BPEL Complementary

Ingegneria dei Processi Aziendali

Modulo 1 - Servizi Web

Unità didattica 1 – Protocolli Web

Ernesto Damiani

Università di Milano

Agenda

- **Business Process 101**
- **Business Processes Automation & Web Service Standards**
- **WS-BPEL Overview**
- **Intalio BPMS BPEL Platform**
- **Future Directions for WS-BPEL Adoption**

Business Processes Defined

A business process is a collection of interrelated tasks, which are designed to deliver a particular result

Types of business processes

- Management processes - processes that govern the operation of a system
- Operational processes - processes that constitute the core activities of the business and delivers the primary value of the organization
- Supporting processes - which support the core processes

A business process can be decomposed into several sub-processes, which have their own attributes, but are aligned with the goal of the overall process

The analysis of business processes typically includes the mapping of processes and sub-processes down to an activity level

Business processes can be automated through BPM software and workflow

Business Process Automation

Key Questions for Workflow Design

- Who Should?
  - Who is involved in the process
- Do What?
  - What tasks/activities need to be performed
- To What?
  - What entities (objects) and data involved
- When?
  - What starts and stops the process
- In What Order?
  - What is the sequence and interrelationship between tasks
- Why?
  - What is the value proposition of using workflow

Workflow Patterns (workflowpatterns.com)

Basic Control Flow

- Sequence
- Parallel Split
- Synchronization
- Exclusive Choice

Advanced Branching and Synchronization

- Multi-Choice
- Structured Synchronizing Merge
- Multi-Merge
- Structured Discriminator
- Blocking Discriminator

Multiple Instance Patterns

- Multiple Instances without Synchronization
- Multiple Instances with a Priori Design-Time Knowledge
- Multiple Instances with a Priori Run-Time Knowledge

State-Based Patterns

- Deferred Choice
- Interleaved Parallel Routing
- Milestone

Cancellation and Force Completion Patterns

Cancel Task
Cancel Case

Interation Patterns

Arbitrary Cycles
Structured Loop
Recursion

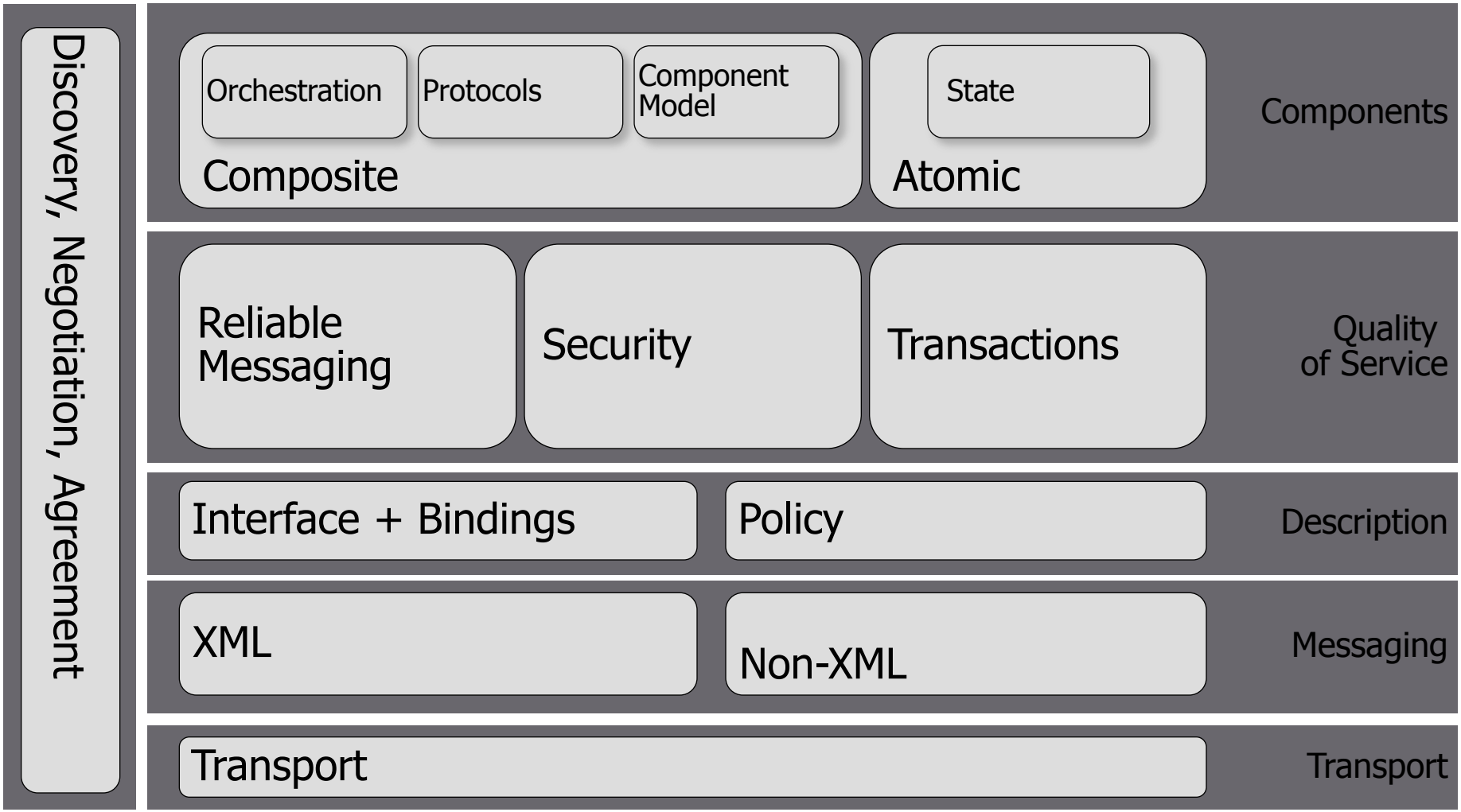
Termination Patterns

Implicit Termination
Explicit Termination

Trigger Patterns

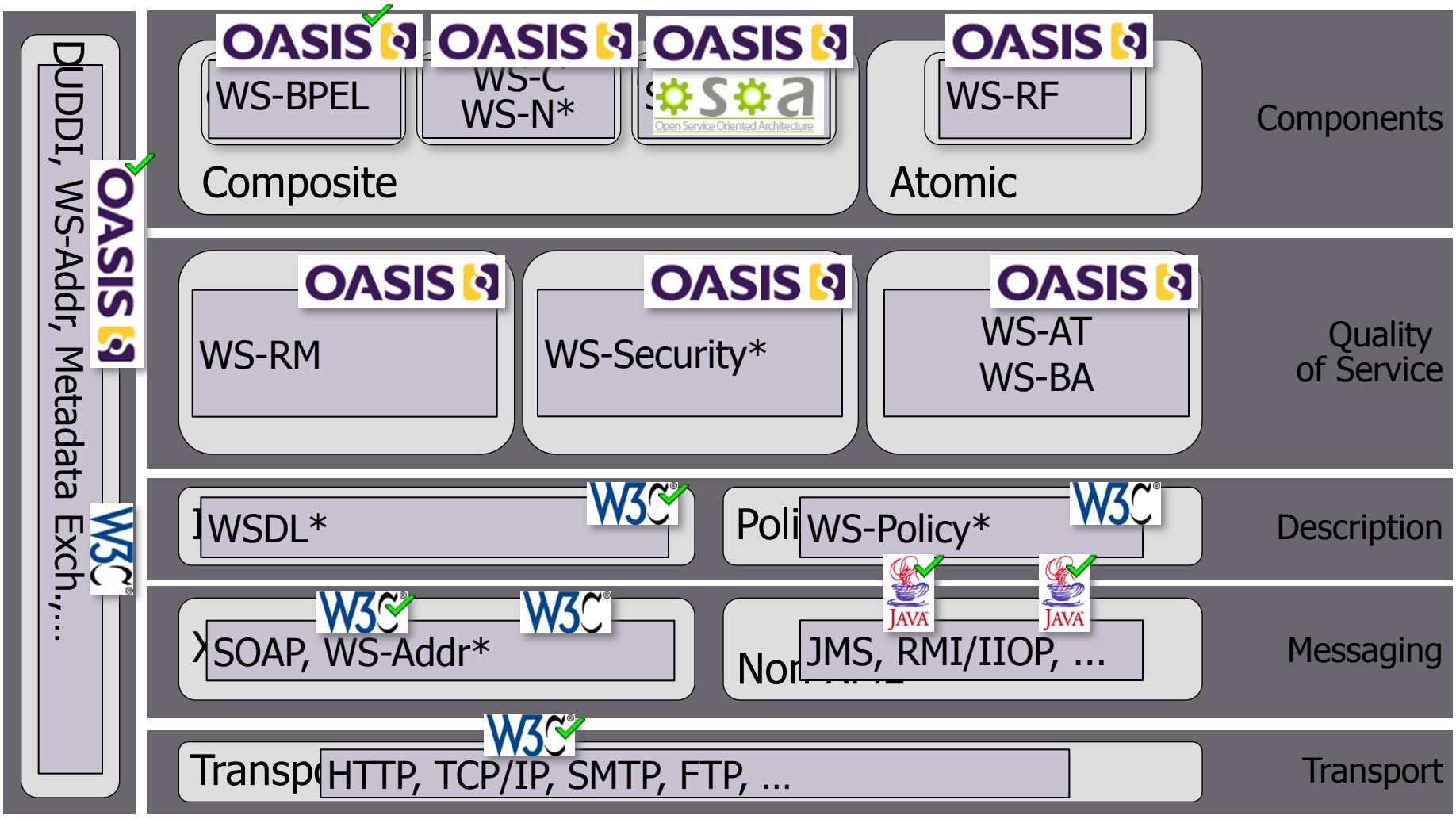
Transient Trigger
Persistent Trigger

Web Services standards for SOA



Web Services standards for SOA

Web Services standards for SOA



Business Process Execution Language (BPEL)

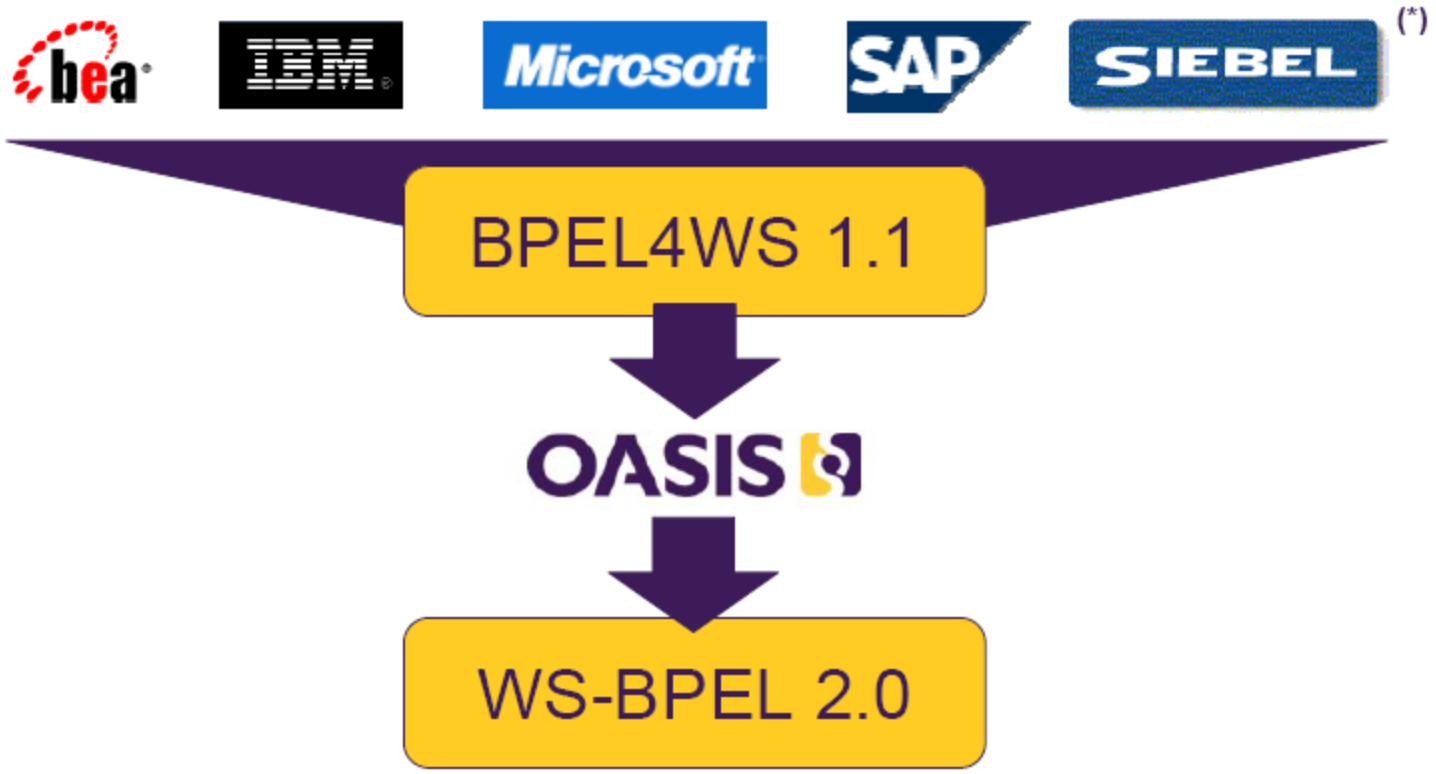
Web Services Business Process Execution Language (WS-BPEL) is a language for describing business processes based on Web Services

- Processes described using WS-BPEL execute functionality by using Web Service interfaces exclusively
- WS-BPEL Specification is administered by OASIS

WS-BPEL is an orchestration language, not a choreography language

- Orchestration specifies an executable process that involves message exchanges with other systems, such that that the message exchange sequences are controlled by the orchestration designer.
- Choreography specifies a protocol for peer-to-peer interactions, defining the legal sequences of messages exchanged with the objective of guaranteeing interoperability
- A choreography is not directly executable
- A choreography can be implemented through an orchestration (i.e. a BPEL process)

BPEL Standard Sponsorship



(*) BPEL4WS 1.1 authors, May 2003

WS-BPEL Language Constructs

- WS-BPEL process definition
- Recursive composition and partner links
- Variables
- Correlation sets
- Basic and structured activities
- Scopes
- Compensation handling

Partner Links Element

WDSL describes functionality of services provided by a partner

Partner Link describes the shape of the relationship with a partner by describing the Port Types used in a peer to peer relationship

Example:

```
<partnerLinks>
  <partnerLink name="Invoice"
    partnerLinkType="inv:InvoiceTyp
    partnerRole="InvoiceServiceProvider"/>
   <partnerLink name="Employee"
    partnerLinkType="emp:EmployeeType"
    partnerRole="EmployeeServiceProvider"/>
</partnerLinks>
```

Reference to WDSL portType element

Variable Element

Variable construct is used to state information related to workflow logic

Variables can contain entire messages and data sets formatted as XSD schema types

Example:

```
<variables>
  <variable name="EmployeeHoursReq
  messageType="emp:getWeeklyHoursRequestMessage
  "/>
</variables>
```

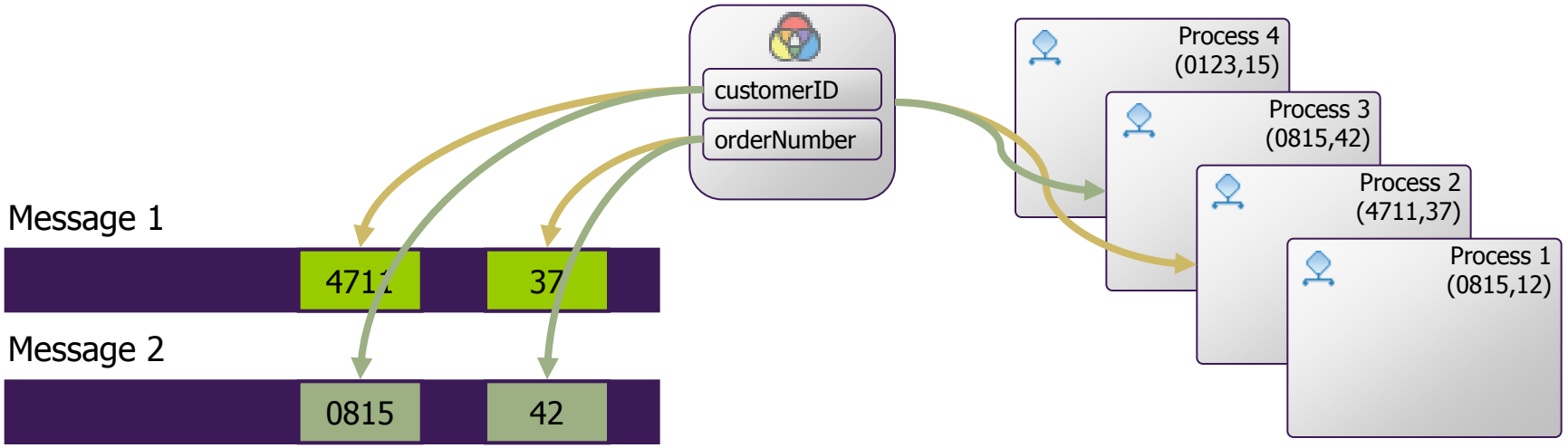
Message Name from Partner Process Definition

Properties and Correlation Sets

How to identify stateful instances via stateless WS interfaces?

A process instance is assigned one or more keys

- Business data is used as key, e.g., customerID
- A key can be compound, e.g., (customerID, orderNumber)
- WS-BPEL calls a key a correlation set – it is used to correlate an incoming message with a process instance



BPEL Syntax Example – Partner Definition

```
<?xml version="1.0" encoding="utf-8"?>
<process name="insuranceSelectionProcess"
         targetNamespace="http://packtpub.com/bpel/example/"
         xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
         xmlns:ins="http://packtpub.com/bpel/insurance/"
         xmlns:com="http://packtpub.com/bpel/company/" >
    <partnerLinks>
      <partnerLink name="client"
                   partnerLinkType="com:selectionLT"
                   myRole="insuranceSelectionService"/>
       <partnerLink name="insuranceA"
                   partnerLinkType="ins:insuranceLT"
                   myRole="insuranceRequester"
                   partnerRole="insuranceService"/>
       <partnerLink name="insuranceB"
                   partnerLinkType="ins:insuranceLT"
                   myRole="insuranceRequester"
                   partnerRole="insuranceService"/>
</partnerLinks>
…
```

BPEL Syntax Example – Variable Definition

```
<variables>
      <!-- input for BPEL process -->
      <variable name="InsuranceRequest"

  messageType="ins:InsuranceRequestMessage"/>
      <!-- output from insurance A -->
      <variable name="InsuranceAResposne"

  messageType="ins:InsuranceResponseMessage"/>
      <!-- output from insurance B -->
      <variable name="InsuranceBResposne"

  messageType="ins:InsuranceResponseMessage"/>
      <!-- output from BPEL process -->
      <variable name="InsuranceSelectionResponse"

  messageType="ins:InsuranceResponseMessage"/>
   </variables>
...
```

BPEL Syntax Example – Process Steps

```
<sequence>
      <!-- Receive the initial request from client -->
      <receive partnerLink="client"
               portType="com:InsuranceSelectionPT"
               operation="SelectInsurance"
               variable="InsuranceRequest"
               createInstance="yes" />
       <flow> <!-- Make concurrent invocations to Insurance A and B -->
          <!-- Invoke Insurance A web service -->
         <invoke partnerLink="insuranceA"
                  portType="ins:ComputeInsurancePremiumPT"
                  operation="ComputeInsurancePremium"
                  inputVariable="InsuranceRequest"
                  outputVariable="InsuranceAResposne" />
          <!-- Invoke Insurance B web service -->
         <invoke partnerLink="insuranceB"
                  portType="ins:ComputeInsurancePremiumPT"
                  operation="ComputeInsurancePremium"
                  inputVariable="InsuranceRequest"
                  outputVariable="InsuranceBResposne" />
       </flow>
```

BPEL Syntax Example – Process Steps (cont')

```
<!-- Select the best offer and construct the response -->
      <switch>
        <case condition="bpws:getVariableData('InsuranceAResposne',
     'confirmationData','/confirmationData/Amount')<= bpws:getVariableData('InsuranceBResposne',
     'confirmationData','/confirmationData/Amount')">
           <assign> <!-- Select Insurance A -->
             <copy>
               <from variable="InsuranceAResposne" />
               <to variable="InsuranceSelectionResponse" />
             </copy>
           </assign>
        </case>
        <otherwise><!-- Select Insurance B -->
           <assign>
             <copy>
               <from variable="InsuranceBResposne" />
               <to variable="InsuranceSelectionResponse" />
             </copy>
           </assign>
        </otherwise>
      </switch><!-- Send a response to the client -->
      <reply partnerLink="client" portType="com:InsuranceSelectionPT"
             operation="SelectInsurance" variable="InsuranceSelectionResponse"/>
    </sequence>
</process>
```

Intalio BPMS Overview

Intalio released a BPMS platform community edition under the Mozilla Public License (MPL) in late 2006

- Lighter version of Enterprise BPMS platform
- BPEL platform based on Apache Geronimo application server
- Graphical developer tool based on Business Process Modeling Notation (BPMN)
- Web site for code, documentation, and tutorials: http://bpms.intalio.com

WS-BPEL Adoption - Products

- Active Endpoints ActiveBPEL
- BEA WebLogic
- Cape Clear Orchestrator
- Intalio/Apache Orchestration Director Engine (Ode)
- IBM WebSphere Process Server
- Microsoft BizTalk Server
- MidOffice BPEL Engine (open source)
- OpenLink Virtuoso Universal Server
- Oracle BPEL Process Manager
- Parasoft BPEL Maestro
- Progress Sonic BPEL Server
- SAP NetWeaver
- Sun eInsight BPM

WS-BPEL Follow-on Work

BPEL4People – human interactions

- http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpel4people/

BPEL-SPE – subprocess coordination protocol

- http://www-128.ibm.com/developerworks/webservices/library/specification/ws-bpelsubproc/

BPELJ – inline Java code in activities and expressions

- http://www-128.ibm.com/developerworks/library/specification/ws-bpelj/

Currency with related standards

- WSDL 2.0, XPath 2.0, XQuery, etc.

BPEL4People & WS-HumanTask - Requirements

- Integration of human-executed activities in Web services-based business processes
- Integration of human-executed activities in SOA-based applications
- Standard-based solution to support interoperability and portability scenarios

BPEL4People & WS-HumanTask - Approach

BPEL4People

- Definition of human interactions within WS-BPEL processes
- Specification built on top of WS-BPEL 2.0

WS-HumanTask

- Definition of service-enabled human tasks and notifications
- Coordination protocol used to control autonomy and life cycle of service-enabled human tasks in an interoperable manner
- Interoperable programming interface enabling task client applications to work with human tasks

Service Component Architecture and WS-BPEL Complementary Technologies

Similarities between SCA and WS-BPEL

- Both are described in a formal language that is based on XML
- Both languages may be used to describe a business service that is implemented by composing together other business services
- Both can describe inbound and outbound service interactions types by WSDL port types

SCA describes the structure of an application

- Components within the business application
- Services offered by components – Service references components depend on
- Connections between components
- Endpoint addresses and communication methods used for the connections
- Policies applied to components and to the connections between them

WS-BPEL describes the logic of a business process

- Sequences of operations which are performed to execute an individual business process
- Services provided and consumed through partnerLinks, that is, abstract interfaces that must be connected to actual endpoints and communication methods through configuration

BPEL Adoption within SOA

Though it is one of the first web service standards published, the mainstream adoption of WS-BPEL has been slow

- BPEL specification was never a complete programming language
- The implementation of web service solutions have focused on HST and JBOWS patterns
- Gaps in functionality have been filled through the use of other programming tools and vendor specific extensions
- Cross version and cross platform interoperability has been problematic to date

WS-BPEL 2.0 standard specification provides promise for improvements in terms of interoperability and functionality, however challenges remain

- WS-HumanTask and BPEL4People highlight the need to add additional semantics to BPEL
- Significant presence of well-established and highly capable legacy workflow products on the market
- Critical mass of web-services to orchestrate are still in development
- No standard graphical notation for WS-BPEL as the OASIS standards committee deemed it out of scope
  - –The question of how to create direct visual representation of BPEL process descriptions still needs to be addressed - Vendor Specific Notations, Business Process Modeling Notation
  - –BPMN to BPEL 2.0 modeling tools

BPEL Process Integration Scenario Microsoft - SAP

SAP focused on formal business process workflows while Microsoft is focused on informal document workflows

- SAP application platform implementation deployed on Netweaver 7.0
  - –Implementation of business process automation using SAP PI which supports BPEL 2.0
- Microsoft collaboration implementation built on Sharepoint 2007
  - –Supports Microsoft WF as work flow engine
  - –Biztalk 2006 ⇔ Netweaver PI to provide cross platform interoperability using BPEL 2.0