguation between terms in XML messages represented in WSDL and concepts in ontology

**Types of Annotation**

- **Functional Semantics**
- **Data Semantics**
- **QoS Semantics**

**Web Services Interfaces**

- A Web service (WS) invocation specifies:
  - The number of input parameters that must be supplied for a proper WS realization and
  - The number of outputs parameters to hold and transfer the results of the WS realization to other tasks.
  - A function to invoke function_foo(x..y)
Adding Semantics to Web Services

Semantics

Web Service Architecture

Semantic Brokering
Specialized brokering services to find Web services

Semantic Discovery
Discovery algorithms that account for semantic information

Semantic Registries
Describe Web services in UDDI registries using semantic concepts

SOAP

- SOAP is an XML Messaging Protocol that allows software running on disparate operating systems, running in different environments to make procedure calls.

- Today's applications communicate using Remote Procedure Calls (RPC) between objects like DCOM and CORBA.
- RPC represents a compatibility and security problem; firewalls and proxy servers will normally block this kind of traffic.
- A better way to communicate between applications is over HTTP, because HTTP is supported by all Internet browsers and servers. SOAP was created to accomplish this.

UDDI

- UDDI stands for Universal Description, Discovery and Integration.
- UDDI serves as a "Business and services" registry and directory and are essential for dynamic usage of Web services.
- A UDDI registry is similar to a CORBA trader, or it can be thought of as a DNS for business applications.
- Is a platform-independent framework for describing services, discovering businesses, and integrating business services by using the Internet.
How UDDI Works?

1. SW companies, standards bodies, and programmers populate the registry with descriptions of different types of services.
2. Businesses populate the registry with descriptions of the services they support.
3. UBR assigns a programmatically unique identifier to each service and business registration.
4. Marketplaces, search engines, and business apps query the registry to discover services at other companies.
5. Business uses this data to facilitate easier integration with each other over the Web.

UDDI and Semantics

- Marketplaces, search engines, and business apps query.
- Semantic UDDI.
- Registry entry.
- Functional Semantics.
- Data Semantics.
- QoS Semantics.
- Web Services.
- Web Service Discovery.
- Selection.

Semantic Discovery of Web Services

Web Services must be located (Discovery) that might contain the desired functionality, operational metrics, and interfaces needed to carry out the realization of a given task.

Discovery New Requirements

- Before.
- Now.
- Tasks.
- Workflow.
- Web Services.
- QoS.
- Search.
- Results.

State of the art in discovery

- UDDI Business Registry.
- Provides non-semantic search.
- Keyword and attribute-based search.
- Which service to select?
- How to select?
- Results.
Present Discovery Mechanism
Keyword and attribute-based search

Web Service Discovery

- UDDI: Keyword and attribute-based search
  - Example: "Quote"
    - Microsoft UBR returned 12 services
    - Human reading of description (Natural Language) help me understand:
      - 6 Entries are to get Famous Quotes
      - 1 Entry for personal auto and homeowners quoting
      - 1 Entry for multiple supplier quotes on all building materials
    - Categorization suggested for UDDI is useful but inadequate (what does the WS do?)
      - 1 Entry for Automobile Manufacturing
      - 1 Entry for Insurance agents, brokers, & service
    - Alternatively read and try to understand WSDL
      - 1 Entry related to security details (Human Understanding)
      - 1 Test Web service for Quotes (which quote?)

Search for services to book an air ticket (using categories)*

- unspsc-org: unspsc:3-1
  - Travel, Food, Lodging and Entertainment Services
    - Travel facilitation
      - Travel agents
      - Travel agencies
    - Hotels: reservations
      - Reservations for hotels in Asia, Australia and New Zealand
      - Your Vacation Specialist Web enabled vacation information
  - Services: 3 records found.
    - Airlines
      - Return air fare from netviagens.com travel agent
  - Providers: 2 records found.

Search for services to book an air ticket (using Keywords)*

- air ticket
  - 1 record with name air tickets booking
- air ticket booking, airtravel, air travel, travel agent, airticketbooking, air ticket booking, travel agency, travel agency
  - 0 records were returned
- travelagent
  - 1 record with name travelagent test
- travelagent test
  - 4 services: BookFlight, cancelFlightBooking etc.
  - Descriptions say that both these services are "XML based Web services"
  - No URL for WSDL
  - Travel
  - 15 records. Purpose/functionality understood from descriptions
    - 2 services: TravelBooks
    - 4 services: TravelInformation
    - 2 services: Reservation and cancellation of travel tickets
    - 1 service: Emergency Services for travellers
    - 1 service: Travel documentation and itinerary
    - 5 services: Description is ambiguous/not present

Web Services Semantic description

- The semantic description of Web services allows
  - To better advertise and subsequently discover Web services
  - And supply a better solution for the selection, composition and interoperation of Web services.

The use of semantics
Benefits Web Service Discovery

- Search engines can better “understand” the contents of a particular page
- More accurate searches
- Additional information aids precision
- Makes it possible to automate searches because less manual “weeding” is needed to process the search results
- Facilitates the integration of several Web services

Semantic Discovery: Overview

- Annotation and Publication
  - WSDL file is annotated using ontologies and the annotations are captured in UDDI

- Discovery
  - Requirements are captured as templates that are constructed using ontologies and semantic matching is done against UDDI entries
    - Functionality of the template, its inputs, outputs, preconditions and effects are represented using ontologies

- Use of ontologies
  - brings service provider and service requestor to a common conceptual space
  - helps in semantic matching of requirements and specifications
Use of ontologies enables shared understanding between the service provider and service requestor.

Adding Semantics to Web Services Standards

For simplicity of depicting, the ontology is shown with classes for both operation and data.

Semantic Publication and Discovery

Semantic Discovery

Syntactic, QoS, and Semantic (Functional & Data) Similarity

Semantic Discovery

Semantic Discovery

Brokering

Semantic Discovery

Discovery in Semantic Web Using Semantics

Semantic Discovery

Syntactic, QoS, and Semantic (Functional & Data) Similarity

Semantic Discovery

Semantic Discovery

Semantic Discovery

Semantic Discovery

Semantic Discovery

Semantic Discovery
### Semantic Brokering Issues
- Structured and non-structured sources
- Read-only
- Transparency
  - Location, schema, language, and ontologies
- Global schema
  - Support for semantic schema integration
- Query models
  - Semantic-based, rule-based, SQL-like, etc
- Semantic Mediators
  - Semantic query analysis and query processing
  - Use wrappers

### Brokering and Semantics
- Find Web services across several UDDIs
- Specialized and optimized brokers for specific domain search
  - Transports, Finances, Education, etc.
- Allow the interpretation of complex requirements
  - Domain semantics
  - Functional semantics
  - Data semantics
  - QoS semantics

### Web Process Architecture
**Semantic Composition**
- Semantic algorithms to compute degree Web services integration

**Web Service Discovery**
- Once the desired Web Services have been found (Discovery), mechanisms are needed to facilitate the resolution of structural and semantic differences (integration)

**Web Service Integration**
- This is because the heterogeneous Web services found in the first step need to interoperate with other components present in a process host

### Semantic Process Composition
**Composition** is the task of combining and linking existing Web Services and other components to create new processes.

**Types of Composition**
- **Static Composition** - services to be composed are decided at design time
- **Dynamic Composition** - services to be composed are decided at run-time

SCET, Semantic Web Process Composition

### Composition of Web Processes
**Web Process Composition**

**Flow Layer**
- Why:
  - Design (composition), analysis (verification), validation (simulation) and execution (exception handling) of the process models
  - To employ mediator architectures for automated composition, control flow and data flow based on requirements
  - To employ user interface to capture template requirements and generate template based on that
- How:
  - Using
    - Functionality/conditions/effects of the participating services
    - Knowledge of conversation patterns supported by the service
    - Mathematical models like process algebra, concurrency formalisms like State Machines, Petri nets, etc.
    - Simulation techniques
- Present Scenario:
  - Composition of Web services is static
  - Dynamic service discovery, run-time binding, analysis and simulation are not supported directly

**Publication**

**Network**
Integration
New Requirements

- When Web services are put together
  - Their interfaces need to interoperate.
  - Structural and semantic heterogeneity need to be resolved*.

- Structural heterogeneity exists because Web services use different data structures and class hierarchies to define the parameters of their interfaces.

- Semantic heterogeneity considers the intended meaning of the terms employed in labeling interface parameters. The data that is interchanged among Web services has to be understood.

* Kashyap and Sheth 1996

Web Services Interfaces

- To enhance the integration, Web services need to have their inputs and outputs associated with ontological concepts (annotation).
- This will facilitate the resolution of structural and semantic heterogeneities.
- Compute the optimal matching (Bondy and Murty, 1976) using semantic information (Cardoso and Sheth, 2002).

Web Service Composition Issues

- Representation of an Abstract Web Process
  - Representing/specified the abstract process in a proper form

- Discovery and Interoperability of Services
  - Need to manually or automatically search for appropriate services
  - The discovered services should interoperate

- Efficiency of a Composed Web Process
  - Need to compose processes which are efficient in terms of performance

- Process Execution
  - Adopting a suitable technique for executing the composed concrete process

- Process Monitoring
  - Using a monitoring technique for run time analysis of the Web process execution

Semantic Web Processes

Questions?

Coffee Break
10 Minutes

NEXT: Composition Languages
NEXT: METEOR-S
Composition Languages

- BPEL4WS
- DAML-S

BPEL4WS

- BPEL4WS - Business Process Execution Language for Web Services is a process modeling language.
  - Developed by IBM, Microsoft, and BEA
  - Version 1.1, 5 May 2003
  - It supersedes XLANG (Microsoft) and WSFL(IBM).
  - It is built on top of WSDL.
    - For descriptions of what services do and how they work, BPEL4WS references port types contained in WSDL documents.

BPEL4WS Introduction

- BPEL4WS was released along with two others specs:
  - WS-Coordination and WS-Transaction*
- WS-Coordination describes how services can make use of pre-defined coordination contexts to subscribe to a particular role in a collaborative activity.
- WS-Transaction provides a framework for incorporating transactional semantics into coordinated activities.

* Fensel and Bussler 2002

BPEL4WS Introduction

- BPEL4WS is a block-structured programming language, allowing recursive blocks but restricting definitions and declarations to the top level.
- The language defines activities as the basic components of a process definition.
- Structured activities prescribe the order in which a collection of activities take place.
  - Ordinary sequential control between activities is provided by sequence, switch, and while.
  - Concurrency and synchronization between activities is provided by flow.
  - Nondeterministic choice based on external events is provided by pick.

Web Services Specification

- DAML-S The service profile ontology describes the functionality of a Web service.

BPEL4WS Introduction

- Process instance-relevant data (containers) can be referred to in routing logic and expressions.
- BPEL4WS defines a mechanism for catching and handling faults similar to common programming languages, like Java.
- One may also define a compensation handler to enable compensatory activities in the event of actions that cannot be explicitly undone.
- BPEL4WS does not support nested process definition.

Let consider the following process.

Let's consider the following process. The WSDL portType offered by the service to its customer.

This section defines the different parties that interact with the business process in the course of processing the order. This section defines the data containers used by the process, providing their definitions in terms of WSDL message types. This section contains fault handlers defining the activities that must be executed in response to faults.

Activities are executed sequentially.

The flow construct provides concurrency and synchronization.

DAML-S provides support for the following elements:

- DAML-S
- DAML (DARPA Agent Markup Language)
- DAML-S: Upper ontology of web services

DAML-S provides support for the following elements:

- Process description.
- Advertisement and discovery of services.
- Selection, composition & interoperation.
- Invocation.
- Execution and monitoring.

DAML-S project home page
DAML-S defines ontologies for the construction of service models:
- Service Profiles
- Process Models
- Service Grounding

Service Profile
An example of Inputs and Outputs

BPEL4WS vs. DAML-S
Comparison

Semantic QoS
Web Processes Quality of Service

Organizations operating in modern markets, such as e-commerce activities, require QoS management.

QoS management is indispensable for organizations striving to achieve a higher degree of competitiveness.
The autonomy of Web services does not allow for designer to identify their operational metrics at design time.

Nevertheless, when composing a process it is indispensable to inquire the Web services operational metrics.

Operational metrics characterize the Quality of Service (QoS) that Web services exhibit when invoked.

What? Formally describes operational metrics of a web service/process

Why? To select the most suitable service to carry out an activity in a process

How? Using QoS model for web services

Composition of processes according to QoS objective and requirements.
Selection and execution of processes based on QoS metrics.
Monitoring of processes to assure compliance with initial QoS requirements.
Evaluation of alternative strategies when QoS requirements are violated.

Operational Metrics Specification

Operational metrics are described using a QoS model represented with a suitable ontology.

The specification of Web services operational metrics allows the analysis and computation processes QoS.

Processes can be designed according to QoS objectives and requirements.

This allows organizations to translate their strategies into their processes more efficiently.
QoS Models

A QoS Model describes non-functional properties of a process.

Which dimensions should be part of a QoS model?

- Time
- Cost
- Fidelity
- Reliability
- Price
- Security
- Availability
- Duration
- Repudiation

QoS Models and Semantics

Use Semantics

QoS in METEOR-S

QoS Creation of Estimates

- To analyze a process QoS, it is necessary to:
  - Create estimated for task QoS metrics and
  - Create estimated for transition probabilities

Once tasks and transitions have their estimates set, algorithms and mechanisms, such as simulation, can be applied to compute the overall QoS of a process.

Web process QoS computation

QoS Estimates for Web Services

WS runtime behavior description can be composed of several classes. For example:

<table>
<thead>
<tr>
<th>QoS Model</th>
<th>Basic class</th>
<th>Distributional class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0.20</td>
<td>0.20</td>
</tr>
<tr>
<td>Cost</td>
<td>0.74</td>
<td>0.74</td>
</tr>
<tr>
<td>Reliability</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>Fidelity</td>
<td>0.63</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Task QoS for an automatic task (SP Fasta task)

Simulate simulation systems

Graph Reduction Techniques

Critical Path Algorithm
### QoS Computation

**Graph Reduction Technique**

![Graph Reduction Technique Diagram](image)

**Reduction of a Sequential System**

- \( T(t) = T(t_1) + T(t_2) \)
- \( C(t) = C(t_1) + C(t_2) \)
- \( R(t) = R(t_1) \cdot R(t_2) \)
- \( F(t_1, t_2) = f(F(t_1), F(t_2)) \)

**Reduction of a Parallel System**

- \( T(t) = \max_{i=1}^{n} T(t_i) \)
- \( C(t) = \sum_{i=1}^{n} C(t_i) \)
- \( R(t) = \prod_{i=1}^{n} R(t_i) \)
- \( F(t_1, t_2) = f(F(t_1), F(t_2), ..., F(t_n)) \)

### QoS Computation

**Graph Reduction Technique**

![Graph Reduction Technique Diagram](image)

**Reduction of a Sequential System**

- \( T(t) = T(t_1) + T(t_2) \)
- \( C(t) = C(t_1) + C(t_2) \)
- \( R(t) = R(t_1) \cdot R(t_2) \)
- \( F(t_1, t_2) = f(F(t_1), F(t_2)) \)

### QoS Computation

**Graph Reduction Technique**

![Graph Reduction Technique Diagram](image)

**Reduction of a Parallel System**

- \( T(t) = \max_{i=1}^{n} T(t_i) \)
- \( C(t) = \sum_{i=1}^{n} C(t_i) \)
- \( R(t) = \prod_{i=1}^{n} R(t_i) \)
- \( F(t_1, t_2) = f(F(t_1), F(t_2), ..., F(t_n)) \)

---

**SCET (Service Composition and Execution Tool)** allows:
- to compose services statically by modeling the process as a digraph in a graphical designer
- stores the process description as WSFL based specification
- allows execution of the composed process using Perl
- supports a simple execution monitoring feature
- supports performance estimation using JSIM simulation

Simulation provides feedback on processes, allowing the composer to modify his process design by:
- Replacing services which do not satisfy the expected runtime behavior with more suitable Web services.
- Modifying the process structure (control flow) based on the simulation runs.

---

1. Miller, Cardoso et al. 2002.
Questions?

Semantics in METEOR-S

- Annotation, Discovery, Composition (in development), and QoS
- Focuses on two issues: **semantic Web services and process composition**.
- Process Composition:
  - Functional perspective
  - Web Service Discovery, handling semantic heterogeneity
  - Operational perspective
  - QoS specification for Web Services and Processes.

METEOR-S components for Semantic Web Services

- **Discovery Infrastructure (MWSDI)**
  - Semantic Annotation and Discovery of Web Services
  - Semantic Peer-to-Peer network of Web Services Registries
- **Composer**
  - SCET: Service Composition and Execution Tool
  - Semantics Process Template Builder and Process Generator
  - QoS Management
    - Specify, compute, monitor and control QoS (SWR algorithm)
- **Orchestrator** (Under development)
  - Analysis and Simulation
  - Execution
  - Monitoring

METEOR-S Web Service Discovery Infrastructure (MWSDI)

- **Functional, Data** and QoS semantics
  - Service Discovery

METEOR-S Web Service Discovery Infrastructure (MWSDI)

Service Selection
- uses Functional, Data and QoS semantics

METEOR-S Web Service Composition Framework (MWSCF)

- needed for the world where business processes never stop changing

MWSCF Architecture

Process Execution
1. Validation and deployment
2. Executing the process using a client

Process Designer
1. Template Construction
   - activity specification using
     - interfaces
     - services
     - semantic activity templates
     - other details
2. Process Generation
   - Service discovery (automatic) and selection (semi-automatic)
   - Data flow

Repositories are used to store
1. Web Service Interfaces
2. Ontologies
3. Process Templates

Repositories
- Web Service Interfaces
- Ontologies
- Process Templates

Process Designer
1. Template Construction
   activity specification using
   - interfaces
   - services
   - semantic activity templates
   - other details
2. Process Generation
   - Service discovery (automatic) and selection (semi-automatic)
   - Data flow

Process Execution
1. Validation and deployment
2. Executing the process using a client

Web Process Life-Cycle

1. Discovery
   - Find matches
   - Rank services
   - Select a service
2. Composition
   - Add Control Flow
   - Data Transformation
   - Data Flow
3. Execution
   - Generate Process
   - Validate Process
   - Execute Process

Semantic Web Process Design

Template Construction

Process Generation
Semantic Web Process Design

Ongoing Projects

- **SWAP**: [http://swap.semanticweb.org/]
  - Share knowledge effectively
  - Combination of Semantic Web and P2P
- **WonderWeb**: [http://wonderweb.man.ac.uk/]
  - Development of a framework of techniques and methodologies that provide an engineering approach to the building and use of ontologies.
  - Development of a set of foundational ontologies covering a wide range of application domains.
  - Development of infrastructures and tool support that will be required by real world applications in the Semantic Web.
Ongoing Projects

- DAML-S: [http://www.daml.org/services/](http://www.daml.org/services/)  
  Set of ontologies to describe functionalities of web services
  Match service requestors with service providers  
  Semantic Matchmaking for Web Services Discovery
  Semi-automatic process for the dynamic composition of web services
  WSDL, UDDI, SOAP  
  Business Process with BPEL4WS

Conclusions

- Semantic Web service Annotation and Discovery  
  - Data semantics  
  - Functional semantics  
  - QoS Semantics
- Web processes vs. Semantic Web processes  
  - BPEL4WS vs. DAML-S  
  - Web process composition  
  - Web services semantic degree of integration  
  - Data, Functional, and QoS similarity
- Web process QoS computation  
  - QoS Models, techniques, and algorithms

Conclusions

- Present Problems in Process Composition  
  - Static discovery of Web Services  
  - Design/deployment-time binding of Web services  
  - Process Composition is based on interfaces of participating services
- Proposition  
  - Semantics is the enabler to address the problems of scalability, heterogeneity (syntactic and semantic), machine understandability faced by Web services
- Semantics for Web Services  
  - Semantics can be applied to different layers of Web Services conceptual stack  
  - Semantics for Web Services can be categorized into at least 4 different dimensions namely Data, Functional, Execution and Quality (QoS)

Conclusions

- Semantics can help address big challenges related to scalability, dynamic environments.  
  But comprehensive approach to semantics will be needed:  
  - Data/information, function/operation, execution, QoS  
  - Semantic (Web) principles and technology bring new tools and capabilities that we did not have in EAI, workflow management of the past

Web Resource for this tutorial
(incl. latest version)

http://lsdis.cs.uga.edu/lib/presentations/SWP-tutorial-resource.htm

References

DAML
http://www.daml.org/services/
http://www.daml.org/2001/03/daml+oil-index
http://www.ksl.stanford.edu/projects/DAML/WebServices/DAMLS-BPEL.html

References

Extensive related work at: IBM, Karlsruhe, U. Manchester, DAML-A (CMU, Stanford, UMD)

- [Sivashanmugam et al.-1] Adding Semantics to Web Services Standards
- [Verma et al.] MWSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services
- [Chandrasekaran et al.] Performance Analysis and Simulation of Composite Web Services
- [Cardoso et al.] Modeling Quality of Service for Workflows and Web Service Processes
- [Silver et al.] Modeling and Simulation of Quality of Service for Composition of Web Services
- [Paolucci et al.] Importing Semantic Web in UDDI

More at: http://lsdis.cs.uga.edu/SWP.htm

Semantic Web Processes

End