

ANEXO D

TUTORIAL BPEL

Introducción

Este documento está compuesto por dos tutoriales, los cuales tratarán de introducir al lector al manejo de dos de las herramientas más importantes dentro de la definición de procesos de negocio mediante BPEL, ActiveBPEL y Oracle BPEL.

Objetivos

- Conocer aspectos acerca de lenguaje de descripción de procesos de negocio BPEL.
- Conocer y manejar aspectos básicos de las herramientas ActiveBPEL y OracleBPEL

Prerrequisitos

- Conocimientos básicos de BPEL
- Conocimientos de Servicios Web

TUTORIAL ActiveBPEL Designer

Introducción

La presente guía constituye una traducción resumida no autorizada de la guía ubicada en el sitio web de ActiveBPEL con algunos cambios. Dirigida y adaptada para estudiantes de la Universidad del Cauca interesados en la definición, prueba y despliegue de procesos BPEL.

Objetivos

- Crear un proyecto de ejemplo
- Crear un nuevo documento del proceso
- Familiarizarse con las herramientas de ActiveBPEL que permiten crear procesos válidos.

Base Teórica

BPEL (Lenguaje de Ejecución de Procesos de Negocio)

Lenguaje utilizado por BDMobIS. Se basa en el secuenciamiento de servicios web para sacar su máximo provecho y llegar a la consecución de un objetivo, es decir, la ejecución de un proceso de negocio, como se dijo previamente. En sus inicios BPEL era simplemente eso, un lenguaje. Este concepto solamente encerraba ideas como el secuenciamiento de un solo proceso en el cual otros ayudantes eran poco aclamados por el sistema. Los procesos de negocios de las empresas eran, por decirlo así, menos complejos que los que hoy en día se definen y es por tal motivo que algún protocolo o método de organización mas complejo era inoficioso. Hoy en día las empresas demandan de herramientas más poderosas en el tratamiento de sus procesos de negocios en donde la intervención de diferentes socios o partners es algo necesario para poder conseguir solucionar un problema crítico interno de una empresa.

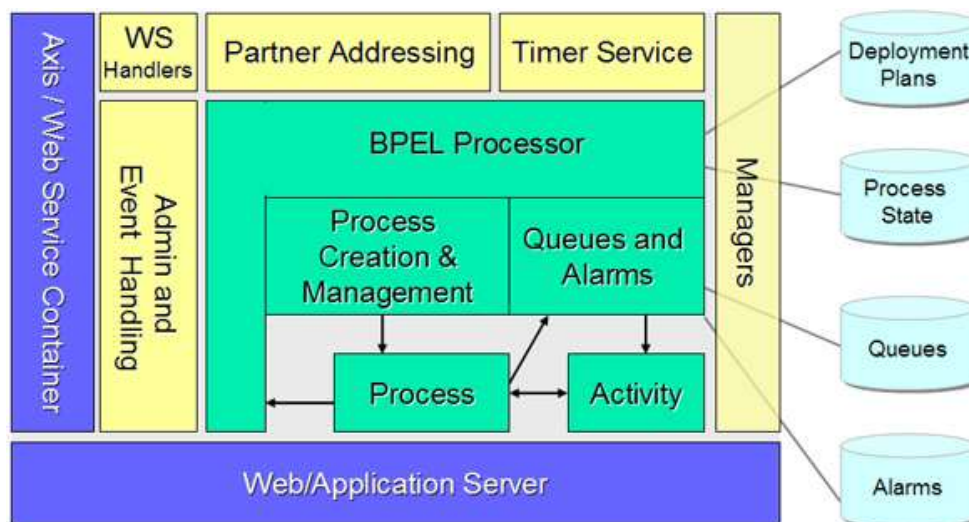
ActiveBPEL

ActiveBPEL permite definir procesos de negocio BPEL a través de la herramienta ActiveBPEL designer. ActiveBPEL Designer es un plug-in para eclipse, que permite crear, probar, desplegar y ejecutar aplicaciones software compuestas. Las

aplicaciones compuestas están basadas en la arquitectura SOA. Las interfaces para todas las aplicaciones componentes se describen utilizando convenciones estándar de servicios (en particular WSDL) y entonces los servicios se tejen en composiciones coherentes de servicios. La meta de la composición es desarrollar una arquitectura de información que sea altamente adaptativa para que envuelva las necesidades de los usuarios.

El Motor Active BPEL

El motor ActiveBPEL es quien ejecuta los procesos



Los elementos de BD a la derecha de la figura representan almacenamiento persistente genérico, es decir que se pueden utilizar diferentes administradores de persistencia.

Iniciar el Motor:

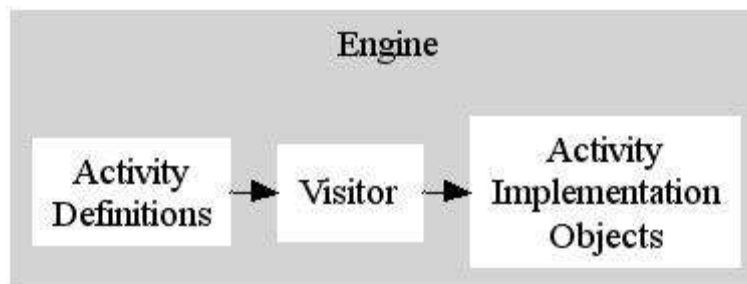
La fábrica de motores administra la creación de un motor ActiveBPEL. La separación de responsabilidades se logra mediante el uso de objetos que manipulan servicios tales como administración de colas, alarmas, y servicios de contadores de tiempo (timers). El siguiente pseudocódigo describe la creación del motor y el soporte de servicios.

```
engine = new Engine(getEngineConfigurationInfo(),
                   createQueueManager(),
                   createProcessStateManager());
engine.setPlanManager(createProcessDeploymentProvider());
createProcessDeploymentManager();
createWorkManager();
createAlarmAndTimerService();
```

El motor se puede configurar con un objeto que proporcione valores por defecto y lea el archivo aeEngineConfig.xml. Un proveedor de despliegue de procesos maneja archivos Process Deployment Descriptor (.pdd) y un administrador de despliegue de procesos maneja la creación del proceso. Un administrador de trabajo programa sus operaciones asíncronas.

Creación de Procesos

Un procesos BPEL nuevo se crea cuando una de sus actividades start se dispara, o mediante un mensaje entrante o a través de una alarma de actividades Pick. Cuando el motor lee una definición de proceso BPEL crea un objeto denominado *definiciones de actividades* que modela al proceso. En este contexto las definiciones son como las clases mientras los objetos de implementación son análogos a las instancias de esas clases.



El objeto *Definiciones de actividades* contiene toda la información requerida para instanciar un objeto *implementación de actividades* BPEL. Los eventos en el motor tienen un valor XPath que indica que actividad en el proceso está disparando el evento. Esos XPath vienen desde las definiciones de actividades.

El motor crea la implementación de objetos usando un patrón Visitor para visitar el modelo de objetos de definición de actividades y crear los objetos implementados desde este modelo. Los archivos de sucesos (log) se almacenan en el directorio $(user.home)/AeBpelEngine/process-logs$, donde $(user.home)$ es la ubicación del directorio del sistema java con la propiedad "user.home". Para activar logging se puede utilizar la página de configuración BpelAdmin que normalmente está en <http://localhost:8080/BpelAdmin/config.jsp>.

Procesos:

Un proceso consiste de

- **Partner links:** describe la relación entre dos WS a nivel de interfaz.

- **Partners:** son entidades que toman parte en una transacción de un WS.
- **Variables:** son contenedores para valores.
- **Grupos correlacionados:** son grupos de datos que únicamente identifica el proceso de negocio. En diferentes tiempos el proceso puede identificar diferentes grupos correlacionados.
- **Fault handlers:** describe que hacer cuando ocurran problemas.
- **Compensation handlers:** describe como dar marcha atrás a procesos de negocio completados.
- **Una actividad de alto nivel:** es una sola actividad BPEL que usualmente contiene otras actividades.

DISPATCHING REQUEST (Solicitud de envío)

Cada proceso BPEL debe tener al menos una actividad start.

Actividades:

Las actividades son los bloques que constituyen un proceso BPEL.

- **Actividades Básicas:** son conceptualmente simples comportamientos como recibir un mensaje, invocar un servicio web y asignar valores a variables.
- **Actividades Estructuradas:** son similares a condiciones y ciclos de los lenguajes de programación.
- **Actividades Especiales:** introducen variables de alcance y manipulan actividades anormales como terminación de procesos y compensación (explícitamente "deshacer" un proceso).

Cada actividad existe dentro de un ambiente que representa un contexto para variables, manipuladores de fallas y manipuladores de compensación. Los ambientes son similares a bloques de lenguajes de programación que introducen nuevas variables de alcance y mecanismos de manipulación de excepciones. Algunas variables como Scope e Invoke generan nuevos ambientes, tanto implícitos como explícitos.

Actividades BPEL

Actividades Básicas

Nombre	Descripción
<receive>	Bloquea y espera mensajes desde un partner
<reply>	Responde al partner que haya enviado el mensaje cuando este sea recibido

<invoke>	Llama algún servicio web, ya sea one-way o request-response
<assign>	Asigna o copia valores a variables
<throw>	Genera una falla
<wait>	Espera hasta que un tiempo particular haya pasado (alarm)
<empty>	Un no-op.

Actividades estructuradas

Nombre	Descripción
<sequence>	Ejecuta hijos en orden
<switch>	Tal como una sentencia "switch" o "case"
<while>	Repite una actividad mientras una condición se cumpla.
<pick>	Bloquea y espera por la llegada de un mensaje.
<flow>	Ejecuta hijos concurrentemente.

Actividades Especiales

Nombre	Descripción
<scope>	Define un nuevo alcance para las variables, manipuladores de fallas y manipuladores de compensación.
<compensate>	Invoca compensación en el interior de un ambiente que ya se completo normalmente.
<terminate>	Inmediatamente finaliza una instancia de un proceso de negocio.

Estados

Cada actividad tiene asociado un estado. Las actividades entran o salen de un estado de acuerdo a las reglas de BPEL. Las actividades también pueden disparar eventos para notificar que cambian de estado. Las constantes de estado están definidas en AeBpelState.

- INACTIVE Cuando el proceso arranca
- READY_TO Lista para ejecutarse. Cuando las actividades están en la cola.
- EXECUTE La condición de unión se evalúa true
- EXECUTING Actualmente ejecutándose
- FINISHED Ejecución finalizada sin fallas
- FAULTED Ejecución finalizada con falla
- DEAD_PATH Removida del path de ejecución debido a la eliminación de path muerto. Cuando una actividad padre llega al estado DEAD_PATH ese estado se propaga a todos sus hijos.
- QUEUED_BY_PARTEN Ejecución puesta en cola por los padres

- TERMINATED Terminada
- UNKNOWN cuando el estado de la actividad es null. Cuando una actividad padre llega a un estado desconocido, entonces los hijos cambian de estado a INACTIVE.

Métodos comunes de las actividades

Las actividades utilizan método para pasar de un estado a otro y para disparar eventos.

- **isReadyToExecute():** Retorna true si la actividad está lista para preparar la ejecución.
- **execute():** Las actividades pueden fijar algunos elementos de soporte requeridos para la ejecución de la actividad.
- **objectCompleted():** procesa cualquier link saliente y luego indica al proceso que la actividad ha sido completada.
- **terminate()** lo llaman los procesos tipo padre cuando hay una falla y la actividad requiere ser finalizada.

Archivos

aeEngineConfig.xml se carga cuando se crea un motor ActiveBPEL.

Archivos de Procesos BPEL.

***.bpr** Es un proceso BPEL almacenado. Contiene el Descriptor de Despliegue del Proceso (.pdd), el WSDL, y archivos relacionados necesarios para desplegar un procesos BPEL.

***.pdd** (Process Deployment Descriptor) indica al motor BPEL acerca de un proceso BPEL. Describe los partner links y los archivos WSDL necesarios.

Para cada proceso BPEL que se quiera desplegar se debe crear un archivo PDD. Cada archivo .bpr tiene su propio archivo .pdd. Los elementos <process> contienen elementos partner links y referencias WSDL.

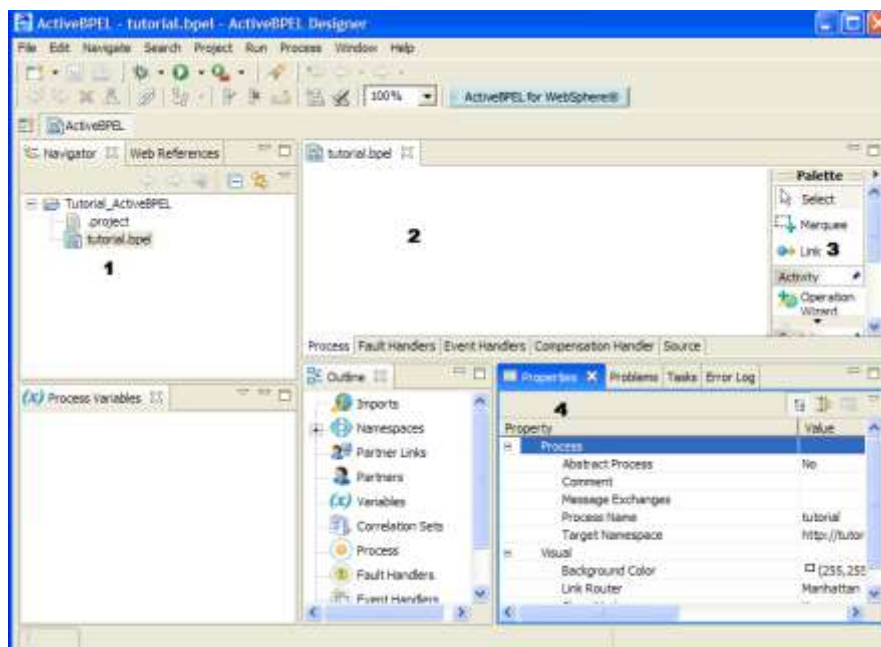
***wsdlCatalog.xml** El catálogo WSDL provee un camino para que el motor BPEL encuentre los archivos WSDL dentro de un archivo de despliegue .bpr. wsdlCatalog.xml reside en el directorio META_INF.

***.pdef.** describe las relaciones entre los partners. No es necesario para todos los procesos BPEL. Cuando se requiere autenticación se utiliza este archivo para proveer la información de autenticación.

XSD es una librería que provee un API para manipular los componentes de un esquema XML.

Entorno de desarrollo de Active BPEL

Como ya se dijo anteriormente ActiveBPEL Designer es un plug-in de eclipse por lo tanto se puede observar su entorno tal como en la siguiente figura



Donde:

1. Corresponde a la vista navigator que muestra los archivos del proceso.
2. Corresponde al editor de procesos.
3. La vista *palette* contiene un grupo de íconos que se pueden arrastrar hacia el editor de procesos para desarrollar una definición de proceso.
4. La vista de propiedades muestra los atributos del objeto seleccionado en el editor de procesos.

Inicio del Tutorial

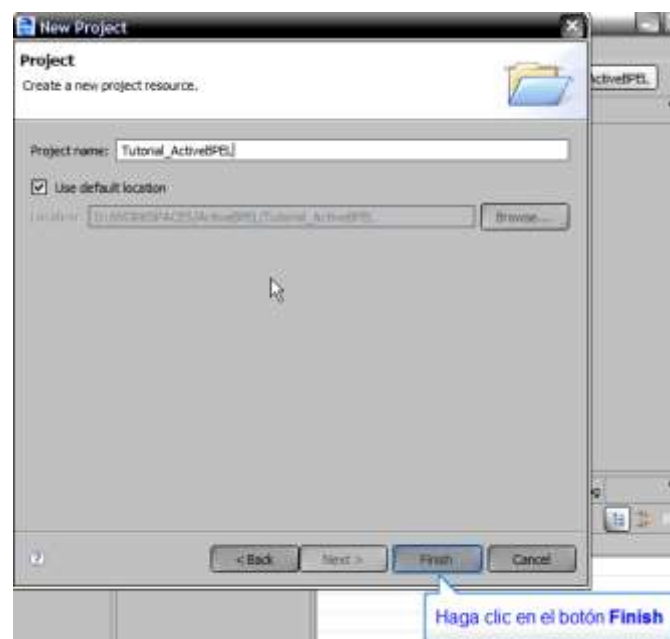
En este tutorial se realiza un ejemplo de aprobación de préstamos, el cual describe las interacciones entre tres participantes, el control del flujo y decisiones basadas en el intercambio de datos. El proceso está basado en un WSDL que describe los mensajes y las operaciones de negocio utilizadas en el proceso.

En ActiveBPEL se puede diseñar un proceso utilizando las técnicas top-down o bottom-up. En la técnica top-down se bosqueja el proceso arrastrando actividades y creando enlaces entre ellas. Luego se adiciona toda la información necesaria para ligar las actividades a una implementación. En la técnica bottom-up se utiliza

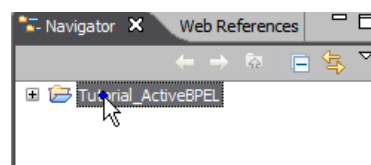
definiciones de la implementación que estén disponibles cuando se empieza la creación del proceso.

Parte 1: Crear un nuevo proceso

1. Crearemos el proyecto Tutorial_ActiveBPEL.
2. Seleccione File-New-Project-Simple-Project y haga clic en siguiente.
3. Escriba Tutorial_ActiveBPEL como nombre del proyecto y haga clic en siguiente.
4. Haga clic en "finish"

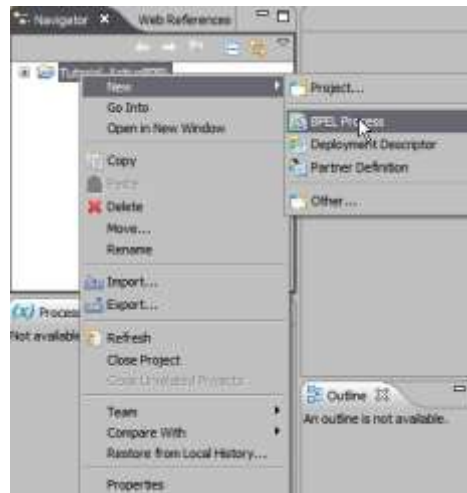


En el navegador se debe mostrar lo siguiente:

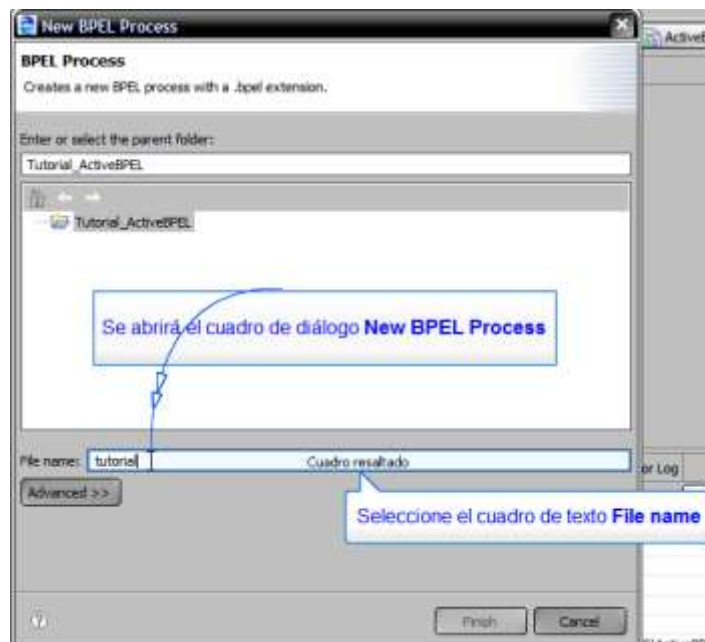


Parte 2: Crear un archivo BPEL de definición del proceso.

1. Haga clic derecho sobre el nodo del proyecto, luego seleccione New>BPEL Process



2. Seleccione la carpeta Tutorial_ActiveBPEL y en el campo de nombre de archivo escriba *tutorial*. La extensión *.bpel* se adicionará automáticamente.



3. Haga clic en el botón “finish”

Se abre un nuevo archivo denominado tutorial.bpel en el Editor de Procesos, tal como se muestra en la siguiente ilustración. Se puede notar además que aparece un mensaje de error que dice , “No activity designated to create instance.” Este mensaje hace parte de la validación BPEL. Por ahora ignoremos ese mensaje.

2 errors, 0 warnings, 0 infos	
Description	Resource
Errors (2 items)	
✘ Container /process is missing required a	tutorial.bpel
✘ No activity designated to create instanc	tutorial.bpel

Parte 3: Planear y diseñar un proceso

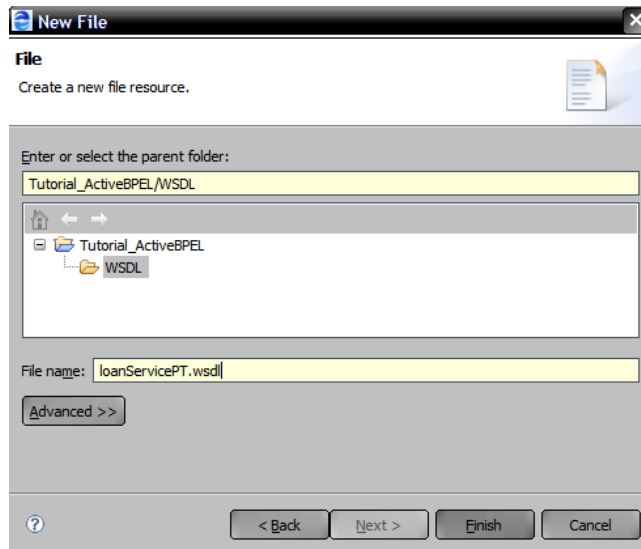
En este paso se utilizará el proceso de ejemplo colocado en la especificación de BPEL4WS [BWS]. En el cual muestra un proceso simple de tres participantes para la aprobación de un crédito.

El proceso de aprobación de crédito inicia recibiendo el requerimiento de una cantidad de dinero en préstamo que realiza un cliente. Si el valor es menor que \$10000, se invoca el servicio Web del asesor para calificar a un cliente como alto o bajo riesgo. Si el cliente es de bajo riesgo entonces se aprueba el préstamo. De lo contrario, el requerimiento se envía a revisión del aprobador. El cliente puede recibir como respuesta la aprobación del asesor o aceptado/rechazado del aprobador.

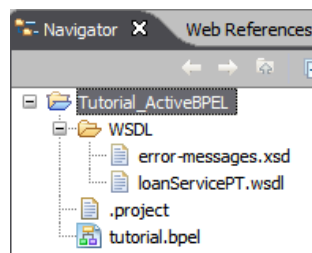
1. Haga clic derecho sobre el nodo del proyecto Tutorial_ActiveBPEL y a continuación haga clic en New>Other>Folder >Next
2. Escriba "WSDL" en Folder Name.



3. Haga clic en Finish
4. Haga clic derecho sobre la carpeta WSDL y a continuación clic en New>Other>File y en el campo File name escriba loanServicePT.wsdl y finalmente haga clic en el botón Finish.



5. De la misma manera cree otro archivo denominado error-messages.xsd, de esta manera el árbol de su proyecto debe quedar como sigue.



6. Edite los archivos con el siguiente código:

- a. loanServicePT.wsdl

```

<wsdl:definitions
  targetNamespace="http://example.com/loan-approval/wsdl/"
  xmlns:ens="http://example.com/loan-approval/xsd/error-messages/"
  xmlns:plnk="http://docs.oasis-open.org/wsbpel/2.0/plnktype"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://example.com/loan-approval/wsdl/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema elementFormDefault="qualified">
      <xsd:import
        namespace="http://example.com/loan-approval/xsd/error-messages/"
        schemaLocation="error-messages.xsd" />
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="creditInformationMessage">
    <wsdl:part name="firstName" type="xsd:string" />
    <wsdl:part name="name" type="xsd:string" />
    <wsdl:part name="amount" type="xsd:integer" />
  </wsdl:message>
  <wsdl:message name="approvalMessage">
    <wsdl:part name="accept" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="riskAssessmentMessage">
    <wsdl:part name="level" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="errorMessage">
    <wsdl:part name="errorCode" element="ens:integer" />
  </wsdl:message>
  <wsdl:portType name="loanServicePT">
    <wsdl:operation name="request">
      <wsdl:input message="tns:creditInformationMessage" />
      <wsdl:output message="tns:approvalMessage" />
      <wsdl:fault name="unableToHandleRequest"
        message="tns:errorMessage" />
    </wsdl:operation>
  </wsdl:portType>

```

```

<wsdl:portType name="riskAssessmentPT">
  <wsdl:operation name="check">
    <wsdl:input message="tns:creditInformationMessage" />
    <wsdl:output message="tns:riskAssessmentMessage" />
    <wsdl:fault name="loanProcessFault"
      message="tns:errorMessage" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:portType name="loanApprovalPT">
  <wsdl:operation name="approve">
    <wsdl:input message="tns:creditInformationMessage" />
    <wsdl:output message="tns:approvalMessage" />
    <wsdl:fault name="loanProcessFault"
      message="tns:errorMessage" />
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="SOAPBinding" type="tns:loanApprovalPT">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="approve">
    <soap:operation soapAction="" style="rpc" />
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:loanapprover"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:loanapprover"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="LoanApprover">
  <wsdl:documentation>Loan Approver Service</wsdl:documentation>
  <wsdl:port name="SOAPPort" binding="tns:SOAPBinding">
    <soap:address
      location="http://localhost:8080/active-
      bpel/services/ApproverWebService" />
  </wsdl:port>
</wsdl:service>
<wsdl:binding name="SOAPBinding1" type="tns:riskAssessmentPT">
  <soap:binding style="rpc"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="check">
    <soap:operation soapAction="" style="rpc" />
    <wsdl:input>
      <soap:body use="encoded" namespace="urn:loanassessor"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="encoded" namespace="urn:loanassessor"
        encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="LoanAssessor">
  <wsdl:documentation>Loan Assessor Service</wsdl:documentation>
  <wsdl:port name="SOAPPort1" binding="tns:SOAPBinding1">
    <soap:address
      location="http://localhost:8080/active-
      bpel/services/AssessorWebService" />
  </wsdl:port>
</wsdl:service>
<plnk:partnerLinkType name="loanPartnerLT">
  <plnk:role name="loanService" portType="tns:loanServicePT" />
</plnk:partnerLinkType>
<plnk:partnerLinkType name="loanApprovalLT">
  <plnk:role name="approver" portType="tns:loanApprovalPT" />
</plnk:partnerLinkType>
<plnk:partnerLinkType name="riskAssessmentLT">
  <plnk:role name="assessor" portType="tns:riskAssessmentPT" />
</plnk:partnerLinkType>
</wsdl:definitions>

```

b. error-messages.xsd

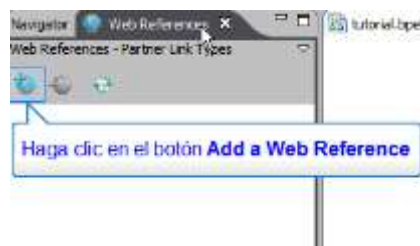
```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://example.com/loan-approval/xsd/error-messages/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://example.com/loan-approval/xsd/error-messages/"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="integer" type="xs:integer"/>
</xs:schema>

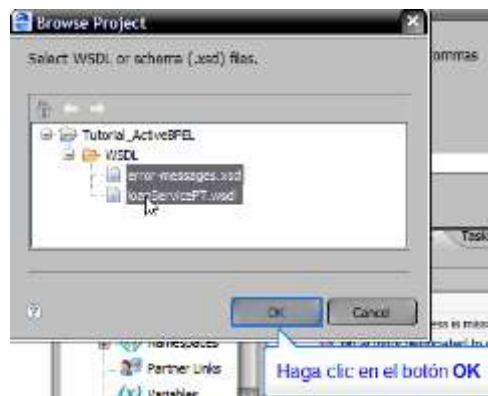
```

Parte 4: Adicionando Referencias Web y Actividades de Interacción Web.

1. Haga clic en la pestaña Web References, que se puede observar en la figura anterior justo al lado del navegador.
2. Haga clic en el ícono Add Web References. Una referencia Web es un archivo WSDL o un esquema. Adicionando archivos WSDL a la vista Web Referencias permite registrar de manera conveniente namespaces, messages, type definitions, simple data y otros elementos necesarios en la definición del proceso. Para este caso utilizaremos el esquema error-messages.xsd y el archivo WSDL loanServicePT.wsdl creados anteriormente.



3. En la ventana Add Web Reference haga clic en el botón Browse.
4. En la ventana Browse Project expanda el proyecto, luego expanda la carpeta WSDL y seleccione los dos archivos que esta contiene y haga clic en "OK"



5. A continuación haga clic en el botón Menú ubicado en la parte superior derecha de la pestaña Web References, tal como se muestra en la siguiente ilustración.



6. Y en el menú emergente haga clic en View Partner Link Types. Ahora tendrá en la pestaña Web References tres Partner Link Types, seleccione

loanPartnerPT y expándalo hasta llegar a la actividad request tal como muestra la siguiente figura.



7. Arrastre la operación Request hacia el Editor de procesos (Ver entorno de desarrollo ActiveBPEL en la parte superior de este documento).
8. En la ventana Operation:Request seleccione Receive-Reply y haga clic en Next



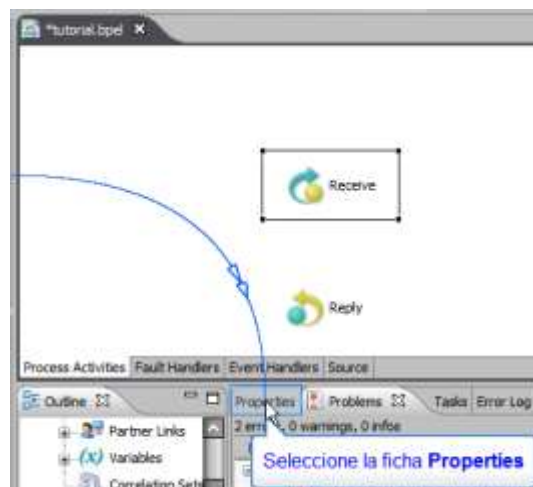
9. ActiveBPEL sugiere el nombre loanPartnerLT en el campo Name, cambie este nombre por customer y haga clic en Next.



10. En variable de entrada en el campo Name cambie creditInformationMessage por request. Haga clic en Next.



11. De la misma manera en la variable de salida cambie approvalMessage por approval y haga clic en Finish.
12. Aparecerán ahora dos actividades receive y reply. Haga clic sobre la actividad Reply y a continuación clic en la pestaña Properties



13. Observe que las propiedades tengan valores válidos en los campos *partner link*, *operation* y *variable*. En el campo Activity Name coloque ReceiveCustomerRequestforLoanAmt.

Activity Name	
Activity Type	Receive
Comment	
Correlations	(none)
Create Instance	No
Documentation	
From Parts	(none)
Message Exchange	
Operation	request
Partner Link	customer
Port Type	loanServicePT
Variable	request

14. Seleccione la actividad Reply, luego seleccione sus propiedades y en la casilla Activity Name coloque AcceptMessageToCustomer.

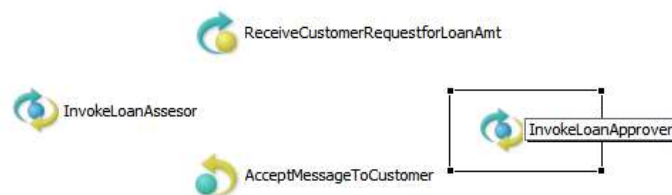


15. A continuación en la pestaña Web References expanda el partner link riskAssessmentLT y encuentre la operación chek y arrástrale hacia el editor de procesos. En la ventana Operation: chek seleccione Invoke y haga clic en Next.

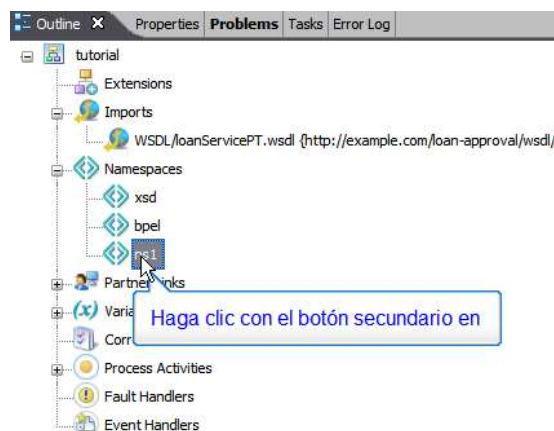


16. En la siguiente ventana en el nombre del partner link cambie el valor por defecto por assessor y haga clic en next.
17. Utilice la variable existente request y haga clic en Next.
18. En la variable de salida cambie el valor por defecto riskAssessmentMessage por risk y haga clic en Finish.
19. En el editor de procesos seleccione la actividad Invoke y cambie su nombre por InvokeLoanAssessor (De la misma manera que se hizo para las anteriores actividades)

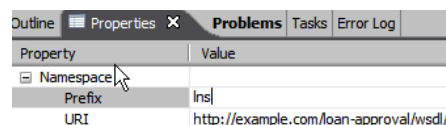
20. En la pestaña WebReferences expanda loanApprovaLLT y encuentre la operación approve y arrástrela hacia el Editor de Procesos.
21. Al igual que la anterior operación seleccione Invoke haga clic en Next.
22. En partner Link cambie el nombre por defecto loanApprovaLLT por approver. Haga clic en Next.
23. Utilice las variables existentes tanto de entrada (request) como de salida (approval) y haga clic en Finish. Cambie el nombre "Invoke" de esta actividad por InvokeLoanApprover. Su editor de procesos deberá contener ahora las 4 actividades así:



24. Cada archivo WSDL declara uno o más namespaces que proveen un context para los mensajes. Los namespaces que se declaran en el archivo WSDL también se deben declarar en el proceso. Seleccione la pestaña Outline y expanda la categoría Namespaces



25. Haga clic derecho sobre ns1 y a continuación en Properties y cambie el valor de prefix de su valor inicial ns1 a Ins.

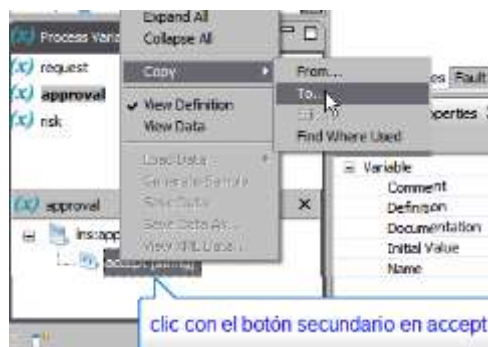


26. El proceso de negocio Loan Approval necesita una actividad más y es una actividad de tipo Assign. En el proceso Loan Approval el asesor debe asignar un valor de tipo congratulations-type al mensaje approval enviado al cliente en

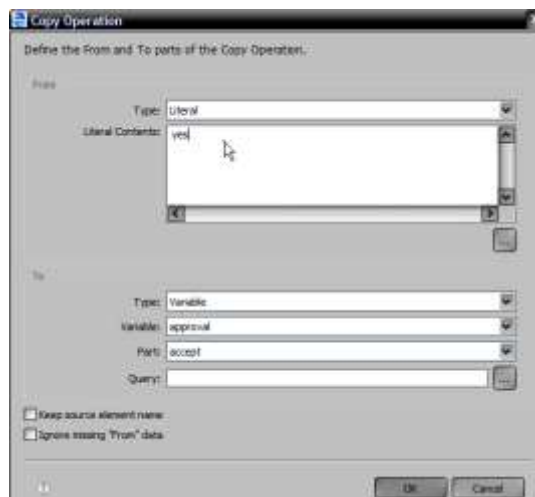
la respuesta (Actividad Reply). Para cumplir esto la actividad Assign realiza una operación de tipo copy. En la pestaña Process variables haga clic derecho sobre la variable approval y seleccione Open.



27. Expanda la variable Ins:approvalMessage y haga clic derecho sobre accept y seleccione Copy>To.

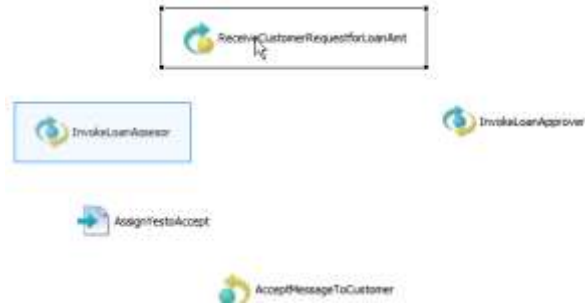


28. En la lista Type Seleccione Literal, escriba yes en Literal Contents y haga clic en OK.



29. La actividad Assign aparecerá en la esquina superior izquierda del Editor de Procesos. A continuación se debe renombrar la actividad Assign con el nombre AssignYestoAccep, tal y como se hizo con las actividades Repty, Receive e Invoke en el paso 4 literal 13.

30. Mueva la actividad AssignYestoAccep justo debajo de la actividad InvokeLoanAssessor, como en la siguiente figura.



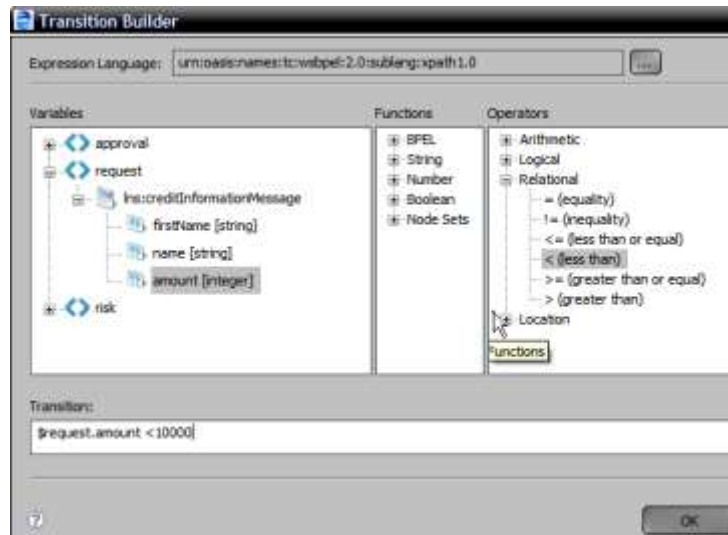
31. Seleccione las actividades ReceiveRequestforLoanAmt y la actividad InvokeLoanAssessor (con la tecla shift presionada), haga clic derecho y seleccione link Activities.



32. Seleccione el nuevo link ya creado y luego en la pestaña propiedades cambie su nombre a receive-to-assess.

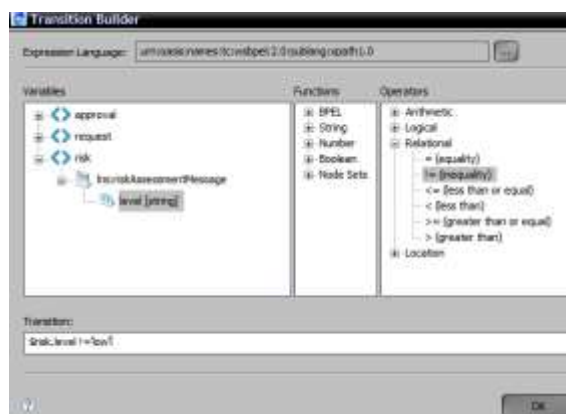
Property	Value
Link	
Comment	
Documentation	
Link Name	receive-to-assess

33. Haga doble clic sobre el enlace receive-to-asses. En la ventana emergente expanda la variable request>Ins:creditInformationMessage y haga doble clic sobre amount. Observe como en la casilla transition condition aparece \$request.amount. Luego en la categoría operators expanda Relational y seleccione < (less than) y escriba 10000. Haga clic en OK.



34. De la misma manera ahora seleccione las actividades ReceiveCustomerRequestforLoanAmt e InvokeLoanApprover y haga clic derecho y luego link activities, en propiedades en la casilla name coloque receive-to-approve y haga doble clic sobre el enlace y coloque la siguiente expresión $\$request.amount \geq 10000$, tal y como se hizo en el anterior enlace.

35. Ahora seleccione las actividades InvokeLoanAssessor e InvokeLoanApprover y haga clic derecho y luego link activities, en propiedades en la casilla name coloque assess-to-approve y haga doble clic sobre el enlace y coloque la siguiente expresión $\$risk.level \neq 'low'$, tal y como se hizo en el anterior enlace. Tenga en cuenta que la expresión $\$risk.level$ se encuentra expandiendo risk como en la siguiente figura, y la palabra low la escribe usted mismo pues corresponde al valor que le vamos a asignar.

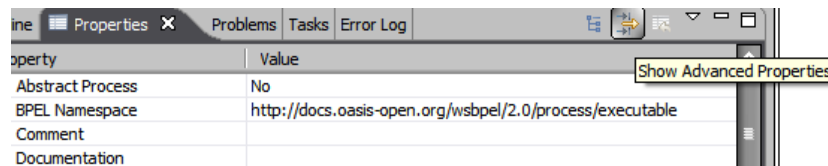


36. De la misma manera seleccione las actividades InvokeLoanAssessor AssignYestoAccept y haga clic derecho y luego link activities, en propiedades en la casilla name coloque assess-to-setMessage y haga doble clic sobre el enlace y coloque la siguiente expresión $\$risk.level = 'low'$, tal y como se hizo en el anterior enlace.

37. Finalmente cree enlaces entre la actividad AssignYestoAccpet y la actividad AcceptMessageToCustomer y un enlace entre la actividad InvokeLoanApprover y la actividad AcceptMessageToCustomer. A estos nuevos enlaces no les haga ningún cambio. Su proceso debe quedar así:



38. Haga clic en una parte en blanco del editor de procesos y luego vaya a la pestaña properties. Haga clic en el botón de propiedades avanzadas.



39. Asegúrese de que el ítem Abstract Process tenga el valor “No” y el ítem Supress join Failure tenga el valor “yes”. Guarde el proyecto, observe la pestaña Problems, no debe encontrar ningún error.

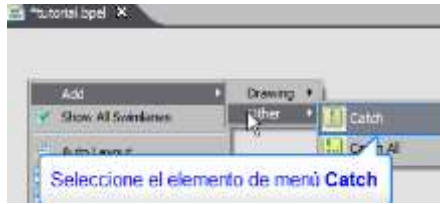
Parte 5: Adicionando Manipulación de Fallas (Fault Handling)

En BPEL la manipulación de fallas es un proceso inverso que deshace el trabajo parcial no exitoso, es decir cuando han ocurrido fallas. Cuando una falla ocurre, el procedimiento normal se termina y el control se transfiere al correspondiente manipulador de fallas.

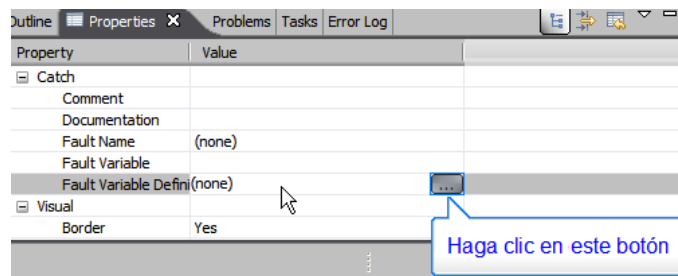
40. Haga clic sobre la pestaña Fault Handlers



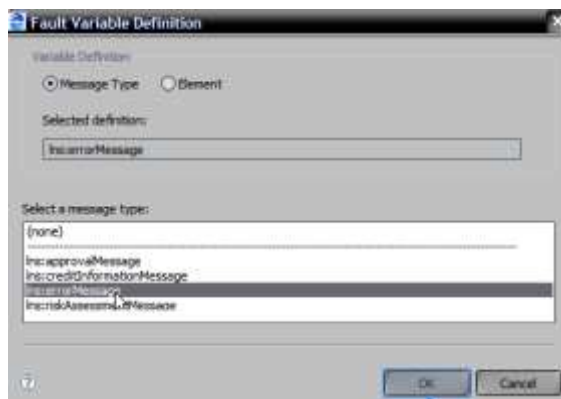
41. En el Editor de Procesos haga clic derecho y seleccione Add>Other>Catch.



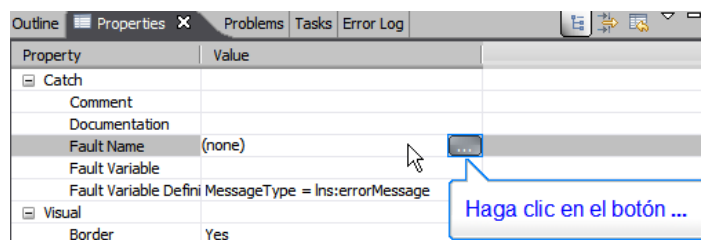
42. Observe que se agregó un nuevo manipulador de fallas, selecciónelo y a continuación haga clic en la pestaña properties. Seleccione Fault Variable Definition y haga clic en el botón al final de la casilla value.



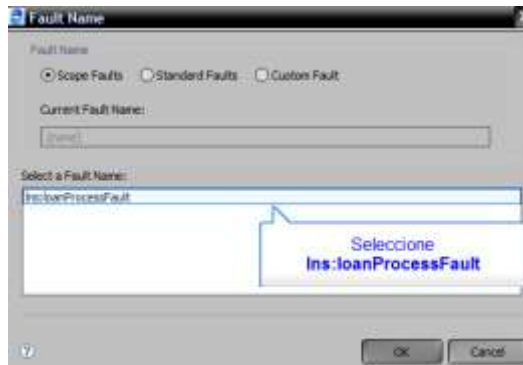
43. Seleccione Ins:errorMessage y haga clic en OK.



44. Vuelva a la pestaña propiedades y cambie el valor de Fault Name al igual que el paso anterior.



45. Seleccione Ins:loanProcessFault y haga clic en OK.

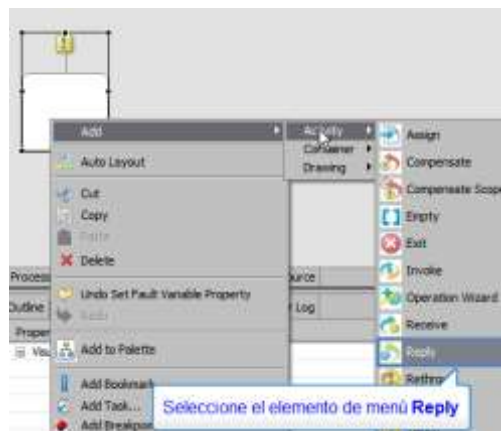


46. Vuelva a la pestaña propiedades y en la casilla Fault Variable escriba "error".

Property	Value
[-] Catch	
Comment	
Documentation	
Fault Name	Ins:loanProcessFault
Fault Variable	error
Fault Variable Defini	MessageType = Ins:errorMessage

47. Observe que en las variables del proceso aparece una nueva variable con el nombre error.

48. Haga clic derecho sobre el Fault Handler y a continuación seleccione Add>Activity>Reply.



49. Aparece una actividad Reply dentro del Fault Handler, selecciónela y haga clic sobre la pestaña properties. A continuación, y en el siguiente orden

- a. seleccione la casilla Partner link Type y cambie su valor a customer.

Property	Value
Fault Name	(none)
Message Exchange	
Operation	(none)
Partner Link	(none)
Port Type	(none)
To Parts	approver assessor
Variable	customer

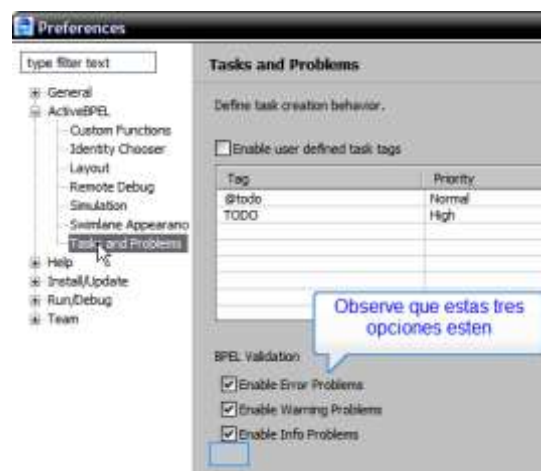
- b. De la misma forma, en la casilla operación seleccione request .
- c. En la casilla Fault name seleccione unableToHandleRequest.
- d. En la casilla Variable seleccione error. De esta manera las propiedades deben quedar así:

Property	Value
Fault Name	unableToHandleRequest
Message Exchange	
Operation	request
Partner Link	customer
Port Type	(none)
To Parts	(none)
Variable	error

50. Guarde el proyecto.

Parte 6: Simular el Proceso.

1. Haga clic en el menú Window>Preferences. Expanda la categoría Active BPEL, seleccione Task and Problems y observe que las opciones Enable Error Problems, Enable Warning Problems y Enable Info Problems estén seleccionadas. Haga clic en OK.



2. En la pestaña problems no debe haber ningún problema.
3. Sobre el editor de procesos haga clic en la pestaña Process Activities.
4. Haga clic en la pestaña Web References, luego en el Menú superior izquierdo de la misma pestaña (Flecha abajo) y seleccione View Messages.



5. En la pestaña Web References aparecen los mensajes. Expanda creditInformationMessage, seleccione la parte amount y haga clic en Add Sample.

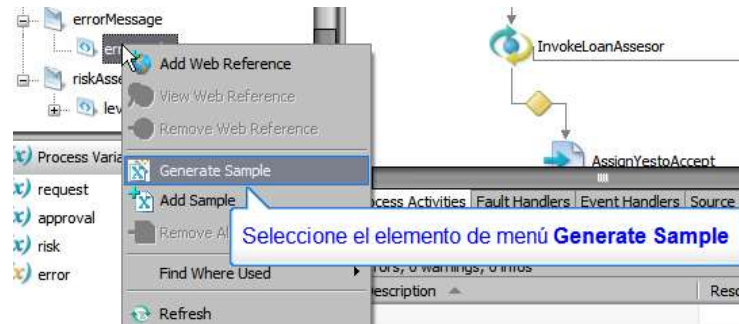


6. En el cuadro de diálogo Add Sample Data escriba 9000.

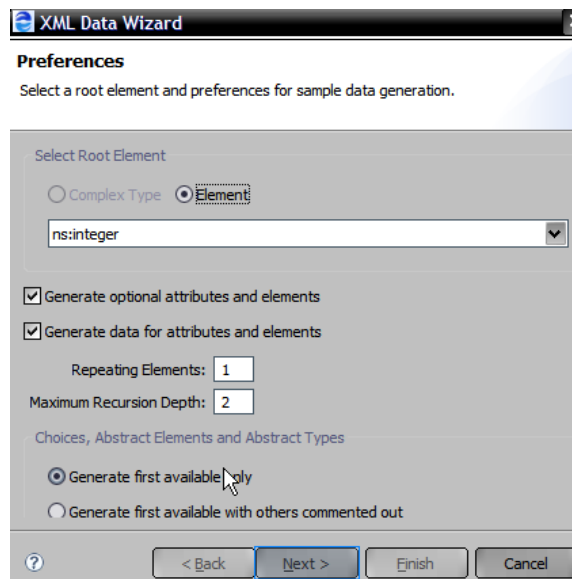


7. Agregue otro valor de ejemplo repitiendo los pasos 4 a 7 para la parte amount pero esta vez escriba 11000.
8. De la misma manera agregue valores de ejemplo para las partes name y firstname, pueden ser sus nombres y apellidos.
9. Como en el caso del mensaje creditInformationMessage, expanda el mensaje riskAssessmentMessage y agregue un valor de ejemplo para la parte level y escriba low.
10. De la misma forma expanda approvalMessage y seleccione la parte accept, adicione un valor de ejemplo y escriba yes.
11. Para el caso del mensaje errorMessage está definida en un tipo de esquema complejo dentro del WSDL loanServicePT.wsdl y por lo tanto requiere datos de ejemplo que contengan las referencias apropiadas para la definición de tipos

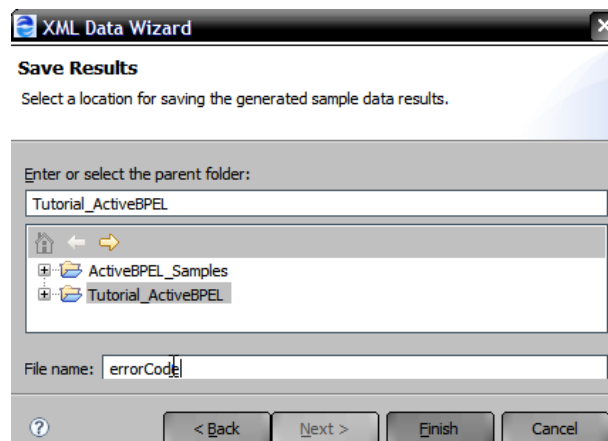
de datos. Para esto, entonces, expanda el mensaje errorMessage, haga clic derecho sobre la parte del mensaje errorCode y seleccione Generate Sample.



12. En el XML Data Wizard seleccione Next.



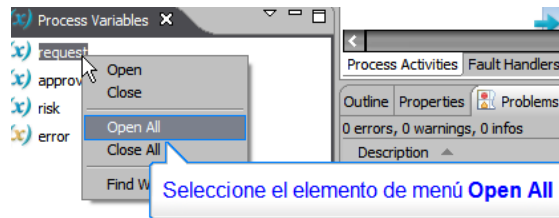
13. A continuación escriba errorCode en el campo File Name y haga clic en Finish.



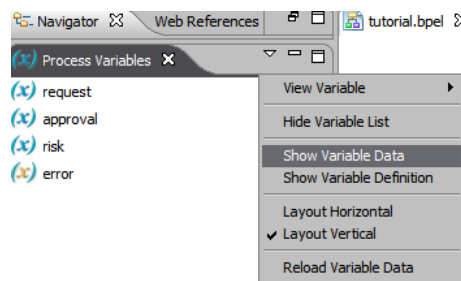
14. Haga clic en el botón refresh content de la pestaña Web References.



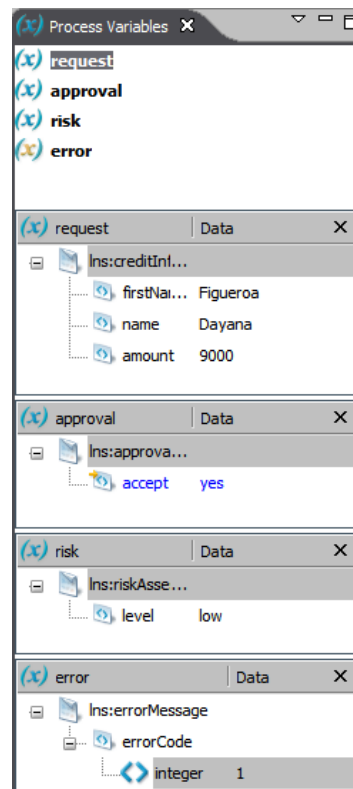
15. A continuación haga clic derecho sobre cualquiera de las variables de la pestaña Process Variables y seleccione Open All.



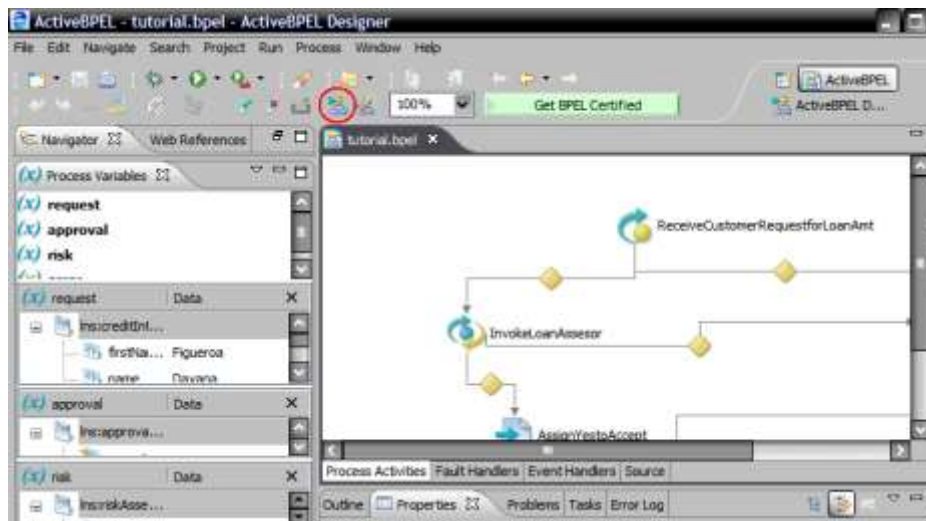
16. Haga clic en el menú de la pestaña Process Variable (Flecha en la esquina superior derecha) y seleccione la opción show Variable Data.



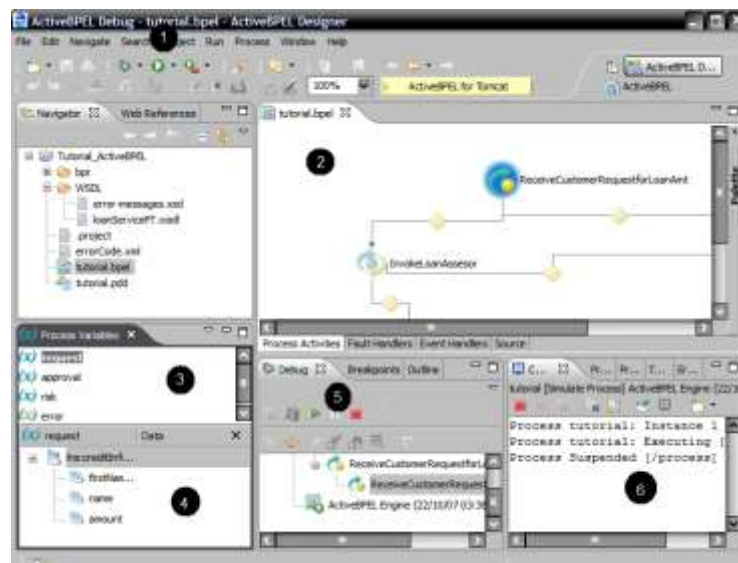
17. Las variables del proceso deben quedar así:



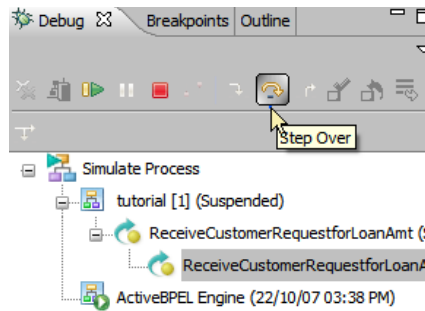
18. Guarde el proyecto, haga clic sobre una parte en blanco del editor de procesos, para activar la barra de herramientas principal, y a continuación haga clic en el icono Start Simulation.



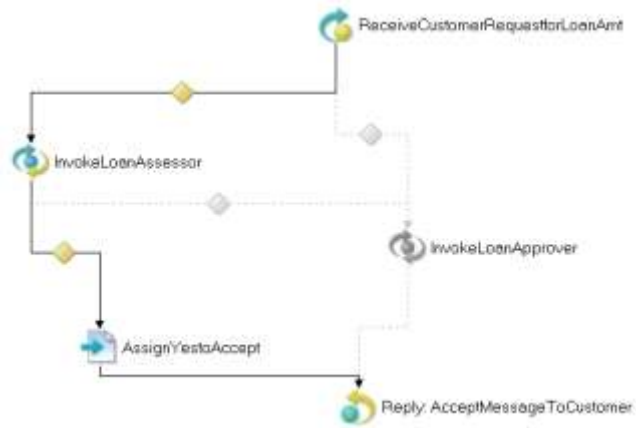
19. Note los cambios en el entorno de ActiveBPEL:



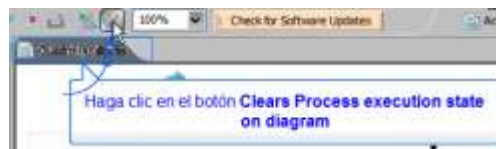
1. La perspectiva cambia al modo Debug.
 2. Se resalta la actividad inicial.
 3. Se resalta la variable actual
 4. Los datos de ejemplo de las variables se ponen en blanco.
 5. La vista Debug se activa para poder ir avanzando en el proyecto.
 6. Se abre la consola para reportar la simulación de eventos.
20. En la vista debug haga clic en el botón Step Over para continuar con la simulación:



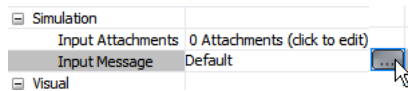
21. Así mismo siga a través de la simulación del proceso inspeccionando las variables del proceso cada vez que una actividad se ejecute. El simulador tomará el camino del loan assessor debido a que la cantidad requerida es menor que \$10000. El editor de procesos tiene que mostrar lo siguiente.



22. Una vez finalizada la ejecución del camino del loan assessor limpiemos el estado de la ejecución del proceso, para esto hacer clic en una parte en blanco del editor de procesos y en la barra de herramientas principal haga clic en el ícono Clear Process Execution State, que está al lado del ícono Start Simulation.



23. A continuación vamos a probar otro camino de ejecución el del approver. Para esto Seleccione la variable ReceiveCustomRequestForLoanAmt, en la pestaña properties en el campo Simulation seleccione Input Message y haga clic en el botón ubicado al final de este.

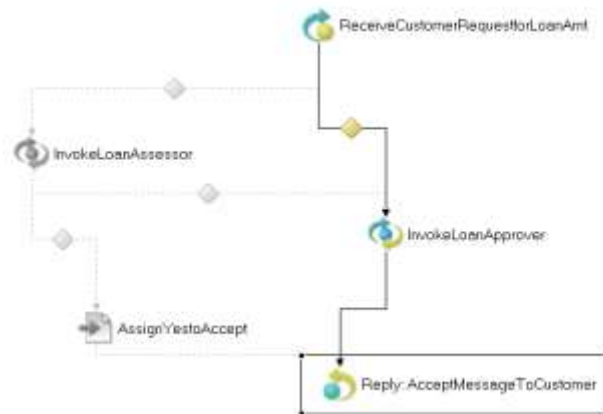


24. En el cuadro Set Simulation Data cambie el valor seleccione amount y haga clic en el botón ubicado a su derecha.



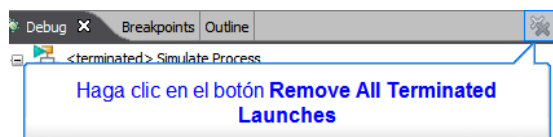
25. Cambie el valor de 9000 a 11000 y haga clic en OK.

26. Ejecute el proceso nuevamente como se explicó en los pasos 18 a 21. El editor de procesos debe quedar así:



27. Limpie el estado de la ejecución como en el paso 22.

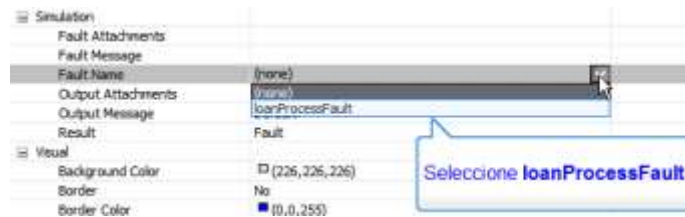
28. Seleccione la pestaña Debug y haga clic en el botón Remove All Terminate Launches.



29. A continuación vamos a probar una ejecución con error. Para esto Seleccione la variable ReceiveCustomRequestForLoanAmt, en la pestaña properties en el campo Simulation seleccione Result y cambie su valor de Normal a Fault.



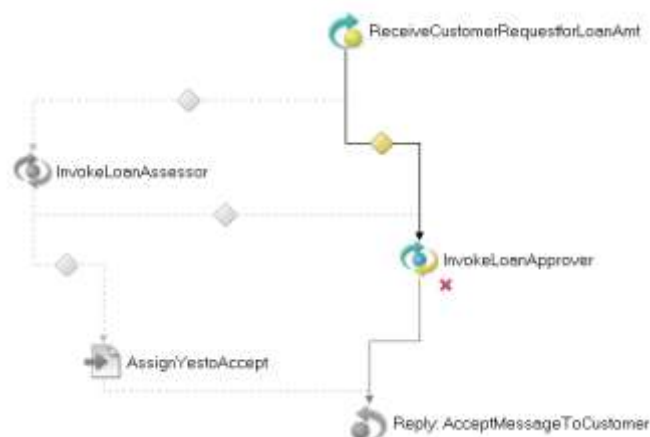
30. Seleccione ahora Fault Name bajo el campo Simulation y seleccione loanProcessFault.



31. Ejecute la simulación como se explica en el paso 18 y observe que esta vez el proceso lanza un FaultHandler.



32. Ahora el editor del proceso mostrará lo siguiente:

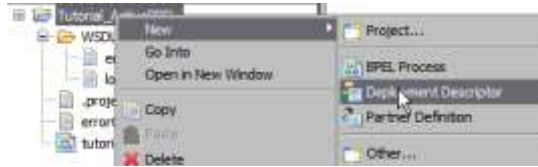


33. Haga clic en la consola y observe los eventos de ejecución.

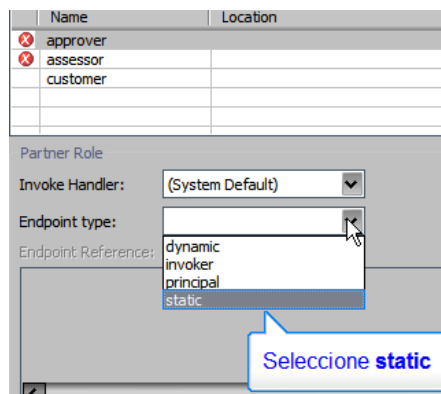
34. Limpie el estado del proceso.

Parte 7: Desplegando el Proceso

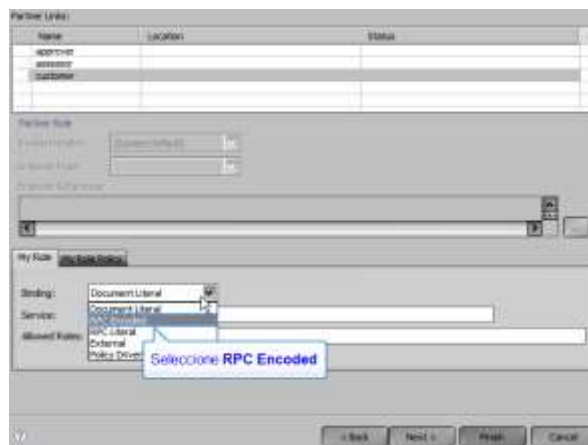
1. Haga clic derecho sobre el árbol del proyecto seleccione New>Deployment Descriptor.



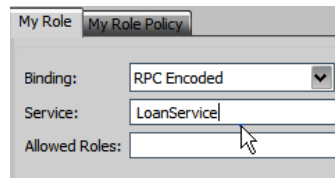
2. Expanda la carpeta Tutorial_ActiveBPEL, seleccione el archivo tutorial.bpel y haga clic en Next y otra vez en Next.
3. En la nueva ventana seleccione approver y cambie su valor Endpoint Type a static.



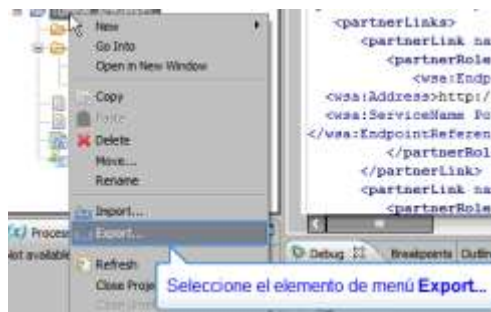
4. Luego seleccione assessor y haga lo mismo que el paso anterior.
5. Seleccione customer y en la casilla Binding seleccione RPC Encoded.



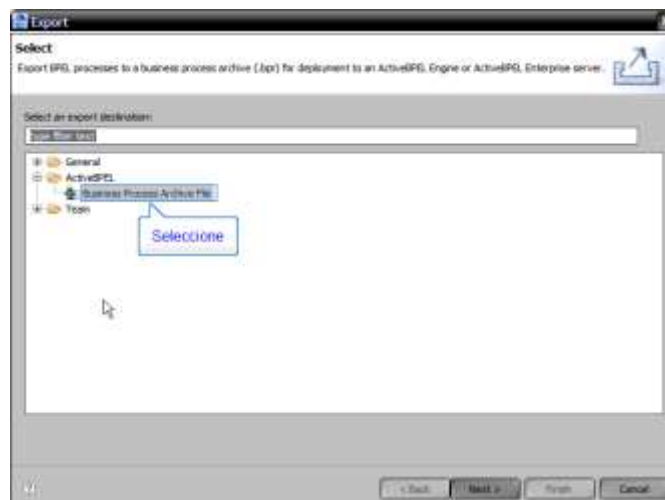
6. Finalmente en la casilla Service escriba LoanService y haga clic en Finish.



7. Haga clic derecho sobre el árbol del proyecto y seleccione New>Other>Folder y haga clic en next.
8. En el campo folder name escriba "bpr" y haga clic en finish.
9. Haga clic derecho sobre el árbol del proyecto y seleccione Export.



10. Expanda la carpeta ActiveBPEL y seleccione Business Process Archive File y haga clic en Next.



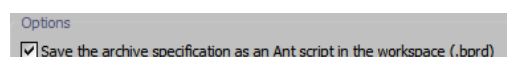
11. Aparecerá entonces el siguiente cuadro de diálogo.



12. En el campo BPR file la ubicación de la carpeta creada en el literal 6 de la parte 7 y escriba como nombre de archivo tutorial.bpr. Haga clic en guardar.



13. En el campo Deployment location seleccione la carpeta *ActiveBPEL_installation* \Server\ActiveBPEL_Tomcat\bpr\ donde ActiveBPEL_installation es el directorio de instalación de Active BPEL, generalmente dentro de Archivos de Programa. Hacer esto permite que ActiveBPEL designer despliegue automáticamente el archivo *tutorial.bpr* en el servidor de aplicaciones.
14. Finalmente seleccione “Save the archive specification as an Ant script in the workspace”.



15. En el campo BPRD File seleccione la carpeta bpr y en File name escriba tutorial.bprd. Este archivo es un script ant el cual se puede ejecutar para

redespregar el archivo bpr cuando se modifique el archivo .bpel o el .pdd. Haga clic en OK, entonces el formulario debe quedar como sigue:



16. Haga clic en Finish.

Parte 8: Ejecutar el Proceso en el Servidor de Aplicaciones.

1. Para este punto es necesario copiar los siguientes archivos a los directorios del servidor de aplicaciones, supongamos que el directorio de instalación de Active BPEL sea `ActiveBPEL_installation` entonces tenemos:

- a. Revise que el archivo `tutorial.bprd` generado en el paso anterior ya se encuentre dentro del directorio:

`ActiveBPEL_installation\Server\ActiveBPEL_Tomcat\bpr\`

- b. Copie el archivo `ActiveBPEL_Tutorial_test_page.war` que se encuentra en:

`ActiveBPEL_installation\Samples\ActiveBPEL_Tutorial_Complete_Files\
ActiveBPEL_Tutorial_test_page.war`

Al directorio `ActiveBPEL_installation\Server\ActiveBPEL_Tomcat\webapps\`

Este archivo sirve como página de prueba para enviar y recibir mensajes del proceso.

- c. Copie el archivo `loan_approval_config.xml` que se encuentra en:

ActiveBPEL_installation \Samples\ActiveBPEL_Tutorial_Complete_Files\
loan_approval_config.xml

Al directorio **ActiveBPEL_installation** \Server\ActiveBPEL_Tomcat\temp\

Este es un archivo de datos utilizados por la página de pruebas.

- d. Copie el archivo *ActiveBPEL_tutorial_web_service.wsr* que se encuentra en:

ActiveBPEL_installation \Samples\ActiveBPEL_Tutorial_Complete_Files\
ActiveBPEL_tutorial_web_services.wsr

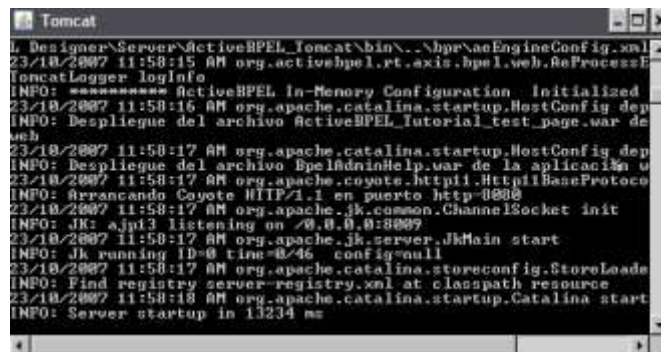
Al directorio **ActiveBPEL_installation** \Server\ActiveBPEL_Tomcat\bpr\

Este es un archivo utilizado también por la página de pruebas.

2. Su directorio de instalación de active BPEL tendrá ahora los archivos necesarios.
3. Para iniciar el servidor de aplicaciones haga doble clic sobre el archivo *startup.bat* que se encuentra en el directorio:

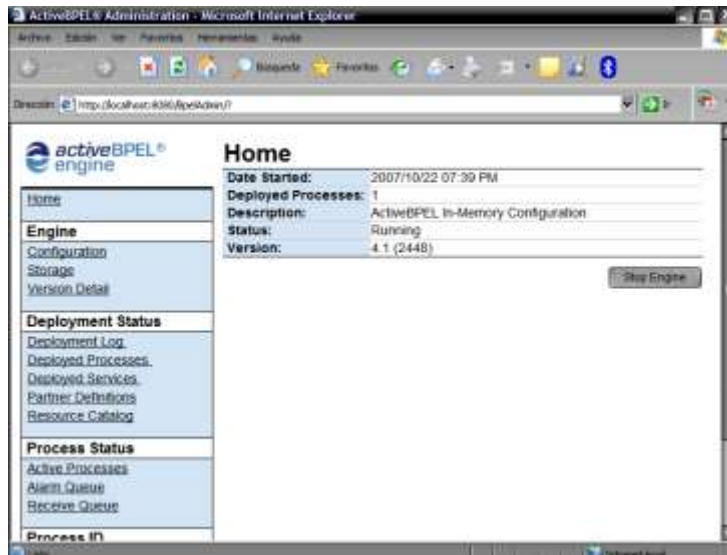
ActiveBPEL_installation \Server\ActiveBPEL_Tomcat\bpr\startup.bat

Si todo está bien observará esta pantalla:



```
Tomcat
I:\Designer\Server\ActiveBPEL_Tomcat\bin\..\bpr\anEngineConfig.xml
23/10/2007 11:58:15 AM org.activebpel.rt.axis.bpel.web.BpProcessE
TomcatLogger logInfo
INFO: ***** ActiveBPEL in-Memory Configuration Initialized
23/10/2007 11:58:16 AM org.apache.catalina.startup.HostConfig dep
INFO: Despliegue del archivo ActiveBPEL_tutorial_test_page.var de
web
23/10/2007 11:58:17 AM org.apache.catalina.startup.HostConfig dep
INFO: Despliegue del archivo BpelAdminHelp.var de la aplicacion w
23/10/2007 11:58:17 AM org.apache.coyote.http11.Http11BaseProce
INFO: Arrancando Coyote HTTP/1.1 en puerto http-8080
23/10/2007 11:58:17 AM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
23/10/2007 11:58:17 AM org.apache.jk.server.JkMain start
INFO: Jk running time=0/46 config=all
23/10/2007 11:58:17 AM org.apache.catalina.storeconfig.StoreLoade
INFO: Find registry server-registry.xml at classpath resource
23/10/2007 11:58:18 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 13234 ms
```

4. Si desea observar la consola administrativa de Active BPEL abra un programa navegador y escriba las siguiente URL. <http://localhost:8080/BpelAdmin>



5. Finalmente para mirar la página de pruebas del servicio loan approval. Ingrese en el navegador la siguiente URL. http://localhost:8080/ActiveBPEL_Tutorial_test_page

6. Deje los valores por defecto y observe que la respuesta entregada por el proceso de negocio luego de evaluar la cantidad de dinero y las respuestas dadas por el Asesor y el Aprobador es “yes”, usted puede seguir probando cambiando los valores y observar las diferentes respuestas entregadas por el proceso de negocio loan approval.

 **ActiveBPEL® Tutorial Test Client**

Congratulations; The ActiveBPEL Tutorial process has been successfully executed. The reply from the process is: "yes"

Loan Request

First Name:

Last Name:

Amount:

Operation:

URI:

Assessor

Response:

Approver

Response:

REFERENCIAS

Página web de Active Endpoints <http://www.active-endpoints.com/>

Oracle BPEL Designer

Introducción

La interoperabilidad entre los diferentes sistemas es algo indispensable hoy en día para el mundo empresarial. Lo que esta práctica propone es la interacción entre dos sistemas (Servicios Web) que unidos a través de un lenguaje (BPEL) prestan una funcionalidad requerida a un usuario final. De esta manera las empresas pueden modelar sus procesos de negocios de manera mas ordenada explotando funcionalidades prestadas por otras empresas aumentando así su productividad y a la vez su competitividad.

Objetivos

1. Proponer las bases para la interacción entre sistemas heterogéneos.
2. Presentar al lenguaje BPEL como la principal herramienta unificadora de tecnologías.
3. Sacar el máximo provecho de las funcionalidades provistas por los Servicios Web.

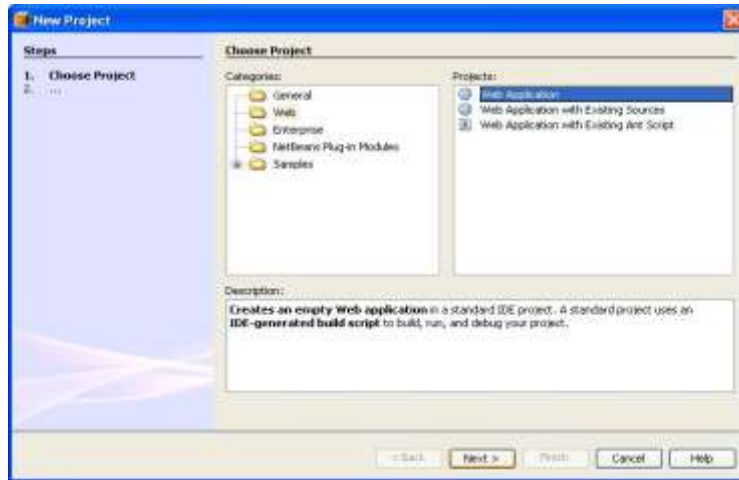
Software Requerido

1. Oracle BPEL Process Manager. (Servidor Collaxa Oracle)
2. NetBeans 5.5. (Sun Java System Application Server)

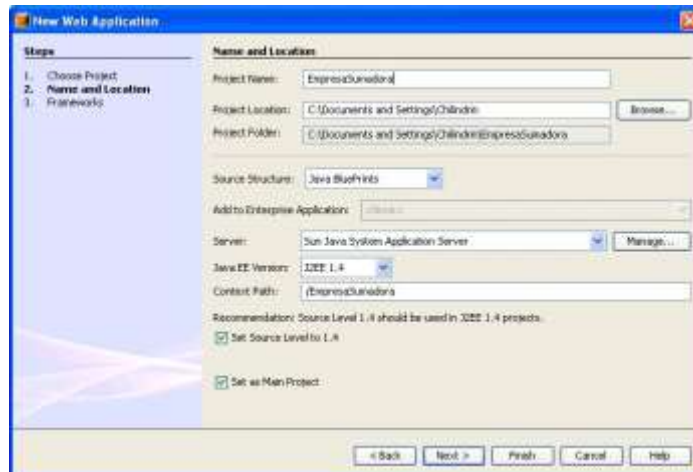
Desarrollo de la Práctica

En la práctica vamos sacar el promedio de dos números ingresados por el usuario. Para ello, mediante BPEL, vamos a utilizar dos servicios web. El primero va a sumar los números ingresados por el usuario. El segundo va a dividirlos sobre dos. En este caso “sacar el promedio de dos números” es visto como un proceso de negocio.

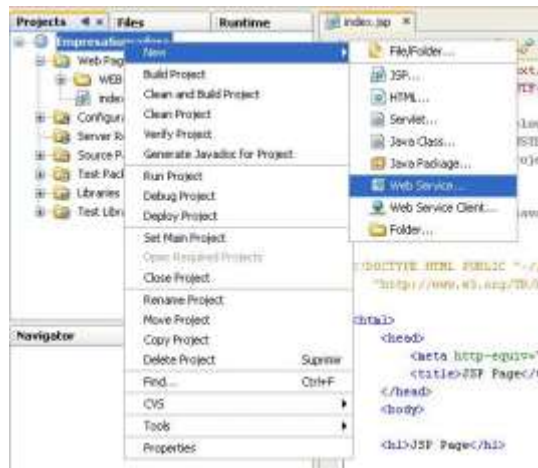
1. **Elaboración de los Servicios Web a utilizar.**
 - a. Servicio Web sumador de dos enteros.



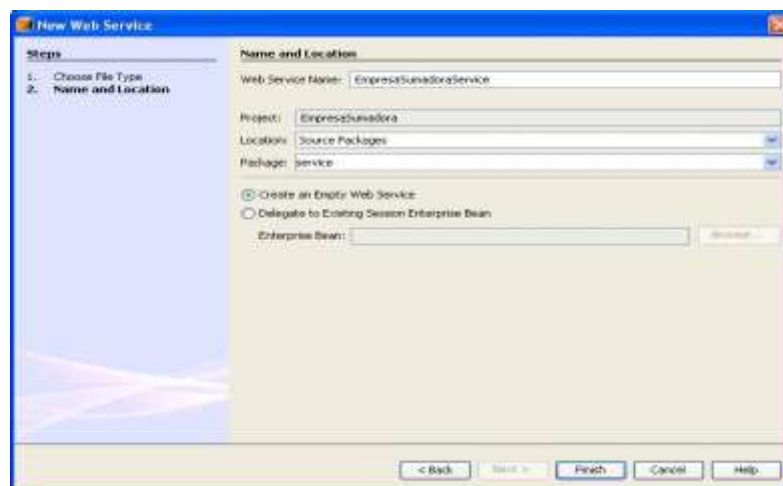
Vamos a crear inicialmente una Aplicación Web a la cual se le va a agregar posteriormente el Servicio Web. Esta aplicación web va a recibir el nombre de “Empresa Sumadora” y ajustamos el nivel de “Java EE Version” a 1.4.



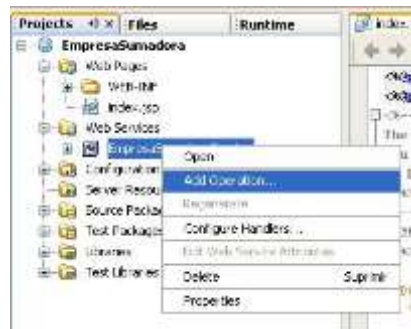
Una vez creada la Aplicación Web vamos a agregarle a esta el Servicio Web el cual recibirá el nombre de “EmpresaSumadoraService”. Esto se hace como lo muestra la siguiente figura.



A continuación NetBeans pregunta por el nombre que queremos darle al Servicio Web, al igual que por el nombre del paquete en donde queremos colocar su Interface y su Implementación. Esto se hace como sigue:



Como se puede apreciar, a nuestro Servicio Web le hemos puesto el nombre de "EmpresaSumadoraService" en el campo "Web Service Name" y en el campo "Package" se ha sugerido el nombre de "service". En este paquete encontraremos entonces la interfaz del servicio web más su implementación. Vamos ahora a agregarle la funcionalidad al Servicio Web que acabamos de crear. Esto se hace como lo muestra la figura.



Esta operación en NetBeans lanza una ventana en la cual se preguntará el nombre que recibirá la operación, los parámetros que recibe dicho método y los parámetros que devuelve.

La implementación de nuestro Web Service debe lucir como sigue:

```

package service;

/**
 * This is the implementation bean class for the EmpresaSumadoraService web service.
 * Created 20/10/2007 01:09:42 PM
 * @author Chilindrin
 */
public class EmpresaSumadoraServiceImpl implements EmpresaSumadoraServiceSEI {

    // Enter web service operations here. (Popup menu: Web Service->Add Operation)
    /**
     * Web service operation
     */
    public int sumarNumerosUsuario(int inta, int intb) throws java.rmi.RemoteException {
        // TODO implement operation
        return inta+intb;
    }
}

```

Y su interfaz:

```

package service;

import java.rmi.Remote;

/**
 * This is the service endpoint interface for the EmpresaSumadoraService web service.
 * Created 20/10/2007 01:09:42 PM
 * @author Chilindrin
 */
public interface EmpresaSumadoraServiceSEI extends Remote {

    /**
     * Web service operation
     */
    public int sumarNumerosUsuario(int inta, int intb) throws java.rmi.RemoteException;
}

```

De esta manera damos por concluida la construcción de nuestro Servicio Web llamado "EmpresaSumadoraService". De la misma manera se hace la

implementación de nuestro segundo Web Service, el cual recibirá el nombre de “EmpresaPromedioService”. Para ello, la implementación cambia ya que la funcionalidad que este servicio prestaría sería la de dividir el dato entrante por dos. Según esto, la implementación sería como sigue.

```
package service;

/**
 * This is the implementation bean class for the EmpresaPromedioService web service.
 * Created 20/10/2007 07:04:56 PM
 * @author Chilindrin
 */
public class EmpresaPromedioServiceImpl implements EmpresaPromedioServiceSEI {

    // Enter web service operations here. (Popup menu: Web Service->Add Operation)
    /**
     * Web service operation
     */
    public int promediarNumerosUsuario(int inta) throws java.rmi.RemoteException {
        // TODO implement operation
        return inta/2;
    }
}
```

Lo que sigue es desplegar nuestros Servicios Web en el Sun Java System Application Server 9. Estos, una vez desplegados, exponen las Ubicaciones URL de sus documentos WSDL, las cuales serán utilizadas posteriormente por el Oracle BPEL Process Manager.

Hecho esto, las URL de los servicios web serían:

<http://localhost:8080/EmpresaPromedio/EmpresaPromedioService?WSDL>
<http://localhost:8080/EmpresaSumadora/EmpresaSumadoraService?WSDL>

Una vez desplegados los servicios vamos a utilizarlos haciendo uso de nuestro lenguaje BPEL. Antes de ello debemos construir el esquema necesario para el proyecto BPEL. Dicho esquema debe tener lo siguiente.

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.globalcompany.com/ns/sales"
  xmlns:po="http://www.globalcompany.com/ns/sales"
  elementFormDefault="qualified">

  <element name="DatosEntrada" type="po:NumerosEntrada"/>
  <element name="DatosSalida" type="po:ResultadoPromedio"/>

  <complexType name="NumerosEntrada">
    <sequence>
      <element name="inta" type="decimal"/>
    </sequence>
  </complexType>
</schema>
```

```

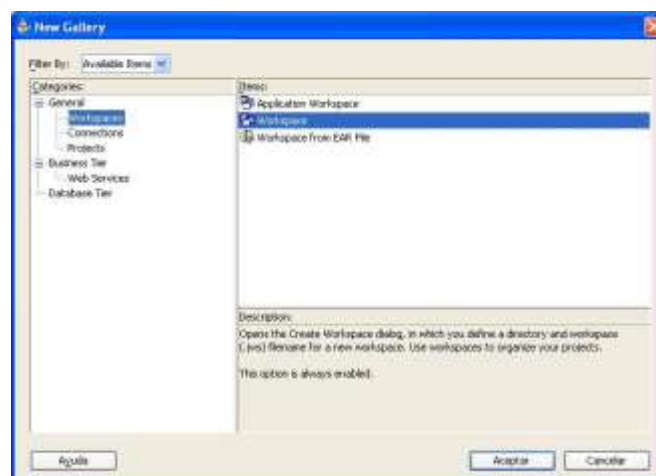
        <element name="intb" type="decimal"/>
    </sequence>
</complexType>

<complexType name="ResultadoPromedio">
    <sequence>
        <element name="resul" type="decimal"/>
    </sequence>
</complexType>

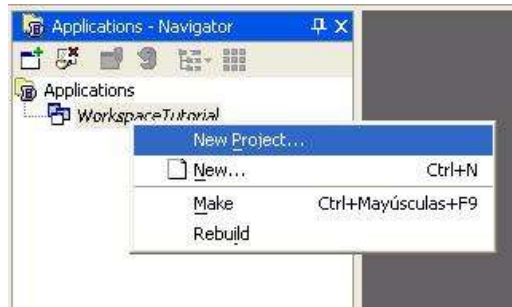
</schema>

```

Archivo que debe quedar bajo el nombre "Tutorial.xsd". Este archivo va a ser necesitado a la hora de crear nuestro proceso BPEL. Ahora, vamos a crear nuestro programa BPEL mediante el cual se van a invocar los Servicios Web anteriormente creados. Esto se hace desde el Oracle BPEL Process Manager. Lo primero que hay que hacer es crear nuestro Workspace. Esto se hace como se muestra a continuación. En el menú "File", se selecciona la opción "New" opción que despliega la ventana siguiente. De "Categories" se debe seleccionar "Workspaces" y de "Items" se debe seleccionar "Workspace". A continuación dar click en "Aceptar".

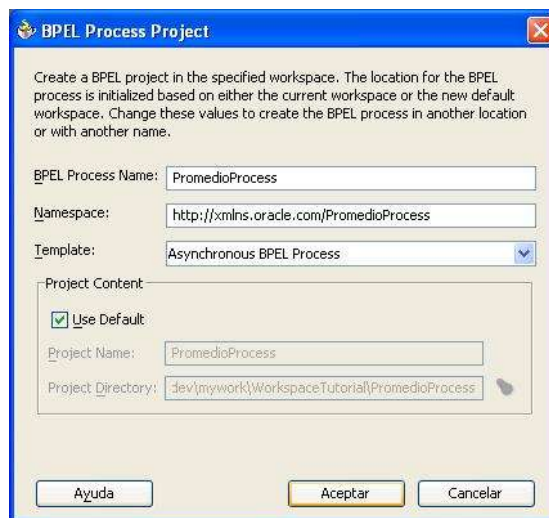


El nombre para este Workspace será "WorkspaceTutorial". Deseleccione la opción de "Add a New Empty Project". Una vez creado el Workspace, se puede trabajar dentro de el con un proyecto BPEL. Ello se hace de la siguiente manera.

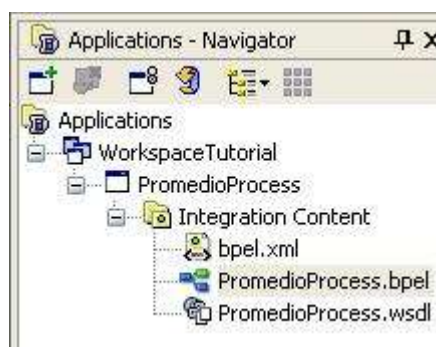


Una vez hecho esto, se debe crear el tipo de proyecto. En el recuadro de “Categories” bajo la sección “General” se debe seleccionar “Projects”. En el recuadro de “Items” se debe seleccionar “BPEL Process Project”. Una vez hecho esto, la siguiente ventana aparece y la siguiente información debe colocarse:

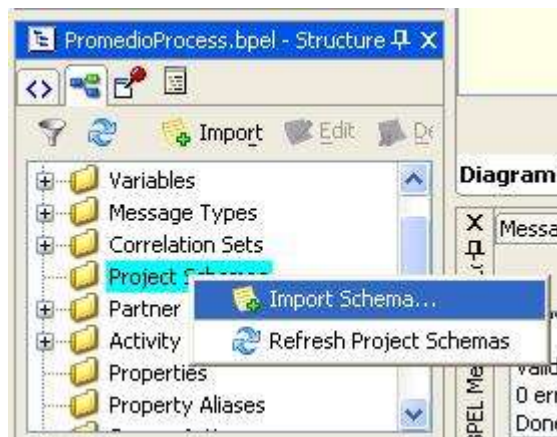
En el campo de “BPEL Process Name” colocar “PromedioProcess”. En el campo de donde se selecciona la plantilla, asegurarse de seleccionar “Asynchronous BPEL Process”. A continuación hacer click en “Aceptar”. Esto se puede ver a continuación.



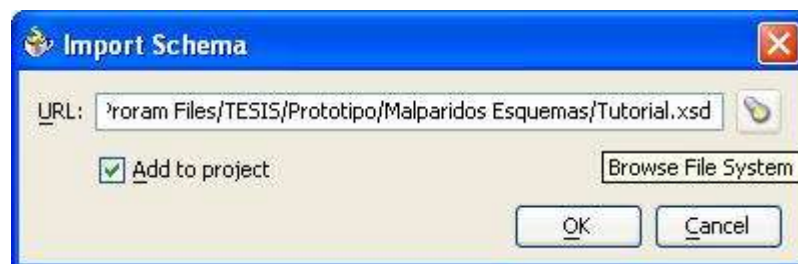
Una vez creado el proyecto BPEL, este debe tener la siguiente vista:



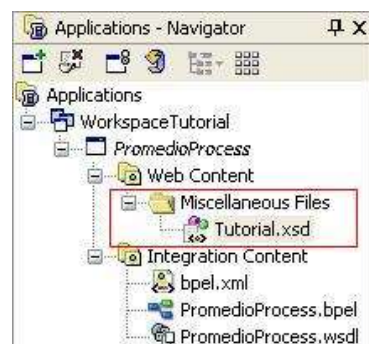
Una vez confirmada la estructura del proyecto, dar doble click sobre el archivo “PromedioProcess.bpel”. Esto desplegará sobre la parte inferior izquierda de la pantalla la estructura de este archivo. Esta estructura sirve para importarle al proyecto el esquema previamente hecho y para editar los mensajes entrantes y salientes del proceso de negocio. Para importar el esquema al proyecto hacer lo que se muestra en la siguiente figura:



Sobre la opción “Project Schema” dar click derecho y seleccionar la opción “Import Schema”.

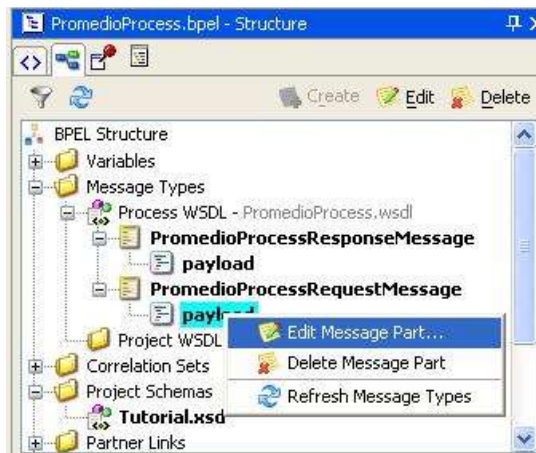


En la ventana que se muestra, buscar la ubicación del archivo “Tutorial.xsd” previamente elaborado y presionar a continuación “OK”. Para confirmar que la importación al proyecto del esquema se ha hecho exitosamente, verificar la estructura del proyecto de la siguiente manera.



Una vez importado el esquema para el proyecto, vamos a decirle al proyecto cuales van a ser sus mensajes salientes y cuales los entrantes. Esto ubicado dentro del esquema recién importado. Esto se hace de la siguiente manera.

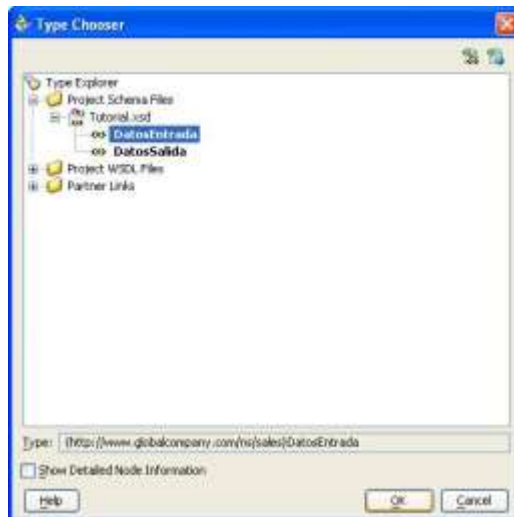
Sobre la parte inferior izquierda, se encuentra ubicada la estructura del archivo BPEL. En dicha estructura hay una parte llamada “Message Types” la cual hay que expandir. Seguidamente expandir también la opción llamada “Process WSDL”. El payload que vamos a modificar esta ubicado dentro de las opciones “PromedioProcessRequestMessage” y “PromedioProcessResponseMessage”. Dar click derecho sobre el payload de “PromedioProcessRequestMessage” como se muestra en la figura.



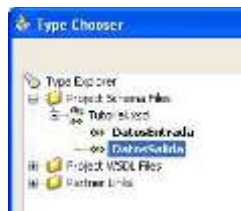
Y a continuación, dando click en la linterna,



seleccionar la entrada para el proceso como se muestra a continuación.



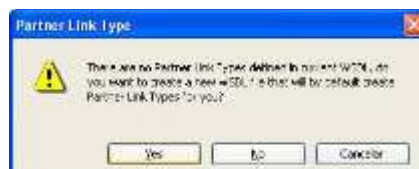
A continuación hacer lo mismo para el “PromedioProcessResponseMessage” pero esta vez seleccionar la siguiente opción.



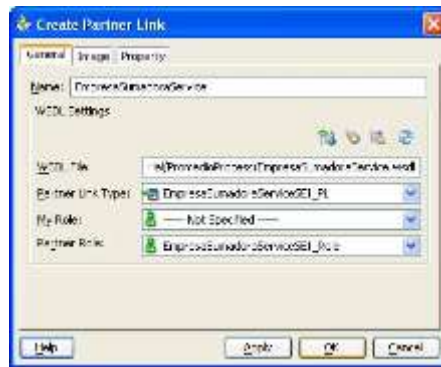
Vamos ahora a importar para nuestro proceso de negocio nuestro primer “Partner Link”. Para ello asegurarse de tener seleccionado en la parte superior derecha la paleta de “Process Activities”. Arrastrar y soltar sobre el costado derecho del proceso un “Partner Link”. Hecho esto, aparece una ventana sobre la cual debe ir la siguiente información. En el campo “Name” teclear “EmpresaSumadoraService”. En el campo “WSDL File” introducir la siguiente información:

<http://localhost:8080/EmpresaSumadora/EmpresaSumadoraService?WSDL>

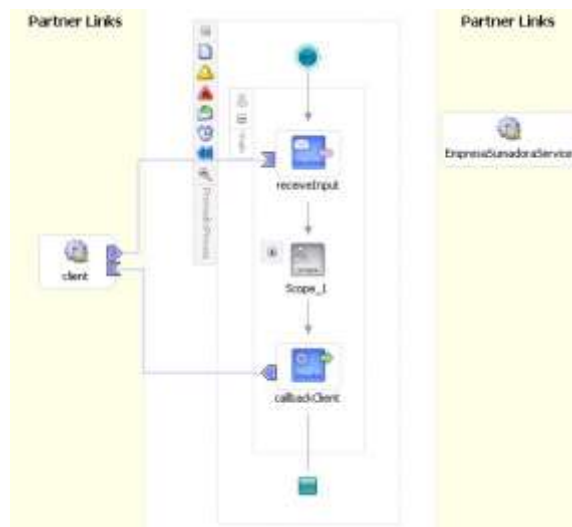
En el campo “Partner Link Type” aceptar la sugerencia que se hace para dicha característica.



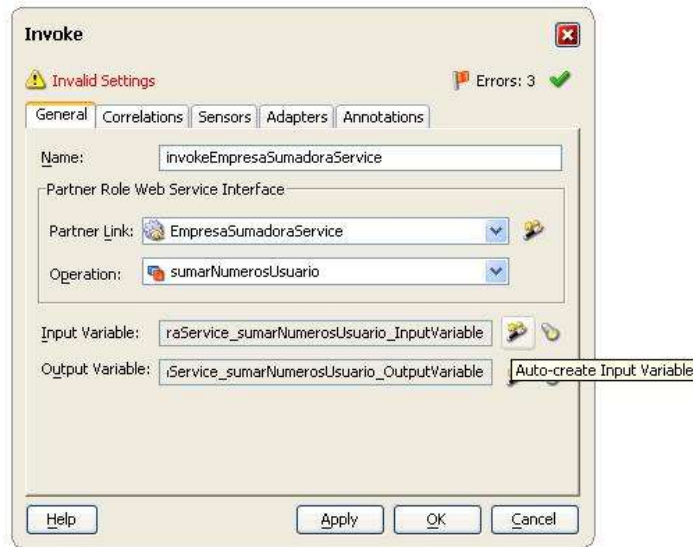
Debido a que el socio a insertar es de naturaleza síncrona, dejar vacío el campo de “My Role” y para la opción de “Partner Role” seleccionar la opción “EmpresaSumadoraServiceSEI_Role”. Todo esto resumido en la siguiente figura.



Una vez hecho esto vamos a insertar una actividad “Scope” para realizar todo lo referente a invocaciones y copiado de variables dentro del proceso. Esto se hace arrastrando y soltando desde “Process Activities” la actividad anteriormente nombrada. Hay que aclarar que esta debe ir colocada justo en medio de las actividades “receiveInput” y “callbackClient”. Esto de la siguiente manera. De color gris se puede apreciar la actividad “Scope” recién insertada al proceso. Dando doble click sobre ella puede cambiarse su nombre, para lo cual se sugiere “invocaciónProcesos”. Dando click sobre el signo más que tiene dicha actividad, esta se expande y da la posibilidad de insertar más actividades en su interior. A continuación vamos a insertar una invocación dentro de nuestra área de trabajo. En el campo de “Name” vamos a teclear lo siguiente: “invokeEmpresaSumadoraService”. En el campo de “Partner Link” vamos a seleccionar “EmpresaSumadoraService”. Una vez hecho esto, el método lo los métodos que este Servicio Web puede hacer, son expuestos automáticamente. A continuación realizamos la creación automática de las variables de entrada y salida para dicha invocación. Todo esto realizado de la siguiente manera.

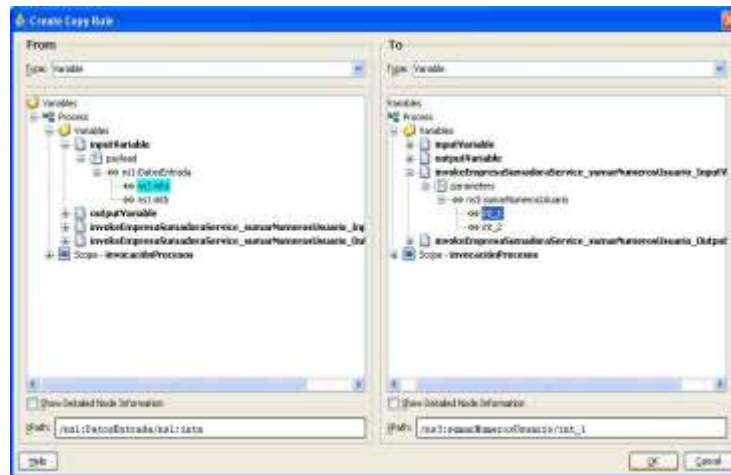


Hay que asegurarse de seleccionar la opción sugerida por la siguiente gráfica. Con esta opción estamos asegurando la auto-creación de las variables de entrada y salida para la invocación que estamos arrastrando hacia el “Scope”.

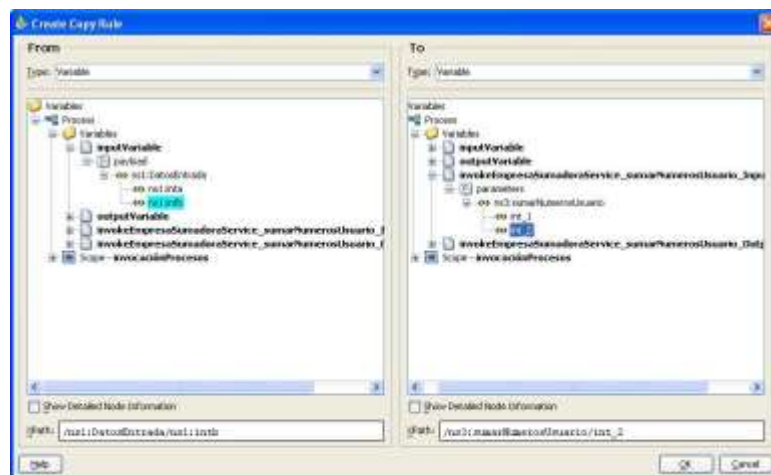


Ahora vamos a asignarle las variables de entrada al Servicio Web Sumador incluido en nuestro proceso de negocio. Esto se hace de la siguiente manera. Arrastrar de la paleta de “Process Activities” la actividad “Assign”. Hay que asegurarse de colocar esta actividad antes de la invocación previamente agregada al proceso. Una vez hecho esto vamos a colocarle a dicha actividad un nombre y se le dirá de donde a donde debe hacer el copiado de variables. En este paso vamos a copiar los valores desde las variables provenientes del esquema previamente elaborado hacia la invocación de este Servicio Web. Esto se hace de la siguiente manera.

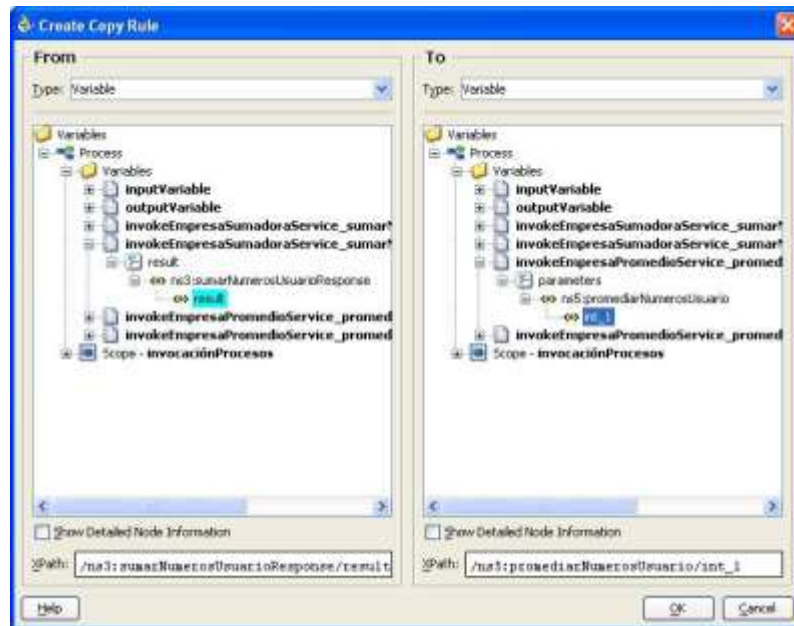
Una vez arrastrada la actividad “Assign”, dar doble click sobre ella. En la pestaña “General” introducir el nombre “assignInvokeSumadorIn”. En la pestaña “Copy Rules” dar click en el botón “Create”. Esto lanza una venta sobre la cual hay que hacer lo que se señala a continuación.



Una vez hecho esto, hay que hacerlo nuevamente, ya que a este Servicio Web tiene como parámetros de entrada dos datos provenientes del esquema. Esta vez, como sigue a continuación:

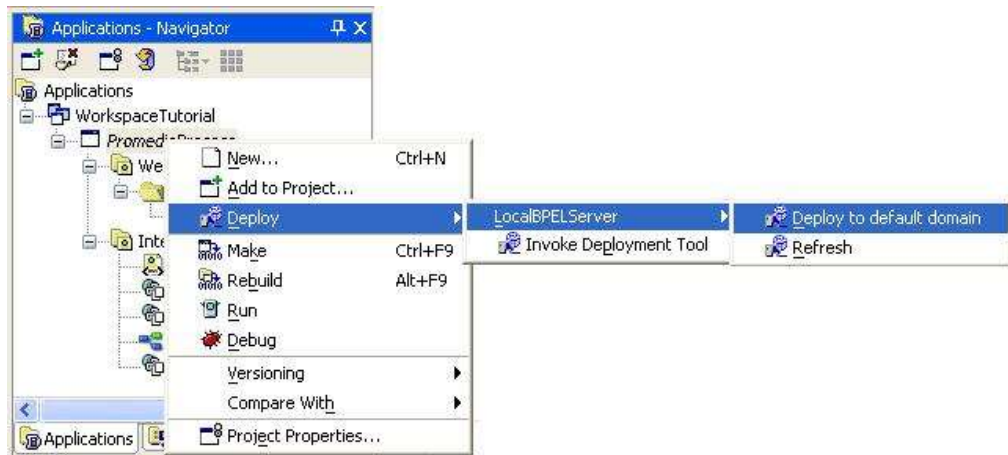


Como se puede apreciar, en el primer caso estamos copiando el contenido proveniente del esquema hacia la primera variable de entrada del Servicio Web. En el segundo caso, estamos copiando el contenido de la segunda variable proveniente del esquema como segundo parámetro del Servicio Web Sumador. A continuación vamos a importar al proceso de negocio el segundo y último Partner Link. Este será el encargado de tomar el resultado saliente del primer Servicio Web y dividirlo por dos. Para ello se hará un proceso exactamente igual al anterior, solo que ahora el copiado de las variables se hará de la siguiente manera.



El tema de las asignaciones es bastante delicado. Ellas son las que evitan que la información se pierda en el proceso de negocio. La primera asignación en nuestro proceso de negocio realizaba el copiado del contenido proveniente del esquema hacia el primer Servicio Web el cual tenía como parámetros de entrada dos variables enteras. En este caso la actividad Assign tenía creadas dos operaciones de copiado debido a las dos variables de entrada que tenía dicho Servicio Web. Ahora, la asignación tiene creada solo una operación de copiado. Esta operación se realiza sobre la variable de salida del primer Servicio Web hacia la variable de entrada del segundo Servicio Web. Recordemos que la primera empresa está prestando solo el servicio de sumar los enteros suministrados por el cliente y la segunda empresa promedia el valor entregado por su antecesora.

Una vez hecho esto, ya se puede verificar nuestro proceso de negocio mediante la consola de prueba con que cuenta el Oracle BPEL Process Manager. Primero se debe desplegar nuestro proyecto al Servidor Collaxa de Oracle. Para ello se debe hacer lo siguiente:



Seguidamente se ingresa la contraseña que por defecto es “bpel”. Cabe aclarar que previo a esto hay que arrancar el servidor. Esto se puede hacer mediante la opción de “Start BPEL PM Server” contenida dentro del Menú del Oracle BPEL Process Manager. La consola también es alcanzable dentro del mismo menú. Una vez dentro de ella, se puede iniciar el proceso ingresando los dos enteros a los cuales se les quiere calcular su promedio. Después de hacer esto, como pista de auditoría, se puede comprobar el resultado exitoso del proceso de negocio, lo cual se puede apreciar así:

Resultado de la Suma:

```

invokeEmpresaSumadoraService - pending
[2007/10/20 22:39:19] Operación bidireccional "sumarNumerosUsuario" llamada en el partner "EmpresaSumadoraService"
<messages>
  <invokeEmpresaSumadoraService_sumarNumerosUsuario_InputVariable>
    <part name="parameters" >
      <sumarNumerosUsuario>
        <int_1>2</int_1>
        <int_2>2</int_2>
      </sumarNumerosUsuario>
    </part>
  </invokeEmpresaSumadoraService_sumarNumerosUsuario_InputVariable>
  <invokeEmpresaSumadoraService_sumarNumerosUsuario_OutputVariable>
    <part name="result" >
      <ns1:sumarNumerosUsuarioResponse>
        <result>4</result>
      </ns1:sumarNumerosUsuarioResponse>
    </part>
  </invokeEmpresaSumadoraService_sumarNumerosUsuario_OutputVariable>
</messages>
  
```

Resultado del Promedio:

invokeEmpresaPromedioService - pending

[2007/10/20 22:39:19] Operación bidireccional "promediarNumerosUsuario" llamada en el partner "EmpresaPromedioService".
[menos](#)

```
<messages>
  <invokeEmpresaPromedioService_promediarNumerosUsuario_InputVariable>
    <part name="parameters" >
      <promediarNumerosUsuario>
        <int_1>4</int_1>
      </promediarNumerosUsuario>
    </part>
  </invokeEmpresaPromedioService_promediarNumerosUsuario_InputVariable>
  <invokeEmpresaPromedioService_promediarNumerosUsuario_OutputVariable>
    <part name="result" >
      <ns1:promediarNumerosUsuarioResponse>
        <result>2</result>
      </ns1:promediarNumerosUsuarioResponse>
    </part>
  </invokeEmpresaPromedioService_promediarNumerosUsuario_OutputVariable>
</messages>
```