

MTIS	Metodologías y Tecnologías de Integración de Sistemas
	Práctica 3
	BPEL: Ejercicio Guiado 2 Control Pick

Ejercicio Guiado 2 Pick activity

Prerrequisitos

1. Instalación del contenedor Web Apache Tomcat 8.0.20
2. Instalación de MySQL 5.x
3. Instalación de Apache ODE 1.3.6
4. Instalación de Eclipse Java EE Mars
5. Máquina virtual de Java 1.8
6. Apache Axis2

Objetivos

El objetivo de estos ejercicios es familiarizarse con el estándar WS-BPEL y sus principales elementos para la composición. Para ello usaremos como entorno de modelado de procesos de Eclipse BPEL Designer y Apache ODE como motor de ejecución de procesos.

1. Enunciado

En este ejercicio se introducirá al alumno a los siguientes elementos de BPEL:

- *Control Pick*

En este ejercicio partiremos de lo aprendido en el ejercicio donde realizamos el proceso BPEL *HolaMundo*. Las partes comunes no serán descritas en el documento y sólo se hará referencia a la introducción de las nuevas actividades.

Con el control *Pick*, se puede especificar qué proceso se ejecuta de acuerdo con el evento recibido. El evento puede ser un evento de mensaje o un evento de alarma. Para cada caso, se puede especificar un conjunto de actividades a realizar. En este tutorial, vamos a examinar el control *Pick*, con dos operaciones: *sayHello* y *sayGoodbye* utilizando el evento mensaje.

1.1 Crear un nuevo proyecto BPEL llamado "*BPEL_Pick*", seleccionando en el menú *File → New → Other... → BPEL 2.0 → BPEL Project* y seleccionar *Next*.

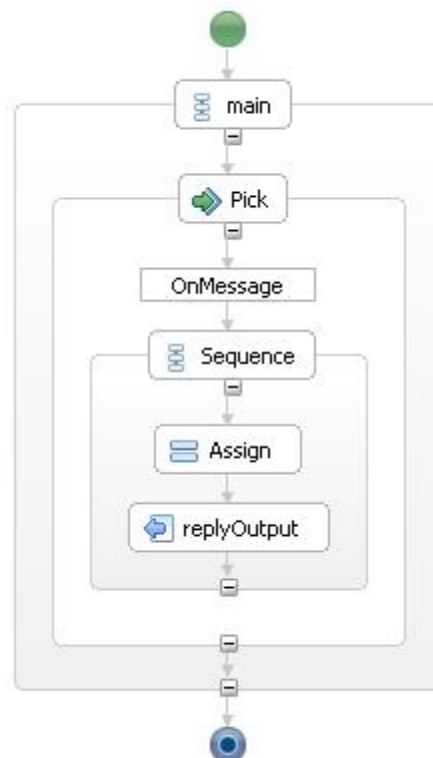
1.2 Crear un archivo BPEL (*New->Others->BPEL 2.0->New BPEL Process File*) denominado *PickSample* con el espacio de nombres *http://pick.bpel.mtis* de tipo síncrono.

1.3 Abre *PickSampleArtifacts.wsdl* con el *design* y con el botón derecho en *PickSample Port Type*, selecciona *Add Operation*, llamala *sayHello*, crea otra

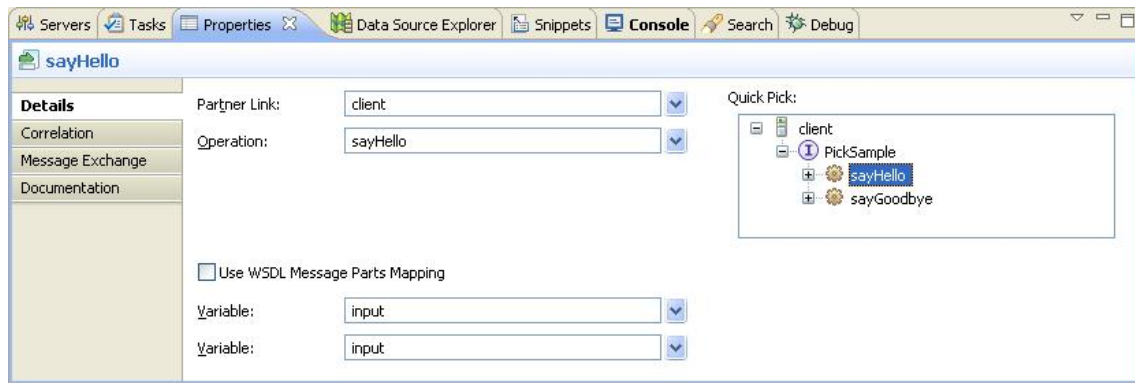
operación y llámala *sayGoodbye*. Eliminar la operación *process*, con el botón derecho. Quedará de esta forma:

PickSample		
sayGoodbye		
input	parameters	sayGoodbye
output	parameters	sayGoodbyeResponse
sayHello		
input	parameters	sayHello
output	parameters	sayHelloResponse

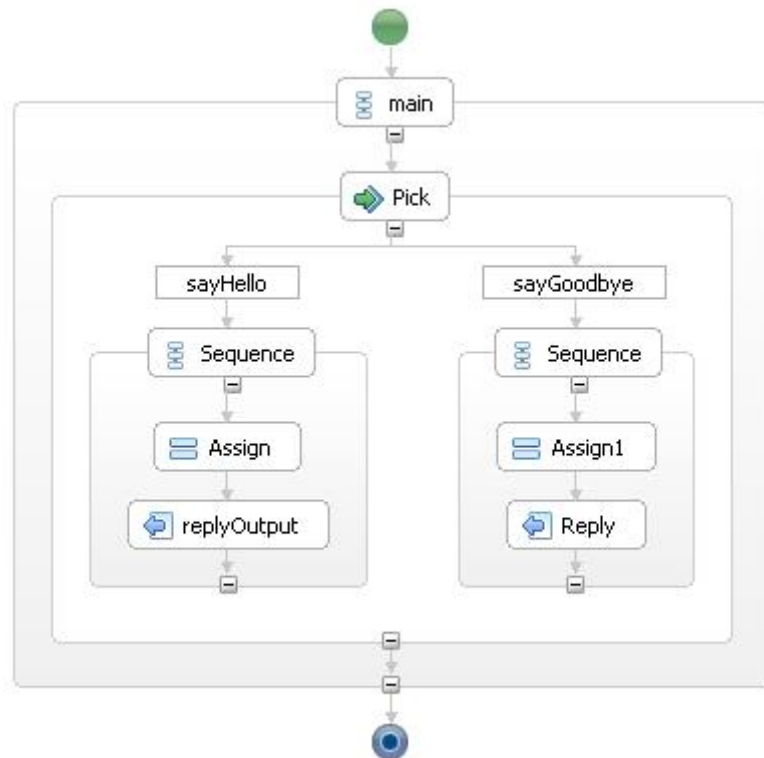
- 1.4 Abre el fichero *PickSample.pbel*, en modo diseño. Pincha y arrastra un control *pick* de la paleta justo antes de *receiveInput*.
- 1.5 En los detalles de *pick*, selecciona la opción *Create a new Process instance if one does not already exist*.
- 1.6 Borra la acción *receiveInput* y la acción *FIX_ME_Add...* y mueve la acción *replyOutput* al evento *OnMessage*, dentro del control *pick*. Pincha y arrastra una acción *Assign* al proceso, antes de *replyOutput*. Una secuencia se creará automáticamente.



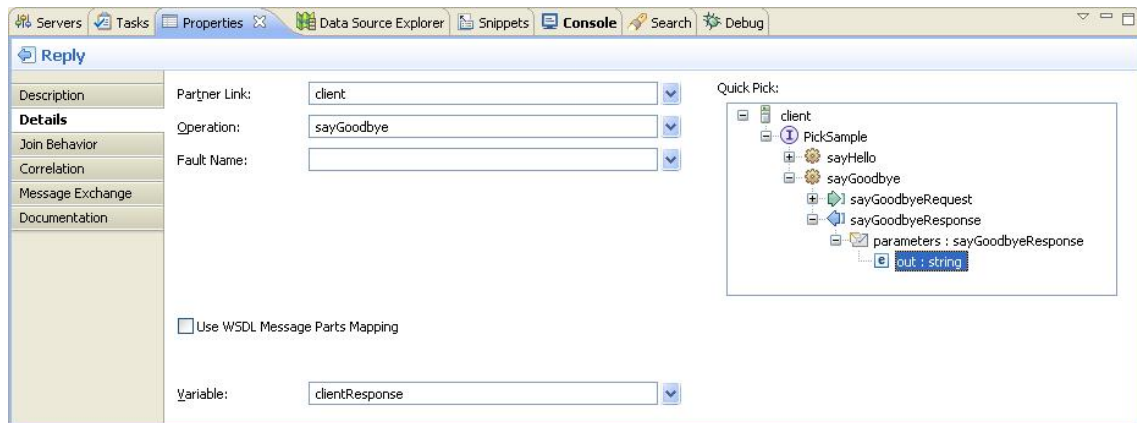
- 1.7 Pincha en el control *onMessage*. En la pestaña *Details*, selecciona la operación *sayHello*.



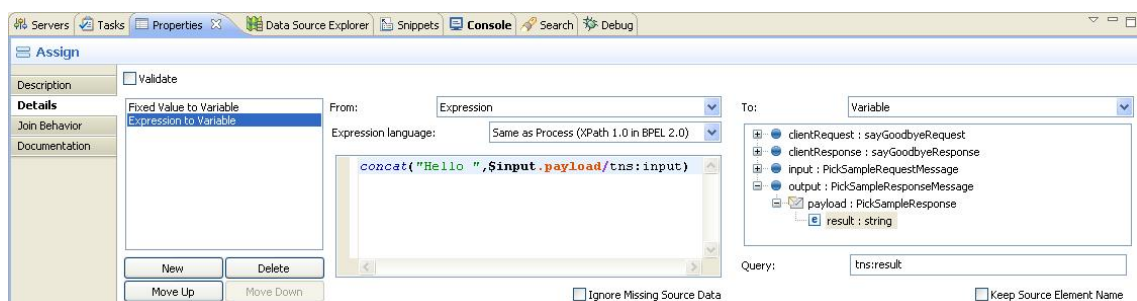
1.8 Botón derecho en el control *Pick*. Selecciona *Add onMessage*. In la pestaña *Detalles*, selecciona la operación *sayGoodbye*. Añade una acción *Assign* y una acción *Reply* a la rama *sayGoodbye*.



1.9 Pincha en la acción *Reply* de la rama *sayGoodbye*, en la pestaña *Detalles*, asigna *PickSample->sayGoodbye->sayGoodbyeResponse->parameters->out* a la variable *clientResponse*.



- 1.10 Pincha en la acción *Assign*, selecciona *Details->New*. Asigna la expresión `concat("Hello ", $input.payload/tns:input)` a la variable `output->payload->result`. Un dialogo pop-up aparecerá preguntando si deseas inicializar la variable, selecciona *Yes*.



- 1.11 Realizar el mismo proceso para la acción *Assign1* de la otra rama, asignar la expresión `concat("Goodbye ", $clientRequest.parameters/tns:in)` a la variable `clientResponse->parameters->out`.
- 1.12 Cambiar la siguiente línea del fichero BPEL, donde indica *process*, cambiar por *sayHello*.

```
<bpel:reply name="replyOutput" partnerLink="client" portType="tns:bp"
operation="process" variable="output" />
<bpel:reply name="replyOutput" partnerLink="client" portType="tns:bp"
operation="sayHello" variable="output" />
```

- 1.13 Ya hemos finalizado la parte de diseño. Ahora vamos al servicio de declaración y registro. Hacemos lo mismo que para el *HelloWorld*:
- 1.14 Guarda los archivos, pulsa el botón derecho en el proyecto *BPEL_Pick*, selecciona *New->Others->BPEL 2.0->Apache ODE Deployment Descriptor*. Pulsa en *Next*. Verifica que el nombre del proyecto sea */BPEL_Pick*. Pulsa *Finish*
- 1.15 Abre el fichero *deploy.xml* file y selecciona *PickSamplePort* como el *Associated port con el Partner Link client*.
- 1.16 Reinicia Eclipse y abre la vista de Server, pulsa el botón derecho en el *Ode v1.x Server at localhost*, selecciona *Add and Remove Projects*. Selecciona el *BPEL_Pick/PickSample.bpel* in la lista de proyectos disponibles y pulsa en *Add* y luego en *Finish*.

- 1.17 Inicia el *ODE server* y mediante el *SOAPUI* lo testeamos. Verás dos operaciones, *sayHello* y *sayGoodbye* que puedes utilizar. Dependiendo de la operación que selecciones, BPEL ejecutará el proceso respectivo.