SUPER: A Quick Overview

John Domingue
“In which of our food manufacturing machines are we processing meat or raw eggs?”

“How many inventory management methods are currently in use?”

“Do we have a cost approval process for items below $200?”
The Critical IT / Process Divide

Business Experts’ Perspective: Processes

Querying the Process Space

Manual Labor

Process Implementation

IT Implementation Perspective
The Critical IT / Process Divide

Business Experts' Perspective: Processes

SCOPE of SUPER

Machine-Accessible Representation of Business Experts' Requirements

Mechanized Mediation based on Machine Reasoning

Machine-Accessible Representation of Processes, Process Fragments, and IT Infrastructure as Semantic Web Services

IT Implementation Perspective
Modelling Stack

- Making sense of a domain\problem
- Communication tool
- What is it all about?

- Visualizing\specifying business process
- Focus: Business Problem
- Who does what, when, how and why?
- Usually multiple layers

- Process execution specification
- Formal, clearly specified grammar
- Focus: Implementation
- Which component is called when, how, by whom with which data?

- Web service encapsulation
- Focus: Implementation
- Which components can and should be exposed how as services?

- Implementation of components
- Programming languages

- Solution maps
- Mind maps
- Ad-hoc modelling techniques
- ...

- Business Scenario Maps
- Event-driven process chains
- Flowchart techniques
- BPMN
- ...

- BPEL
- ...

- WS*
- ...

- ...
Modelling Stack

- Making sense of a domain\problem
- Communication tool
- What is it all about?

- Visualizing\specifying business process
- Focus: Business Problem
- Who does what, when, how and why?
- Usually multiple layers

- Process execution specification
- Formal, clearly specified grammar
- Focus: Implementation
- Which component is called when, how, by whom with which data?

- Web service encapsulation
- Focus: Implementation
- Which components can and should be exposed as services?

- Implementation of components
- Programming languages
## Content on Demand

<table>
<thead>
<tr>
<th>Industry Value Chain</th>
<th>Design &amp; Build Infrastructure</th>
<th>Operate Infrastructure</th>
<th>Develop &amp; Promote Products &amp; Services</th>
<th>Sell &amp; Fulfill</th>
<th>Bill &amp; Collect</th>
<th>Assist Customers</th>
<th>Customers &amp; Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppliers &amp; Partners</td>
<td>Network Life-Cycle Management</td>
<td>Demand and Supply Planning</td>
<td>Investment Management</td>
<td>Network Design and Build</td>
<td>Operations and Maintenance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Marketing Analytics &amp; Product Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales &amp; Service Fulfillment</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loss and Opportunity Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sales and Order Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Customer Field Service Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Logistics Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Content On Demand

**Digital Rights Management & Content Procurement**

**Content Distribution & Billing**

**Enterprise Management & Support**
Modelling Stack

- Making sense of a domain\problem
- Communication tool
- What is it all about?

- Solution maps
- Mind maps
- Ad-hoc modelling techniques
- ...

- Visualizing\specifying business process
- Focus: Business Problem
- Who does what, when, how and why?
- Usually multiple layers

- Business Scenario Maps
- Event-driven process chains
- Flowchart techniques
- BPMN
- ...

- Process execution specification
- Formal, clearly specified grammar
- Focus: Implementation
- Which component is called when, how, by whom with which data?

- BPEL
- ...

- Web service encapsulation
- Focus: Implementation
- Which components can and should be exposed as services?

- WS*
- ...

- Implementation of components

- Programming languages
- ...

© SUPER 17.06.2007  SBPM Workshop @ ESWC 2007, Innsbruck, Austria
Business Process Components

Events trigger functions

Functions generate events

Diagram:
- Receipt booking request → Event
- Event
- Check booking request → Function
- Function
- Booking request checked → Event
- Event
- Create flight booking → Function
- Function
- Offer alternative flight
- Alternative flight offered → Event
- Event
- Flight booking created
Business Process Components

Functions generate and process services

Data is processed in functions

Employees are assigned to organisational units

Employees are responsible for functions

Service
Booking form
Receipt booking request

Check booking request
Mrs. Miller
Call center

Flight data
Booking request
Booking request checked

Alternative flight offer
Service

Service

Create flight booking
Client data
Offer alternative flight

Mrs. Walker
Plane ticket
Flight booking created

Mr. Green
Flight disposition

Flight disposition

Client data

Altemative flight offered
Modelling Stack

- Making sense of a domain\problem
- Communication tool
- What is it all about?

- Visualizing\specifying business process
- Focus: Business Problem
- Who does what, when, how and why?
- Usually multiple layers

- Process execution specification
- Formal, clearly specified grammar
- Focus: Implementation
- Which component is called when, how, by whom with which data?

- Web service encapsulation
- Focus: Implementation
- Which components can and should be exposed as services?

- Implementation of components

- Solution maps
- Mind maps
- Ad-hoc modelling techniques

- Business Scenario Maps
- Event-driven process chains
- Flowchart techniques
- BPMN

- BPEL

- WS*
Programming in the Large
Workflow System

Choreography/Flow/Process

Web Services/Functions

Application Server
Programming in the Small

Application

<process name="Mediation Example - Ordering BPEL Snippet - 1" suppressJoinFailure="yes" targetNamespace="...">
  <sequence>
  <receive name="Initial_Receive" createInstance="true"/>
  <invoke name="Invoke_Check_Order_Consistency"/>
  <switch>
  <case condition="bpws:getVariableData('consistency', '') = 'OK'">
  <flow>
  <invoke name="Invoke_Update_Provisioning_Systems_Subprocess"/>
  <invoke name="Invoke_CustomerReply_Confirmation_Note"/>
  </flow>
  </case>
  <otherwise>
  <invoke name="Invoke_CustomerReply_Rejection_Note"/>
  </otherwise>
  </switch>
  <reply name="Final_Reply"/>
  </sequence>
</process>
Modelling Stack

- Making sense of a domain\problem
- Communication tool
- What is it all about?

- Visualizing|specifying business process
- Focus: Business Problem
- Who does what, when, how and why?
- Usually multiple layers

- Process execution specification
- Formal, clearly specified grammar
- Focus: Implementation
- Which component is called when, how, by whom with which data?

- Web service encapsulation
- Focus: Implementation
- Which components can and should be exposed how as services?

- Implementation of components

- Solution maps
- Mind maps
- Ad-hoc modelling techniques
- ...

- Business Scenario Maps
- Event-driven process chains
- Flowchart techniques
- BPMN
- ...

- BPEL
- ...

- WS*
- ...

- Programming languages
- ...

© SUPER 17.06.2007

SBPM Workshop @ ESWC 2007, Innsbruck, Austria
Objectives that a client wants to achieve by using Web Services

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

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

Connectors between components with mediation facilities for handling heterogeneities
Objectives that a client wants to achieve by using Web Services

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

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

Connectors between components with mediation facilities for handling heterogeneities
WSMO Top Level Notions

Objectives that a client wants to achieve by using Web Services

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

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

Connectors between components with mediation facilities for handling heterogeneities
Objectives that a client wants to achieve by using Web Services

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

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

Connectors between components with mediation facilities for handling heterogeneities
Objectives that a client wants to achieve by using Web Services

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

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

Connectors between components with mediation facilities for handling heterogeneities