

Eclipse BPEL designer y Apache ODE

Comienzo de la práctica	22-11-2016
Entrega de la práctica	6-12-2016

1 Introducción

El objetivo de este documento es aprender, de forma tutorial, a desplegar un proceso BPEL –por ejemplo, uno que reciba una cadena de entrada de un *cliente* y le responda devolviéndole exactamente la misma entrada. Se recomienda utilizar el IDE Eclipse (funciona para la versión 4.6 "Neon"), el plugin BPEL Designer (1.0.5), Tomcat y ODE 1.3.6.

1.1 Instalación del plugin BPEL

Se debe instalar el plug-in desde el propio IDE de eclipse:

- *Help* → *Install new software* → *Add*
- Bpel site: <http://download.eclipse.org/bpel/site/1.0.5>
- Seleccionar todos los items que aparezcan en la ventana “Available software to install” → *Eclipse BPEL Designer* e instalarlos

BPEL Designer también se encuentra como un proyecto en GitHub, que se puede clonar y crearse un repositorio local, pero no funciona bien para todas las versiones de Eclipse. Para clonar el proyecto aludido hemos de ir a la vista *Windows* del navegador de proyectos de Eclipse y seleccionar: *Show View* → *Other* → *Git*.

Una vez en la vista *Git Repositories*, seleccionaremos “*Clone a Git Repository*”:

- URI: `https://git.eclipse.org/r/bpel/org.eclipse.bpel`
- Host: `git.eclipse.org`
- Repository path: `/r/bpel/org.eclipse.bpel`
- Protocol: `https`

Seleccionamos todos los paquetes que se nos ofrezcan, que se instalarán, y cambiamos a la vista *BPEL*.

Instalación del servidor ODE

Ahora abriremos la perspectiva *Servers* en nuestro IDE y en el cambas seleccionaremos *new*, tras clicar con el botón derecho del ratón:

- New: *Server*
- Seleccionaremos *Ode v1.x server*, dentro del paquete Apache
- Hemos de descargarnos el desplegable `ode.war` del sitio de distribución oficial de Apache (ver 2 en créditos) y ubicarlo en *webapps* de la distribución de Tomcat que tengamos instalada en nuestro disco duro.
- Pulsando *NEXT* se nos presentará un widget que nos pide información de la ubicación de `ode.war` y de la distribución de *Tomcat* que tengamos instalada localmente. Por ejemplo, indicaremos algo como lo siguiente:
 - JRE: `jre7`¹
 - ODE's home directory: `%TOMCAT_HOME%/webapps/ode`
 - Tomcat's home directory: `%TOMCAT_HOME%`
 - Server address: `localhost`
 - Port: `8080`
 - VM Arguments: `-Xms128m -Xmx512m -XX:MaxPermSize=256m`
- Clicando 2 veces sobre el nombre del servidor seleccionaremos el enlace `LaunchConfiguration` y accediendo a la pestaña *classpath*, añadiremos como *External JARs* el archivo `tomcat-juli.jar`² debajo de *User entries' state*.

Si tras ejecutar el servidor *Ode v1.x server* (icono verde), nos aparece la etiqueta *[Started]*, entonces estaremos seguros de que el servidor está bien configurado.

2 Despliegue de un proceso en BPEL

Nos crearemos un proyecto BPEL nuevo: *File*→*New*→*Other*→*BPEL 2.0*→*BPEL Project*. Por ejemplo, lo llamaremos: *ODE_Prueba* y seleccionaremos como *runtime* objetivo a *Apache ODE 1.x* y su configuración por defecto. Terminamos esta parte pulsando *Finish*.

Ahora vamos a crear el proceso BPEL, para lo cual, sobre la carpeta *ODE_Prueba\bpelContent*, seleccionaremos del menú *File: New*→*Other*→*BPEL 2.0*→*New BPEL process file*.

- *Creation mode*: Create a BPEL process from template
- *BPEL Proces Name*: `HolaMundo`
- *Name Space*: `http://holaMundo`

¹ `jre8`, a veces, da problemas

² se encuentra en `%TOMCAT_HOME%/bin`

Seleccionaremos la plantilla: *Synchronous BPEL Process* y cambiaremos la dirección del servicio a: `http://localhost:8080/ode/processes/HolaMundo`, suponiendo que nuestro entorno Tomcat se ejecuta en el puerto 8080. La sustitución en la pantalla que se nos presenta puede ser:

- *Template*: Synchronous BPEL Process
- *Service Name*: HolaMundoService (autocompletado)
- *Port Name*: HolaMundoPort
- *Service Address*: `http://localhost:8080/ode/processes/HolaMundo`
- *Binding Protocol*: SOAP

Mantendremos las declaraciones anteriores o similares y *Finish*.

Proceso BPEL

Si nos fijamos en el navegador de proyectos, observaremos que se crearon 3 archivos en la carpeta *bpelContent* y que se abre el *editor BPEL* con el archivo *HolaMundo.bpel* para editar dicho proceso. Clicaremos en el *canvas* para abrir el panel *Palette*, que podremos ver en la parte derecha. Seleccionamos *Show Palette* o *Palette View*, para ver lo que muestra la figura 1.

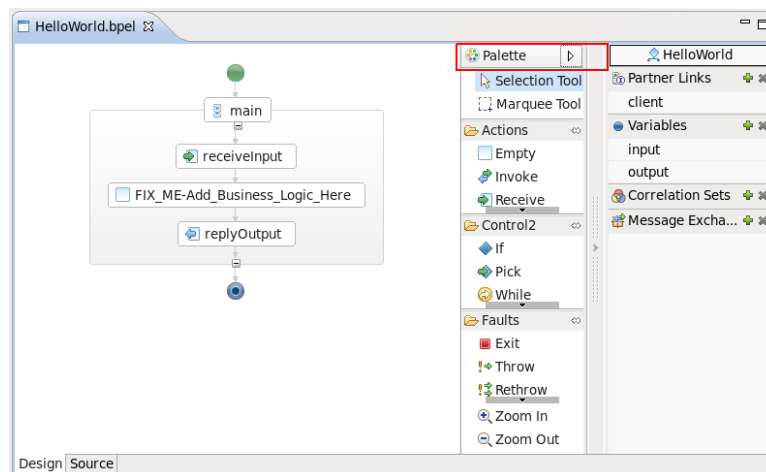


Figura 1: Palette View y proceso BPEL mostrado por el Editor gráfico BPEL

Arrastramos la acción *Assign* que se encuentra dentro del apartado *Actions* de la pestaña *Palette*, la unimos a la acción *receiveInput* sobre el gráfico del proceso BPEL y luego eliminamos el cuadrado *FIX_ME-Add_Business_Logic_Here* del gráfico. El diseño del proceso *BPEL* nos quedaría como muestra la figura 2:

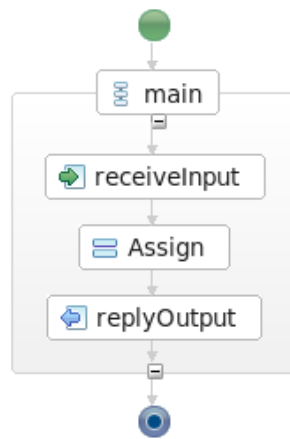


Figura 2: Proceso HolaMundo con el Editor de procesos BPEL

Ahora hay que seleccionar la actividad *Assign* con el botón derecho del ratón y seleccionar *Show in properties* del menú desplegable que aparece. En la pestaña de *Properties*, seleccionaremos *Details* y clicaremos en el botón *New* para, de esta manera, poder definir un operador de asignación (`copy from: to:`) de BPEL. Asignaremos, seleccionando en las ventanas (`from input→payload→input` `to output→payload→result`). Por último, a la pregunta sobre inicialización de la variable le respondemos “yes”.

Ver en la figura 3 cómo se realiza esta asignación gráficamente dentro del editor de procesos BPEL.

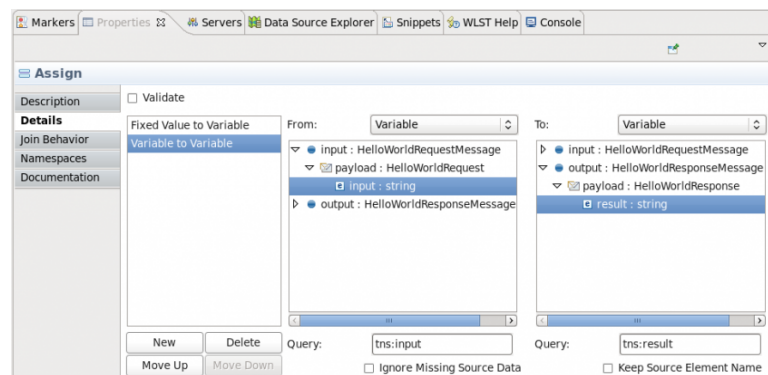


Figura 3: Asignación con copia de una variable dentro de una acción de proceso

Archivo WSDL

Ahora hay que proporcionar un *puerto* y una *ligadura* (“binding”) para comunicar con el servicios. Si clicamos 2 veces en el archivo *HolaMundoArtifacts.wsdl* que nos muestra el navegador de proyectos, dentro de la carpeta *bpelContent*, aparecerá el *editor de WSDL*.

Seleccionaremos *Show Properties* en el menú que nos aparece con el botón derecho sobre *HolaMundoService*. Clicando en *HolaMundoPort* aparece un menú desplegable que se abre. Hemos de asegurarnos que la dirección del puerto está asignada a: `http://localhost:8080/ode/processes/HolaMundo`. La sustitución en la pantalla que se nos presenta puede ser:

- **Name:** HolaMundoPort
- **Binding:** HolaMundoBinding
- **Address:** `http://localhost:8080/ode/processes/HolaMundo`
- **Protocol:** SOAP

La ventana del editor nos indicará que su contenido está *sucio* (**HolaMundoArtifact*), por tanto, hemos de salvar el editor WSDL con la definición que acabamos de terminar.

The screenshot shows the BPEL Deployment Editor interface. The 'General' tab is active, showing 'This process is' set to 'activated' and an unchecked checkbox for 'Run this process in memory'. Below this is the 'Inbound Interfaces (Services)' section, which contains a table with the following data:

Partner Link	Associated Port	Related Service	Binding Used
client	HelloWorldPort	//helloworld.localhost}HelloWorld	HelloWorldBinding

Below the inbound interfaces is the 'Outbound Interfaces (Invokes)' section, which also contains a table with the same headers but no data rows.

Figura 4: Vista de *deploy.xml* en el Editor de Despliegue BPEL

Despliegue del servicio

Hemos creado un servicio muy básico que acepta una cadena del cliente y se la devuelve, mostrándosela. Como todo SW hay que desplegarlo para que puedan acceder a él los clientes. Clicamos en la carpeta *bpelContent* y seleccionamos *File→New→Others→BPEL 2.0→BPEL Deployment Descriptor*. Ahora hacemos doble-clic en el archivo *deploy.xml* que se encuentra dentro de la carpeta *bpelContent* y se nos abrirá un nuevo editor: *ODE Deployment Descriptor Editor*, tal como muestra la figura 4.

En *editor de despliegue BPEL* nos mostrará la tabla denominada *Inbound Interfaces* (ver 4). Esta tabla contiene las interfaces que proporciona el proceso, especificando el *servicio*, *puerto* y la *ligadura* que queremos utilizar para cada *PartnerLink* que aparece en las filas de esta tabla. En este ejemplo el *Partner Link* se llamará *client*.

Clicaremos en el campo debajo de la columna *Associated Port* de la tabla anterior y nos aparecerá un cursor de despliegue para seleccionar *HolaMundoPort*. Finalmente, clicaremos en *Related Service* y se rellenará automáticamente, junto con los otros campos de la fila.

La ventana del editor nos indicará que su contenido está *sucio* y hemos de salvar el editor con el archivo *deploy.xml*.

Tenemos, por tanto, el modelado del servicio simple *HolaMundo* terminado ya, que se compone de los siguientes elementos:

1. Un proyecto BPEL: *ODE_Prueba*
2. Un proceso BPEL: *HolaMundo.bpel*
3. El archivo con el servicio Web: *HolaMundoArtifact.wsdl*
4. Descriptor de despliegue del servicio: *deploy.xml*

Ahora tendremos que añadir el proyecto BPEL al servidor *Ode v1.x server* arrancado.

Ejecución del servicio

Iremos a la vista *Servers* de nuestro IDE y arrancaremos el servidor ODE (si no lo hemos hecho ya al comenzar este tutorial) y con el botón derecho seleccionaremos *Add and Remove*, nos aparecerá un menú partido con *ODE_Prueba* en la columna izquierda (*Available*), que añadiremos a la columna derecha (*Configured*). Si apareciese una ventana de notificación o si el menú partido tuviera su columna izquierda vacía, hay que cerrar el *runtime* de Eclipse y volver a configurar el servidor *Ode v1.x*, a través del enlace *launch and configuration* de su menú de configuración. Por último, seleccionamos *Add y Finish*.

Si el despliegue del servicio *HolaMundo* se ha realizado correctamente, observaremos que se escriben unos mensajes similares a los siguientes en la consola, si clicamos sobre la pestaña correspondiente de nuestro IDE:

```
11:44:49 INFO [BpelServerImpl] Registered process {http://holamundo.localhost}HolaMundo-1.  
11:44:49 INFO [DeploymentPoller] Deployment of artifact ODE_Prueba successful:  
[{http://holamundo.localhost}HolaMundo-1]
```

Consola de administración de ODE

Si escribimos la dirección `http://localhost:8080/ode/` en el navegador (ver figura 5) y seleccionamos el enlace “Services” (`http://localhost:8080/ode/deployment/services/`), aparecerá la lista de los servicios instalados en el puerto 8080 de nuestra máquina local. Se puede acceder también desde la página inicial de la consola seleccionando *Deployment browser* → *Process services* mostrándose la pantalla de la figura 6.

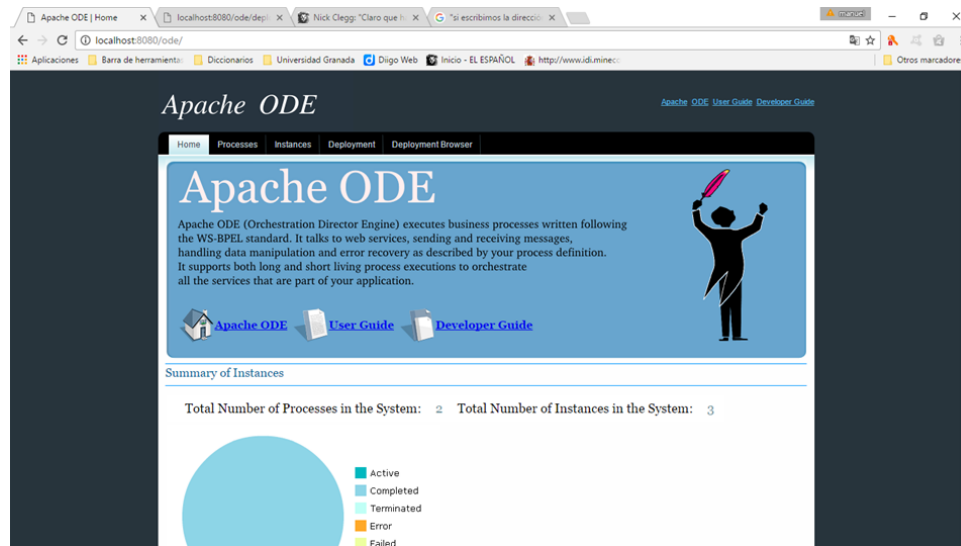


Figura 5: Vista de la consola administrativa de ODE abierta localmente con el navegador

Si el despliegue del servicio desarrollado en el tutorial se realizó correctamente, ha de aparecer la entrada *HolaMundo* en la lista de servicios siguiente:

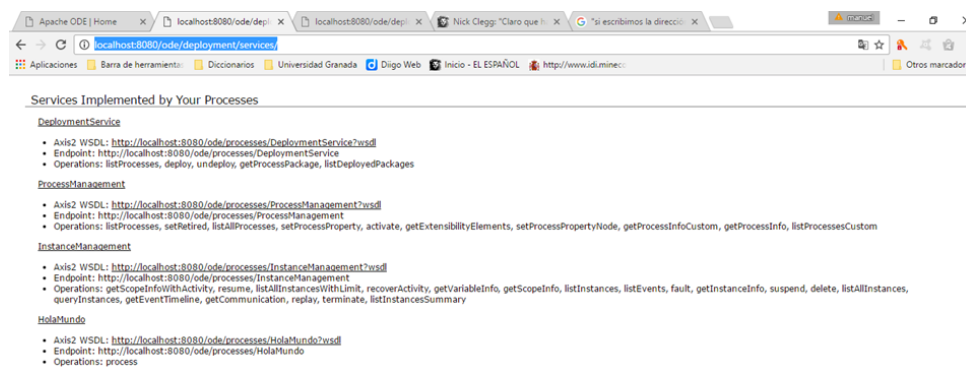
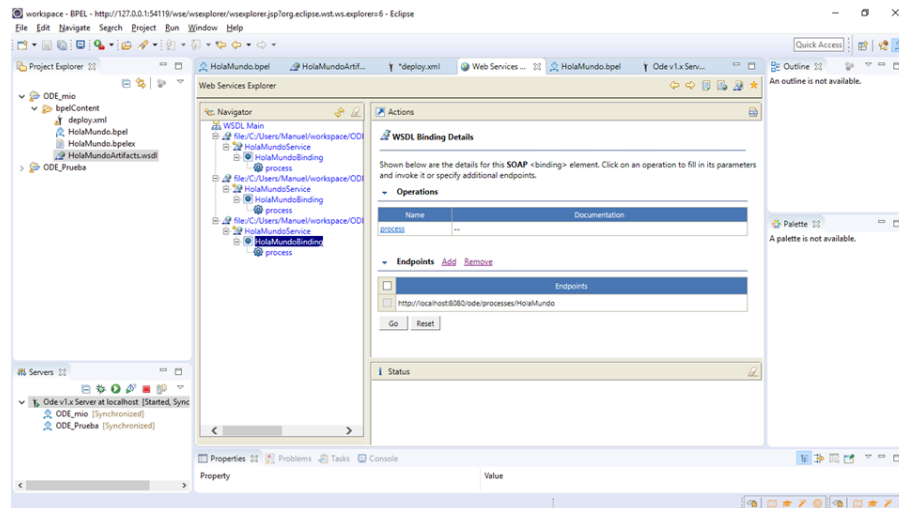
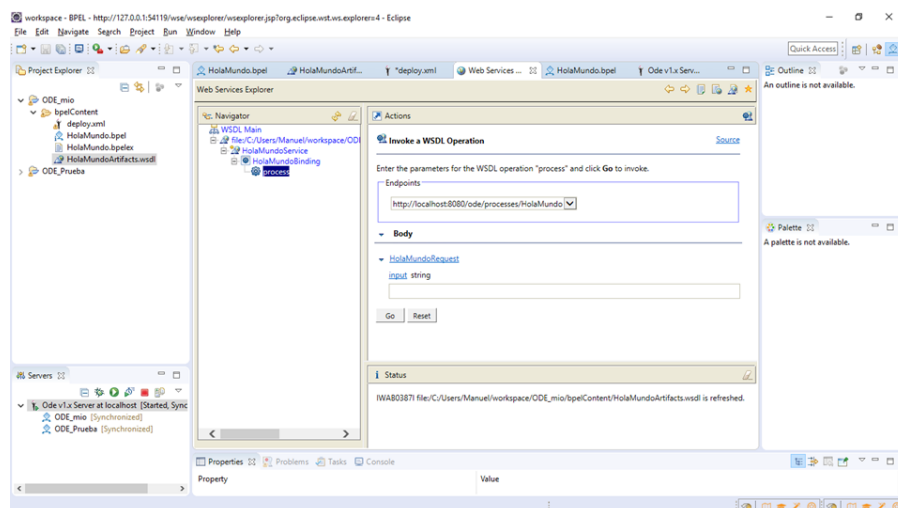
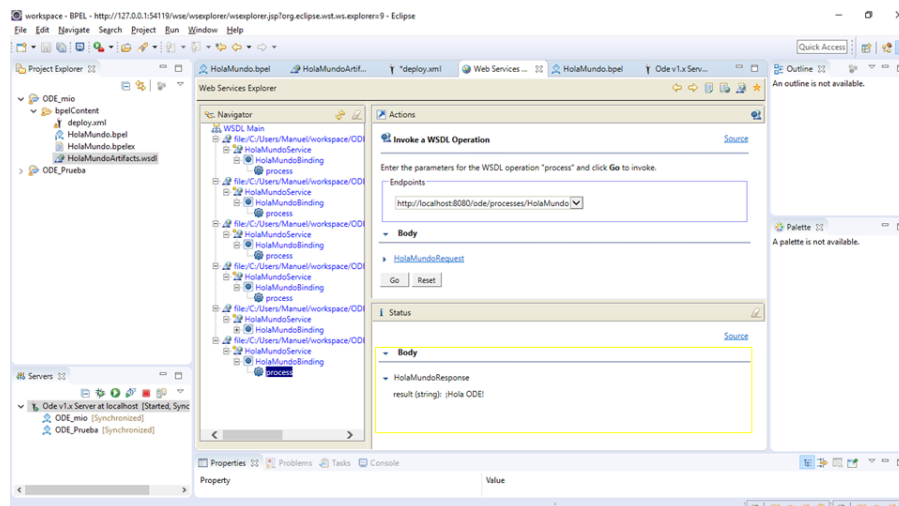


Figura 6: Servicios desplegados en el servidor *Ode v1.x* localmente

3 Ejecución de proceso BPEL con el navegador de servicios Web de Eclipse

Ahora buscamos en el explorador de proyectos de Eclipse el archivo *HolaMundoArtifacts.wsdl* que seleccionamos con el botón derecho, nos aparecerá un menú desplegable: *Web Services*→*Test with Web Services Explorer* y entonces aparecerá el *explorador de servicios Web* de Eclipse. En la parte izquierda del navegador, hay que expandir el árbol debajo de `file:///.../ODE_Prueba/bpelContent/HolaMundoArtifacts.wsdl` hasta que se muestra `process`. El resultado ha de ser similar a la captura que muestra la figura 7.

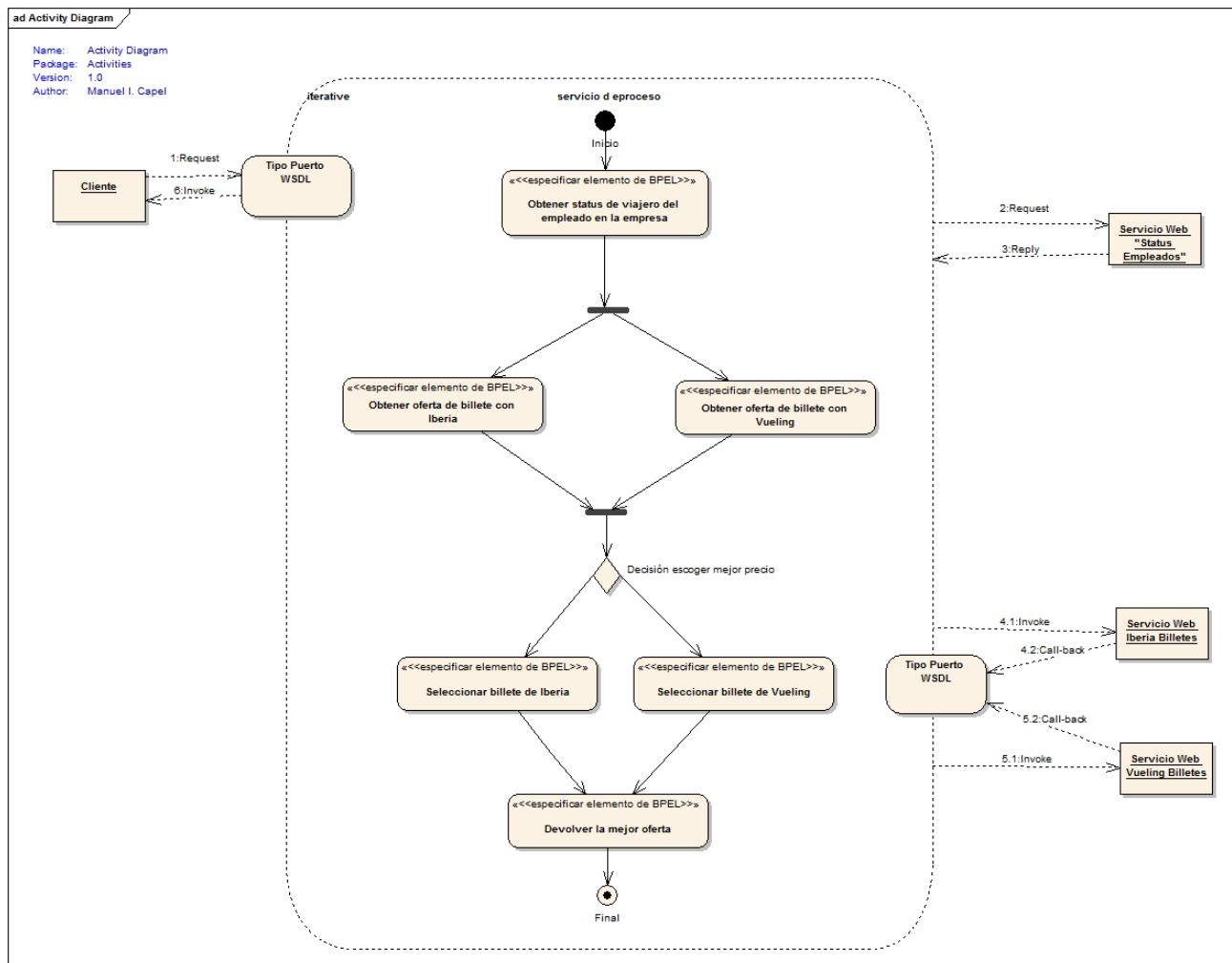
Ahora hay que clicar en el enlace `process` de la tabla *Operations* para que nos aparezca un cuadro de texto de sustitución para los parámetros de la operación cuyo comportamiento queramos probar y poder llamarla o especificar puntos finales (*endpoints*) adicionales. Aparecerá la captura de pantalla, que se muestra en la figura 8; escribimos el mensaje “¡Hola ODE!” en la ventana de texto y pulsamos el botón “Go”, entonces nos ha de aparecer el resultado de la ejecución de este servicio simple en la ventana inferior (ver figura 9).

Figura 7: Explorador de servicios Web de Eclipse mostrando el servicio-ejemplo *HolaMundoService*Figura 8: Interfaz de ejecución del servicio *HolaMundo* en el explorador de servicios Web de EclipseFigura 9: Resultado de la ejecución del servicio *HolaMundo* en el explorador de servicios Web

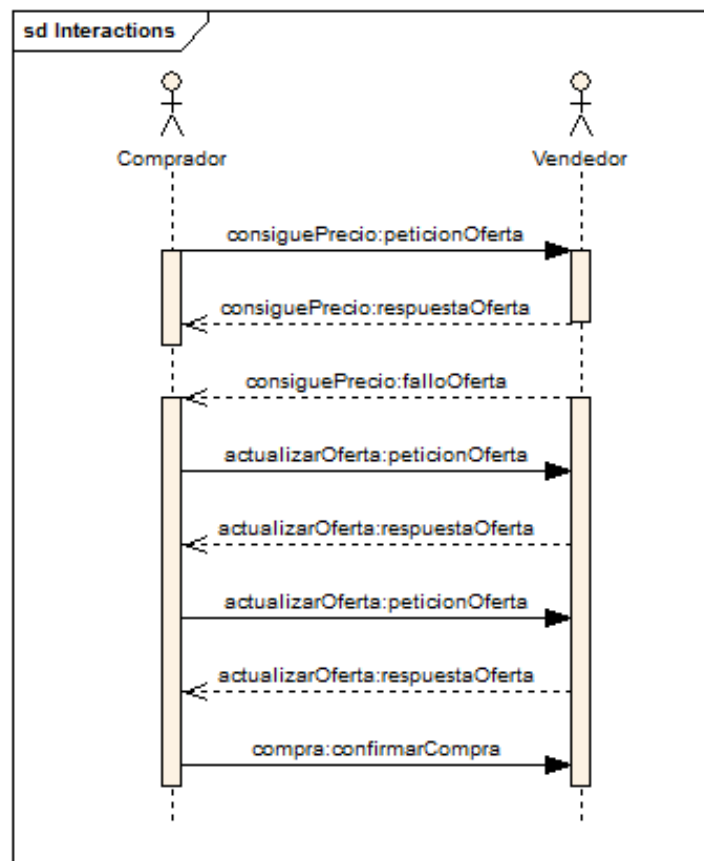
4 Práctica

Se propone el modelado con BPEL 2.0, utilizando *BPEL Designer* para Eclipse, de los siguientes procesos de negocio:

1. Especificar utilizando BPEL el proceso de negocio (cuyo diagrama de actividad se muestra en la siguiente figura) y que se describe informalmente como sigue: “El cliente invoca al proceso de negocio, especificando el nombre del empleado, el destino de su viaje, la fecha de salida, y la fecha de regreso. El proceso de negocio BPEL comprueba primero el *status* del empleado para viajar (clase turista, business, avión privado); suponemos que existe un servicio Web contra el que se puede realizar esta consulta. Después, el proceso BPEL comprobará el precio del billete con 2 líneas aéreas: Iberia y Vueling. Supongamos otra vez que ambas compañías proporcionan un servicio Web contra el que se pueden realizar estas gestiones. El proceso BPEL seleccionará el precio más bajo y devolverá un plan de viaje al cliente”.



2. Orquestar, de forma simplificada, el mercadeo entre un comprador y un vendedor de un producto solicitado, de acuerdo con el diagrama de interacción que se muestra en la siguiente figura. El comprador comienza pidiendo un precio al vendedor y el vendedor responde con un precio para el producto o una excepción si no conoce el artículo que le demandan o no estuviera disponible en el almacén. El comprador continua pidiendo precio al vendedor y entra en un comportamiento repetitivo con actualizaciones (del precio del artículo) hasta que decide comprar el artículo cuando considera que se le ofrece el mejor precio. Se pide en este ejercicio desarrollar la descripción completa de la orquestación que se ha descrito anteriormente entre el comprador y el vendedor.



Créditos

1. <http://www.oracle.com/technetwork/articles/matjaz-bpell-090575.html>
2. <http://www.apache.org/dyn/closer.cgi/ode/apache-ode-war-1.3.6.zip>
3. http://www-inf.int-evry.fr/SIMBAD/courses/doku.php?id=teaching_assistant:bpel:deploying_hello_world_bpel_process_with_eclipse_bpel_designer_and_apache_ode
4. <http://localhost:8080/ode/index.html>
5. <http://ode.apache.org/userguide/>