

Information Systems 2

6. Business Process Modelling II: Process Languages

Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Institute for Business Economics and Information Systems
& Institute for Computer Science
University of Hildesheim
<http://www.ismll.uni-hildesheim.de>

Business Process Standard Families

organization	acronym	standard name
OASIS / BPMI	BPMN WS-BPEL	Business Process Modelling Notation Web Services Business Process Execution Language (formerly BPEL4WS or just BPEL)
WfMC	XPDL WfXML WAPI	XML Process Definition Language Workflow XML Workflow Application Programming Interface
W3C	WS-CDL WSCI WSCL	Web Services Choreography Description Language Web Services Choreography Interface Web Services Conversation Language
OMG	BPDM BPRI BPSS	Business Process Definition Metamodel Business Process Runtime Interface Business Process Specification Schema
IBM Microsoft BEA	WSFL XLANG BP4J	Web Services Flow Language Process Definition for Java

BPMI: Business Process Modelling Initiative.

WfMC: Workflow Management Coalition.

1. WS-BPEL

2. Implementing BPEL Processes with Apache ODE

Overview

- WS-BPEL is a web service orchestration and execution language.
- WS-BPEL is an XML application. Its namespace is <http://docs.oasis-open.org/wsbpel/2.0/process/executable>
- WS-BPEL can use
 - the XML Schema type system.
 - XPath and XSLT query / processing languages and
 - WSDL webservice descriptions.
- WS-BPEL is managed by OASIS, its actual version is WS-BPEL 2.0 (April 3, 2007), it can be found at <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- WS-BPEL originally was called *Business Process Execution Language for Web Services (BPEL4WS)*, v1.0 (2002), which was based on
 - XLANG (2001; developed by IBM, BEA, Sun, OASIS, SAP, INNODATA/ISOGENE)
 - WSFL (Web Services Flow Language; IBM)

An Example / 1. WSDL

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <wsdl:definitions
3   targetNamespace="http://www.cgnm.de/examples/bpel/Travel.wsdl"
4   xmlns="http://schemas.xmlsoap.org/wsdl/"
5   xmlns:tns="http://www.cgnm.de/examples/bpel/Travel.wsdl"
6   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
7   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
8   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
9   xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype">
10
11 <wsdl:message name="TravelRequest">
12   <wsdl:part name="from" type="xsd:string"/>
13   <wsdl:part name="to" type="xsd:string"/>
14 </wsdl:message>
15
16 <wsdl:message name="TravelResponse">
17   <wsdl:part name="price" type="xsd:double"/>
18 </wsdl:message>
19
20 <wsdl:portType name="TravelPortType">
21   <wsdl:operation name="searchTravel">
22     <wsdl:input message="tns:TravelRequest" name="TestIn"/>
23     <wsdl:output message="tns:TravelResponse" name="TestOut"/>
24   </wsdl:operation>
25 </wsdl:portType>
26
27 <wsdl:binding name="TravelSoapBinding" type="tns:TravelPortType">
```

```
28 <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
29 <wsdl:operation name="searchTravel">
30   <soap:operation soapAction="" style="rpc"/>
31   <wsdl:input>
32     <soap:body
33       namespace="http://www.cgnm.de/examples/bpel/Travel.wsdl"
34       use="literal"/>
35   </wsdl:input>
36   <wsdl:output>
37     <soap:body
38       namespace="http://www.cgnm.de/examples/bpel/Travel.wsdl"
39       use="literal"/>
40   </wsdl:output>
41 </wsdl:operation>
42 </wsdl:binding>
43 <wsdl:service name="TravelService">
44   <wsdl:port name="TravelPort" binding="tns:TravelSoapBinding">
45     <soap:address location="http://localhost:8080/ode/processes/Travel"/>
46   </wsdl:port>
47 </wsdl:service>
48
49 <plnk:partnerLinkType name="TravelPartnerLinkType">
50   <plnk:role name="me" portType="tns:TravelPortType"/>
51   <plnk:role name="you" portType="tns:TravelPortType"/>
52 </plnk:partnerLinkType>
53 </wsdl:definitions>
54
```

Figure 1: The Web Service Description Travel.wsdl

An Example / 2. BPEL

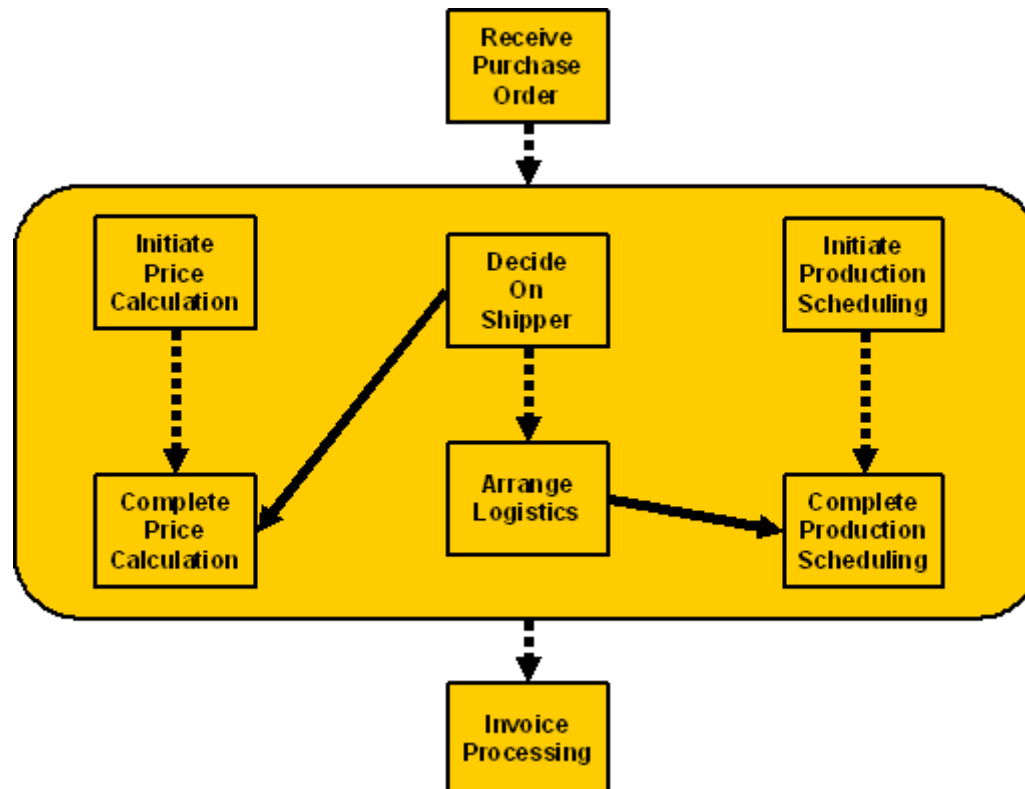
```
1 <process name="Travel"
2   targetNamespace="http://www.cgnm.de/examples/bpel/Travel"
3   xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
4   xmlns:tns="http://www.cgnm.de/examples/bpel/Travel"
5   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
6   xmlns:test="http://www.cgnm.de/examples/bpel/Travel.wsdl"
7   queryLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0"
8   expressionLanguage="urn:oasis:names:tc:wsbpel:2.0:sublang:xpath2.0">
9
10 <import location="Travel.wsdl"
11   namespace="http://www.cgnm.de/examples/bpel/Travel.wsdl"
12   importType="http://schemas.xmlsoap.org/wsdl/" />
13
14 <partnerLinks>
15   <partnerLink name="TravelPartnerLink"
16     partnerLinkType="test:TravelPartnerLinkType"
17     myRole="me" />
18 </partnerLinks>
19
20 <variables>
21   <variable name="req" messageType="test:TravelRequest"/>
22   <variable name="resp" messageType="test:TravelResponse"/>
23 </variables>
24
25 <sequence>
26   <receive name="start"
27     partnerLink="TravelPartnerLink"
```



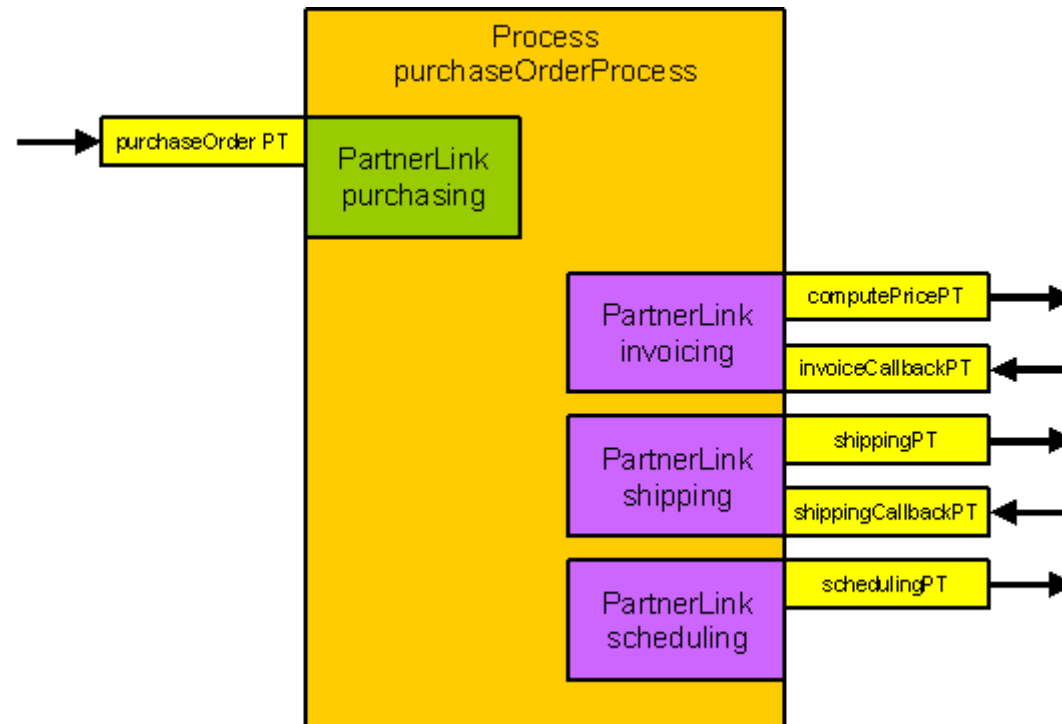
```
28     portType="test:TravelPortType"  
29     operation="searchTravel"  
30     variable="req"  
31     createInstance="yes"/>  
32  
33 <assign name="assign1">  
34     <copy>  
35         <from><literal>344.50</literal></from>  
36         <to variable="resp" part="price"/>  
37     </copy>  
38 </assign>  
39  
40 <reply name="end"  
41     partnerLink="TravelPartnerLink"  
42     portType="test:TravelPortType"  
43     operation="searchTravel"  
44     variable="resp"/>  
45 </sequence>  
46 </process>
```

Figure 2: The Business Process Execution Travel.bpel

A Business Process Made From Components



Component Processes are Mapped to Partner Links



WS-BPEL Activities

element	short description
<receive>	wait for message
<reply>	send a message
<invoke>	invoke a one-way or request-response operation
<pick>	wait for one out of several messages or timeout
<assign>	update variables with new data
<validate>	validate values of a variables against their definitions
<if>	select one activity from choices
<while>	repeat some activity
<repeatUntil>	repeat some activity
<forEach>	repeat some activity
<exit>	immediately end a business process instance
<wait>	wait for some time or until some timepoint
<empty>	do nothing
<sequence>	perform several activities sequentially
<flow>	perform several activities concurrently
<scope>	define nested activity
<compensate>	start compensation on all inner scopes
<compensateScope>	start compensation on a specified inner scope
<throw>	throw a fault
<rethrow>	rethrow a fault
<extensionActivity>	define a new activity type

1. WS-BPEL

2. Implementing BPEL Processes with Apache ODE

Apache Orchestration Director Engine (ODE)

- Apache Orchestration Director Engine (ODE) is an WS-BPEL engine.
- Apache ODE is open source (v1.3.2 / 11.5.2009 or v1.2 / 3.7.2008) and can be downloaded at <http://ode.apache.org>
- Apache ODE comes ready to install as war-archive bundled with the Axis2 webservice engine.
- To run a web application archive (war-archive), you will need a servlet container such as Apache Tomcat.

Installing Apache ODE

```
tar xzf apache-tomcat-6.0.16.tar.gz
unzip apache-ode-war-1.2.zip
cp apache-ode-war-1.2/ode.war apache-tomcat-6.0.16/webapps/
cd apache-tomcat-6.0.16/
./bin/startup.sh
```

Now the Axis2 engine is running on port 8080.

You can get a list of deployed web services by visiting

<http://localhost:8080/ode/>

with a web browser.

An Example / 3. Deployment Descriptor

To deploy a business process in ODE, one simply has to copy the three files

- `Travel.wsdl`,
- `Travel.bpel` **and**
- `deploy.xml`

in a subdirectory `travel` of the process directory

`apache-tomcat-6.0.16/webapps/ode/WEB-INF/processes/`

```
1 <deploy xmlns="http://www.apache.org/ode/schemas/dd/2007/03"
2   xmlns:pns="http://www.cgnm.de/examples/bpel/Travel"
3   xmlns:wns="http://www.cgnm.de/examples/bpel/Travel.wsdl">
4
5   <process name="pns:Travel">
6     <active>true</active>
7     <provide partnerLink="TravelPartnerLink">
8       <service name="wns:TravelService" port="TravelPort"/>
9     </provide>
10  </process>
11 </deploy>
```

Figure 5: The Apache ODE Deployment Descriptor `deploy.xml`