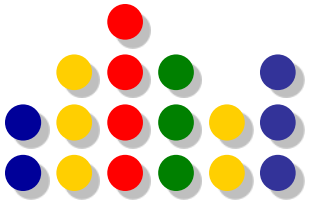
Semantic Web Processes:
Semantics Enabled Annotation, Discovery, Composition and Orchestration of Web Scale Processes



Jorge Cardoso[1], Amit Sheth[2,3]
[1]University of Madeira
[2]LSDIS Lab, Computer Science, University of Georgia
[3] Semagix, Inc

4rd International Conference on Web Information Systems Engineering (WISE 2003),
December 10th to 12th, 2003,
Rome, Italy.

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, scalability, 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)

Semantics

Web Processes

Web Process Composition

Web Process QoS

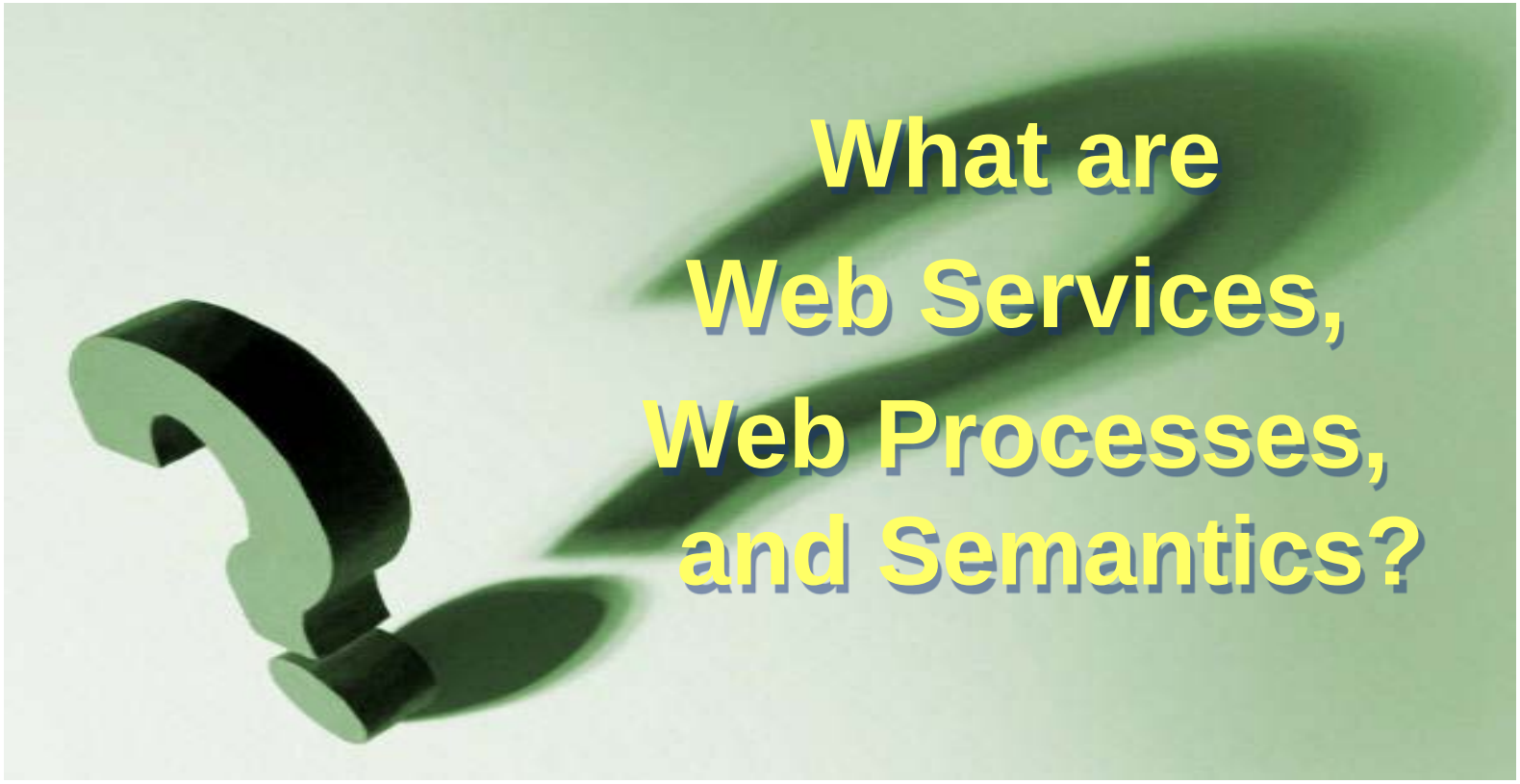
Web Services

Web Service Annotation

Web Service Discovery

Web Service QoS

The Basics



Web Services: Definition

Web Services

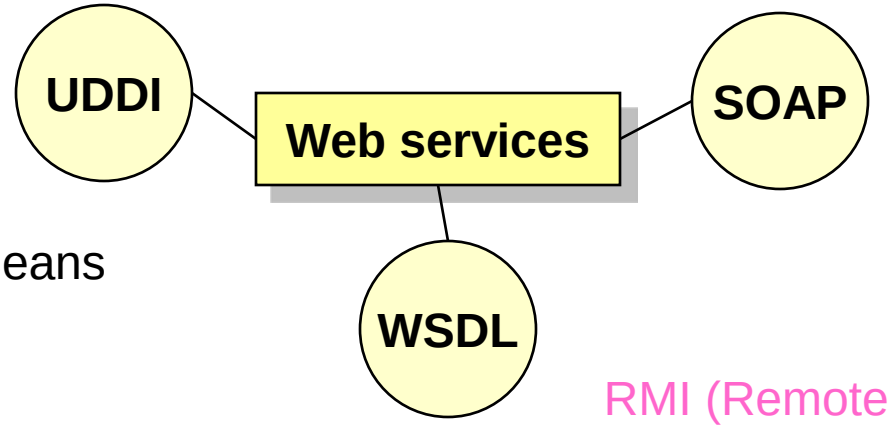
"Web services are a new breed of Web application. They are **self-contained**, **self-describing**, modular applications that can be **published**, **located**, and **invoked** across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes. …

Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service."

IBM web service tutorial

Why Web Services?

Web Services



Jini

Enterprise Java Beans

RMI (Remote Method Invocation)

Microsoft DCOM

CORBA (Common Object Request Broker Architecture)

Open Software Foundation DCE (Distributed Computing Environment)

Sun ONC/RPC (Open Network Computing)

IP, UDP, TCP

Why Web services?

Web Services

Feature	CORBA	Web Services
Data Model	Object Model	SOAP Message exchange model
Client Server Coupling	Tight Coupling	Loose Coupling
Parameter Passing	Pass by reference/value	Pass by value only
Type Checking	1.Static + Runtime type checking (Regular) 2. Runtime type checking only (DII)	RunTime type checking only
State	Stateful	1. Stateless, Uncorrelated (Web Services) 2. Stateful (Web Process)
Firewall Traversal	Work in Progress	Uses HTTP port 80
Service Discovery	CORBA naming/trading Service	UDDI
Communication Mode	1-way, 2-way sync 2-way async	2-way sync (Web Services) 1-way, 2-way sync, 2-way async (Web Process)

Gokhale et al, Reinventing the Wheel ? CORBA vs Web-services;
Sheth and Miller, Web Services: Incremental Technical Advance with Huge Practical Impact

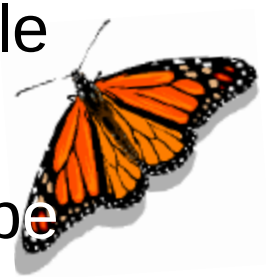
What are Web Processes (1)?

- **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**

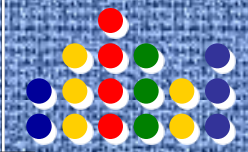
What are Web Processes ? (2)

Web Processes

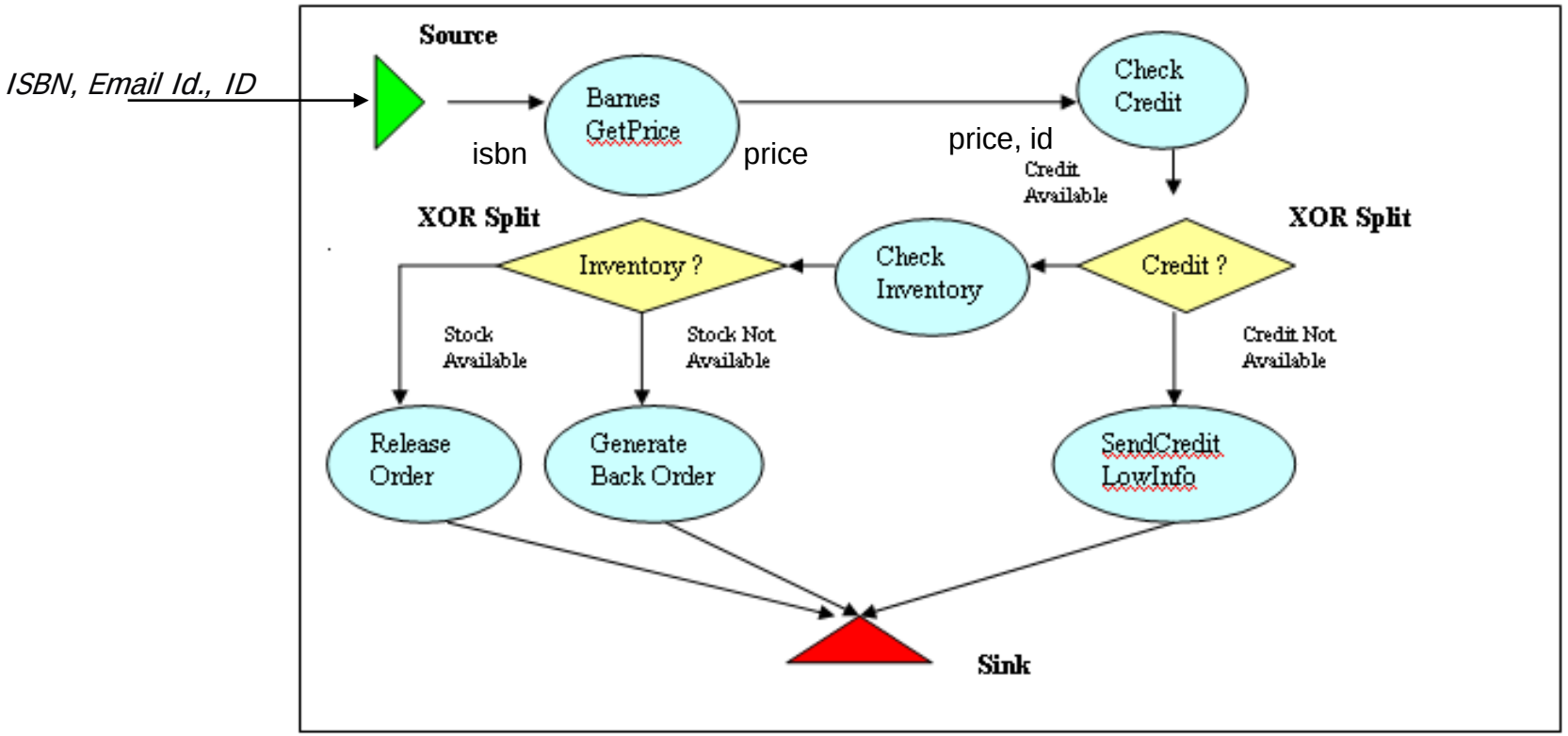
- Web processes describe how Web services are connected to create reliable and dependable business solutions
- Web processes allow businesses to describe sophisticated processes that can both consume and provide Web services
- The role of Web processes within the enterprise is to simplify the integration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



Web Process
An Example



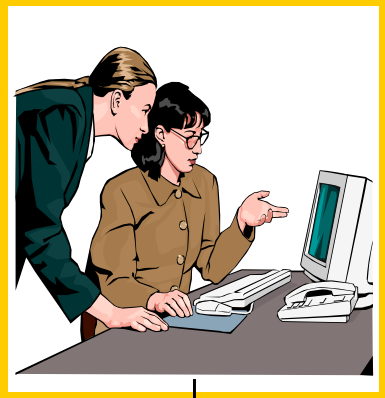
- Graphical example of a web process

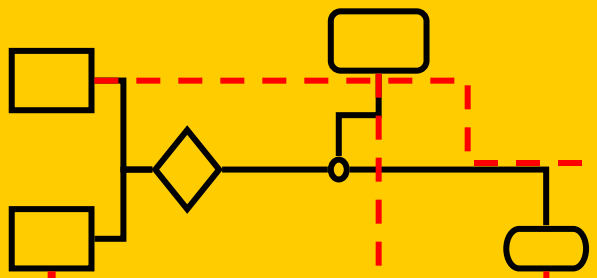


The BarnesBookPurchase process

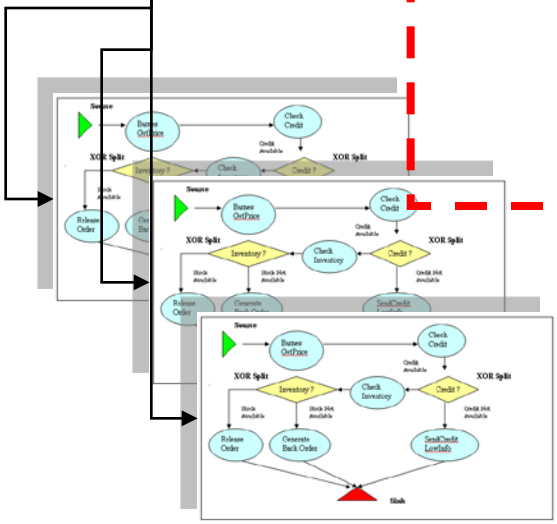
Web Processes Composition

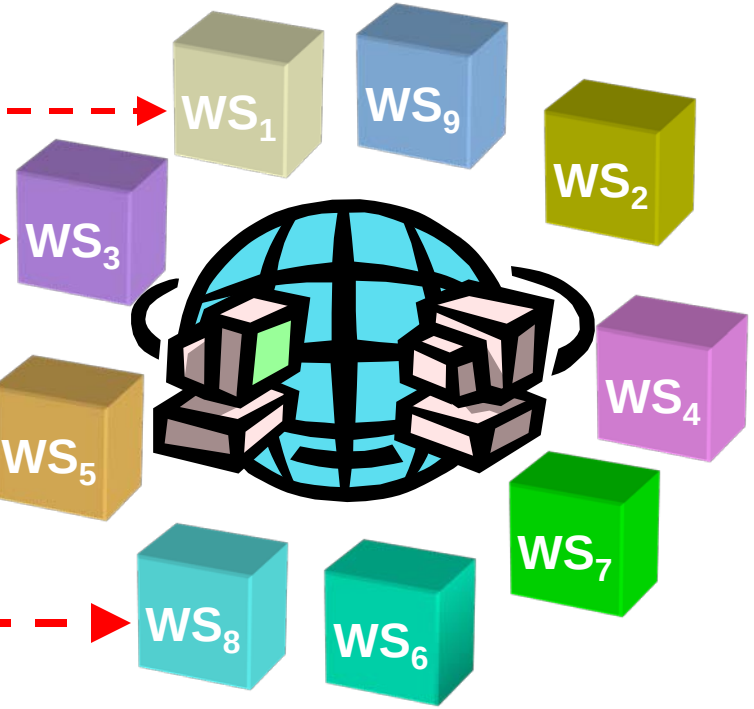
Web Processes





Web Process Design





Web services

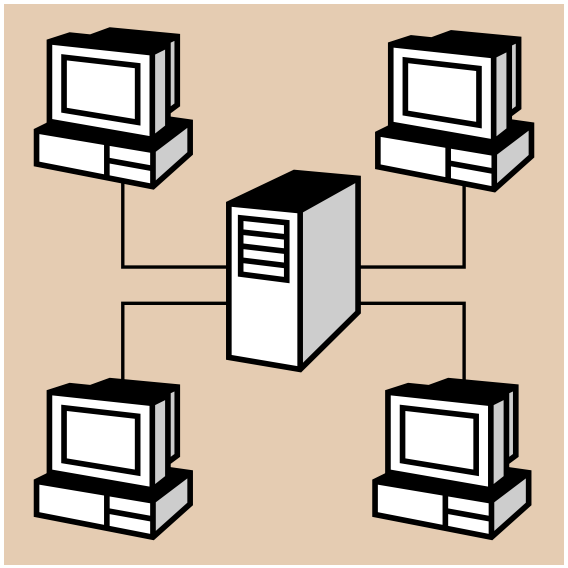
Architectures for Web Processes*

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



Workflows

B2B

Distributed Workflows

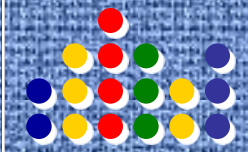
E-Services

Web Processes

Enterprise | Inter-Enterprise | Global

Processes driving the Networked Economy

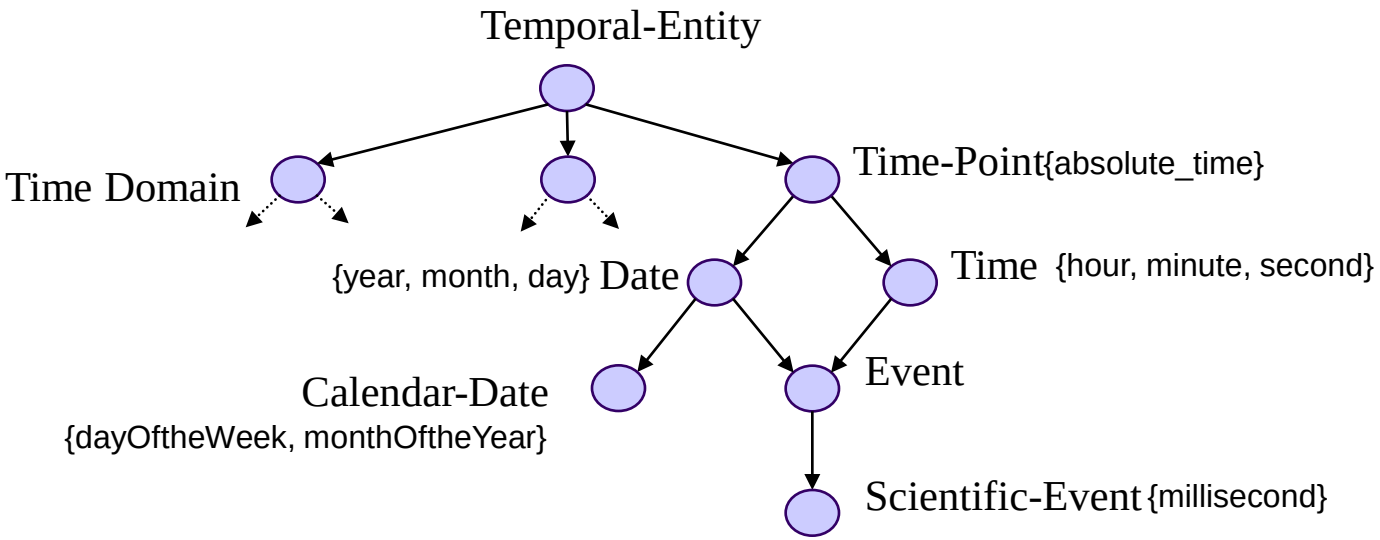
BIG Challenges



- **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
  - Demands: 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

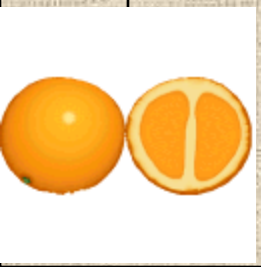
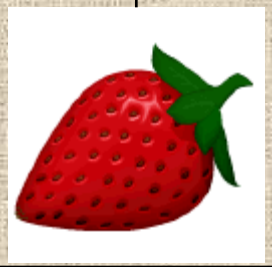
Annotation of Web Services

Web Process Composition

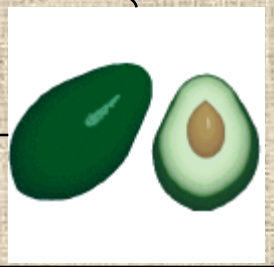
Semantic Web

Web Service Discovery

Web Processes Quality of Service

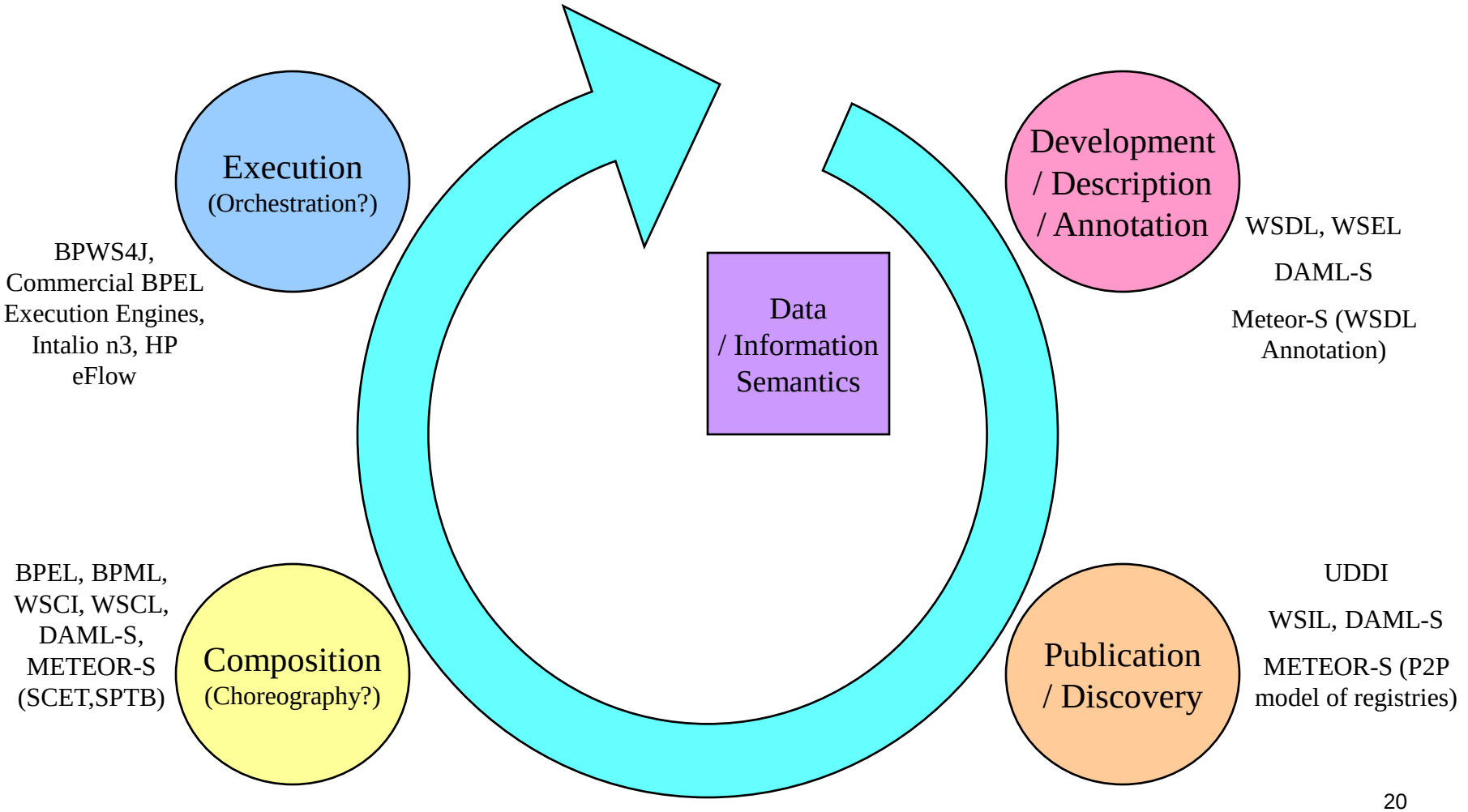




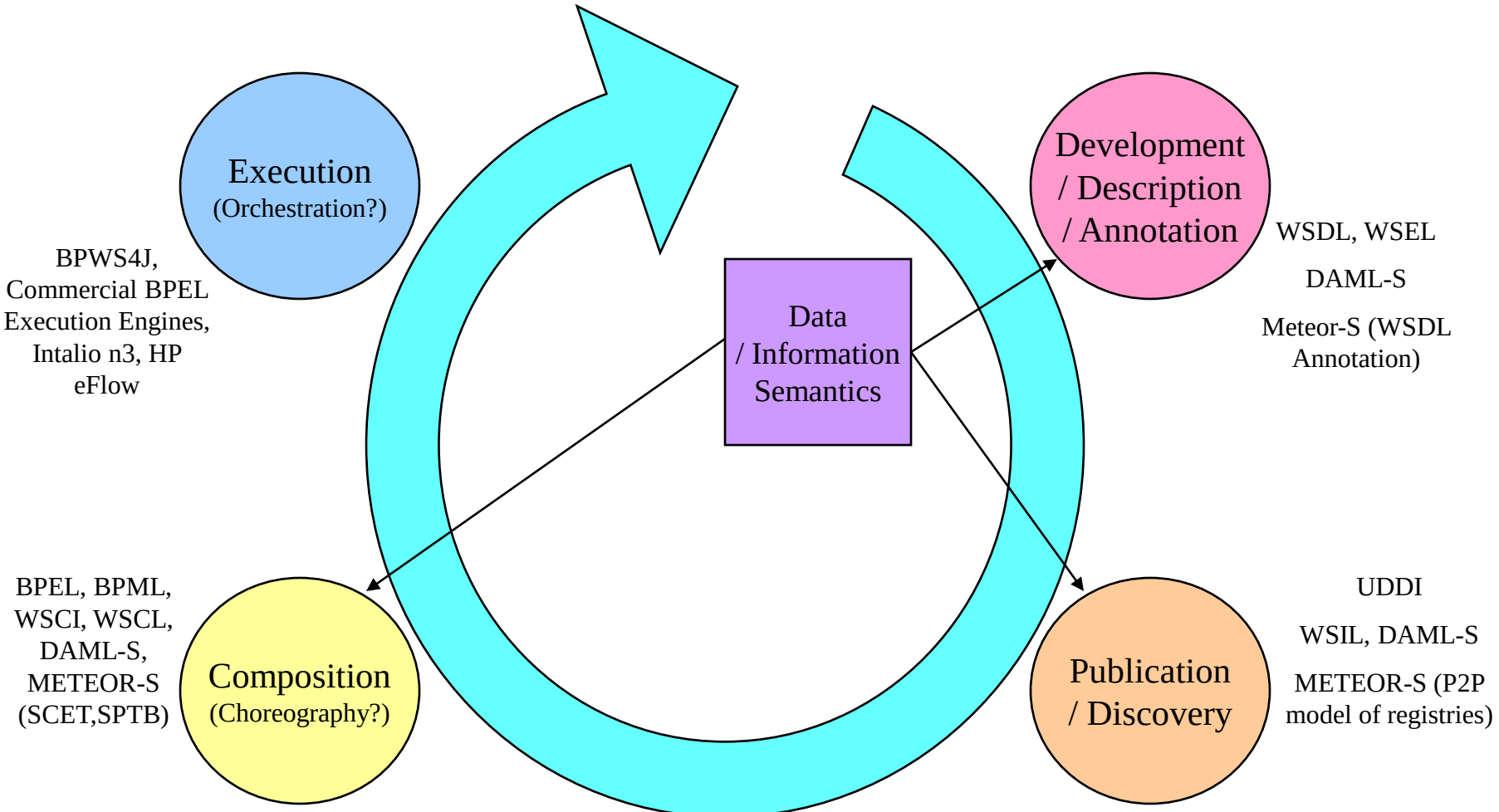

Semantics for Web Processes

- 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

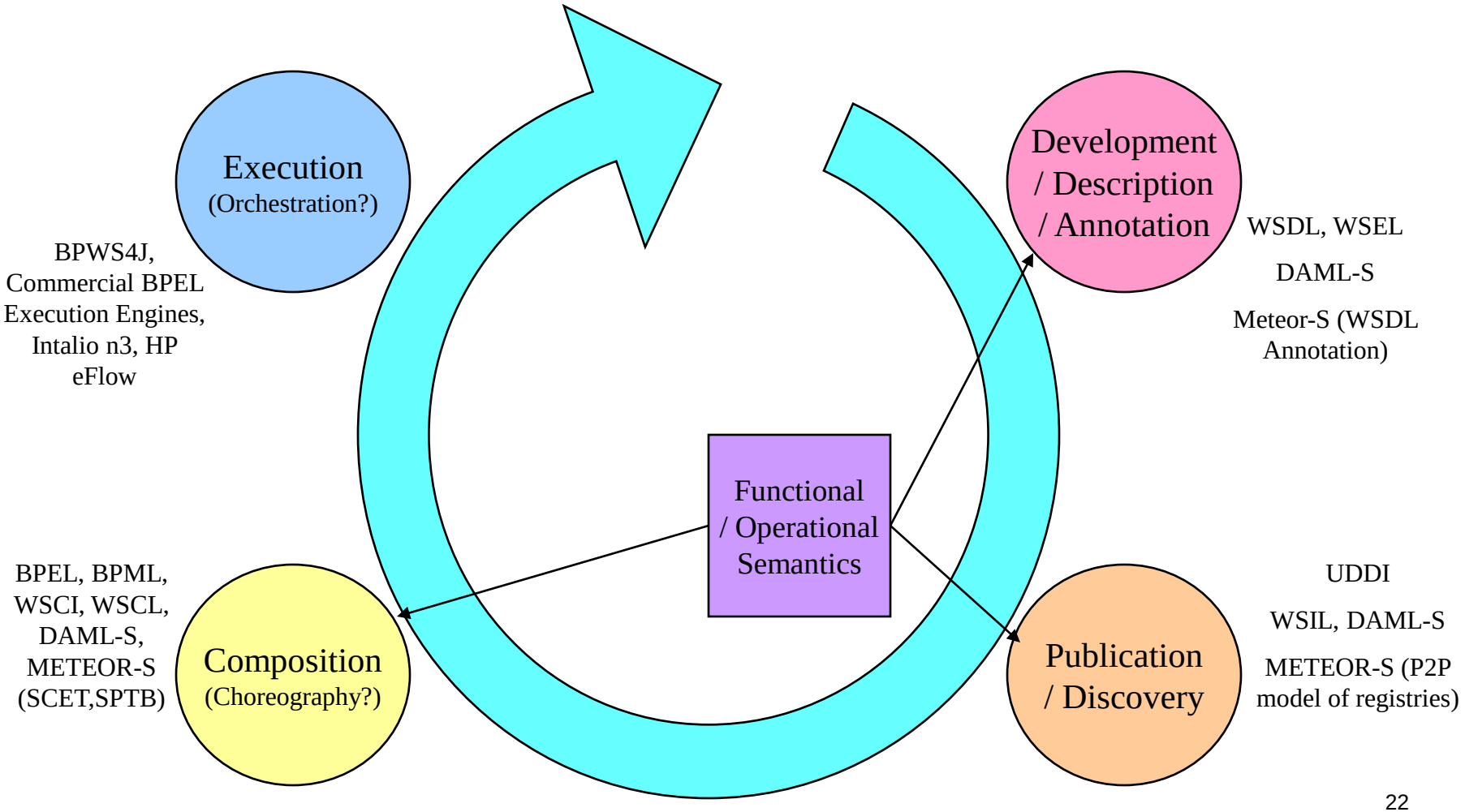
Semantics for Web Process Life-Cycle



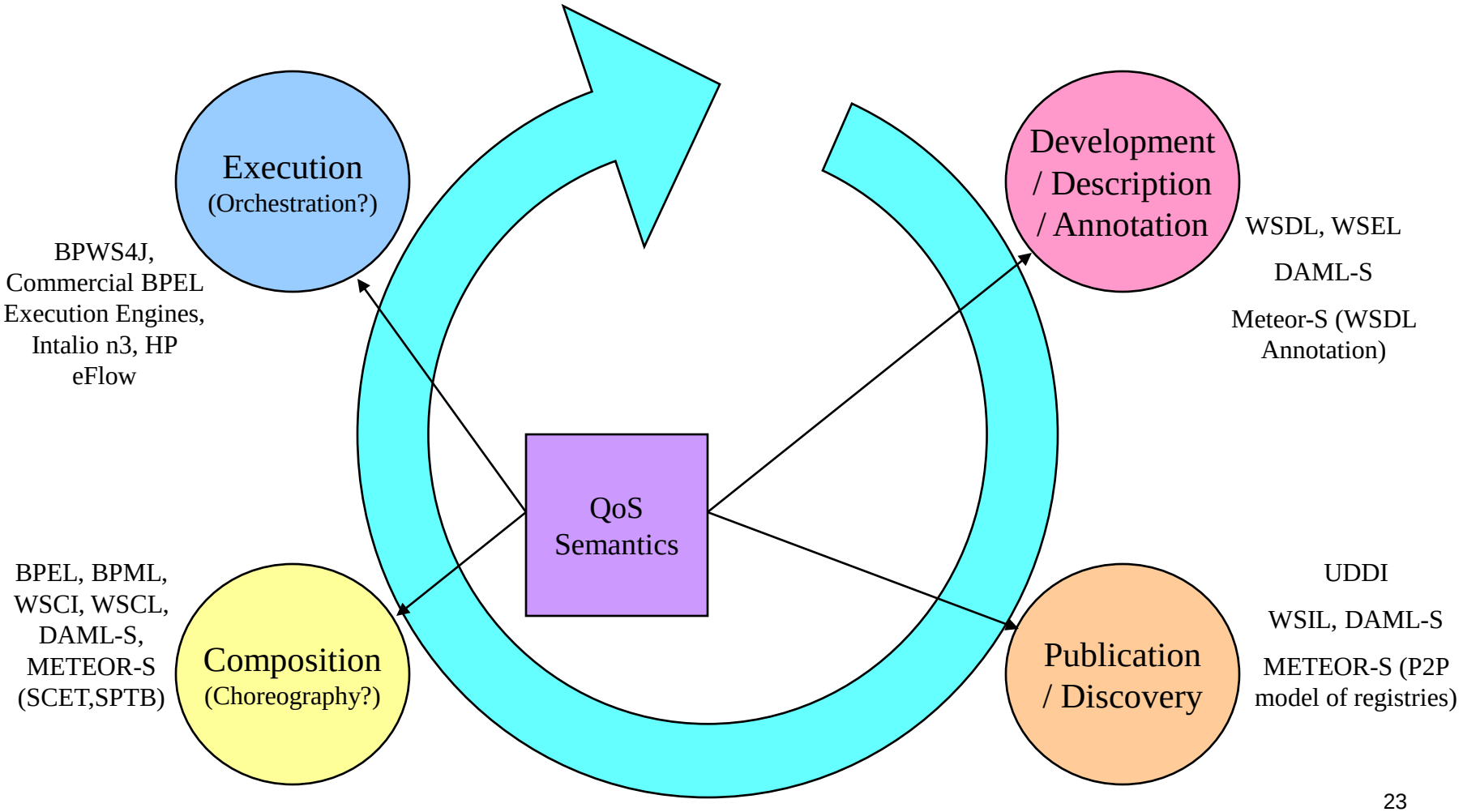
Semantics for Web Process Life-Cycle



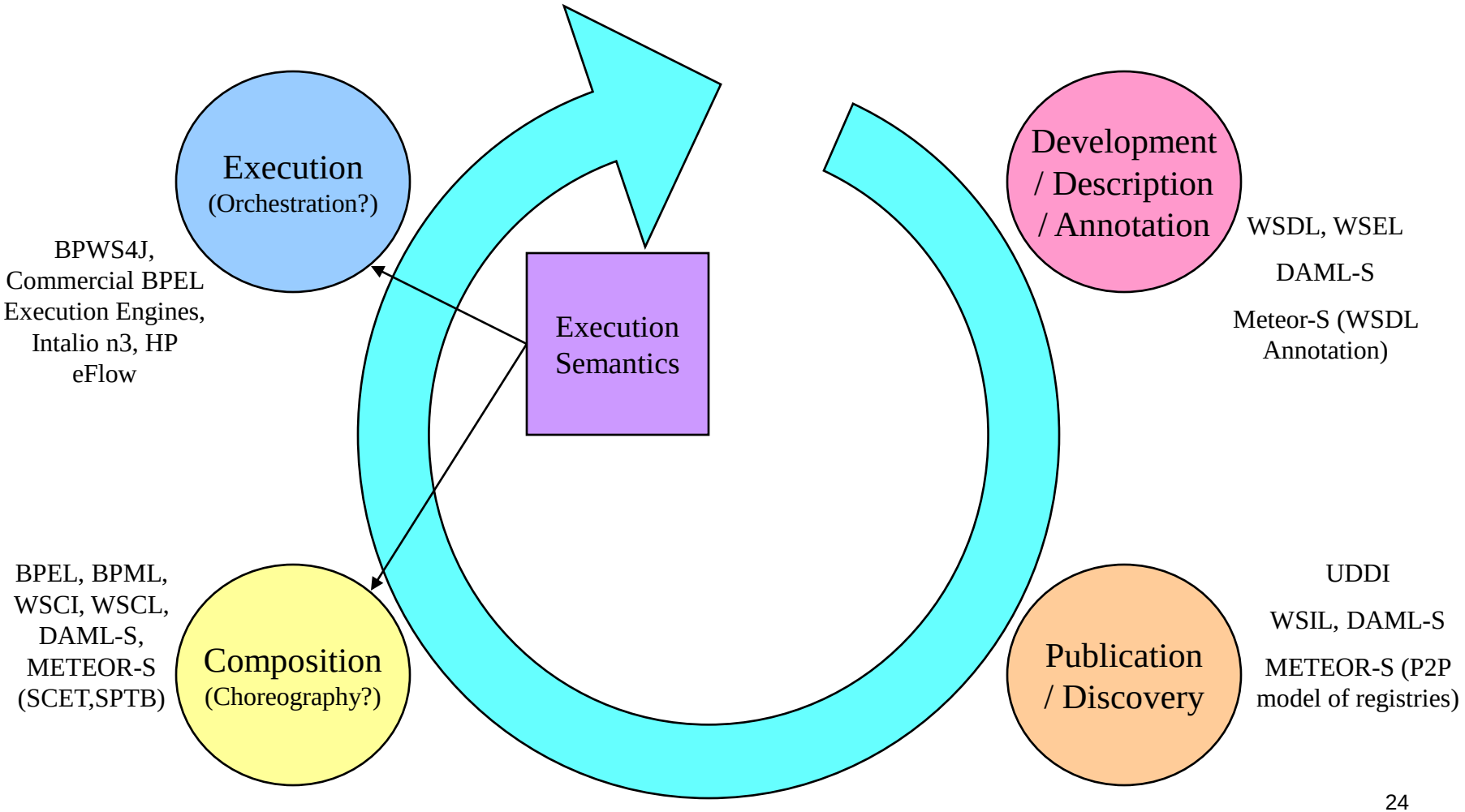
Semantics for Web Process Life-Cycle



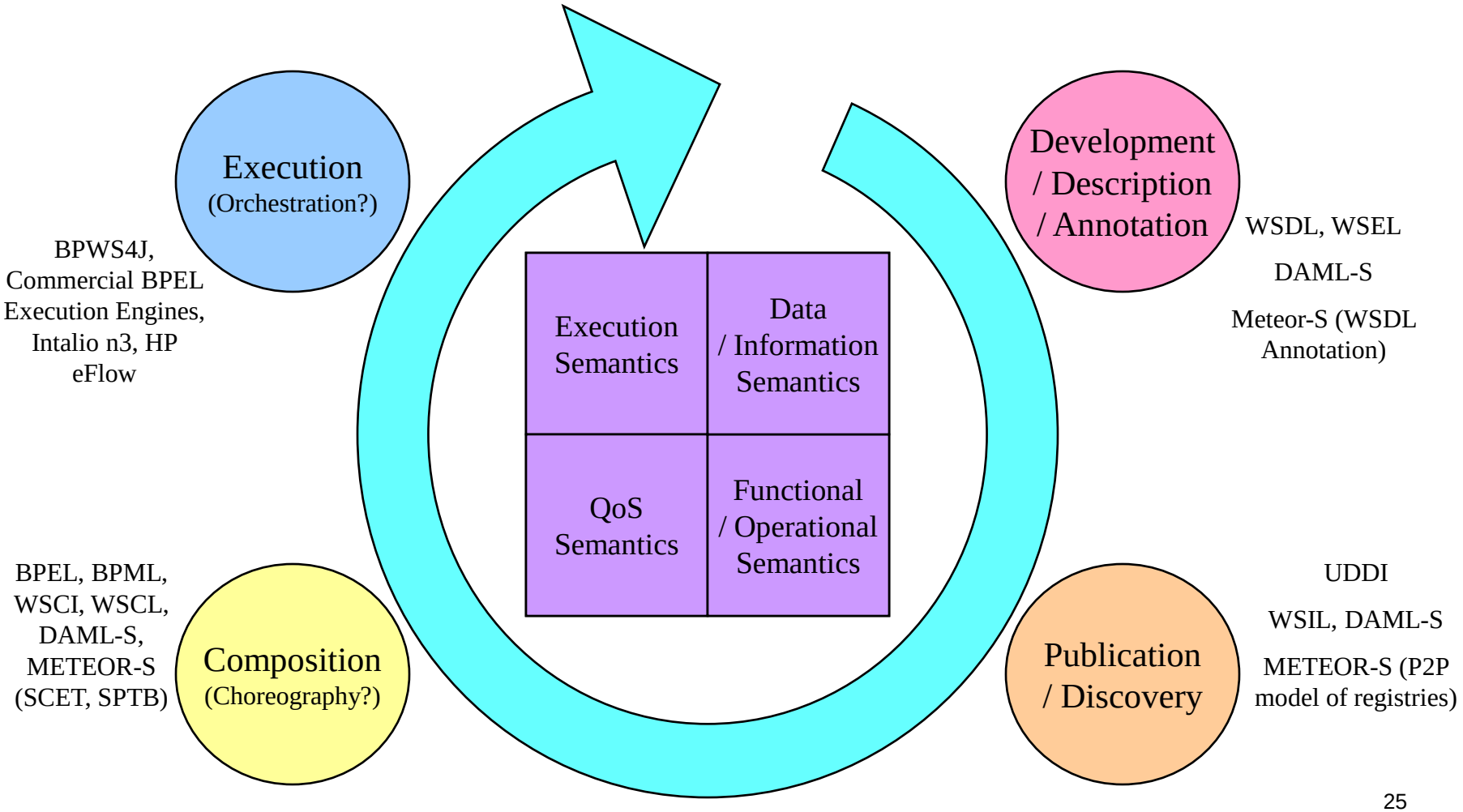
Semantics for Web Process Life-Cycle



Semantics for Web Process Life-Cycle



Semantics for Web Process Life-Cycle



Web Processes Architecture

How can semantics be explored ???

...

Composition

Brokering

Discovery

UDDI

WSDL

SOAP

XML

Web Servers

HTTP/HTTPS

TCP/IP-SSL

Semantics

Web Process Architecture

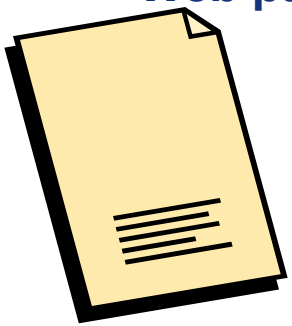
Semantic Web servers

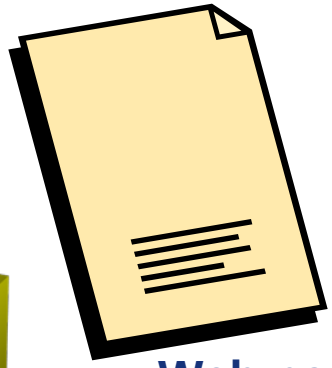
Associate ontology based
semantic layers to web resources

Semantic Web browsers

Making sense of page contents
Supporting the *interpretation* of web pages

Web page



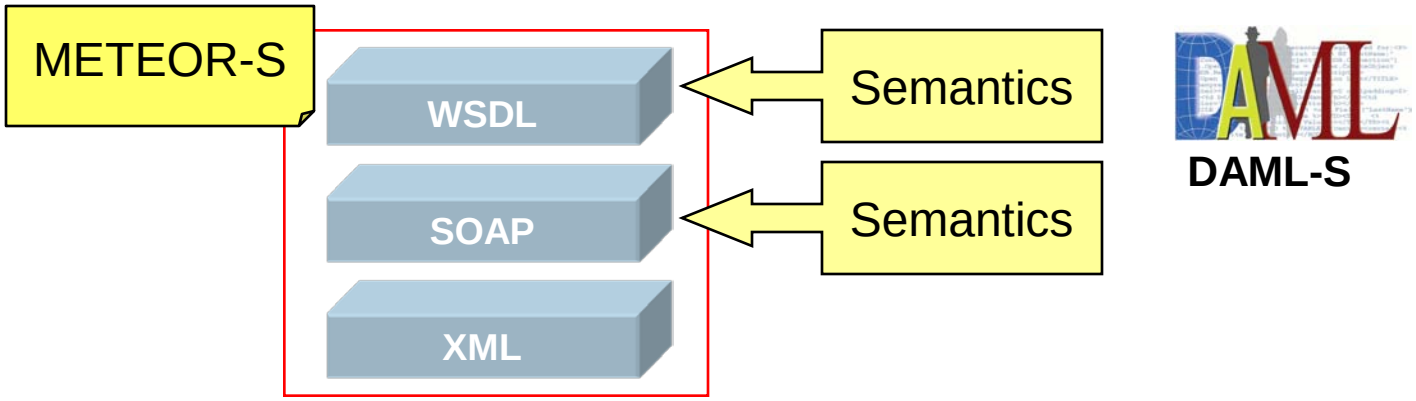


Web page

Web Process Architecture

Web service Semantic Annotation

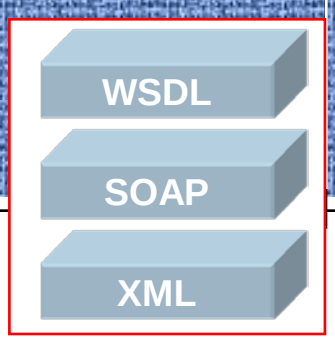
Associate ontological concepts
to Web service descriptions



Adding Semantics to Web Services Standards , Semantic Annotation of Web Services

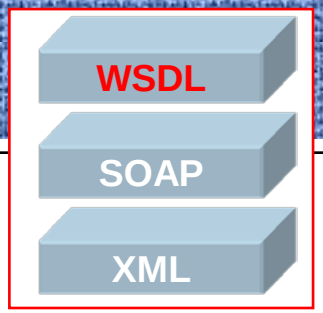
Semantics

Web Services



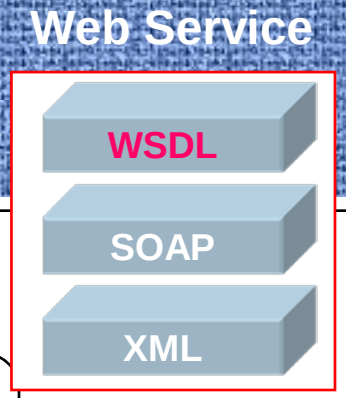
- **WSDL** defines services as collections of network endpoints or *ports*. A port is defined by associating a network address with a binding; a collection of ports define a service.
- **SOAP** is a message layout specification that defines a uniform way of passing XML-encoded data. It also defines a 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.

WSDL



- WSDL stands for Web Services Description Language
- WSDL is an XML document
- WSDL is used to describe Web services
- WSDL is also used to locate Web services

WSDL



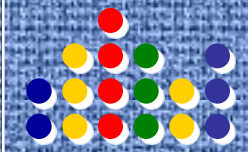
```
<definitions>
    <types>
          definition of types..
    </types>

    <message>
          definition of messages...
    </message>

    <portType>
          <operation> ..... </operation>
          <operation> ..... </operation>
    </portType>

    <binding>
          definition of binding....
    </binding>

    <service>
          <port>....</port>
          <port>....</port>
    </service>
</definitions>
```

Abstract Description

Concrete Description

From S. Chandrasekaran's Talk

Semantic Annotation of Web Services

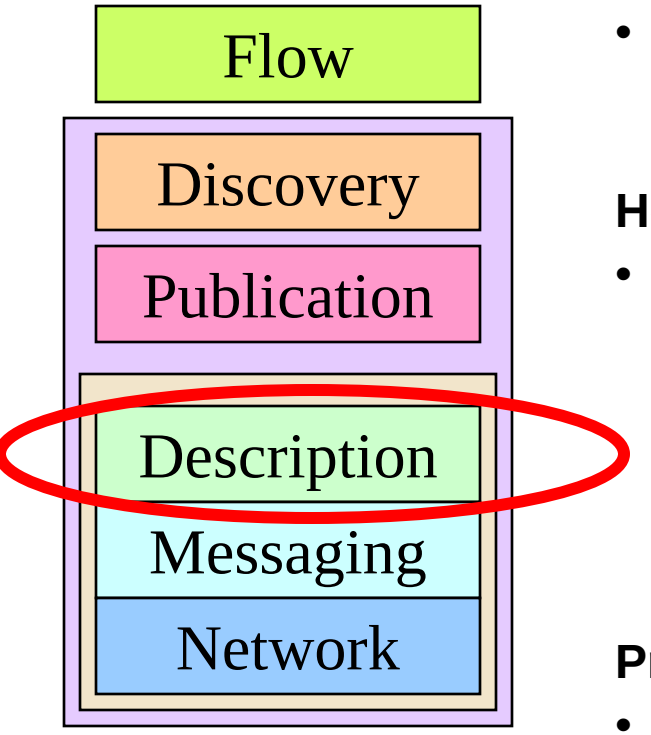


Annotation of Web Services

- 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.

Semantics at Description Layer



Description Layer:

Why:

- Unambiguously understand the functionality of the services and the semantics of the operational data

How:

- Using Ontologies to semantically annotate WSDL constructs (conforming to extensibility allowed in WSDL specification version 1.2) to sufficiently explicate the semantics of the
  - data types used in the service description and
  - functionality of the service

Present scenario:

- WSDL descriptions are mainly syntactic (provides operational information and not functional information)
- Semantic matchmaking is not possible

Adding Semantics to Web Services Standards , Semantic Annotation of Web Services

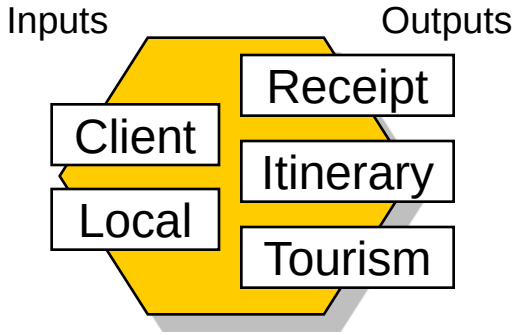
How to Annotate ?

- 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

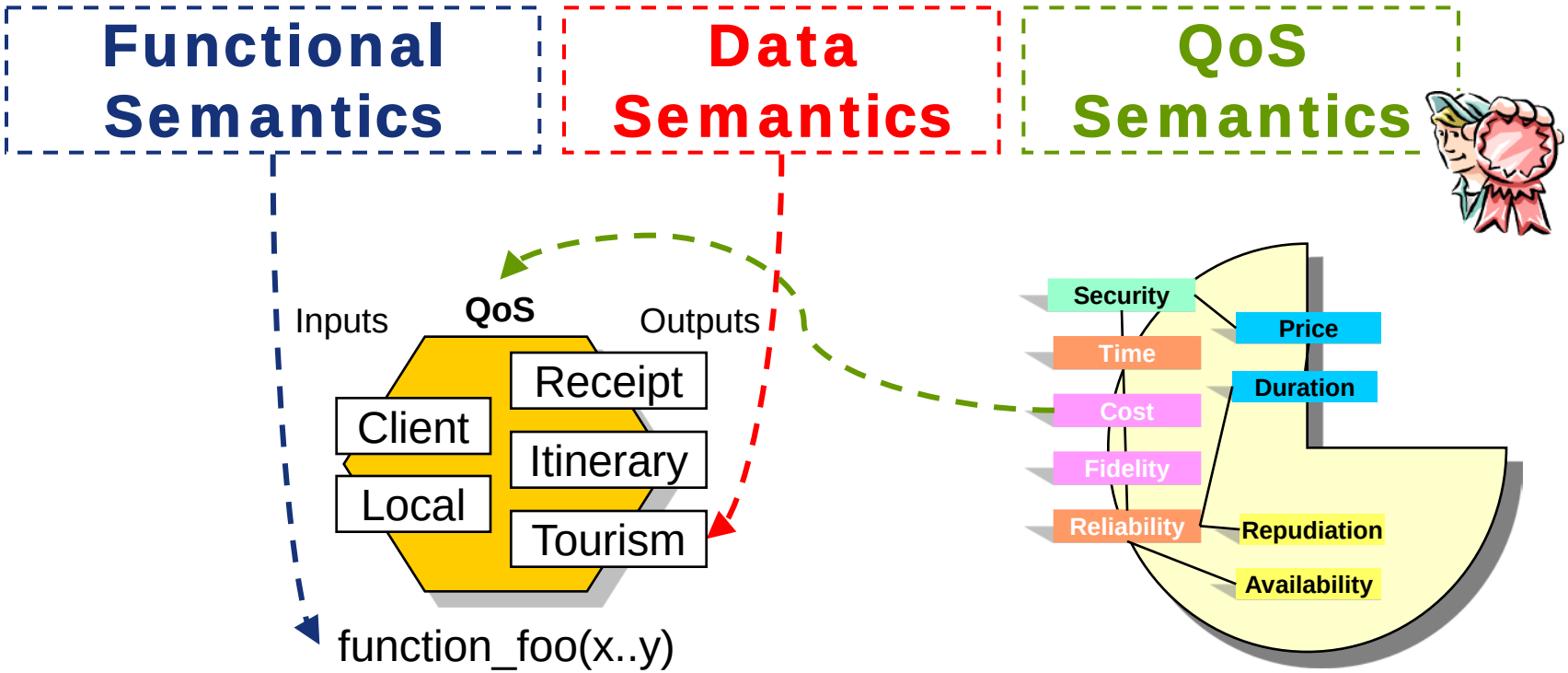
Web Services
Interfaces

- 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)

Types of Annotation



Adding Semantics to Web Services

```
<xsd:complexType name="Date">
<xsd:sequence>
  <xsd:element name="year" type="xsd:integer" />
  <xsd:element name="month" type="xsd:integer" />
  <xsd:element name="day" type="xsd:byte" />
</xsd:sequence>
</xsd:complexType>
```

WSDL

Ontologies

Data Semantics

Ω = Time - Ontology

Temporal-Entity

Time Interval

Time Domain

Time-Point {absolute_time}

{year, month, day} Date

Time {hour, minute, second}

Calendar-Date

{dayOftheWeek, monthOftheYear}

Event

Scientific-Event {millisecond}

Ω = Local ontology

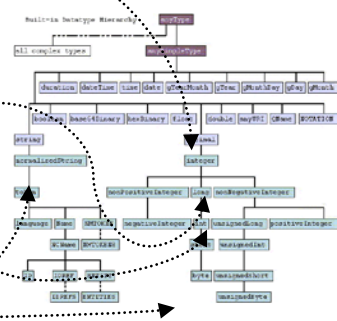
Coordinates {x, y}

Area {name}

City

Forrest

XML Schema Data type hierarchy



Web Service

Interfaces

Inputs

Outputs

Name

Year

Date

Duration

City

Get Conference Information

QoS Ontology

QoS Semantics

Quality

Min

Functional Semantics

Information Function

Conference Information Functions

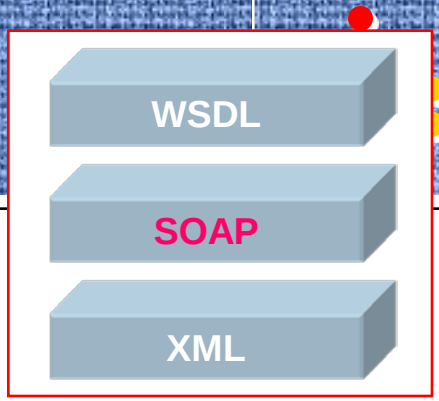
Get Information

Get Date

WSDL

```
<portType name="ConferenceInformation">
<operation name="getInformation">
  <input message="tns:Data" />
  <output message="tns:ConferenceInformation" />
</operation>
```

SOAP



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

```
<SOAP:Envelope
   xmlns:SOAP='http://schemas.xmlsoap.org/soap/envelope/'
   SOAP:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:v='http://www.topxml.com/soapworkshop/'>
   <SOAP:Header>
           <v:From SOAP:mustUnderstand='1'>
                   cdix@soapworkshop.com
           </v:From>
   </SOAP:Header>
   <SOAP:Body>
           <v:DoCreditCheck>
                   <ssn>123-456-7890</ssn>
           </v:DoCreditCheck>
   </SOAP:Body>
</SOAP:Envelope>
```

Header

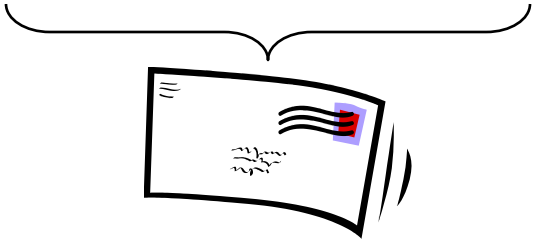
Body

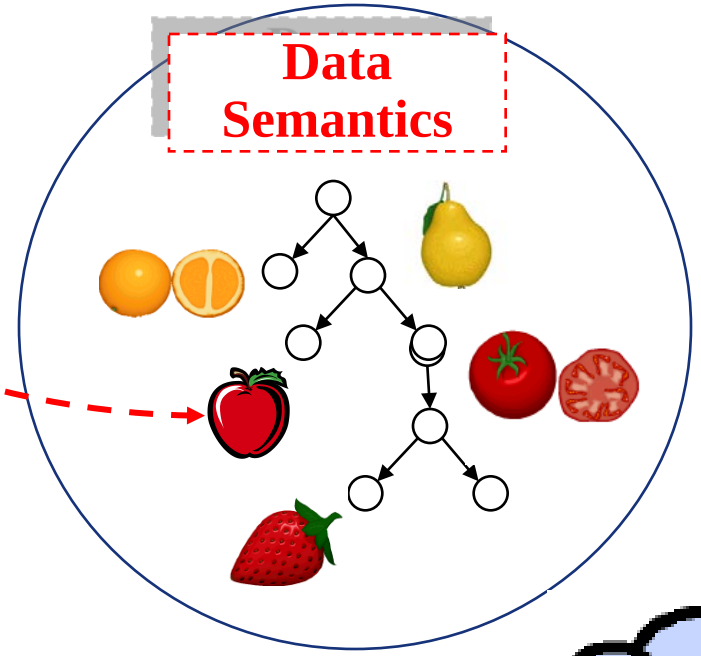
Why SOAP?

- 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.

SOAP - Annotation

```
<soap:Body>
  <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
                    <m:Item>Apples</m:Item>
          </m:GetPrice>
</soap:Body>
…
</soap:Envelope>
```





Server

SOAP over HTTP

XML

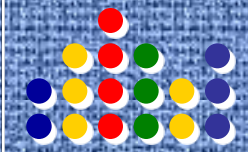
Client

SOAP over HTTP

XML

Internet

Web Process Architecture



Semantic Brokering

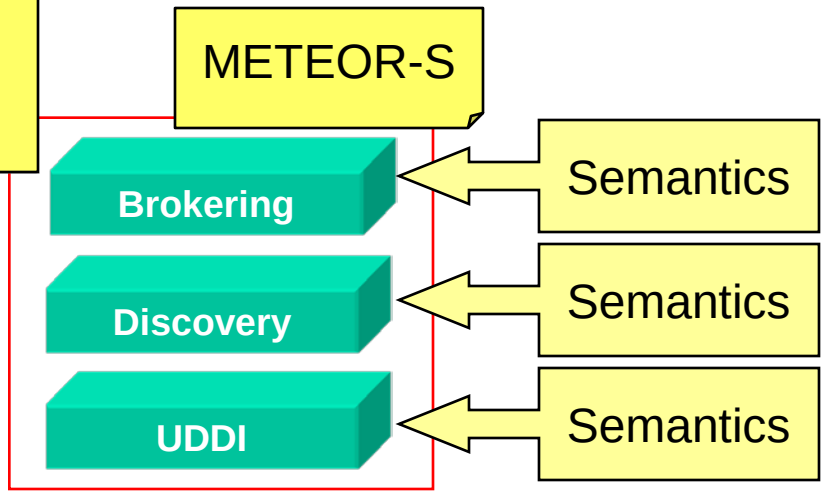
Specialized brokering services to find Web services

Semantic Discovery

Discovery algorithms that account for semantic information

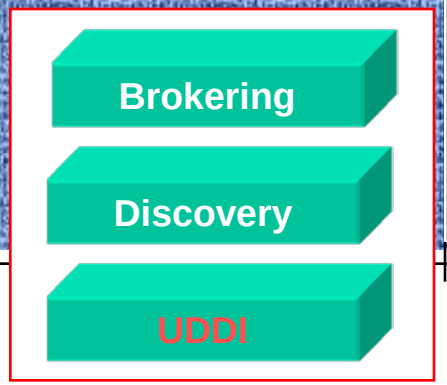
Semantic Registries

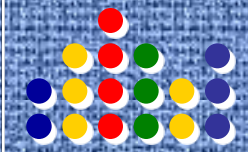
Describe Web services in UDDI registries using semantic concepts



Semantics

UDDI



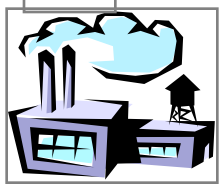


- **UDDI** stands for Universal Description, Discovery and Integration
- **UDDI** serves as a "Business and services" registry and directory and are essential for dynamic usage of Web services
- A **UDDI** registry is **similar to a CORBA trader**, or it can be thought of as a DNS for business applications.
- 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

UDDI Business Registry

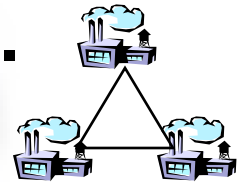
Business Registrations

Service Type Registrations

3. UBR assigns a programmatically unique identifier to each service and business registration

4. 

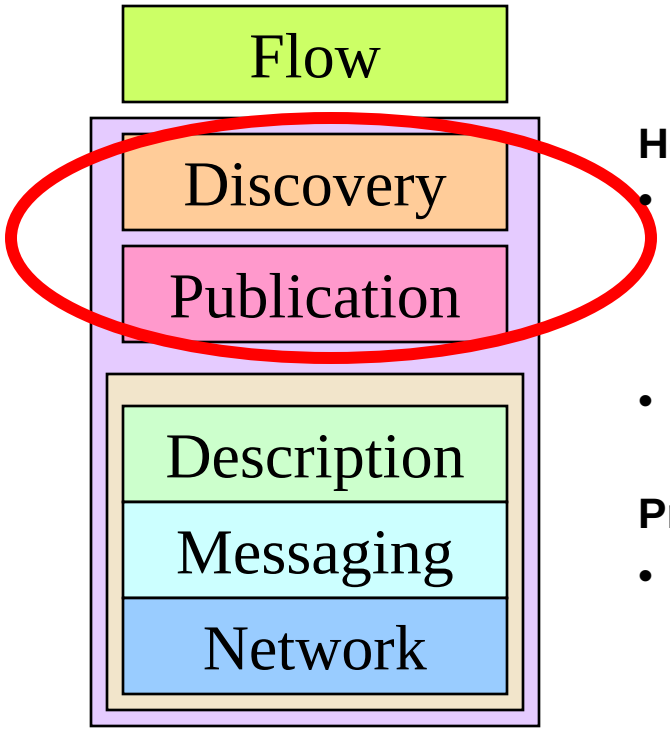
Marketplaces, search engines, and business apps query the registry to discover services at other companies

5. 

Business uses this data to facilitate easier integration with each other over the Web

Source : http://www.uddi.org/pubs/UDDI_Overview_Presentation.ppt

Semantics at Publication and Discovery Layers



Publication and Discovery Layers:

Why:

- Enable scalable, efficient and dynamic publication and discovery (machine processable / automation)

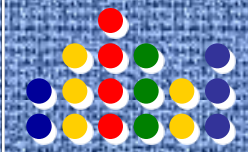
How:

- Use of ontology to categorize registries based on domains and characterize them by maintaining the
  1. properties of each registry
  2. relationships between the registries
- Capturing the WSDL annotations in UDDI

Present scenario:

- Suitable for simple searches (like services offered by a provider, services that implement an interface, services that have a common technical fingerprint etc.)
- Categories are too broad
- Automated service discovery (based on functionality) and selecting the best suited service is not possible

UDDI and Semantics



Marketplaces, search engines, and business apps query

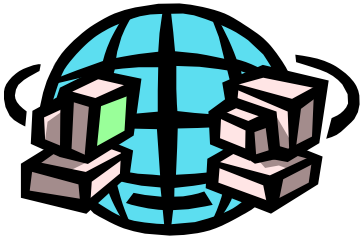



Semantic UDDI

Registry entry

Functional Semantics | **Data Semantics** | **QoS Semantics**

Internet



WS_9 WS_4 WS_2 WS_7 WS_8

Inputs

QoS

Outputs

Client
Local
Receipt
Itinerary
Tourism

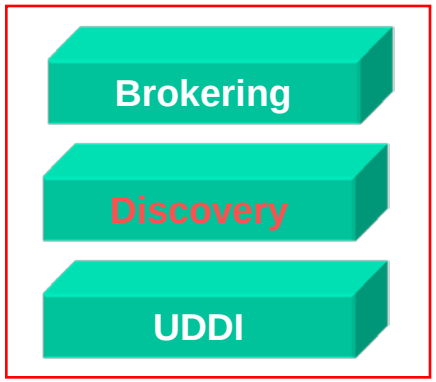
function_foo(x..y)

Security
Time
Cost
Fidelity
Reliability
Price
Duration
Repudiation
Availability

Semantic Discovery of Web Services

Web Service Discovery

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.

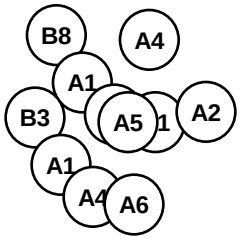


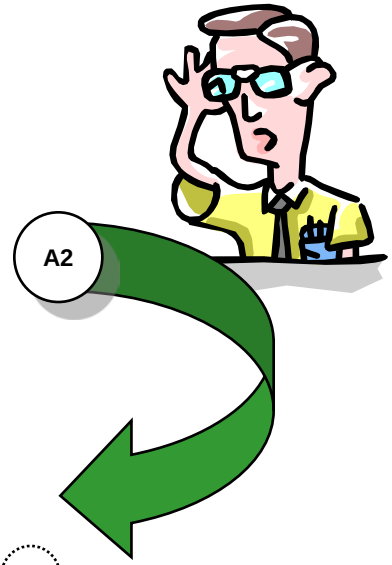
Discovery
New Requirements

Web Service Discovery

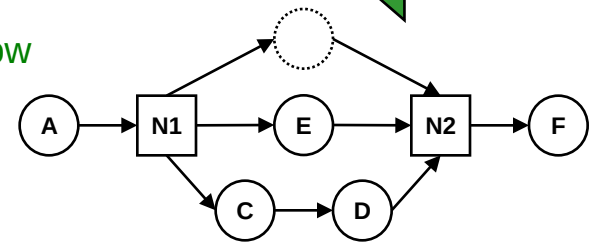
Before | **Now**

Tasks



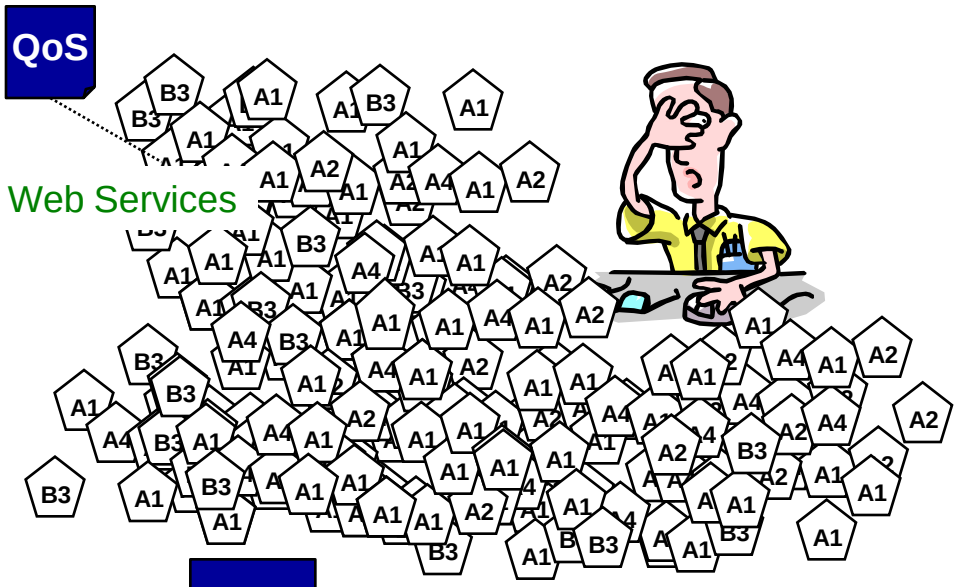


Workflow



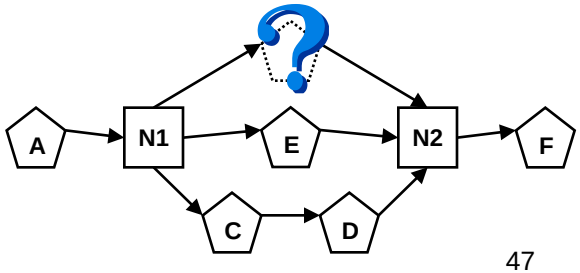
QoS

Web Services



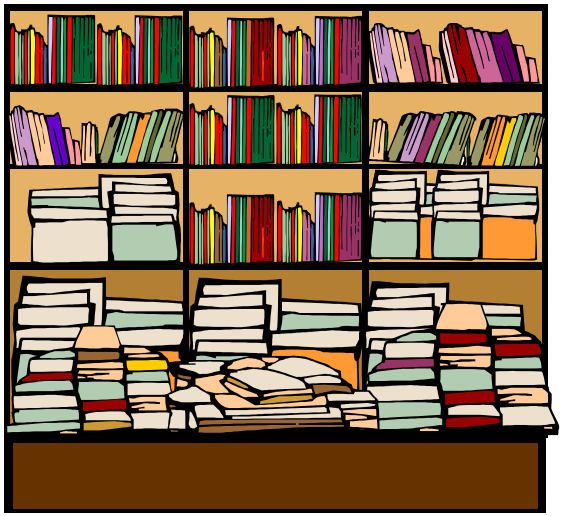
QoS

Web Process



State of the art in discovery

UDDI Business Registry

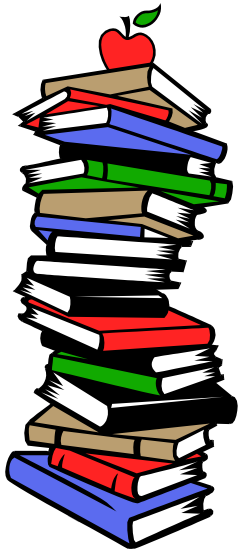


Provides non-semantic search

Search

Keyword and attribute-based match

Results



Search retrieves lot of services (irrelevant results included)

Selection

Which service to select ?
How to select?

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?)

Present Discovery Mechanism
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
- Services: 3 records found.
  - AirFares
    Returns air fares from netviagens.com travel agent
  - Hotel reservations
    Reservations for hotels in Asia, Australia and New Zealand
  - Your Vacation Specialists
    Web enabled vacation information
- Providers: 2 records found.

* Search carried out in one of the Universal Business Registries

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

- air ticket
  - 1 record with name air tickets booking
- airticket, ticketbooking, airtravel, air travel, travel agent, airticketbooking, air ticket booking, travel agency, travelagency
  - 0 records were returned
- travelagent
  - 1 record with name 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 cancallation of travel tickets
    - 1 service : Emergency Services for travellers
    - 1 service : Travel documentation and itinerary
    - 5 services : Description is ambiguous/not present

* Search carried out in one of the Universal Business Registries

The use of semantics
Benefits

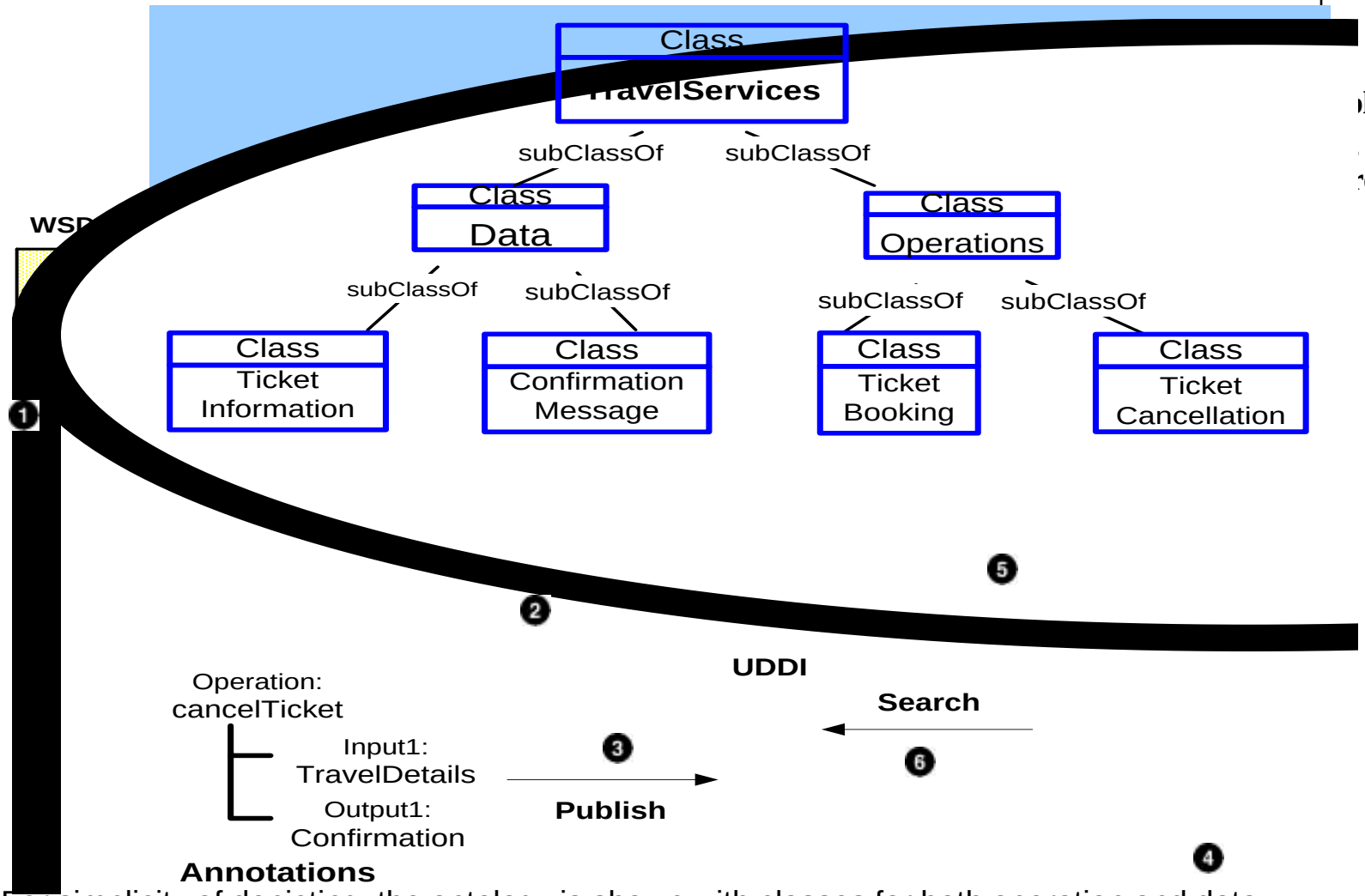
Web Service Discovery

- 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

Semantic Publication and Discovery



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

Discovery in Semantic Web
Using Semantics

Web Service Discovery

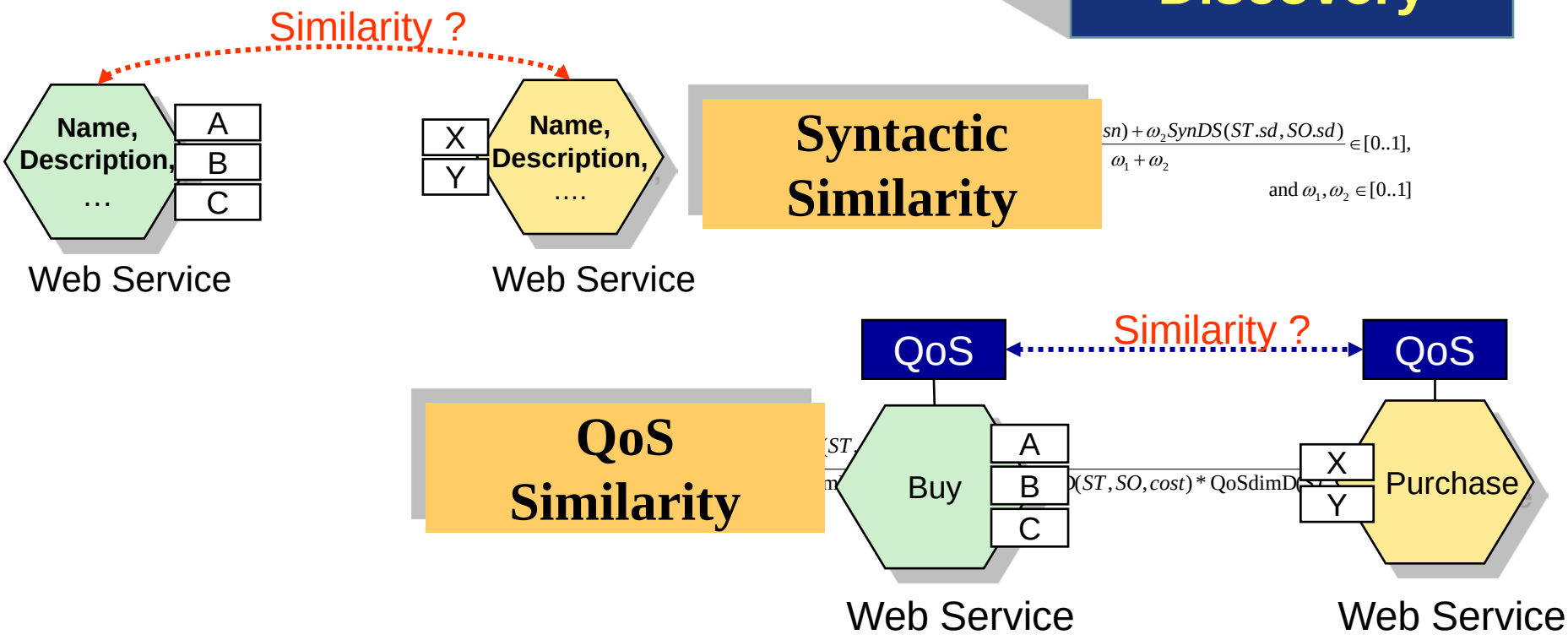
- Functionality: What capabilities the distributor expects from the service (Functional semantics)
- Inputs: What the distributor can give to the to the Manufacturer's service (Data semantics)
- Outputs: What the distributor expects as outputs from the service (Data semantics)
- QoS: Quality of Service the distributor expects from the service (QoS semantics)

(Functional semantics)
(Data semantics)
(QoS semantics)
(Syntactic description)

- Description: Natural language description of the service functionality (Syntactic description)

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

Web Service Discovery



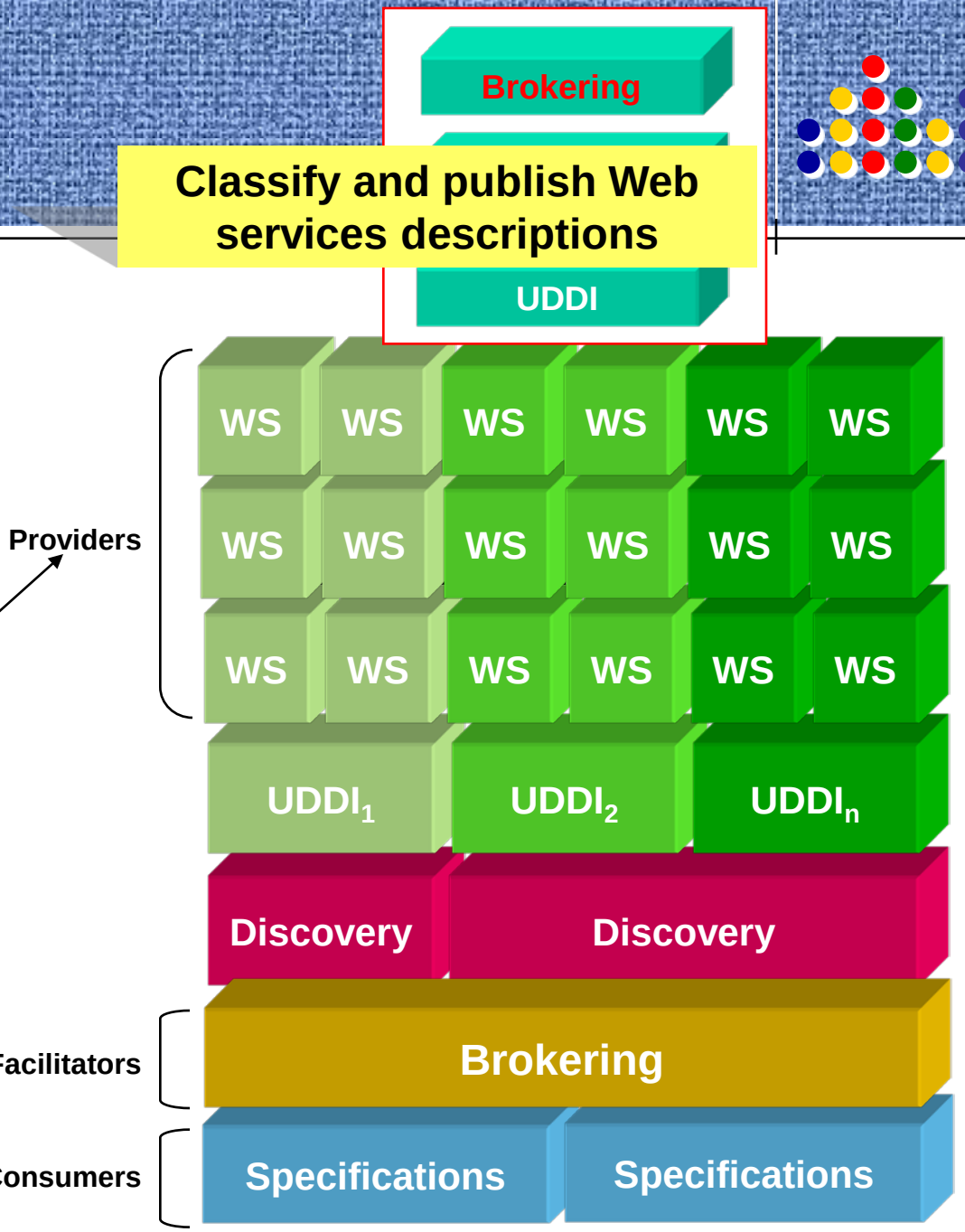
Brokering

The key players of brokering are the *service providers*, *service consumers*, and *facilitators*

Providers advertise their web services

Facilitators matches subscriptions to advertised services

Consumers register web services needs



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

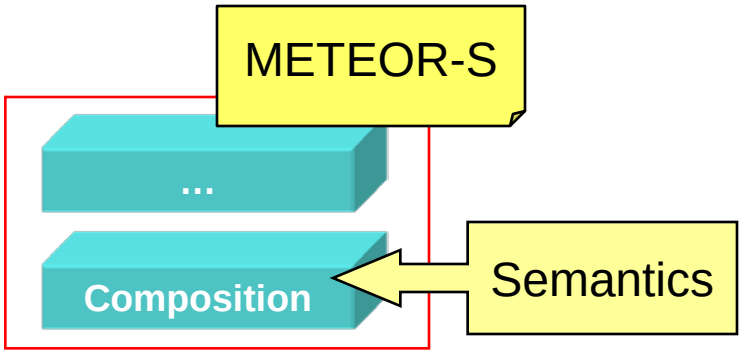
Brokering and Semantics

- 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



Semantics

Semantic Process Composition

Web 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

Composition of Web Processes

Web Process Composition

Web Service Discovery

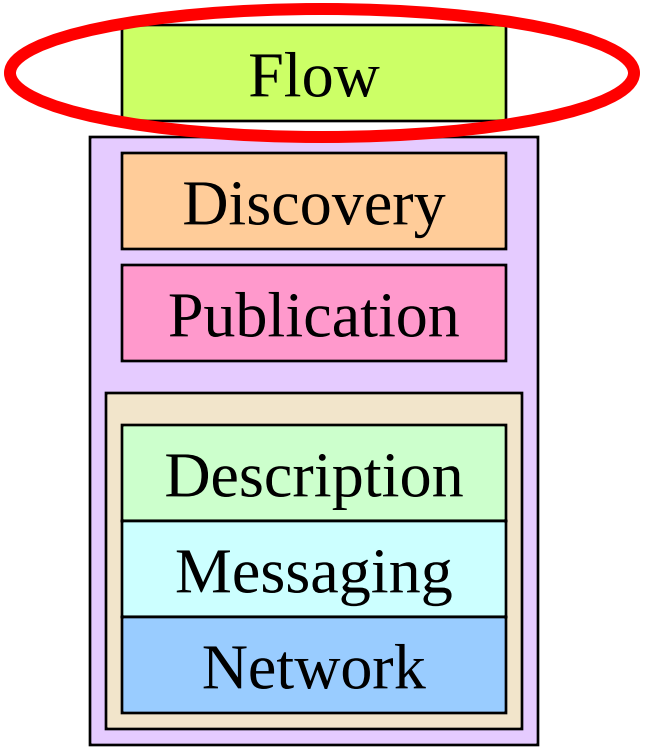
Web Service Integration

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

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

Semantics at Flow Layers

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:

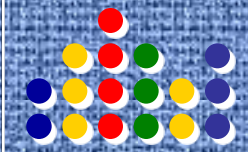
- Using
  - Functionality/preconditions/effects of the participating services
  - Knowledge of conversation patterns supported by the service
  - Formal mathematical models like process algebra, concurrency formalisms like State Machines, Petri nets etc.
  - Simulation techniques

Present Scenario:

- Composition of Web services is static.
- Dynamic service discovery, run-time binding, analysis and simulation are not supported directly

Integration
New Requirements

Web Process Composition



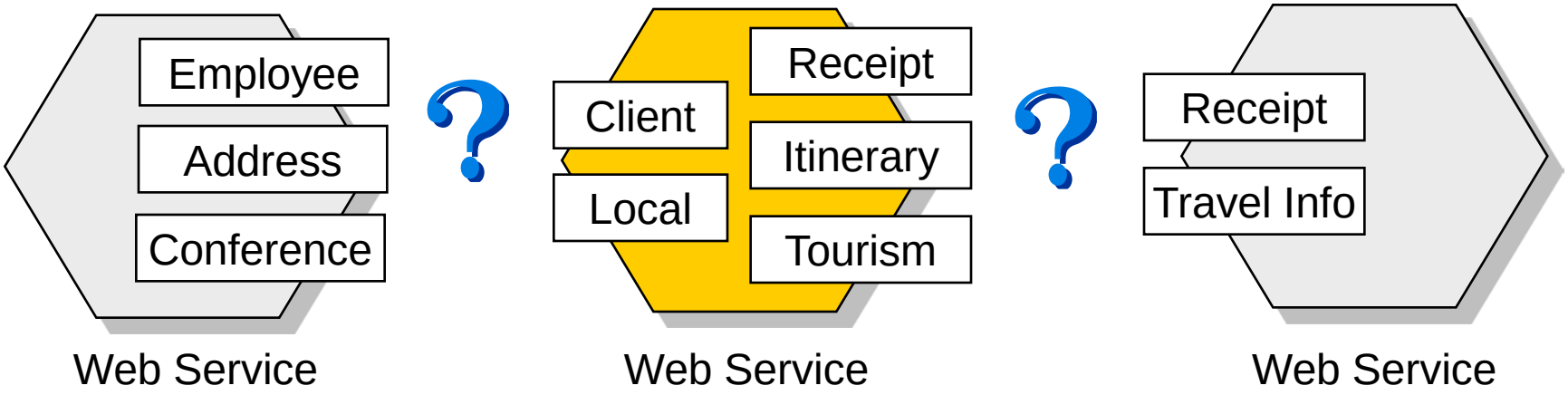
- When Web services are put together
  - Their interfaces need to interoperate.
  - Structural and semantic heterogeneity need to be resolved*.
- **Structural heterogeneity** exists because Web services use different data structures and class hierarchies to define the parameters of their interfaces.
- **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

Integration
New Requirements

Web Process Composition

How to establish data connections between Web Services interfaces?



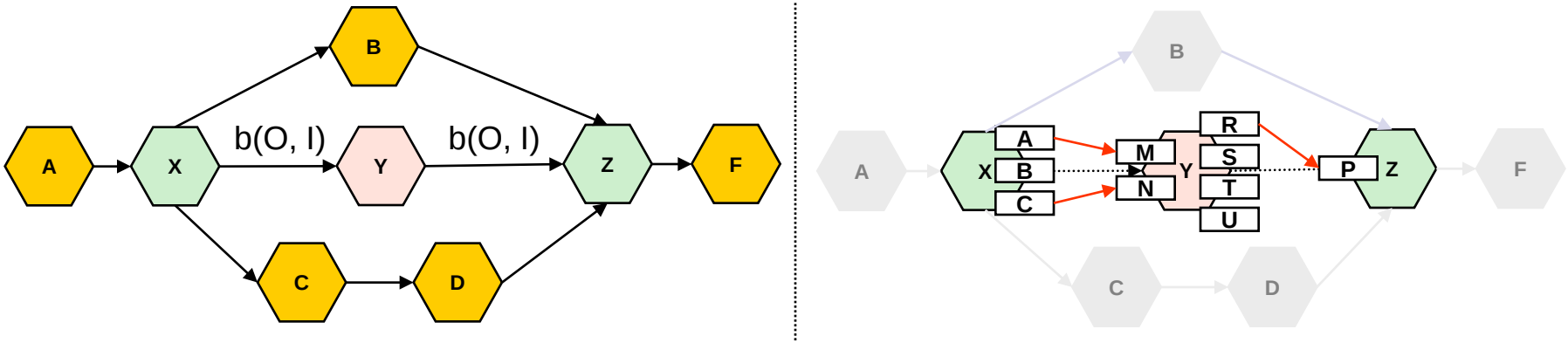
How to establish data connections between the different data structures and class hierarchies of the interface parameters?

How to understand the intended meaning of the terms used in labeling interface parameters?

Web Services
Interfaces

Web Process Composition

- 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)

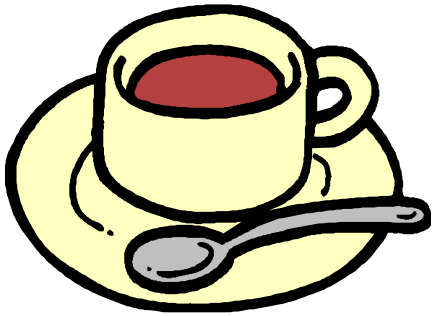


Bipartite graph. Each edge has a weight (semantic similarity).

Semantic Web Processes

Questions?

Semantic Web Processes



Coffee Break

10 Minutes

NEXT: Composition Languages

NEXT: METEOR-S

Composition Languages

- BPEL4WS
- DAML-S

BPEL4WS
Introduction

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 supercedes XLANG (Microsoft) and WSFL(IBM).
- It is build on top of WSDL.
  - For descriptions of what services do and how they work, BPEL4WS references port types contained in WSDL documents.

Web Services
Specification

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

* Fensel and Bussler 2002

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.

*http://www-106.ibm.com/developerworks/webservices/library/ws-coor/, http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/

BPEL4WS
Introduction

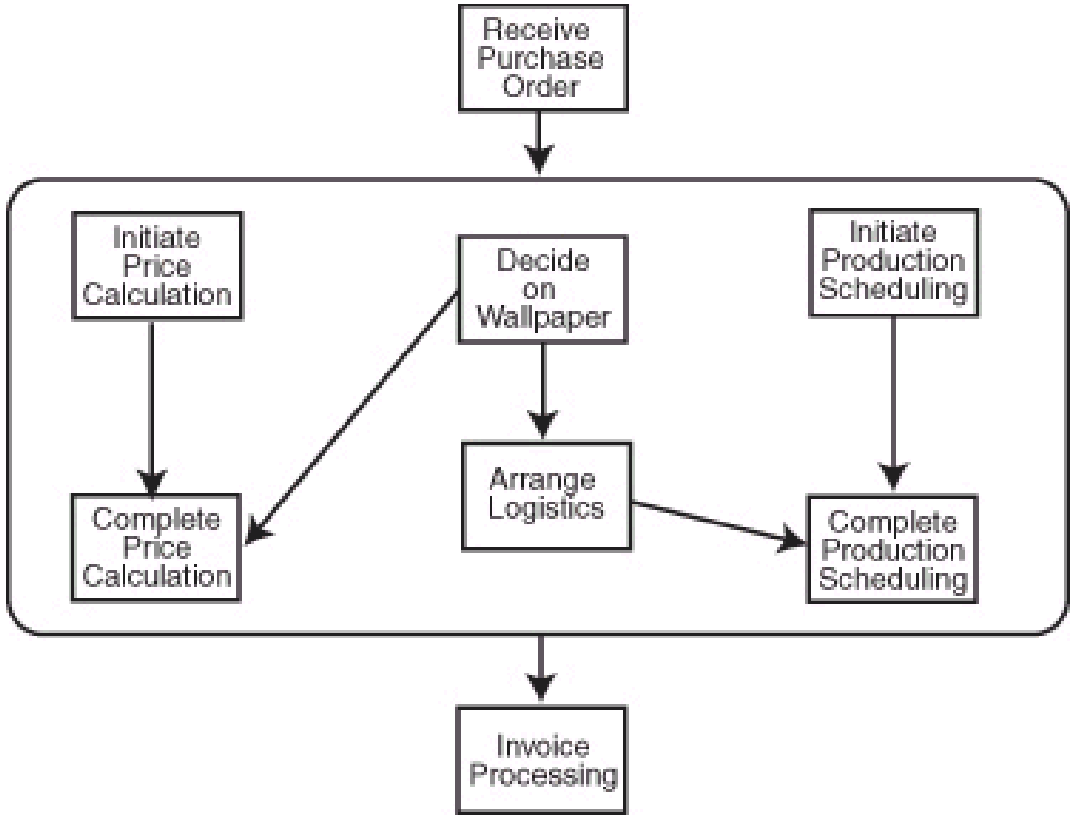
- BPEL4WS is a block-structured programming language, allowing recursive blocks but restricting definitions and declarations to the top level.
- The language defines activities as the basic components of a process definition.
- Structured activities prescribe the order in which a collection of activities 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.

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.

BPEL4WS
An Example

Let consider the following process.



[http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/]

BPEL4WS
An Example – WSDL definitions

```
<definitions targetNamespace="http://manufacturing.org/wsdl/purchase"
      xmlns:sns="http://manufacturing.org/xsd/purchase"
…
<message name="POMessage">
   <part name="customerInfo" type="sns:customerInfo"/>
   <part name="purchaseOrder" type="sns:purchaseOrder"/>
</message>
…
<message name="scheduleMessage">
   <part name="schedule" type="sns:scheduleInfo"/>
</message>

<portType name="purchaseOrderPT">
   <operation name="sendPurchaseOrder">
      <input message="pos:POMessage"/>
      <output message="pos:InvMessage"/>
      <fault name="cannotCompleteOrder"
             message="pos:orderFaultType"/>
   </operation>
</portType>
…
<slnk:serviceLinkType name="purchaseLT">
   <slnk:role name="purchaseService">
       <slnk:portType name="pos:purchaseOrderPT"/>
   </slnk:role>
</slnk:serviceLinkType>
…
</definitions>
```

Messages

The WSDL portType offered by the service to its customer

Roles

BPEL4WS
An Example – The process

```
<process name="purchaseOrderProcess"
         targetNamespace="http://acme.com/ws-bp/purchase"
…
   <partners>
      <partner name="customer"
               serviceLinkType="lns:purchaseLT"
               myRole="purchaseService"/>
   …
   </partners>

   <containers>
      <container name="PO" messageType="lns:POMessage"/>
      <container name="Invoice"
                 messageType="lns:InvMessage"/>
   …
   </containers>

   <faultHandlers>
      <catch faultName="lns:cannotCompleteOrder"
             faultContainer="POFault">
         <reply   partner="customer"
                  portType="lns:purchaseOrderPT"
                  operation="sendPurchaseOrder"
                  container="POFault"
                  faultName="cannotCompleteOrder"/>
      </catch>
   </faultHandlers>
…
```

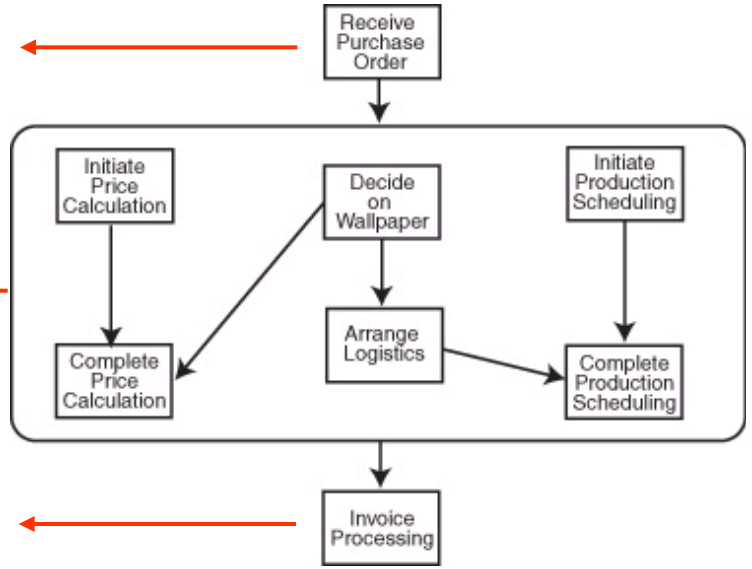
This section defines the different parties that interact with the business process in the course of processing the order.

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.

BPEL4WS
An Example – The process

```
…
    <sequence>

      <receive partner="customer"
               portType="lns:purchaseOrderPT"
               operation="sendPurchaseOrder"
               container="PO">
      </receive>

      <flow>
      …
      </flow>

      <reply partner="customer"
             portType="lns:purchaseOrderPT"
             operation="sendPurchaseOrder"
             container="Invoice"/>
   </sequence>

</process>
```



BPEL4WS
An Example – The process

```
<flow>
```

The flow construct provides concurrency and synchronization

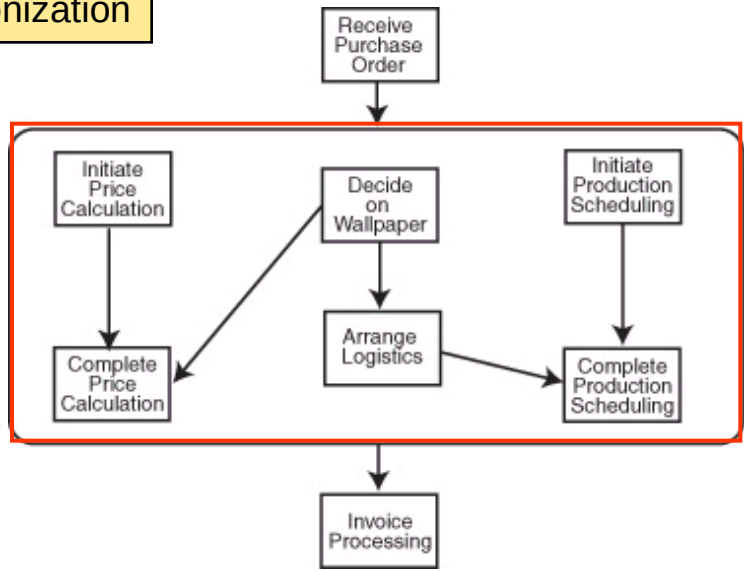
```
        <links>
           <link name="ship-to-invoice"/>
           <link name="ship-to-scheduling"/>
        </links>

        <sequence>
         …
```

Activities are executed sequentially

```
           <invoke  partner="shippingProvider"
                    portType="lns:shippingPT"
                    operation="requestShipping"
                    inputContainer="shippingRequest"
                    outputContainer="shippingInfo">
              <source linkName="ship-to-invoice"/>
           </invoke>

           <receive partner="shippingProvider"
                    portType="lns:shippingCallbackPT"
                    operation="sendSchedule"
                    container="shippingSchedule">
              <source linkName="ship-to-scheduling"/>
           </receive>
        </sequence>
…
<flow>
```

Activity Call

Activity call



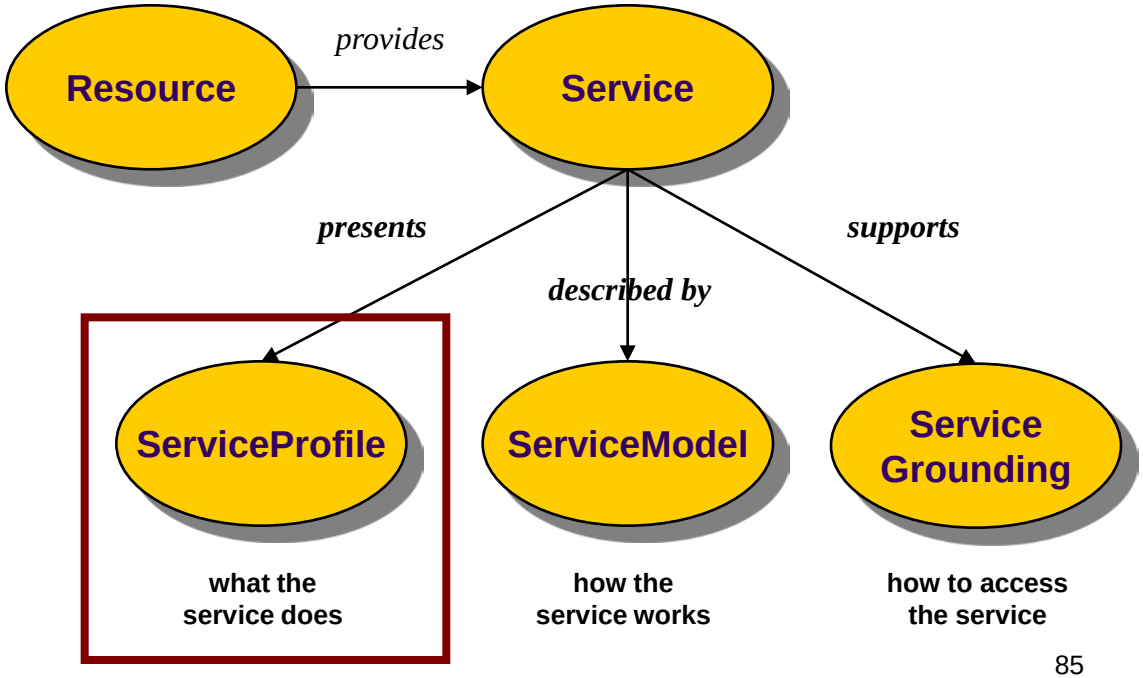
DAML-S
Introduction

DAML-S

- 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
Ontologies

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

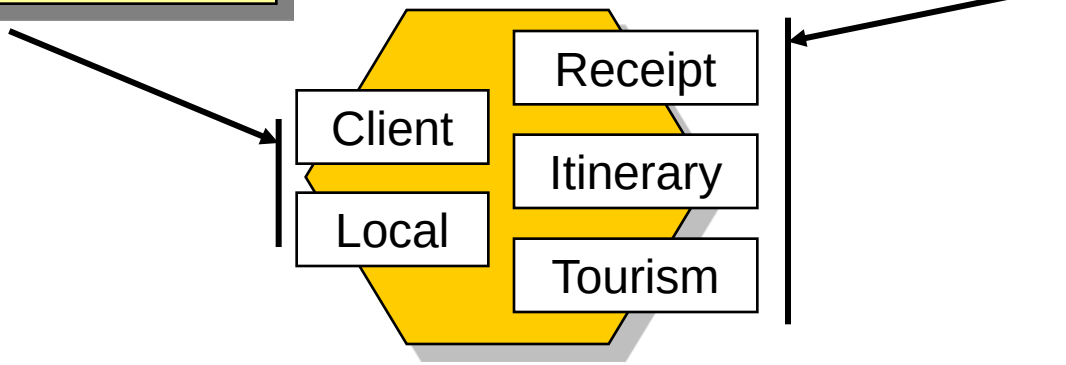


DAML-S
Service Profile

The Service Profile provides details about a service.

Inputs. Inputs that should be provided to invoke the service.

Outputs. Outputs expected after the interaction with the service.



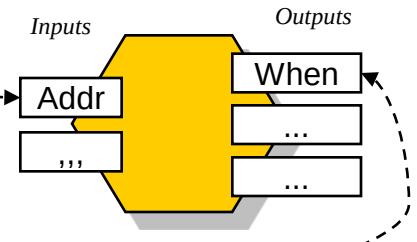
Web Service

Preconditions. Set of conditions that should hold prior to the service being invoked.

Effects. Set of statements that should hold true if the service is invoked successfully.

Service Profile
An example of Inputs and Outputs

```
...
<!ENTITY temporal "http://ovid.cs.uga.edu:8080/scube/daml/Temporal.daml">
<!ENTITY address "http://ovid.cs.uga.edu:8080/scube/daml/Address.daml">
...
<input>
    <profile:ParameterDescription rdf:ID="Addr">
    <profile:parameterName> Addr </profile:parameterName>
    <profile:restrictedTo rdf:resource="&address;#Address"/>
    <profile:refersTo rdf:resource="&congo;#congoBuyReceipt"/>
    </profile:ParameterDescription>
</input>
...
<output>
    <profile:ParameterDescription rdf:ID="When">
    <profile:parameterName> When </profile:parameterName>
    <profile:restrictedTo rdf:resource="&temporal;#Date"/>
    <profile:refersTo rdf:resource="&congo;#congoBuyReceipt"/>
    </profile:ParameterDescription>
< output >
...
```



BPEL4WS vs. DAML-S Comparison

- BPEL4WS relates closely to the ServiceModel (Process Model) component of DAML-S.
- DAML-S defines preconditions and effects
  - This enables the representation of side effects of Web services.
  - It also enables a better reasoning about the composition of services.
- DAML-S classes provide a richer representation of services
  - Classes allow reasoning draw properties from inheritance and other relationships to other DAML-S classes.

BPEL4WS vs. DAML-S Comparison

- The DAML-S ServiceProfile and ServiceModel provide sufficient information to enable
  - The automated discovery, composition, and execution based on well-defined descriptions of a service's inputs, outputs, preconditions, effects, and process model.
- BPEL4WS has complicated semantics for determining whether an activity actually happens in a block.
- BPEL4WS defines mechanisms for catching and handling faults and for setting compensation handlers.
- BPEL4WS includes WS-Coordination and WS-Transaction to provide a context for pre-defined transactional semantics.

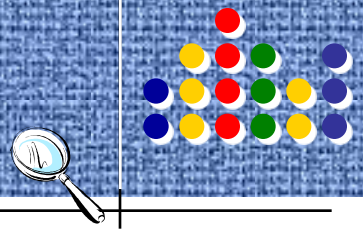
Semantic QoS

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.

Discovery
New Requirements

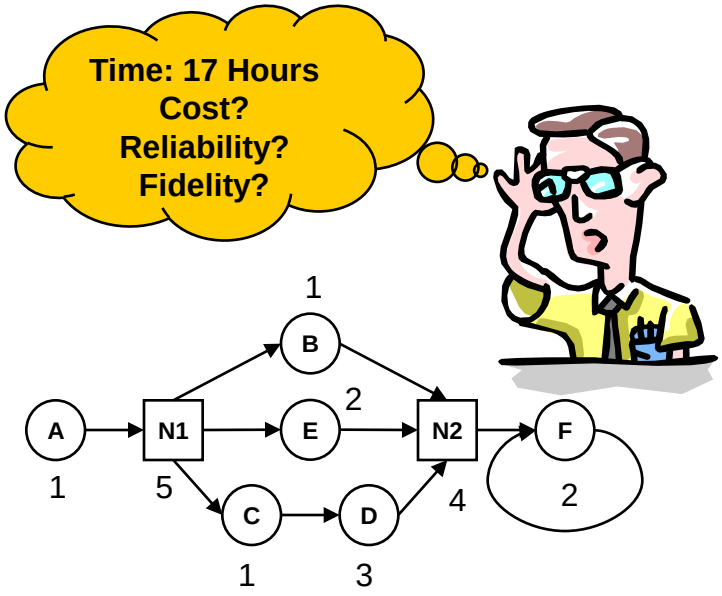


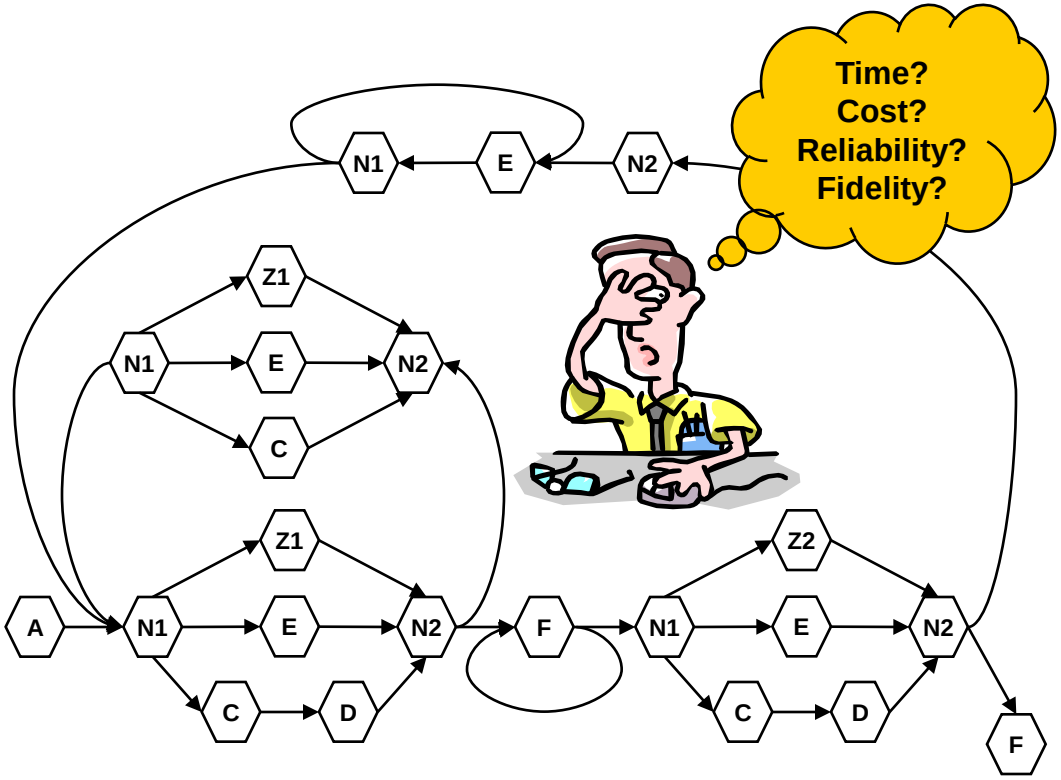
- The autonomy of Web services does not allow for designer to identify their operational metrics at design time.
- Nevertheless, when composing a process it is indispensable to inquire the Web services operational metrics.
- Operational metrics characterize the Quality of Service (QoS) that Web services exhibit when invoked.

QoS
New Requirements

Quality of Service

Before	Now





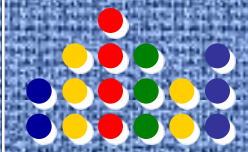
QoS Semantics

QoS

- 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

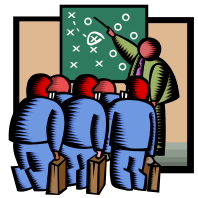
[Cardoso and Sheth, 2002]

QoS
Benefits



QoS

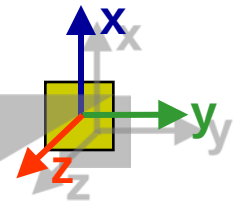
- **Composition** of processes according to QoS objective and requirements.
- **Selection and execution** of processes based on QoS metrics.
- **Monitoring** of processes to assure compliance with initial QoS requirements.
- **Evaluation** of alternative strategies when QoS requirements are violated.

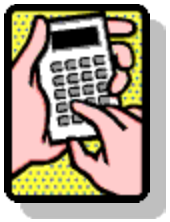


Semantic WP QoS
Research Issues

QoS

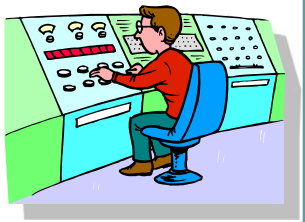
Specification. What dimensions need to be part of the QoS model for processes?





Computation. What methods and algorithms can be used to compute, analyze, and predict QoS?

Monitoring. What king of QoS monitoring tools need to be developed?

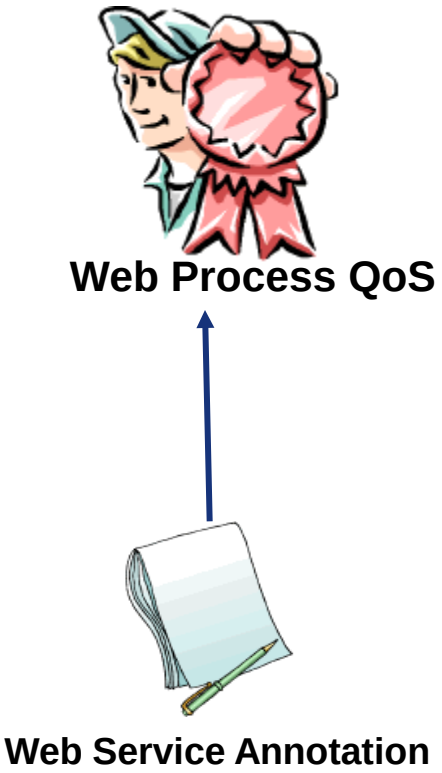


Control. What mechanisms need to be developed to control processes, in response to unsatisfactory QoS metrics?

Web Services
QoS Specification

QoS

- 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

QoS

A QoS Model describes non-functional properties of a process

Which dimensions should be part of a QoS model?

- Time
- Cost
- Fidelity
- Reliability

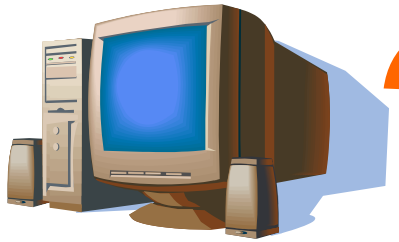
- Price
- Duration
- Repudiation
- Reliability
- Availability

- Security

QoS Models and Semantics

QoS

Use Semantics



? ?

Z#$%&!

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

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

QoS in METEOR-S

QoS

Design

QoS Model

QoS Estimates for Tasks/Web services

QoS Estimates for Transitions

SWR algorithm

Stochastic Process

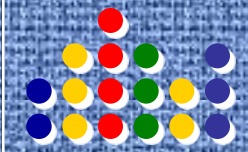
QoS Computation

Enact

Simulation

Log

QoS
Creation of Estimates

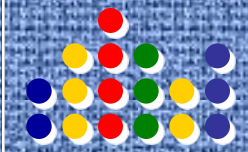


QoS

- 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.

QoS
Estimates for Web Services



QoS

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

QoS Model

Basic class

Distributional class

	Basic class			Distributional class
	Min value	Avg value	Max value	Dist. Function
Time	0.291	0.674	0.895	Normal(0.674, 0.143)
Cost	0	0	0	0.0
Reliability	-	100%	-	1.0
Fidelity.a_i	0.63	0.81	0.92	Trapezoidal(0.7,1,1,4)

Task QoS for an automatic task (SP FASTA task)

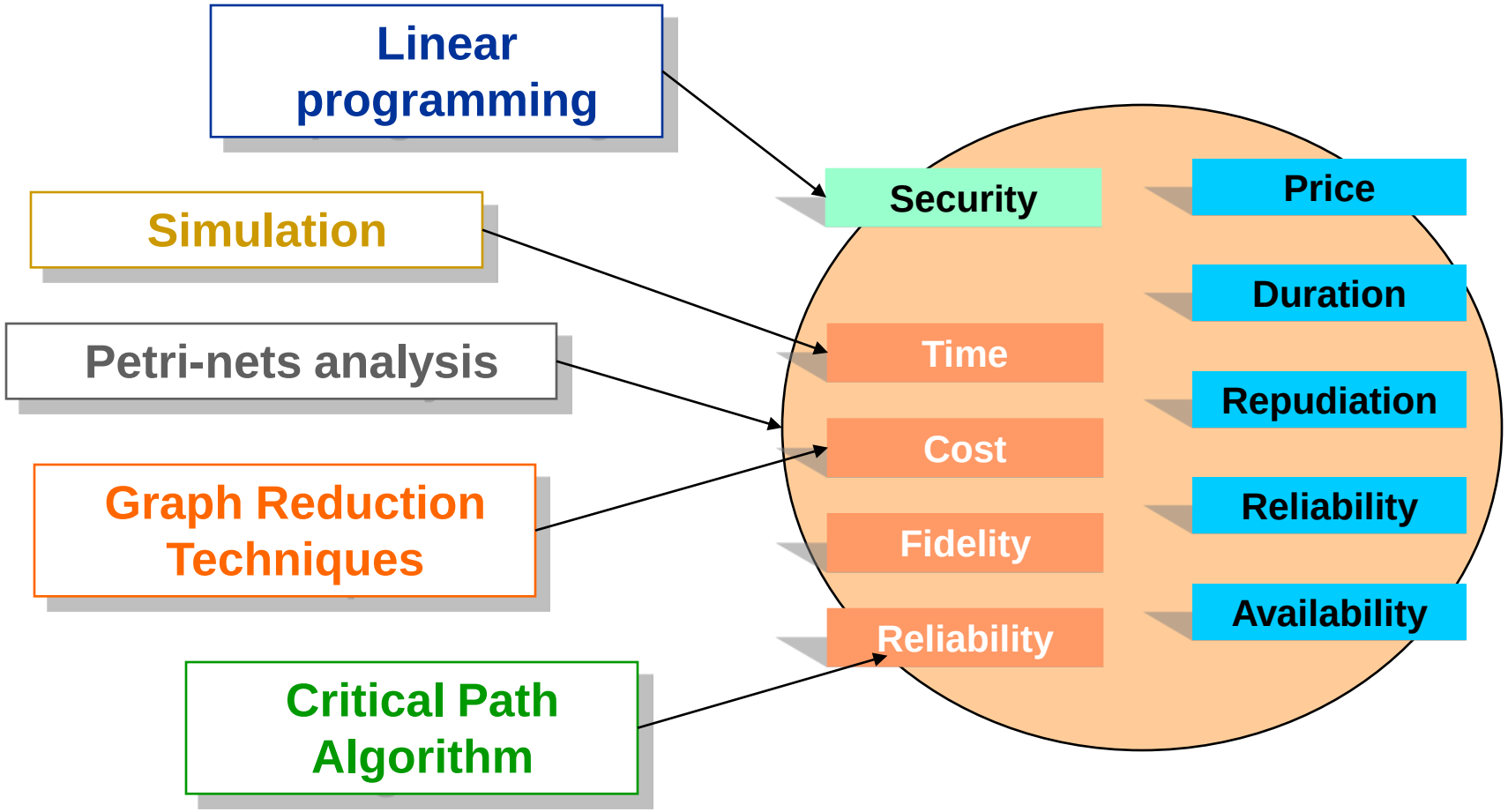
mathematical methods

simulation systems

Web process QoS computation

QoS

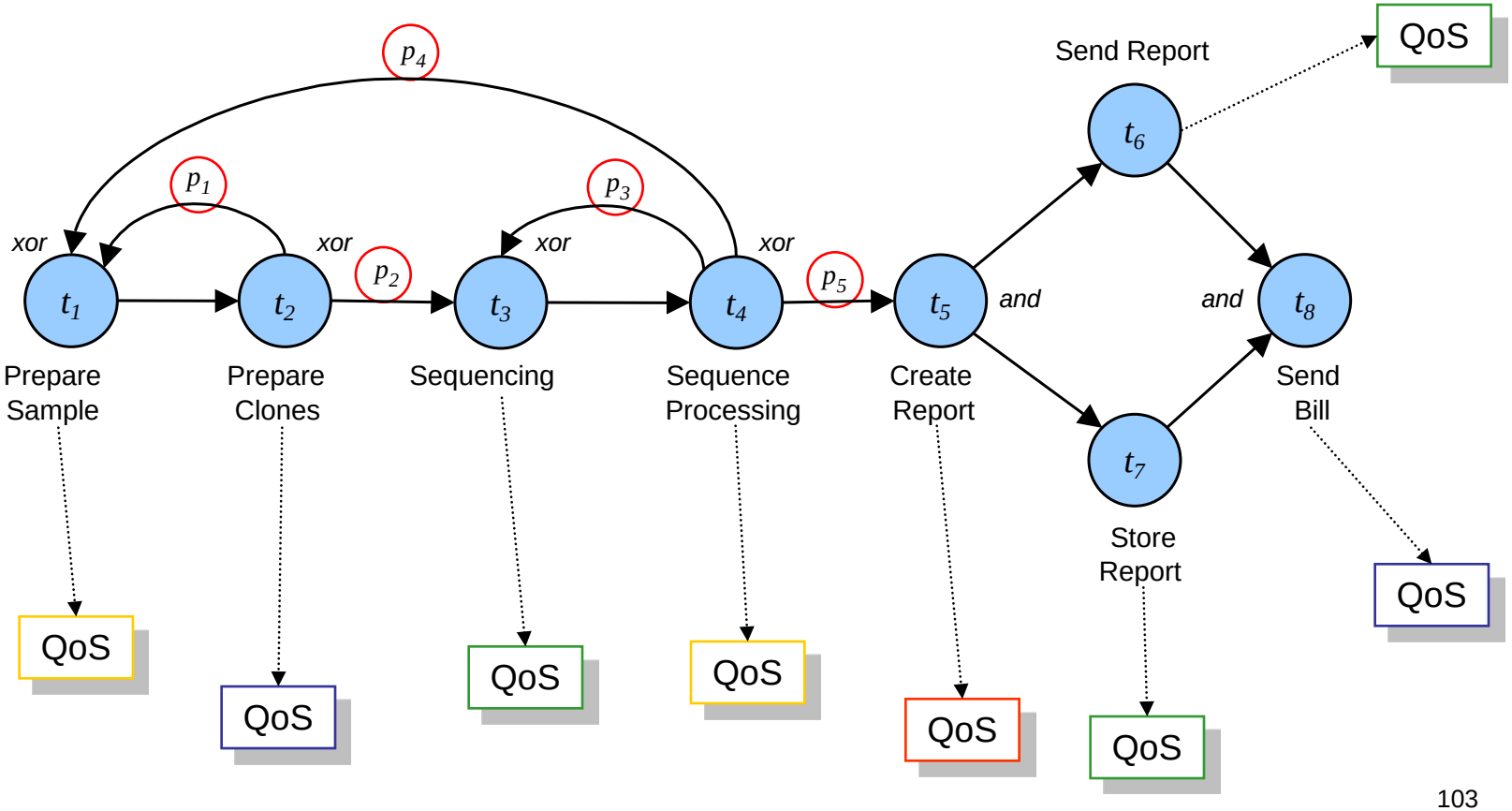
Design time| Runtime



QoS Computation

QoS

Graph Reduction Technique



QoS Computation

QoS

Graph Reduction Technique

(a) (b)

Reduction of a Sequential System

$$T(t_{ij}) = T(t_i) + T(t_j)$$

$$C(t_{ij}) = C(t_i) + C(t_j)$$

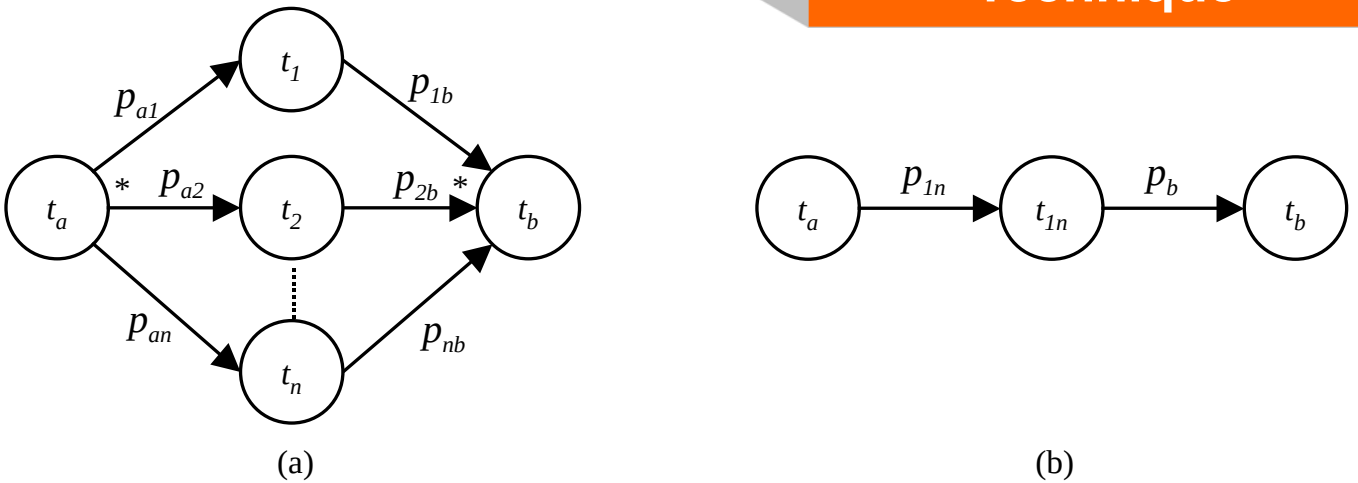
$$R(t_{ij}) = R(t_i) * R(t_j)$$

$$F(t_{ij}).a_r = f(F(t_i), F(t_j))$$

QoS Computation

QoS

Graph Reduction Technique



Reduction of a Parallel System

$$T(t_{1n}) = \text{Max}_{I \in \{1..n\}} \{T(t_i)\}$$

$$C(t_{1n}) = \sum_{1 \le i \le .n} C(t_i)$$

$$R(t_{1n}) = \prod_{1 \le i \le .n} R(t_i)$$

$$F(t_{1n}).a_r = f(F(t_1), F(t_2), \ldots, F(t_n))$$

QoS Computation

QoS

Simulation

- While mathematical methods can be effectively used, another alternative is to utilize simulation analysis[1].
- Simulation can play an important role in tuning the QoS metrics of processes by exploring "what-if" questions.
- In our project, these capabilities involve a loosely-coupled integration between the METEOR WfMS and the JSIM simulation system[2].

[1]Miller, Cardoso *et al.* 2002, [2]Nair, Miller *et al.* 1996; Miller, Nair *et al.* 1997; Miller, Seila *et al.* 2000.

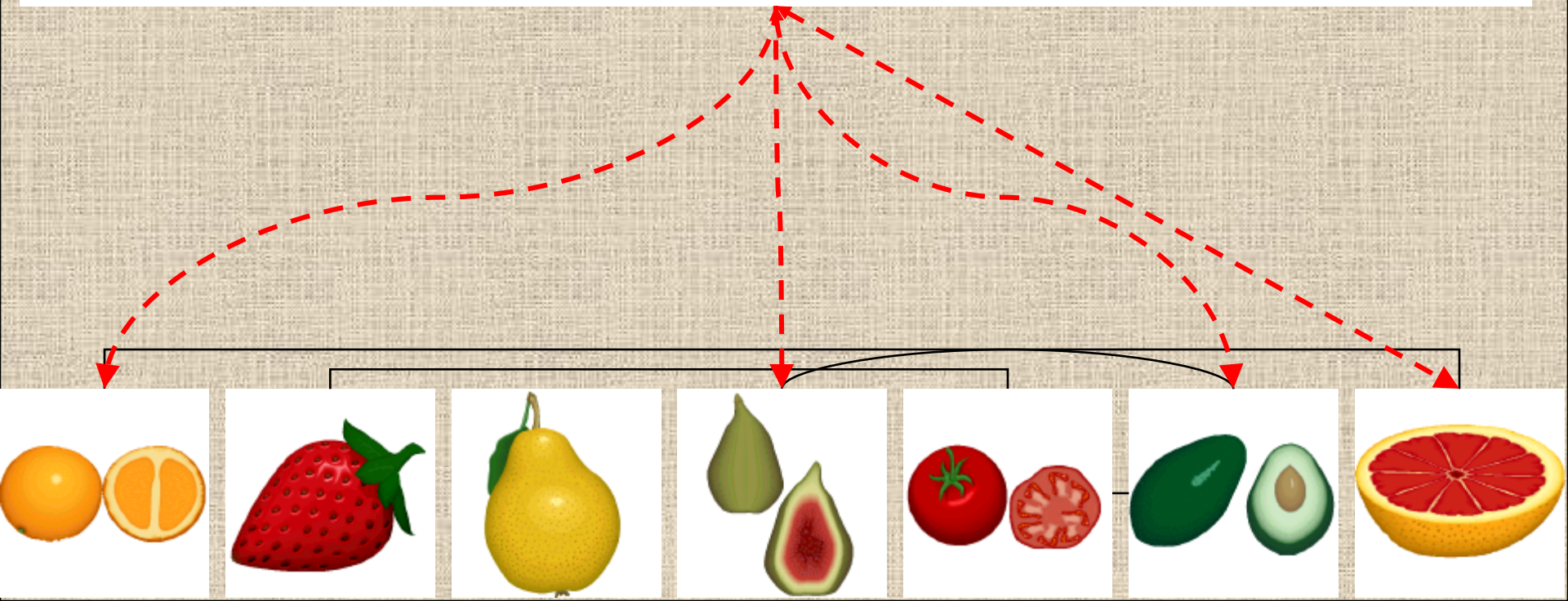
Semantic Web Processes

Questions?

NEXT: METEOR-S Project @ LSDIS lab

Systems and Applications

METEOR-S Project @ LSDIS lab



METEOR-S Project @ LSDIS lab

- METEOR-S exploits Workflow, Semantic Web, Web Services, and Simulation technologies to meet these challenges in a practical and standards based approach.
  - Applying Semantics in Annotation, Quality of Service, Discovery, Composition, Execution of Web Services
  - Adding semantics to different layers of Web services conceptual stack
  - Use of ontologies to provide underpinning for information sharing and semantic interoperability

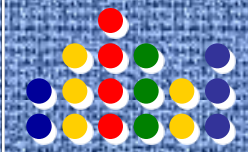
http://swp.semanticweb.org, http://lsdis.cs.uga.edu/proj/meteor/swp.htm

METEOR-S components for Semantic Web Services

- **Discovery Infrastructure** (MWSDI)
  - Semantic Annotation and Discovery of Web Services [1]
  - Semantic Peer-to-Peer network of Web Services Registries [2]
- **Composer**
  - SCET: Service Composition and Execution Tool [3]
  - **Semantics Process Template Builder and Process Generator** [4]
  - QoS Management
    - Specify, compute, monitor and control QoS (SWR algorithm) [5]
- **Orchestrator** (Under development)
  - Analysis and Simulation [6]
  - Execution
  - Monitoring [6]

[1] [Sivashanmugam et al.-1], [2] [Verma et al.], [3] [Chandrasekaran et al.], [4] [Sivashanmugam et al.-2], [5] [Cardoso et al.], [6] [Silver et al.]

METEOR-S Web Service Discovery Infrastructure (MWSDI)



- uses 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

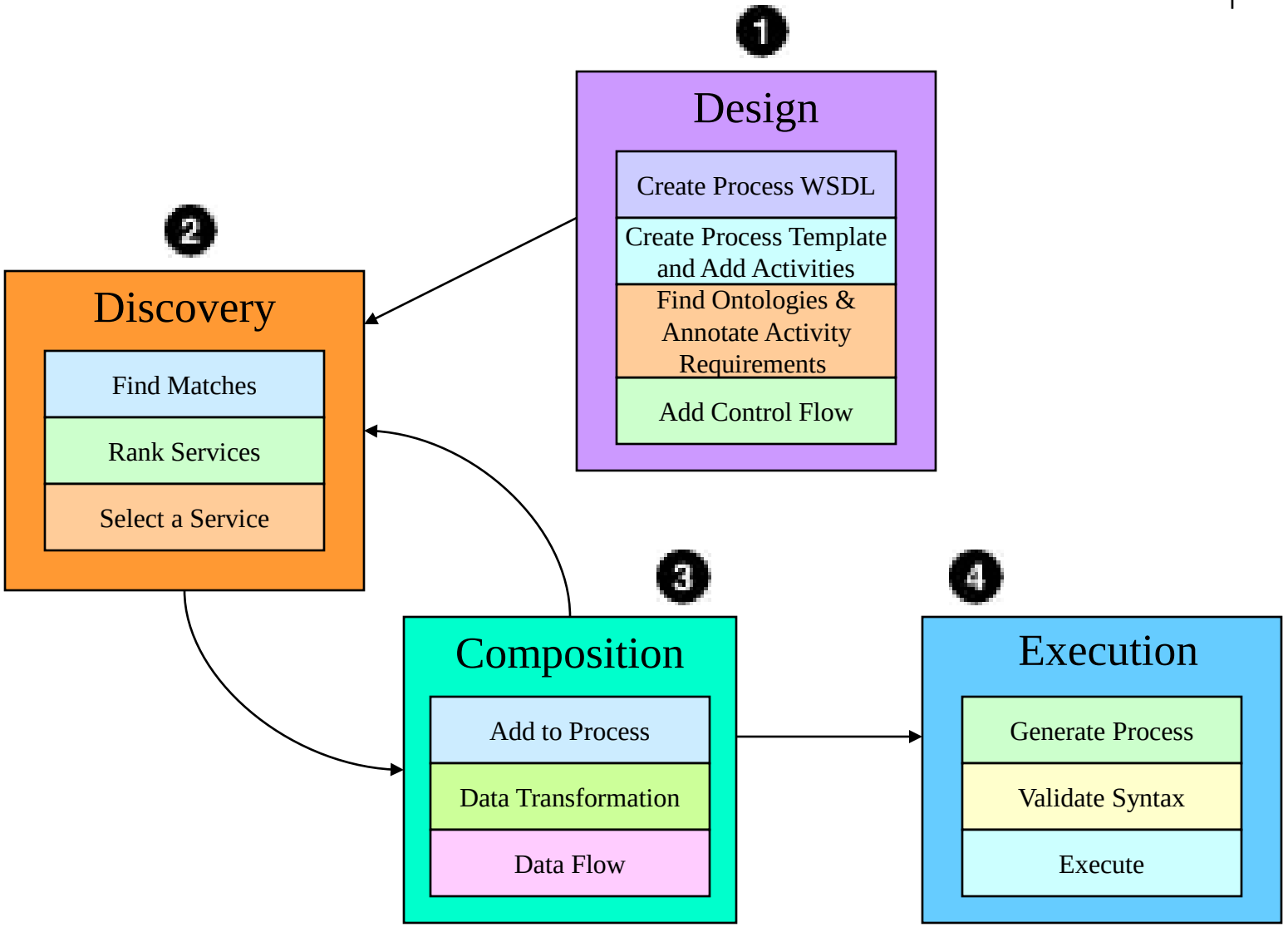
Process Designer

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

Web Process Life-Cycle



Semantic Web Process Design

Semantic Web Process Designer

View Process WSDL | View Template | Generate Process | View BPEL Tree | List Ontologies

Control Flow | Data Flow | Process Variables | Service Selection | List Activities

Process Details | Add Web Services | Add Activity Interface | Add Semantic Activity Template | Interface Browser

Activity Name: ierySupplierPartner

Decomposable: ☐

Ontology URL: .edu/~kaarthik/LSDIS-FunctionalOnt.daml

Operation Concept: eForOrderToyParts

Disovery URL: ~server/RegistryServerServlet

Discovery Specifications: C:\Thesis\discovery\disc3.xml [Open]

Ranking Details: C:\Thesis\ranking\rank1.xml [Open]

Qos Specifications: C:\Thesis\qos\qos5.xml [Open]

Add Message | Add Precondition | Add Effect

Collect | Update | Show Services

MessagePart Name: input-1

MessagePart Category: Input

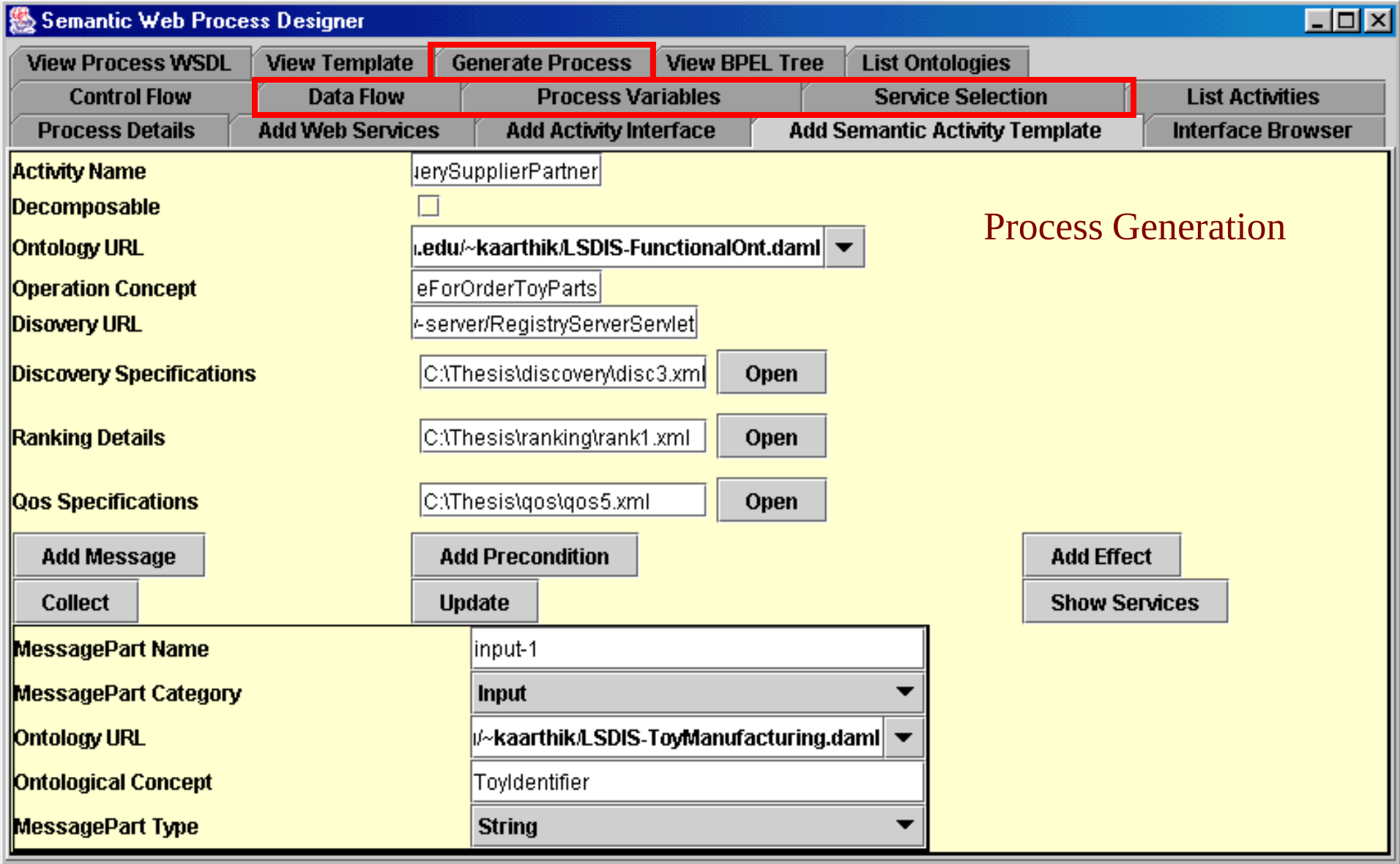
Ontology URL: l/~kaarthik/LSDIS-ToyManufacturing.daml

Ontological Concept: ToyIdentifier

MessagePart Type: String

Template Construction

Semantic Web Process Design



Process Generation

Semantic Web Process Design

Semantic Web Process Designer

View Process WSDL | View Template | Generate Process | View BPEL Tree | List Ontologies

Control Flow | Data Flow | Process Variables | Service Selection | List Activities

Process Details | Add Web Services | Add Activity Interface | Add Semantic Activity Template | Interface Browser

Activity Name	ıerySupplierPartner
Decomposable	☐
Ontology URL	ı.edu/~kaarthik/LSDIS-FunctionalOnt.daml ▼
Operation Concept	eForOrderToyParts
Disovery URL	~server/RegistryServerServlet
Discovery Specifications	C:\Thesis\discovery\disc3.xml Open
Ranking Details	C:\Thesis\ranking\rank1.xml Open
Qos Specifications	C:\Thesis\qos\qos5.xml Open

Add Message | Add Precondition | Add Effect

Collect | Update | Show Services

MessagePart Name	input-1
MessagePart Category	Input ▼
Ontology URL	ı/~kaarthik/LSDIS-ToyManufacturing.daml ▼
Ontological Concept	ToyIdentifier
MessagePart Type	String ▼

Semantic Web Process Design

Semantic Web Process Designer

View Process WSDL | View Template | Generate Process | View BPEL Tree | List Ontologies

Control Flow | Data Flow | Process Variables | Service Selection | List Activities

Process Details | Add Web Services | Add Activity Interface | Add Semantic Activity Template | Interface Browser

Update Activities | Hotel | List Services | Select Service | Save Details

Business Name	Service Name	Operation Name	WSDL URL	Ranking Value
BusinessNo6	HotelReservation	bookHotel	http://lsdis.cs.uga.edu/proj/meteors/wsdls/Hotel...	0.666666666...
BusinessSeven	Business7HotelService	bookHotel	http://lsdis.cs.uga.edu/proj/meteors/wsdls/Hotel...	0.733333333...
Demo1_NewBusiness2	TestHotelService2	bookHotel	http://lsdis.uga.edu/proj/meteors/wsdls/DontSel...	0.333333333...
Demo1_NewBusiness3	TestHotelService3	bookHotel	http://lsdis.uga.edu/proj/meteors/wsdls/DontSel...	0.333333333...
Demo1_NewBusiness1	TestHotelService1	bookHotel	http://lsdis.uga.edu/proj/meteors/wsdls/HotelSer...	0.666666666...
BusinessSeven	Business7HotelService	bookHotel	http://lsdis.cs.uga.edu/proj/meteors/wsdls/Hotel...	0.733333333...
Demo1_NewBusiness2	TestHotelService2	bookHotel	http://lsdis.uga.edu/proj/meteors/wsdls/DontSel...	0.333333333...
Demo1_NewBusiness3	TestHotelService3	bookHotel	http://lsdis.uga.edu/proj/meteors/wsdls/DontSel...	0.333333333...

Semantic Web Process Design

Semantic Web Process Designer

View Process WSDL | View Template | Generate Process | View BPEL Tree | List Ontologies

Control Flow | Data Flow | Process Variables | Service Selection | List Activities

Process Details | Add Web Services | Add Activity Interface | Add Semantic Activity Template | Interface Browser

Source	From	Target	To	Expression
Assembly	(http://www.w3.org/2001/XMLSchema) : OutDate	RawMaterialDeliver...	(http://www.w3.org...	☐
Expr	'AL-465'	RawMaterialDeliver...	(http://www.w3.org...	☑
				☐
				☐
				☐
				☐
				☐
				☐

Save | Assign | Clear

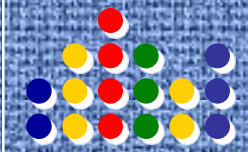
Source Activity: Assembly ▼ Target Activity: RawMaterialDeliveryInterface ▼ Load Activities

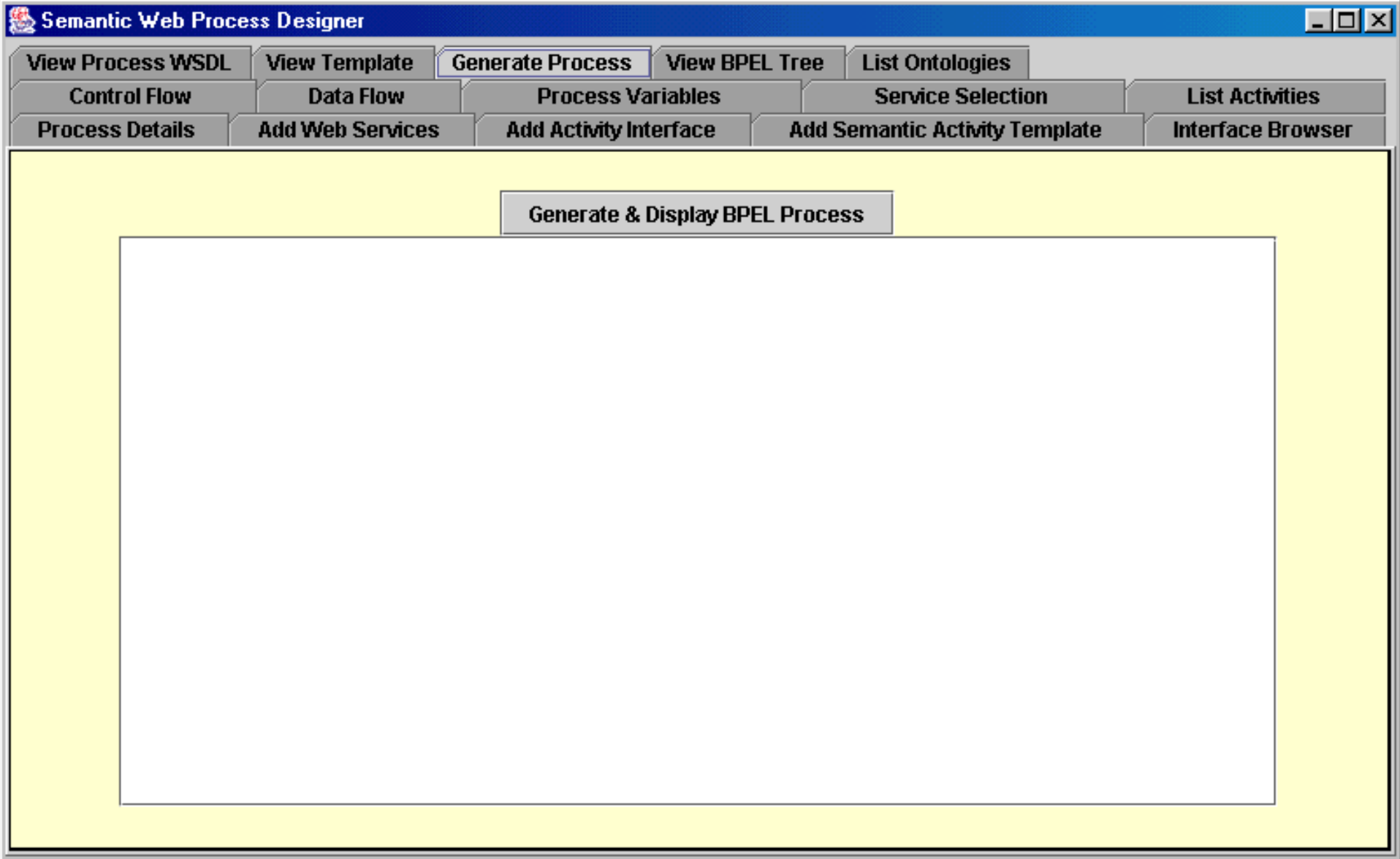
- Service
  - assemblyLine
    - Output Messages
      - (http://www.w3.org/2001/XMLSchema) : OutDate

Input Messages

- (http://www.w3.org/2001/XMLSchema) : DelvieryLocation
- (http://www.w3.org/2001/XMLSchema) : PickupDate
- (http://www.w3.org/2001/XMLSchema) : PickupLocation
- (http://www.w3.org/2001/XMLSchema) : DeliveryMeans

Semantic Web Process Design





Semantic Web Process Design

Semantic Web Process Designer

View Process WSDL | View Template | Generate Process | View BPEL Tree | List Ontologies

Control Flow | Data Flow | Process Variables | Service Selection | List Activities

Process Details | Add Web Services | Add Activity Interface | Add Semantic Activity Template | Interface Browser

Generate & Display BPEL Process

```
<?xml version="1.0" encoding="UTF-8"?>
<process xmlns="http://schemas.xmlsoap.org/ws/2002/07/business-process/" xmlns:NS1="http://lsdis.cs.uga.edu/Confe
 <partners><partner name="caller" serviceLinkType="NS1:sampleConferenceArrangerSLT"/><partner name="service-pro
 <containers>
  <container messageType="NS1:arrange4ConferenceRequest" name="receive"/>
  <container messageType="NS2:getConferenceDetailsRequest" name="ConferenceDetails-request"/>
  <container messageType="NS2:getConferenceDetailsResponse" name="ConferenceDetails-response"/>
  <container messageType="NS3:bookHotelRequest" name="Hotel-request"/>
  <container messageType="NS3:bookHotelResponse" name="Hotel-response"/>
  <container messageType="NS4:bookAirTicketRequest" name="AirTicketTo-request"/>
  <container messageType="NS4:bookAirTicketResponse" name="AirTicketTo-response"/>
  <container messageType="NS4:bookAirTicketRequest" name="AirTicketReturn-request"/>
  <container messageType="NS4:bookAirTicketResponse" name="AirTicketReturn-response"/>
  <container messageType="NS1:arrange4ConferenceRequest" name="response"/>
 </containers>

 <sequence name="sequence-1">
  <receive container="receive" createInstance="yes" name="receive" operation="arrange4Conference" partner="caller" po
  <assign name="ConferenceDetails">
  <copy><from container="receive" part=" ConferenceId"/><to container="ConferenceDetails-request" part=" ConferenceId
```

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

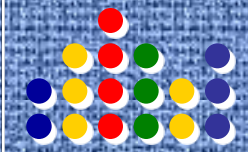
- **DAML-S:** http://www.daml.org/services/
  - Set of ontologies to describe functionalties of web services
- **DAML-S Matchmaker:** http://www-2.cs.cmu.edu/%7Esoftagents/daml_Mmaker/daml-s_matchmaker.htm
  - Match service requestors with service providers
  - Semantic Matchmaking for Web Services Discovery
- **Web Service Composer:** http://www.mindswap.org/~evren/composer/
  - Semi-automatic process for the dynamic composition of web services
- **Web Services:** http://www-106.ibm.com/developerworks/webservices/
  - WSDL, UDDI, SOAP
  - Business Process with BPEL4WS

Conclusions

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

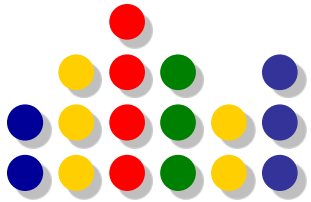
More at: http://lsdis.cs.uga.edu/proj/meteor/SWP.htm

Semantic Web Processes

Questions?

Web Resource for this tutorial

(incl. latest version)



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

References

DAML

http://www.daml.org/services/

http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/

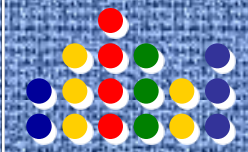
http://www.daml.org/2001/03/daml+oil-index

http://www-106.ibm.com/developerworks/webservices/library/ws-coor/

http://www-106.ibm.com/developerworks/webservices/library/ws-transpec/

http://www.ksl.stanford.edu/projects/DAML/Webservices/DAMLS-BPEL.html

References



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

- [Kreger] http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf
- [Sivashanmugam et al.-1] Adding Semantics to Web Services Standards
- [Sivashanmugam et al.-2] Framework for Semantic Web Process Composition
- [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
- [UDDI-v3] http://uddi.org/pubs/uddi-v3.00-published-20020719.htm

Semantic Web Processes

End