

BPEL4WS

Business Process Execution Language for
Web Services

Themen:

- Einleitung
 - Was ist BPEL4WS?
 - Warum BPEL4WS?
 - Stellung in der „Web-Service-Architektur“
- BPEL4WS Spezifikation
 - Einführendes Beispiele
 - Sprachelemente (Syntax u. Semantik)
- BPEL4WS in der Praxis
 - BPEL4WS-Prozesse entwickeln, veröffentlichen, benutzen

Einleitung

- Was ist BPEL?
 - Sprache zur Beschreibung von Geschäftsprozessen
 - Basiert auf XML bzw. WSDL
 - Definiert Interaktion von Geschäftsprozessen innerhalb und zwischen Unternehmen
 - Beispiele für Prozesse:
 - Bestellwesen
 - Mahnwesen
 - Ablauf-Management
 - Implementierungsmöglichkeit von Workflows

Einleitung

- Hintergrund von BPEL
 - Entwickelt von: Microsoft, IBM und BEA
 - BPEL4WS Version 1.0 im Juli 2002 veröffentlicht
 - Standard zur Beschreibung v. Geschäftsprozessen?
 - Vorfahren:
 - Microsoft: XLANG
 - IBM: Web Services Flow Language (WSFL)
 - BPEL kombiniert Vorteile von XLANG und WSFL:
 - Grafikorientierte Prozess-Entwicklung (XLANG)
 - Strukturierte Prozess-Entwicklung (WSFL)

1. Einleitung

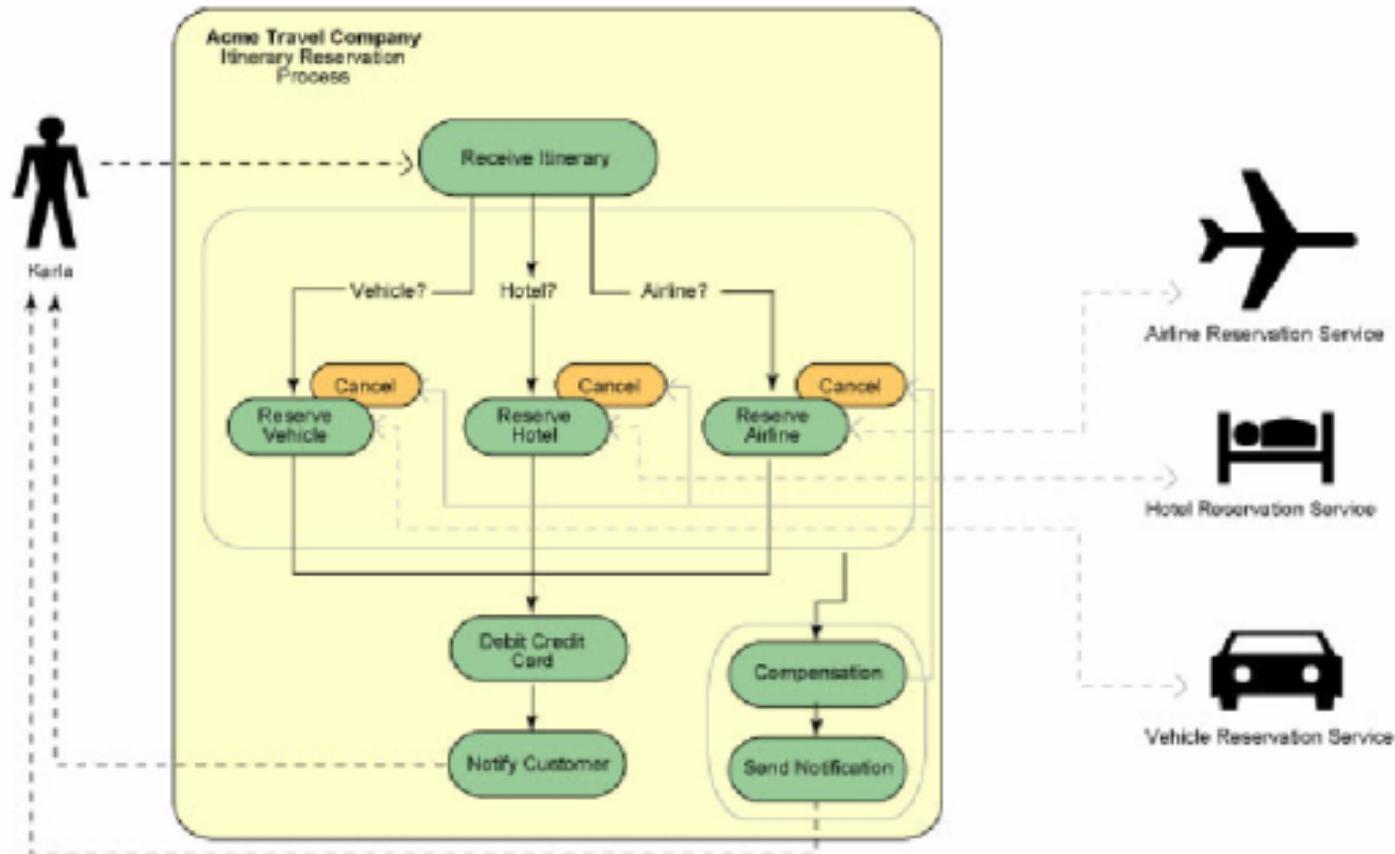
- Warum BPEL?
 - Web-Service-Spezifikationen wie SOAP, WSDL oder UDDI bieten keine Möglichkeit komplexe (Geschäfts) Prozesse zu beschreiben.
 - Keine Möglichkeit einer plattformunabhängigen Kommunikation zwischen Web Services
 - Keine Architektur, um Geschäftsprozesse allgemein auszudrücken, die Web Services verwenden.

1. Einleitung

- Was bietet BPEL?
 - Ermöglicht die Benutzung von mehreren Web Services innerhalb eines Prozesses
 - Ein in BPEL geschriebener Geschäftsprozess ist selbst ein Web Service.
 - Transparenz: Bietet wie ein gewöhnlicher Web Service seine Dienste an, WSDL-Schnittstelle
 - Einsatz von BPEL zur:
 - Implementierung ausführbarer Geschäftsprozesse,
 - Beschreibung nicht-ausführbarer abstrakter Prozesse

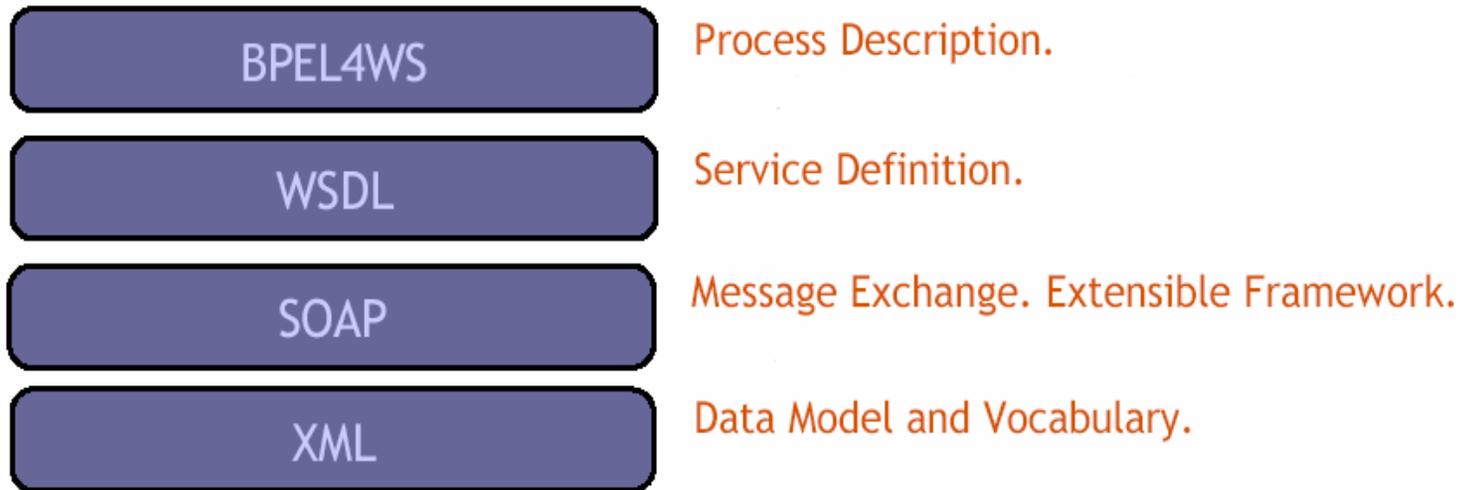
1. Einleitung

- Ein Geschäftsprozess in einer Reisebüro:



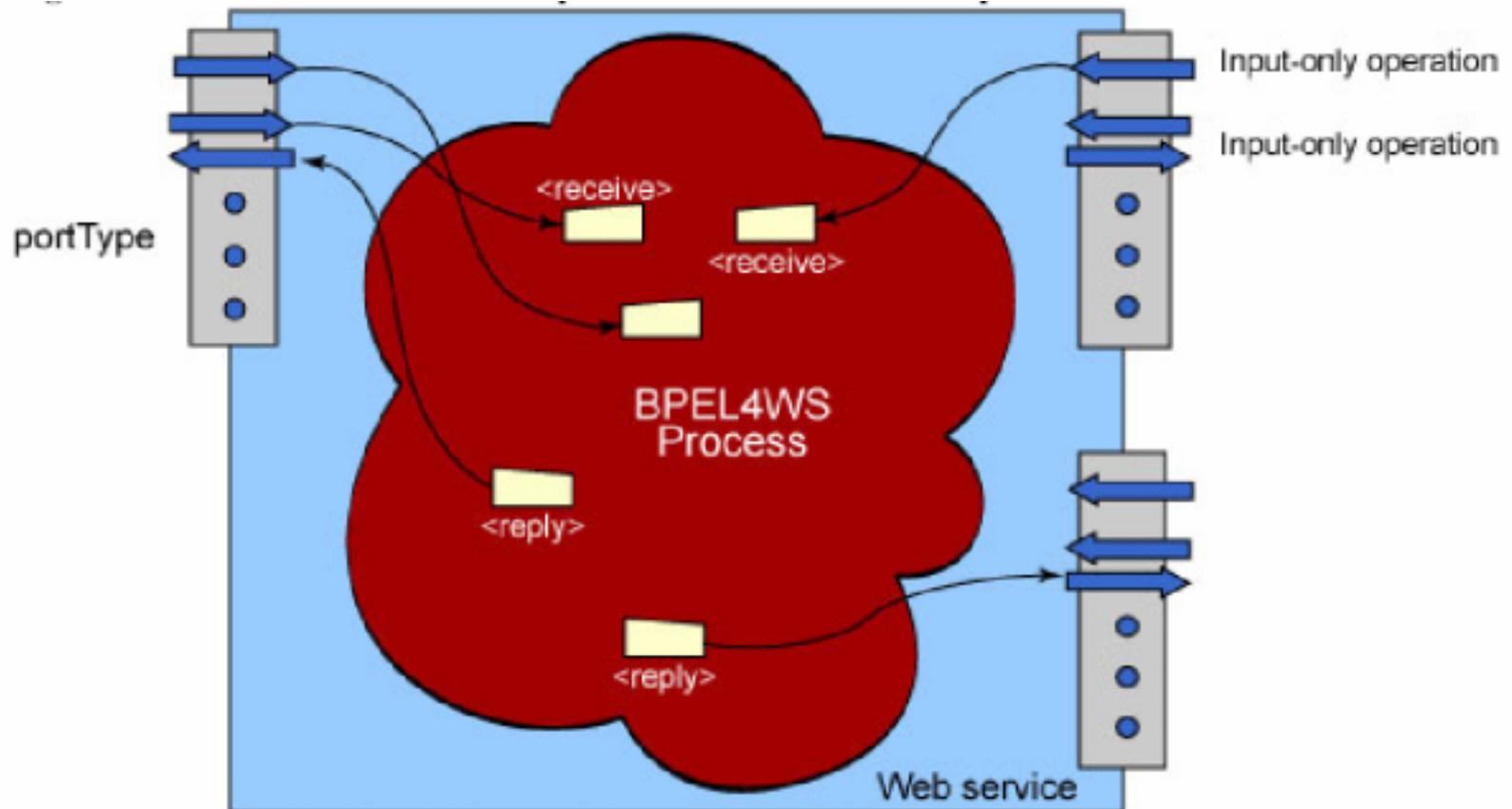
1. Einleitung

- Einordnung von BPEL in die vorhandene Web-Service-Architektur:
- BPEL-NameSpace: „<http://schemas.xmlsoap.org/ws/2002/07/business-process/>“

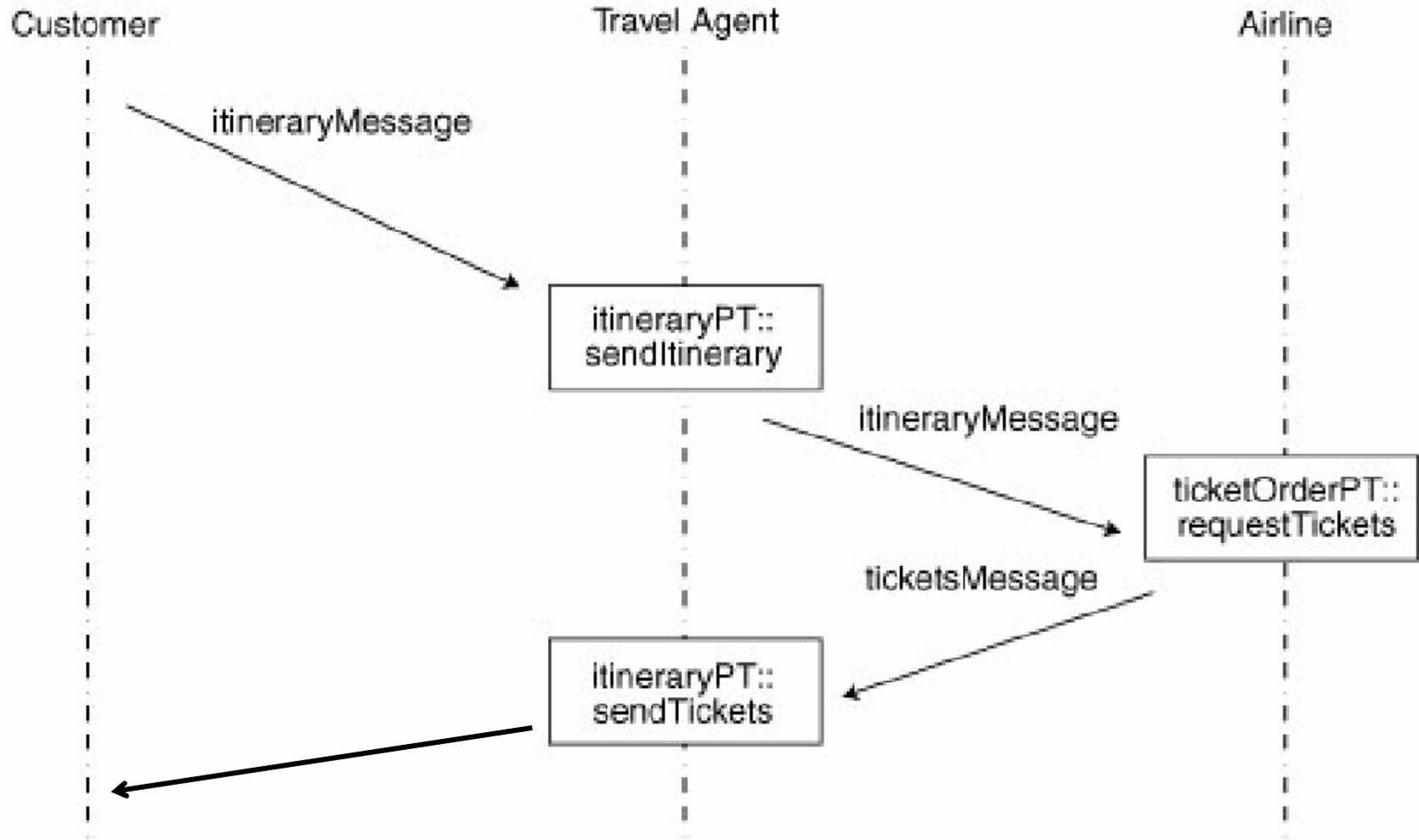


2. BPEL4WS - Überblick

- Beziehung zur WSDL:



2. BPEL4WS - Beispiel



2. BPEL4WS – <partner> u. <container>

```
<process name="ticketOrder">
  <partners>
    <partner name="customer"
      serviceLinkType="agentLink"
      myRole="agentService"/>
    <partner name="airline"
      serviceLinkType="buyerLink"
      myRole="ticketRequester"
      partnerRole="ticketService"/>
  </partners>

  <containers>
    <container name="itinerary" messageType="itineraryMessage"/>
    <container name="tickets" messageType="ticketsMessage"/>
  </containers>
```

2. BPEL4WS – <flow>,<receive>,<invoke>

```
<flow>
```

```
  <links>
```

```
    <link name="order-to-airline"/>
```

```
    <link name="airline-to-agent"/>
```

```
  </links>
```

```
  <receive partner="customer" portType="itineraryPT"
```

```
    operation="sendItinerary" container="itinerary">
```

```
    <source linkName="order-to-airline"/>
```

```
  </receive>
```

```
  <invoke partner="airline" portType="ticketOrderPT"
```

```
    operation="requestTickets" inputContainer="itinerary">
```

```
    <target linkName="order-to-airline"/>
```

```
    <source linkName="airline-to-agent"/>
```

```
  </invoke>
```

2. BPEL4WS - <reply>

```
<receive partner="airline"  
    portType="itineraryPT"  
    operation="sendTickets"  
    container="tickets"  
    <target linkName="airline-to-agent"/>  
</receive>
```

```
<reply partner="customer"  
    portType="travelPT"  
    operation="sendTickets,,  
    container="tickets"/>
```

```
</flow>  
</process>
```

2. BPEL4WS - <serviceLinkType>

- <serviceLinkTypes> definieren als Child-Elemente Rollen, welche bestimmte Partner im Geschäftsprozess zugeordnet werden.
- Rollen beinhalten bereits definierte WSDL-PortTypes, dessen Operationen dem Rolleninhaber zur Verfügung stehen.
- Beispiel:

```
<serviceLinkType name="buyerLink">
  <role name="ticketRequester">
    <portType name="itineraryPT"/>
  </role>
  <role name="ticketService">
    <portType name="ticketOrderPT"/>
  </role>
</serviceLinkType>
```

2. BPEL4WS – weitere Sprachelemente

- `<assign>` - Element
 - Speichern oder Kopieren von Daten in `<container>`-Elemente.
 - Beispiel:

```
<assign>
  <copy>
    <from container="callerRequest" part="symbol"/>
    <to container="tempContainer" part="symbol"/>
  </copy>
</assign>
```

2. BPEL4WS – weitere Sprachelemente

- `<sequence>` - Element
 - Aktivitäten innerhalb eines `<sequence>`-Elements werden in ihrer Reihenfolge ausgeführt
 - Syntax:
`<sequence standard-attributes>`
`standard-elementsactivity+`
`</sequence>`

2. BPEL4WS – weitere Sprachelemente

- `<switch>` - Element
 - Steuerungselement mit dessen Hilfe zwischen Aktivitäten gewählt werden kann.

- Syntax:

```
<switch standard-attributes>  
  standard-elements  
  <case condition="bool-expr">  
    activity  
  </case>  
  <otherwise>  
    activity  
  </otherwise>  
</switch>
```

2. BPEL4WS – weitere Sprachelemente

- `<while>` - Element
 - Solange Bedingung erfüllt ist angegebene Aktivitäten ausführen.
 - Syntax:
`<while condition="bool-expr" standard-attributes>`
`standard-elementsactivity`
`</while>`

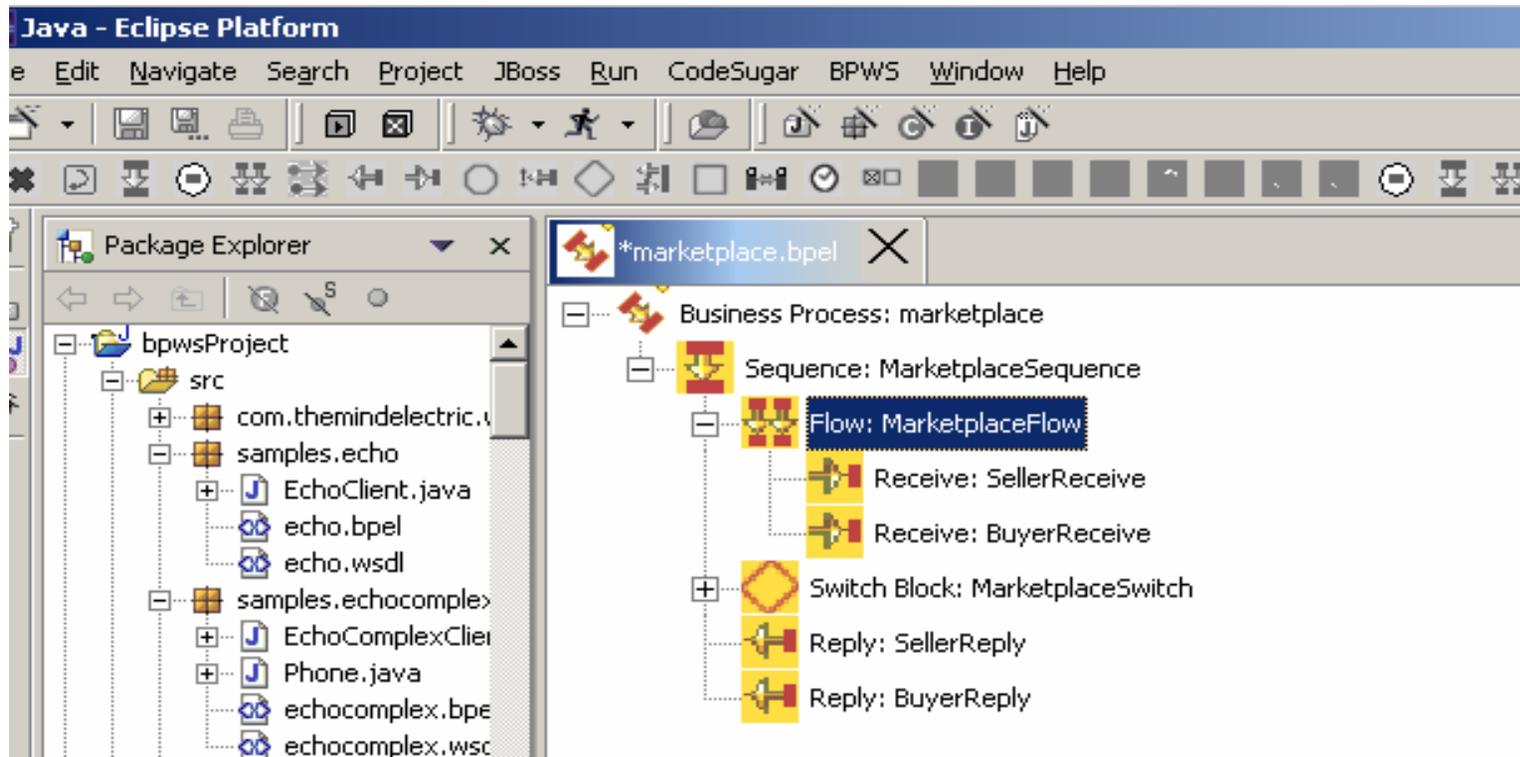
3. BPEL4WS in der Praxis

- BPWS4J (1)
 - Erste (Java-) Implementierung einer BPEL-Engine von IBM (alphaWorks)
 - Versteht BPEL-Prozesse und kann sie ausführen.
 - Einsetzbar in verschiedene Application Server:
 - WebSphere Application Server 4.0
 - Apache Tomcat 4.0
 - Enthält einige nützliche Beispiele
 - Homepage: <http://www.alphaworks.ibm.com/tech/bpws4j>

3. BPEL4WS in der Praxis

- BPWS4J (2)

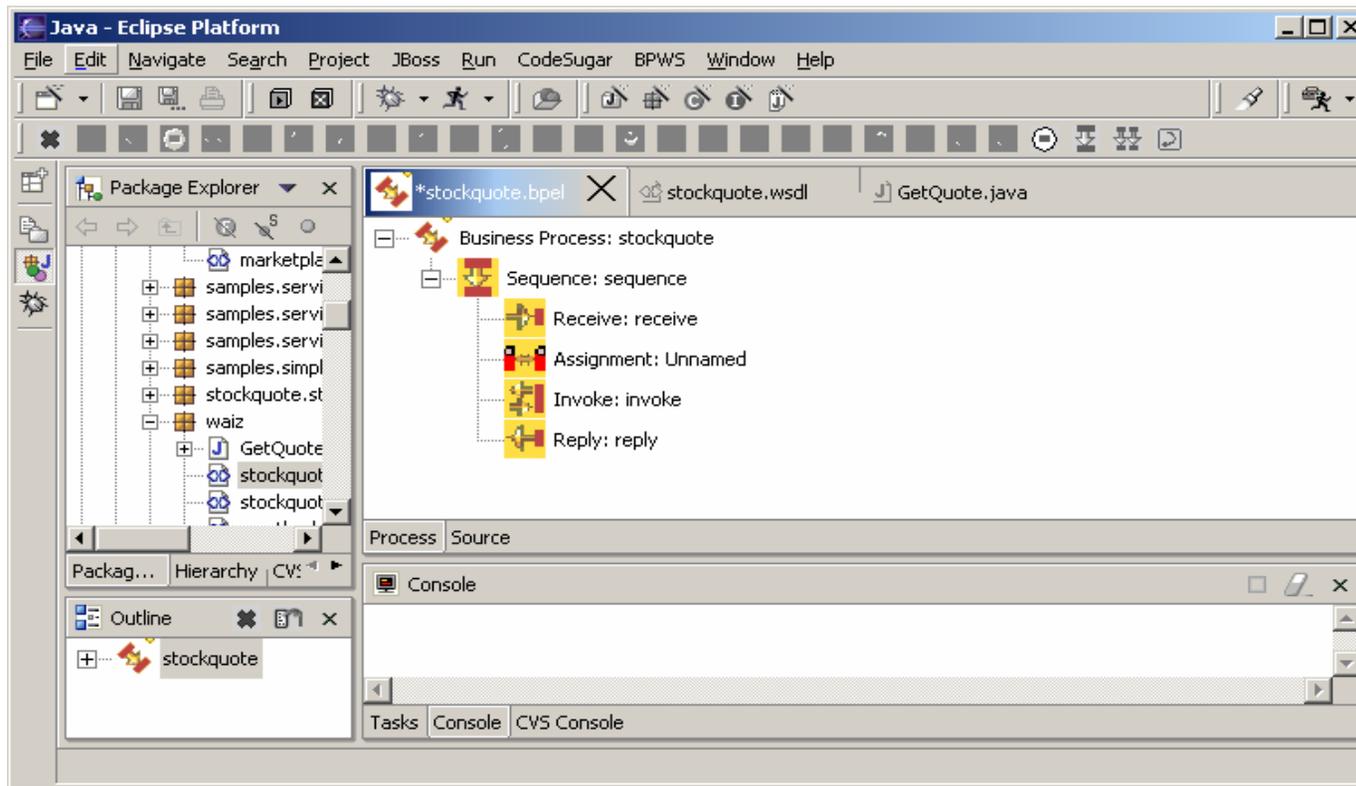
- Enthält auch ein Eclipse-Plugin zur Visualisierung und Bearbeitung von BPEL Prozessen.



3. BPEL4WS in der Praxis

• Beispiel Prozess:

- Ein einfacher BPEL-Prozess, der die Anfrage eines Aktienkurses an einer Web Service weiterleitet (stockquote.bpel):



3. BPEL4WS in der Praxis

- Veröffentlichen einer Schnittstelle für diesen Prozess (stockquote.wsdl):

```
<message name="request">
    <part name="symbol" type="xsd:string"/>
</message>
<message name="response">
    <part name="quote" type="xsd:float"/>
</message>
<portType name="StockQuotePT">
    <operation name="getQuote">
        <input message="tns:request"/>
        <output message="tns:response"/>
    </operation>
</portType>
```

3. BPEL4WS in der Praxis

- Nach Angabe aller WSDL-Quellen für fremdbenutzte Web Service wird der BPEL-Prozess von BPWS4J veröffentlicht:

IBM Business Process Execution Language for Web Services
Java Runtime

Configure Processes

List

Deploy

Un-deploy

List of Deployed Processes

Here are the deployed processes:

- Process ID (QName): {urn:stockquote:stockQuoteService}stockQuoteServiceBP
External WSDL: [\[click here\]](#)
Partner mappings:
 - provider: {http://www.theminelectric.com/wsdl/net.xmethods.services.stockquote.StockQuote/}net.xmethods.services.stockquote.StockQuoteService1c79dfc-f1dbe48599--8000

Channels:

- Apache SOAP
SOAP Address: http://localhost:8080/bpws4j/soaprpcrouter
SOAP Action URI:
Method Namespace URIs:
 - urn:stockquote:stockQuoteService#stockQuoteServiceBP#caller#urn:stockquote:stoc

3. BPEL4WS in der Praxis

- Java Client, um den veröffentlichten Prozess aufzurufen:

```
URL url = new URL("http://localhost:8080/bpws4j/soaprpcrouter");
```

```
String symbol = "IBM";
```

```
// Build the call.
```

```
Call call = new Call();
```

```
call.setTargetObjectURI("urn:stockquote:stockQuoteService#...");
```

```
call.setMethodName("getQuote");
```

```
call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
```

```
Vector params = new Vector();
```

```
params.addElement(new Parameter("symbol", String.class, symbol, null));
```

```
call.setParams(params);
```

```
// Make the call:
```

```
Response resp = call.invoke(url, "");
```

```
System.out.println(symbol + ": " + result.getValue());
```

3. BPEL4WS in der Praxis

- Ergebnis der Anfrage:
 - IBM: 86.92
- Apache AXIS:
 - Benutze Java-Framework, die SOAP komplett kapselt und somit einfache Zugriffsmöglichkeiten auf Web Services ermöglicht.
 - Nützliche Klassen:
 - WSDL2Java
 - Java2WSDL

Literatur Hinweise

- BPEL4WS Spezifikation:
 - <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- Einführende Artikel und Dokumente:
 - <http://www.ebpml.org/bpel4ws.htm>
 - <http://www-106.ibm.com/developerworks/library/ws-bpelwp/>
- Tutorial (vierteilig) :
 - <http://www-106.ibm.com/developerworks/library/ws-bpelcol1/>
 - <http://www-106.ibm.com/developerworks/webservices/library/ws-bpelcol2/>
- Software:
 - BPWS4J: <http://www.alphaworks.ibm.com/tech/bpws4j>
 - Apache Tomcat: <http://jakarta.apache.org/tomcat/>
- Freie Web Services:
 - <http://www.xmethods.com>