Understanding Links to Partners

Prof. Vipul K. Dabhi
Department of Information Technology,
D. D. University

Partner Links

- A partner link type declares how two parties interact and what each party offers.
- BPEL Process interact with external web-services in two ways:
  - BPEL process invokes operations on other web services
  - BPEL process receives invocations from clients.

Partner Links

- Links to all parties BPEL interacts with are called partner links.
- Partner links can be links to web services that are invoked by BPEL process. These are called invoked partner links.
- Partner links can also be links to clients. These are called client partner links.

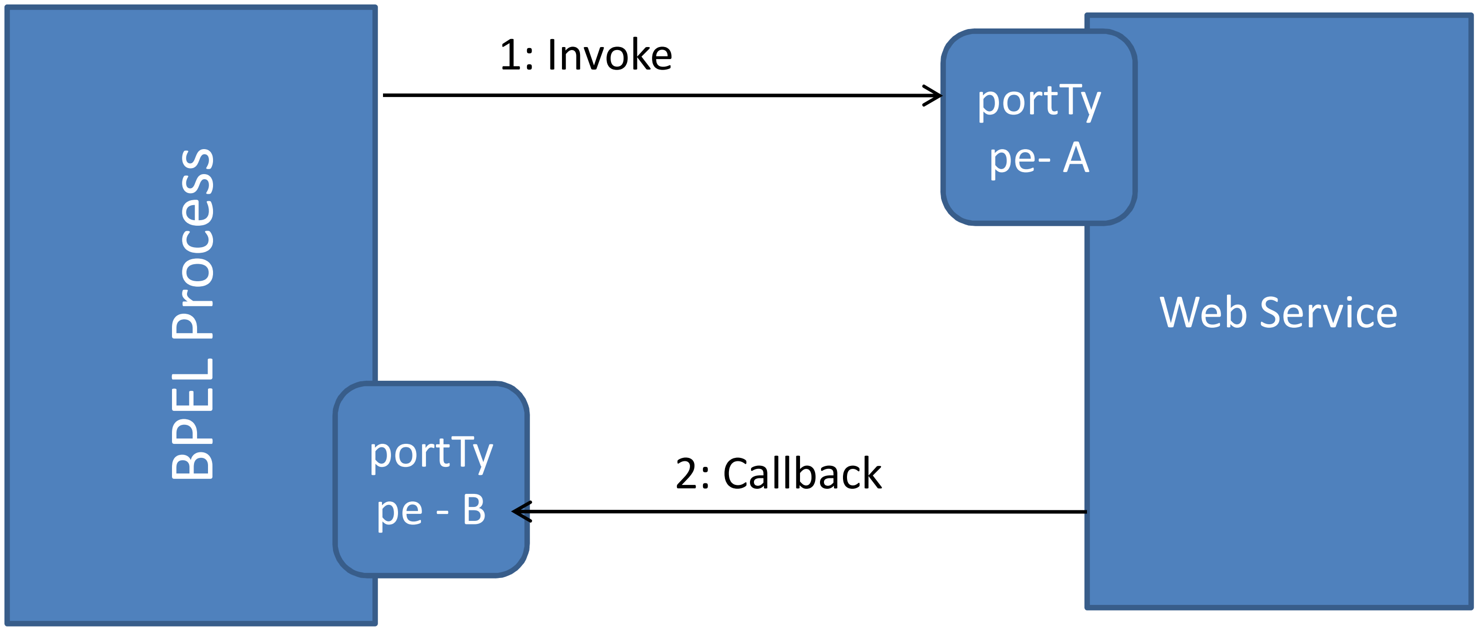
Client Partner Links

- BPEL treats clients as partner links for two reasons:
  - Support for asynchronous interactions
  - BPEL process can offer services: These services can be used by more than one client. The process may wish to distinguish between different clients and to offer them only functionality they are authorized to use.

Partner Links

- Partner links describe links to partners, where partners might be:
  - Services invoked by process
  - Services that invoke the process
  - Services that have both roles – they are invoked by process and they invoke the process (Asynchronous communication)

Asynchronous Callback



Roles in Partner Links

- A partner link type must have at least one role and can have at most two roles.
- For each role we must specify a portType that is used for interactions.

Example

- <partnerLinkType name=“insuranceLT” xmlns=http://schmeas.xmlsoap.org/ws/2003/05/partnerlink/>

```
<role name=“insuranceService”>
    <portType name=“ins:ComputeInsurancePremiumPT”/>
</role>
<role name=“insuranceRequester”>
    <portType name=“ins:ComputeInsurancePremiumCallbackPT”/>
</role>
</partnerLinkType>
```

Example

- <partnerLinkType name="insuranceLT" xmlns=http://schmeas.xmlsoap.org/ws/2003/05/partnerlink/>

```
<role name="insuranceService">
    <portType name="ins:ComputeInsurancePremiumPT"/>
</role>
</partnerLinkType>
```

Sometimes, we may not need to specify two roles. For ex. When we use synchronous request/response operations.

If we specify only one role, we express willingness to interact with the service, but do not place any additional requirements on service.

partnerLinkTypes in WSDL

- It is important to note that partnerLinkTypes are not part of BPEL process specification documentation.
- This is because partnerLinkTypes belong to service specification and not the process specification.
- Therefore they could be placed in WSDL document that describes the partner web-service.
- Partner link types use the WSDL extensibility mechanism.

Partner Links in BPEL Process Definition

- Partner links are concrete references to services that a BPEL business process interacts with.
- They are specified near the beginning of the BPEL process definition document, just after the <proess> tag.

-
  ```
  <process>
      <partnerLinks>
      <partnerLink>
      </partnerLink>
        </partnerLinks>
  </process>
  ```

Partner Links in BPEL Process Definition

- For each partner link, we have to specify:
  - Name: Serves as a reference for interactions via that partner link.
  - partnerLinkType: Defines the type of partner link
  - myRole: Indicates the role of BPEL process itself.
  - partnerRole: Indicates the role of partner.

- We define both roles (myRole and partnerRole) only if partnerLinkType specifies two roles (In case of asynchronous communication).
- If the PartnerLinkType specifies only one role, the partnerLink also has to specify only one role.

Example

- <partnerLinks>

```
<partnerLink name="insurance"
                    partnerLinkType="tns:insuranceLT"
                    myRole="insuranceRequester"
                   partnerRole="insuranceService"/>
</partnerLinks>
```

The role of BPEL process (myRole) is described as insurance requester and the partner role is described as insurance service.

BPEL Process Tag

- The <process> tag requires that we specify certain attributes.
  - Name: Specifies the name of BPEL business process.
  - targetNamespace: Specifies the target namespace for the business process definition.
  - Xmlns: the namespace used by BPEL.
  - queryLanguage: query language used for node selection in assignments, properties and other uses. Default is Xpath 1.0
  - abstractProcess: Specifies whether the process is abstract or executable. The default is no, which means it is executable.

Variables

- BPEL business processes model the exchange of messages between involved web services. Messages are exchanges as operations are invoked.
- When business process invokes the operation and receives the result, we often want to store that result for subsequent invocations.
- BPEL provides variables to store and maintain the state.
- Variables are used to store the messages that are exchanged between business process partner or to hold data that relates to the state of process.

Variables

- Each variable has to declared before it can be used.
- When we declare a variable, we must specify the variable name and type.
- To specify type we have to specify one of following attributes:
  - messageType: a variable that can hold a WSDL message
  - Element: a variable that can hold an XML schema element
  - Type: a variable that can hold an XML Schema simple type

Variable Example

- <variables>

```
    <variable name=“InsuranceRequest”
                    messageType=“ins:InsuranceRequestMessage”/>
     <variable name=“PartialInsuranceDescription”
                    messageType=“ins:InsuranceDescription”/>
     <variable name=“LastName”
                    type=“xsd:string”/>
</variables>
```

First two declaration assume that the corresponding messageType and element have been declared in WSDL.

Providing Interface to BPEL Processes

<invoke>, <receive>, <reply>

- <receive> : business process usually waits for the initial message to start the process. Another typical use of <receive> is to wait for callbacks.
- <reply> : a BPEL process can send a response, if process is modeled as synchronous.
- All three activities use the same three basic attributes:
  - partnerLink: Specifies which partner link will be used
  - portType : Specifies the used port type
  - operation: Specifies name of operation to invoke (<invoke>), to wait for being invoked (<receiver>) or name of operation which has been invoked but is synchronous and requires a reply (<reply>).

<invoke>

- When business process invokes an operation on web service, it sends a set of parameters.
- These parameters are modeled as input messages for invocation, we use inputVariable attribute to store it.
- If we invoke a synchronous request/response operation, it returns a result. To store it in a variable, <invoke> provides another attribute called outputVariable.

<invoke>

```
<invoke partnerLinke="insuranceA"
                portType="ins:ComputeInsurancePremium"
                operation="ComputeInsurancePremium"
                inputVariable="InsuranceRequest"
                outputVariable="InsuranceAResponse">
</invoke>
```

<receive>

- Business process needs to store the incoming message and it can use the variable attribute to specify a suitable variable.
- Another attribute for <receive> activity is the createInstance attribute, which is related to business process lifecycle and instructs BPEL engine to create a new instance of the process.

<receive>

```
<receive partnerLink="client"
                portType="com:InsuranceSelectionPT"
                operation="SelectInsurance"
                variable="InsuranceRequest"
                createInstnce="yes">
            </receive>
```

Because this is initial <receive> activity, the createInstance attribute is used.

<reply>

- <reply> is always related to initial <receive> through which BPEL process started.
- Using <reply> we can return the answer or we can return a fault message.
- When we use <reply> to return a response for synchronous process we have to define only one additional attribute – name of variable where the response is stored.

<reply>

```
<reply partnerLink="client"
          portType="com:InsuranceSelectionPT"
          operation="SelectInsurance"
          variable="InsuranceSelectionResponse"
</reply>
```

Assignments

- To copy data between variables, expressions, and partner link endpoint references BPEL provides the <assign> activity.
- With in <assign> we can perform one or more <copy> commands.
- For each <copy> we have to specify the source <from> and destination <to>.

Assignments

```
<assign>
        <copy>
                <from…./>
                <to…./>
        </copy>
        <copy>
                <from…./>
                <to…./>
        </copy>
</assign>
```