Lab 8: Business Process Execution Language (BPEL)
Mapping between BPMN and BPEL

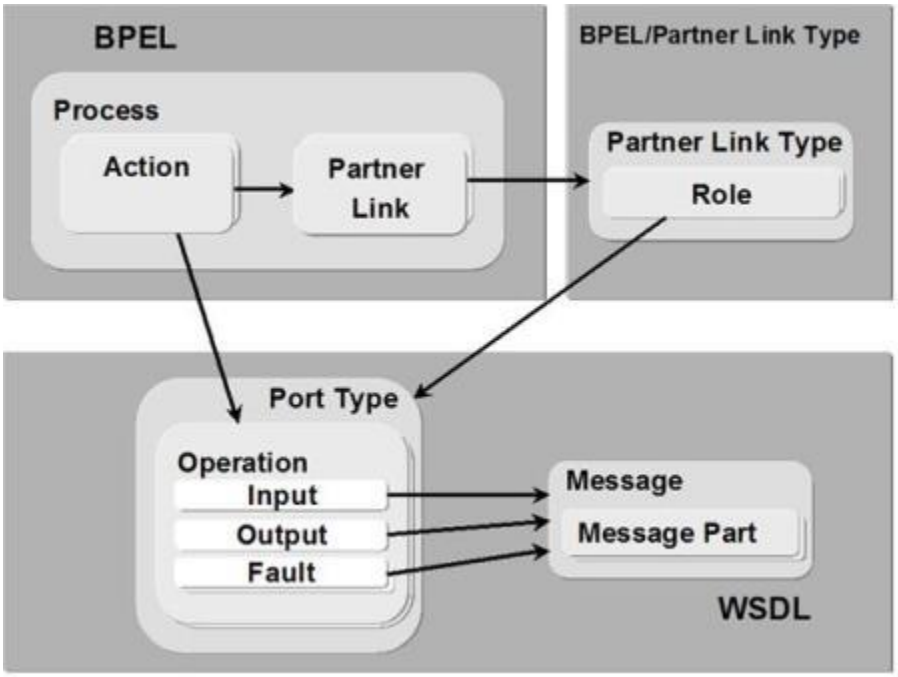
Introduction

Developing the web services and exposing the functionality (via WSDL) is not sufficient

Example Scenario:

Concert ticket purchase Web service has 3 operations, which need to be performed in the following order

  - Getting a price quote
  - Purchase a ticket
  - Confirmation and cancellation
- We also need a way to orchestrate these functionality in the right order
- Web services are described in WSDL
- We need a way to orchestrate these operations with multiple web services in the right order to perform a Business process
  - Sequencing, conditional behavior etc.
- BPEL provides standard-based orchestration of these operations



Necessary features:

Business processes need to support some functionalities such as:

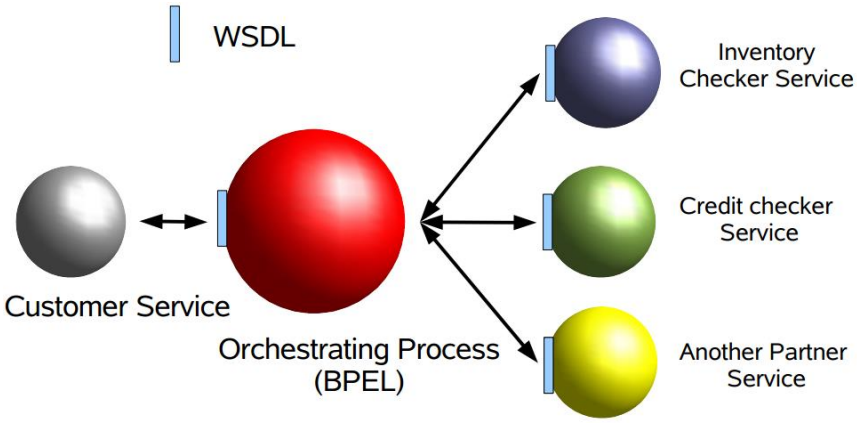
- Co-ordinate asynchronous communication between services
- Correlate message exchanges between parties
- Implement parallel processing of activities
- Implement compensation logic (Undo operations)
- Manipulate/transform data between partner interactions
- Support for long running business transactions and activities
- Handle exception handling
- Need for universal data model for message exchange

BPEL

- BPEL is an XML-based language that models a business process as a composition from a set of elementary web services**.**
- BPEL includes basic and structured activities, variables, partner links and handlers.
- It is XML-based language used to specify business processes based on Web Services
- BPEL processes describe Long running, state-ful, transactional, conversations between two or more partner web services
- BPEL is key to implementing SOA (Service Oriented Architecture)
  - Conversational
  - Mostly Async
  - XML Document-based
  - Orchestrated

There are many tools to represent BPEL using designer editor to facilitate to the end user how these tags are written; e.g. Eclipe, NetBeans (version 6.5.1 has samples while other version don't), Oracle, IBM

BPEL relations to partners

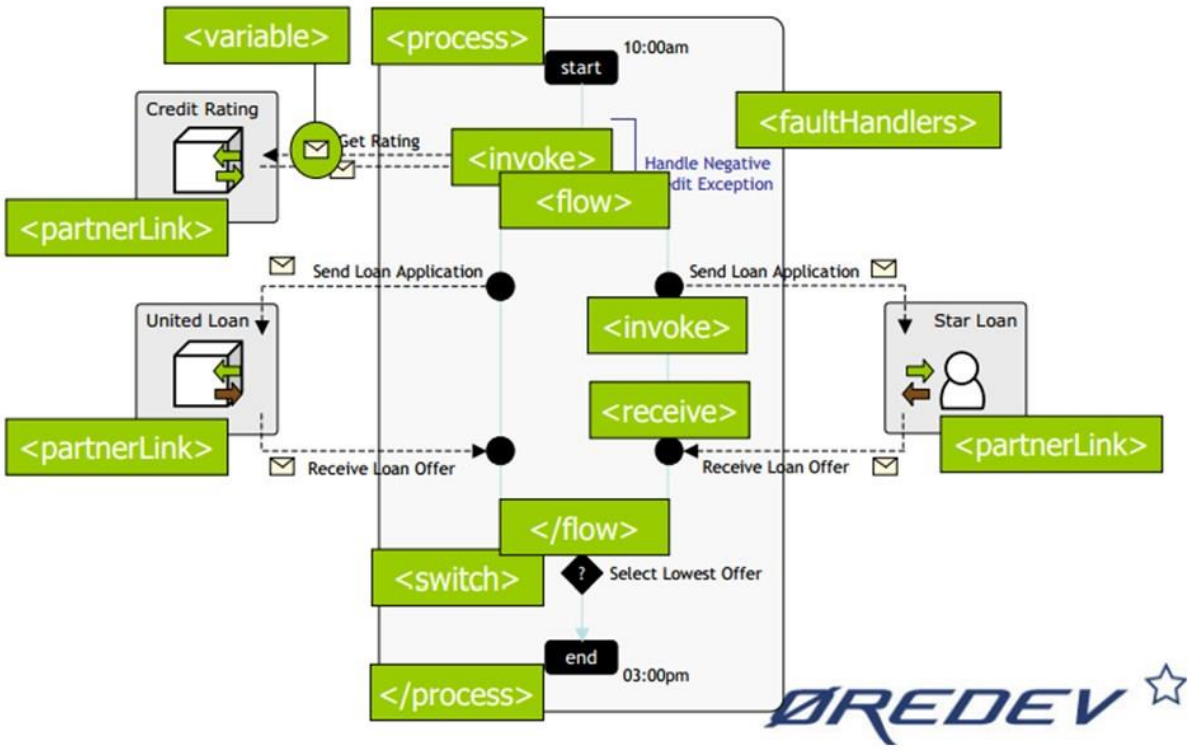


There are several web services, which together can perform the process provided by customer service. BPEL is the one component responsible of orchestrating theses web services and handle their communications with WSDL interface.

BPEL document structure

```
<process>
      <!-- Definition and roles of process participants -->
      <partnerLinks> ... </partnerLinks>

      <!-- Data/state used within the process -->
      <variables> ... </variables>

      <!-- Properties that enable conversations -->
      <correlationSets> ... </correlationSets>

      <!-- Exception handling -->
      <faultHandlers> ... </faultHandlers>

      <!-- Error recovery or undoing actions -->
      <compensationHandlers> ... </compensationHandlers>

      <!-- Concurrent events with process itself -->
      <eventHandlers> ... </eventHandlers>

      <!-- Business process flow -->
      (activities)*
</process>
```

BPEL activities



Basic Activities

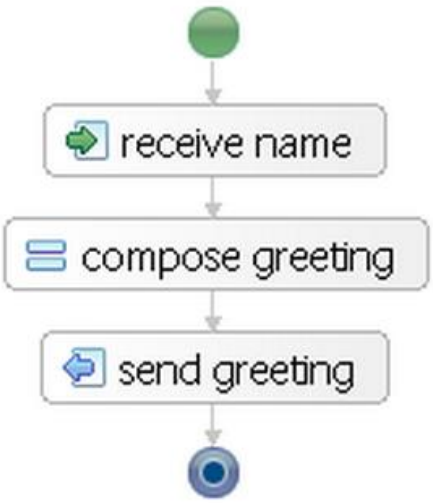
<invoke>	- Invokes a one-way or request/response operation on a PortType offered by a partner.
<receive>	- Does a blocking wait for a matching message to arrive - It can be responsible of instantiating of the business process.
<reply>	- Sends a message in reply to a message that was received through a <receive>. - The combination of a <receive> and a <reply> forms a request-response operation on the WSDL portType for the process.
<assign>	- Provides a method for data manipulation, such as copying the contents of one variable to another. - Copy operations enable you to transfer information between variables, expressions, endpoints, and other elements.
<throw>	- Generates a fault from inside the business process.
<wait>	- Allows a process to specify a delay for a certain period or until a certain deadline is reached. - A typical use of this activity is to invoke an operation at a certain time. - This activity enables you to wait for a given time period or until a certain time has passed. - Exactly one of the expiration criteria must be specified.
<empty>	- Enables you to insert a no-operation instruction into a process. - It is useful when you must use an activity that does nothing (for example, when a fault must be caught and suppressed).
<exit>	- Enables you to immediately end all currently running activities on all parallel branches without involving any termination handling, fault handling, or compensation handling mechanisms.

Structured Activities

<flow>	- Performs activities in parallel
<sequence>	- Performs activities in sequential order
<scope>	- Encloses multiple activities in a single scope
<pick>	- Waits for the occurrence of one event in a set of events and performs the activity associated with that event. - The occurrence of the events is often mutually exclusive (the process either receives an acceptance or rejection message, but not both). - If multiple events occur, the selection of the activity to perform depends on which event occurred first. - If the events occur nearly simultaneously, there is a race and the choice of activity to be performed is dependent on both timing and implementation.
<if>	- Conditional choice of activities
<while>	- Supports repeated performance of a specified iterative activity. - The iterative activity is repeated until the given while condition is no longer true.
<repeatUntil>	- Use this activity if the body of an activity must be performed at least once. - Condition in the repeatUntil activity is evaluated after the body of the activity completes. - Condition is evaluated repeatedly (and the body of the activity processed) until the provided Boolean condition is true.
<foreach>	- Processes multiple sets of activities sequentially or in parallel.

BPEL – Business Process Examples illustration

Example1:



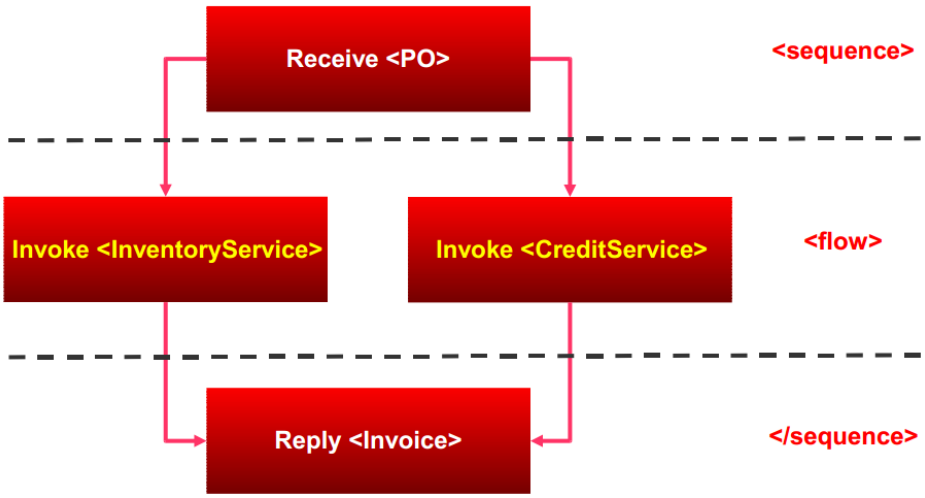
Definitions part:

```
<definitions targetNamespace="http://jbpm.org/examples/hello"
             xmlns:tns="http://jbpm.org/examples/hello"
             xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
             xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns="http://schemas.xmlsoap.org/wsdl/" >
      <!-- carries the name of a person -->
      <message name="nameMessage">
            <part name="name" type="xsd:string" />
      </message>
      <!-- carries the greeting -->
      <message name="greetingMessage">
            <part name="greeting" type="xsd:string" />
      </message>
      <!-- describes the interface presented to callers -->
      <portType name="Greeter">
            <operation name="sayHello">
                  <input message="tns:nameMessage" />
                  <output message="tns:greetingMessage" />
            </operation>
      </portType>
      <!-- characterizes the relationship between the greeter and its caller -->
      <plt:partnerLinkType name="Greeter-Caller">
            <plt:role name="Greeter">
                  <plt:portType name="tns:Greeter" />
            </plt:role>
      <!-- the Caller does not provide services to the Greeter,  this is why we omit the
"Caller" role -->
      </plt:partnerLinkType>
</definitions>
```

Process definition:

```
<process name="HelloWorld" targetNamespace="http://jbpm.org/examples/hello"
         xmlns:tns="http://jbpm.org/examples/hello"
         xmlns:bpel="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
         xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
      <partnerLinks>
            <!-- establishes the relationship with the caller agent -->
            <partnerLink name="caller" partnerLinkType="tns:Greeter-Caller"
                         myRole="Greeter" />
      </partnerLinks>
      <variables>
            <!-- holds the incoming message -->
            <variable name="request" messageType="tns:nameMessage" />
            <!-- holds the outgoing message -->
            <variable name="response" messageType="tns:greetingMessage" />
      </variables>
      <sequence name="MainSeq">
            <!-- receive the name of a person -->
            <receive name="ReceiveName" operation="sayHello" partnerLink="caller"
                     portType="tns:Greeter" variable="request" createInstance="yes" />
            <!-- compose a greeting phrase -->
            <assign name="ComposeGreeting">
                  <copy>
                        <from expression="concat('Hello, ',
                                    bpel:getVariableData('request', 'name'), '!')" />
                        <to variable="response" part="greeting" />
                  </copy>
            </assign>
            <!-- send greeting back to caller -->
            <reply name="SendGreeting" operation="sayHello" partnerLink="caller"
                   portType="tns:Greeter" variable="response" />
      </sequence>
</process>
```

Business Process Example 2: (self-exercise)



Sample Activities in BPEL

```
<sequence>
      <receive partnerLink="customer"  portType="lns:purchaseOrderPT"
               operation="sendPurchaseOrder" variable="PO"  createInstance="yes" />
      <flow>
            <invoke partnerLink="inventoryChecker" portType="lns:inventoryPT"
                    operation="checkINV" inputVariable="inventoryRequest"
                    outputVariable="inventoryResponse" />
            <invoke partnerLink="creditChecker" portType="lns:creditPT"
                    operation="checkCRED" inputVariable="creditRequest"
                    outputVariable="creditResponse" />
      </flow>
      ...
      <reply partnerLink="customer" portType="lns:purchaseOrderPT"
             operation="sendPurchaseOrder" variable="invoice"/>
</sequence>
```

References

1. Netbeans IDE 6.5.1 source (To check samples of BPEL) https://netbeans.org/downloads/6.5.1/start.html?platform=windows&lang=en&option=all&version=6.5.1
2. SOA using Open ESB, BPEL, and NetBeans http://docs.huihoo.com/openesb/soabpelopenesb.pdf
3. IBM BPEL process, last accessed 24/4/2014 http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wbit.help.prodovr.doc/topics/cbpelproc.html
4. Oracle BPEL Process Activities and Services, last accessed 24/4/2014 http://docs.oracle.com/cd/E17904_01/integration.1111/e10224/bp_appx_ref.htm
5. Oracle Transforming BPMN into BPEL, last accessed 24/4/2014 http://www.oracle.com/technetwork/articles/dikmans-bpm-101437.html
6. Web Services Business Process Execution Language Version 2.0, last accessed 24/4/2014 http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html
7. OpenESB, last accessed 24/4/2014 http://www.open-esb.net/