

Servicios Web (III)

Norberto Fernández
Departamento de Ingeniería Telemática
<http://www.it.uc3m.es/berto/>



Tecnologías de Distribución de Contenidos
UC3M



1

Procesos de Negocio

- Organizaciones constan de departamentos especializados en cumplir ciertas funciones
 - Contabilidad, RR.HH., *Marketing*, etc.

- Actividad de organización es el resultado de actividad coordinada de sus unidades organizativas
 - Parte de ellas pueden estar externalizadas
 - Ej.: Logística

Tecnologías de Distribución de
Contenidos - UC3M

2

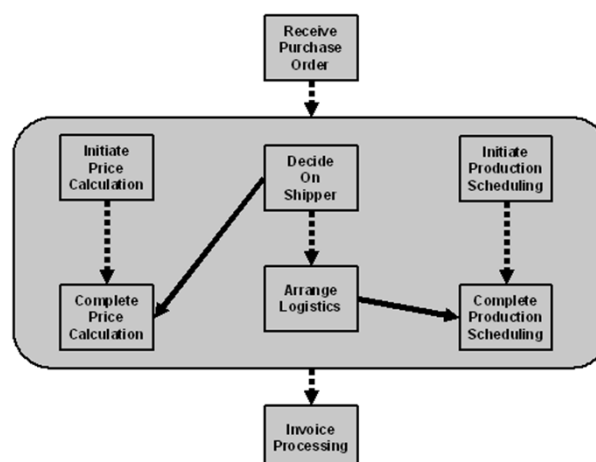
Procesos de Negocio

- Procesos de negocio:
 - Conjunto de actividades lógicamente relacionadas que llevadas a cabo permiten alcanzar un determinado resultado
 - Afectan a distintas unidades organizativas
 - ... y potencialmente a distintas organizaciones
 - Necesario coordinar los sistemas de información de las distintas unidades/organizaciones
 - “Conectar” sus servicios web

Tecnologías de Distribución de
Contenidos - UC3M

3

Procesos de Negocio



Fuente: WS-BPEL 2.0 OASIS Standard

Tecnologías de Distribución de
Contenidos - UC3M

4

WS-BPEL

¿Qué es WS-BPEL?

- BPEL: *Business Process Execution Language*
- Vocabulario XML para especificar procesos de negocio
 - Composición (*orchestration*) de servicios web
 - Resultado: un nuevo servicio web
 - Permite composición recursiva
- Similar a lenguaje de programación
 - Bucles, condicionales, variables, gestión fallos, ...
 - Interpretado por un motor de ejecución de BPEL

WS-BPEL: Historia

- Antecedentes:
 - WSFL (*Web Services Flow Language*) (IBM)
 - XLANG (Microsoft)
- A partir de anteriores → BPEL4WS:
 - v1.0 (Julio 2002) (IBM, Microsoft, BEA)
 - v1.1 (Mayo 2003) (incorporan más, como SAP)
 - Primeros intérpretes comerciales con cierto éxito
 - Se envía a OASIS para estandarización
 - Origen de WS-BPEL v2.0 (Abril de 2007)

Elementos básicos (I)

- Estructura de procesos
 - *process*
- Declaración de relaciones con servicios
 - *partnerLinks* / *partnerLink*
- Estado (variables)
 - *variables* / *variable*

Elementos básicos (II)

- Comportamiento (actividades):
 - Comunicación con WS:
 - *receive, invoke, reply*
 - Control de flujo:
 - *sequence, if-else, flow, while, repeatUntil, forEach*
 - Manipulación de datos:
 - *assign, copy, query*
 - Gestión de excepciones:
 - *faultHandlers, catch, catchAll*
- Hay más elementos

Estructura de procesos

- Elemento: *process*
 - Elemento raíz del documento XML
 - Nombre del proceso (atributo *name*)
 - Declaraciones de espacios de nombres

```
<process name="purchaseOrderProcess"  
  targetNamespace="http://example.com/ws-bp/purchase"  
  xmlns="http://docs.oasis-open.org/wsbpel/2.0/process/executable"  
  xmlns:lns="http://manufacturing.org/wsd1/purchase">  
</process>
```

Fuente: WS-BPEL 2.0 OASIS Standard

Relaciones con servicios

- Elementos: *partnerLinks* / *partnerLink*
- Definen los servicios con los que el proceso tiene interacción
 - Es llamado por / llama a
 - Como mínimo tiene que haber uno que llama al proceso
- Hacen referencia a *partnerLinkTypes* definidos en WSDL
 - Indican los roles de cada una de las partes (servicios) que interactúan y los *portTypes* implicados

Tecnologías de Distribución de
Contenidos - UC3M

11

Relaciones con servicios

- *partnerLinkTypes*

```
<wsdl:portType name="sellerPT">  
  <wsdl:operation name="submit">  
    <wsdl:input message="tns:sellerData" />  
  </wsdl:operation>  
</wsdl:portType>  
  
<wsdl:portType name="sellerAnswerPT">  
  <wsdl:operation name="answer">  
    <wsdl:input message="tns:sellerAnswerData" />  
  </wsdl:operation>  
</wsdl:portType>
```

Fuente:
WS-BPEL 2.0
OASIS Standard

```
<plnk:partnerLinkType name="sellerAuctionHouseLT">  
  <plnk:role name="auctionHouse" portType="tns:sellerPT" />  
  <plnk:role name="seller" portType="tns:sellerAnswerPT" />  
</plnk:partnerLinkType>
```

Tecnologías de Distribución de
Contenidos - UC3M

12

Relaciones con servicios

- *partnerLinks* / *partnerLink*

```
<partnerLinks>
  <partnerLink name="seller"
    partnerLinkType="as:sellerAuctionHouseLT"
    myRole="auctionHouse"
    partnerRole="seller" />
  <partnerLink name="buyer"
    partnerLinkType="as:buyerAuctionHouseLT"
    myRole="auctionHouse"
    partnerRole="buyer" />
  <partnerLink name="auctionRegistrationService"
    partnerLinkType="as:auctionHouseAuctionRegistrationServiceLT"
    myRole="auctionHouse"
    partnerRole="auctionRegistrationService" />
</partnerLinks>
```

Fuente: WS-BPEL 2.0 OASIS Standard

Variables

- Elementos: *variables* / *variable*
- Tipos:
 - Tipos de mensaje WSDL
 - Tipos XML Schema (simples o complejos)
 - Elementos XML Schema
- Lenguaje de manipulación de variables:
 - XPath 1.0

Variables

```
<variables>
  <variable name="myVar1" messageType="myNS:myWSDLMessageDataType" />
  <variable name="myVar1" element="myNS:myXMLElement" />
  <variable name="myVar2" type="xsd:string" />
  <variable name="myVar2" type="myNS:myComplexType" />
</variables>
```

Fuente: WS-BPEL 2.0 Primer

Comunicación con WS

- Elementos: *receive*, *invoke*, *reply*
- *Receive*
 - Recibir mensajes de *partner* externo
 - Especificar:
 - Si se crea o no una nueva instancia de proceso
 - Variable donde se dejará la información de entrada

```
<receive name="receiveInput"
  partnerLink="client"
  operation="Submit"
  variable="ClientSubmission"
  createInstance="yes" />
```

Fuente: WS-BPEL 2.0 Primer

Comunicación con WS

■ *Invoke*

- Llamada a servicio *partner*
- Tipos: *request/response* (bloqueante) o *one-way* (no bloqueante)
 - Según especifiquen *inputVariable* / *outputVariable*

```
<invoke name="ValidateWeeklyHours"
  partnerLink="Employee"
  operation="GetWeeklyHoursLimit"
  inputVariable="EmployeeHoursRequest"
  outputVariable="EmployeeHoursResponse" />
```

Fuente: WS-BPEL 2.0 Primer

Comunicación con WS

■ *Reply*

- Enviar resultados al invocador del proceso
 - Si interacción *request / response*
- En ocasiones puede que el resultado sea un fallo
- Especificar:
 - Variable que contiene la información a mandar
 - En caso de error, el fallo a generar (atributo *faultName*)

```
<reply partnerLink="client"
  operation="SubmitTimesheet"
  variable="TimesheetSubmissionResponse" />
```

Fuente: WS-BPEL 2.0 Primer

Control de flujo

- Elementos: *sequence*, *if-else*, *flow*, *while*, *repeatUntil*, *forEach*
- *sequence*: ejecución secuencial de actividades
- *flow*: ejecución concurrente de actividades
- *if-else*: ejecución condicional de actividades
- *while*, *repeatUntil*, *forEach*: ejecución iterativa de actividades

Control de flujo

```
<sequence>  
  <flow>  
    <invoke partnerLink="Seller" ... />  
    <invoke partnerLink="Shipper" ... />  
  </flow>  
  <invoke partnerLink="Bank" name="transferMoney" ... />  
</sequence>
```

Fuente: WS-BPEL 2.0 OASIS Standard

Control de flujo

```
<if name="isOrderBiggerThan5000Dollars">
  <condition>
    $order > 5000
  </condition>
  <invoke name="calculateTenPercentDiscount" ... />
</elseif>
  <condition>
    $order > 2500
  </condition>
  <invoke name="calculateFivePercentDiscount" ... />
</elseif>
<else>
  <reply name="sendNoDiscountInformation" ... />
</else>
</if>
```

Fuente: WS-BPEL 2.0 Primer

Control de flujo

```
<while>
  <condition>
    $iterations > 3
  </condition>
  <invoke name="increaseIterationCounter" ... />
</while>
```

Fuente:
WS-BPEL 2.0
Primer

```
<repeatUntil>
  <invoke name="increaseIterationCounter" ... />
  <condition>
    $iterations > 3
  </condition>
</repeatUntil>
```

```
<forEach parallel="no" counterName="N" ...>
  <startCounterValue>1</startCounterValue>
  <finalCounterValue>5</finalCounterValue>
  <scope>
    <documentation>check availability of each item ordered</documentation>
    <invoke name="checkAvailability" ... />
  </scope>
</forEach>
```

Manipulación de datos

- Elementos: *assign*, *copy*, *query*
- *assign*: asignación de valores a variables
- *copy*: copia de valores de variables
 - Desde origen (*from*) a destino (*to*)
- *query*: selección de la información a copiar
 - Ej.: Parte de un mensaje
 - La selección se hace con XPath 1.0

Manipulación de datos

```
<assign>
  <copy>
    <from variable="TimesheetSubmissionFailedMessage" />
    <to variable="EmployeeNotificationMessage" />
  </copy>
</assign>
```

```
<assign>
  <copy>
    <from variable="Input" part="operation1Parameter">
      <query>
        creditCardInformation
      </query>
    </from>
    <to variable="CreditCardServiceInput" />
  </copy>
</assign>
```

Fuente: WS-BPEL 2.0 Primer

Gestión de excepciones

- Elementos: *faultHandlers*, *catch*, *catchAll*
- Se captura la situación excepcional que se produce y se ejecuta cierto código
 - Como os *catch* de Java

```
<faultHandlers>
  <catch faultName="BookOutOfStockException"
    faultVariable="BookOutOfStockVariable">
    ...
  </catch>
  <catchAll>...</catchAll>
</faultHandlers>
```

Fuente: WS-BPEL 2.0 Primer

Tecnologías de Distribución de
Contenidos - UC3M

25

Otros elementos

- *wait* → esperar un determinado tiempo

```
<wait><!-- wait for three days and ten hours to go by ... -->
  <for>'P3DT10H'</for>
</wait>
```

Fuente: WS-BPEL 2.0 Primer

- *exit* → Finalizar el proceso
- *throw* → Lanzar fallo

```
<throw xmlns:FLT="http://example.com/faults"
  faultName="FLT:OutOfStock" />
```

Fuente: WS-BPEL 2.0 OASIS Standard

Tecnologías de Distribución de
Contenidos - UC3M

26

```

<process name="HelloWorld" targetNamespace="http://jbpm.org/examples/hello"
xmlns:tns="http://jbpm.org/examples/hello"
xmlns:bpel="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">

<partnerLinks>
  <!-- establishes the relationship with the caller agent -->
  <partnerLink name="caller" partnerLinkType="tns:Greeter-Caller" myRole="Greeter" />
</partnerLinks>

<variables>
  <!-- holds the incoming message -->
  <variable name="request" messageType="tns:nameMessage" />
  <!-- holds the outgoing message -->
  <variable name="response" messageType="tns:greetingMessage" />
</variables>

<sequence name="MainSeq">

  <!-- receive the name of a person -->
  <receive name="ReceiveName" operation="sayHello" partnerLink="caller"
portType="tns:Greeter" variable="request" createInstance="yes" />

  <!-- compose a greeting phrase -->
  <assign name="ComposeGreeting">
    <copy>
      <from expression="concat('Hello, ', bpel:getVariableData('request', 'name'), '!')" />
      <to variable="response" part="greeting" />
    </copy>
  </assign>

  <!-- send greeting back to caller -->
  <reply name="SendGreeting" operation="sayHello" partnerLink="caller"
portType="tns:Greeter" variable="response" />

</sequence>
</process>

```

Fuente: <http://docs.jboss.com/jbpm/bpel/v1.1/userguide/tutorial.hello.html> 27

Referencias

- Business Process Execution Language for Web Services (2nd Ed.)
 - Packt Publishing
 - ISBN: 9781904811817
- WS-BPEL 2.0 (OASIS standard)
 - <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>
- WS-BPEL 2.0 Primer
 - <http://docs.oasis-open.org/wsbpel/2.0/Primer/wsbpel-v2.0-Primer.html>

