

Business Process Execution Language

Christian Vollmer <christian.vollmer@udo.edu>

Oliver Garbe <oliver.garbe@udo.edu>

Aufbau

- Was ist BPEL?
- Wofür ist BPEL gut?
- Wie funktioniert BPEL?
- Wie sieht BPEL aus?

Was ist BPEL?

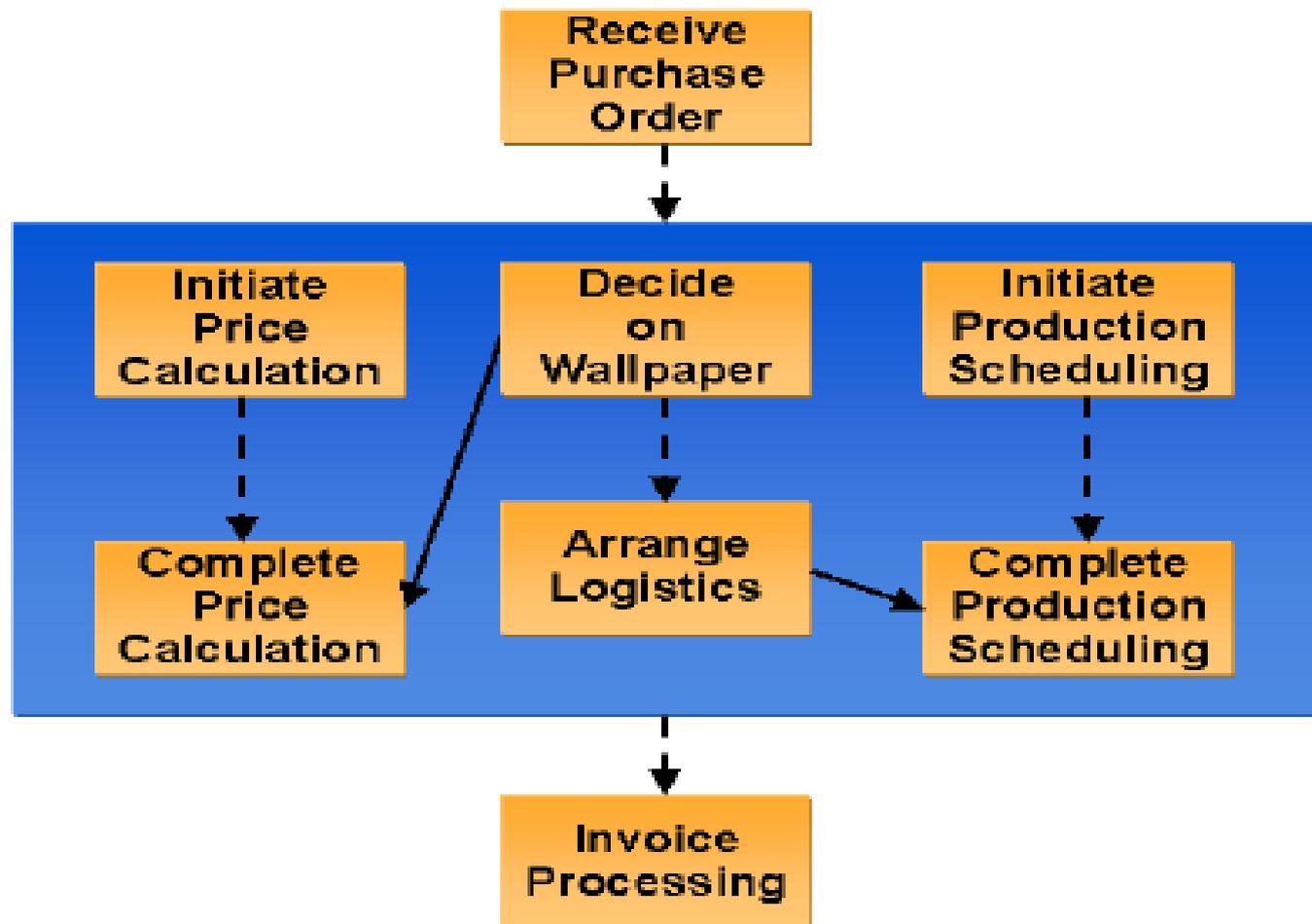
- Sprache zur Beschreibung der Steuerung und Koordination (orchestration) mehrerer Web Services in einem Geschäftsprozess
 - implementiert in XML
 - ausführbar (durch verschiedene BPEL engines)

Was ist BPEL?

- Sprache zur Beschreibung der Steuerung und Koordination (orchestration) mehrerer Web Services in einem Geschäftsprozess
 - implementiert in XML
 - ausführbar (durch verschiedene BPEL engines)
- die einzelnen Webdienste werden durch den BPEL-Prozess zusammengefasst
 - der BPEL-Prozess steuert die gesamte Abfolge
 - er enthält die Ablauflogik, um verschiedene Dienste aufzurufen, die gemeinsam einen Job erledigen

Beispiel

Geschäftsprozess



Entwicklung

- 2002 unter dem Namen BPEL4WS von IBM, Microsoft und BEA eingeführt
 - es wurde Ideen aus den Vorgängersprachen XLANG und WSFL zusammengeführt

Entwicklung

- 2002 unter dem Namen BPEL4WS von IBM, Microsoft und BEA eingeführt
 - es wurde Ideen aus den Vorgängersprachen XLANG und WSFL zusammengeführt
- im Mai 2003 an OASIS zur Standardisierung übergeben
 - nicht-gewinnorientierte Organisation, die sich mit der Weiterentwicklung von E-Business- und Web-Service-Standards beschäftigt

Entwicklung

- 2002 unter dem Namen BPEL4WS von IBM, Microsoft und BEA eingeführt
 - es wurde Ideen aus den Vorgängersprachen XLANG und WSFL zusammengeführt
- im Mai 2003 an OASIS zur Standardisierung übergeben
 - nicht-gewinnorientierte Organisation, die sich mit der Weiterentwicklung von E-Business- und Web-Service-Standards beschäftigt
- 2007 mit Version 2.0 in WS-BPEL umbenannt

Benutzte Standards

- Web Services Description Language (WSDL)
 - XML-Spezifikation zur Beschreibung der Schnittstellen von Web Services
 - beschreibt das Format der Anfrage- und Antwortnachrichten
 - der BPEL-Prozess selbst ist auch als Web Service nach WSDL-Modell implementiert

Benutzte Standards

- Web Services Description Language (WSDL)
 - XML-Spezifikation zur Beschreibung der Schnittstellen von Web Services
 - beschreibt das Format der Anfrage- und Antwortnachrichten
 - der BPEL-Prozess selbst ist auch als Web Service nach WSDL-Modell implementiert
- SOAP
 - Protokoll für den Austausch XML-basierter Nachrichten über HTTP

Benutzte Standards

- UDDI
 - Universal Description, Discovery and Integration
 - zentrale Registrierungsdatenbank für Web Services
 - Suche nach einem Web Service möglich

Benutzte Standards

- UDDI
 - Universal Description, Discovery and Integration
 - zentrale Registrierungsdatenbank für Web Services
 - Suche nach einem Web Service möglich
- XPATH
 - zum Schreiben von Ausdrücken und Suchanfragen

Motivation

- Integration häufig ein Problem in Unternehmen
 - innerhalb des Unternehmens: Enterprise Application Integration
 - mit Partnern: Business-to-Business Integration

Motivation

- Integration häufig ein Problem in Unternehmen
 - innerhalb des Unternehmens: Enterprise Application Integration
 - mit Partnern: Business-to-Business Integration
- Trend zur Service-orientierten Architektur
 - Anwendungen werden als Web Service realisiert
 - nur lose verbundene, dynamische Interaktionen
 - unterschiedliche Plattformen
 - an keiner Stelle Kontrolle über den kompletten Ablauf

Motivation

- WSDL definiert nur ein zustandsloses Interaktionsmodell
 - Nachrichtenaustausch entweder durch synchronen oder nicht in Beziehung stehenden asynchronen Aufruf

Motivation

- WSDL definiert nur ein zustandsloses Interaktionsmodell
 - Nachrichtenaustausch entweder durch synchronen oder nicht in Beziehung stehenden asynchronen Aufruf
- Viele Geschäftsprozesse brauchen eine robustere Interaktion
 - Nachrichtenaustausch zwischen zwei Services über Minuten, Stunden oder Tage

Wofür ist BPEL gut?

- zustandsabhängige langlebige Interaktion zwischen Web Services

Wofür ist BPEL gut?

- zustandsabhängige langlebige Interaktion zwischen Web Services
- BPEL kann eine Serie von Aktivitäten ausführen, die mit der Zeit eintreten und mit internen sowie externen Services interagieren

Wofür ist BPEL gut?

- zustandsabhängige langlebige Interaktion zwischen Web Services
- BPEL kann eine Serie von Aktivitäten ausführen, die mit der Zeit eintreten und mit internen sowie externen Services interagieren
- den untergeordneten Services wird die Organisation abgenommen

Beschränkungen

- bei sehr kleinen Prozessen (<5 Services) und einfachen Abläufen eventuell zu hoher Aufwand

Beschränkungen

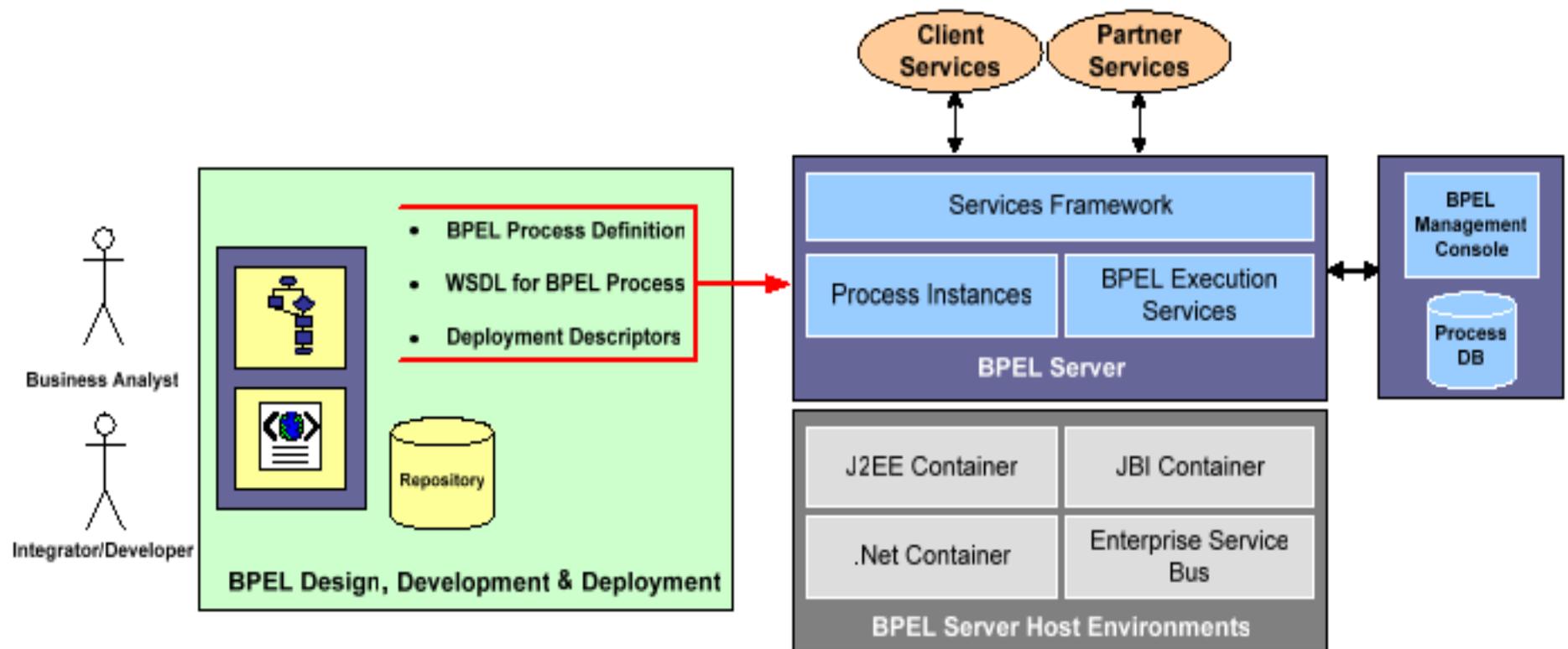
- bei sehr kleinen Prozessen (<5 Services) und einfachen Abläufen eventuell zu hoher Aufwand
- standardmäßig keine direkte Benutzerinteraktion vorgesehen
 - ermöglicht durch Erweiterung BPEL4People

Beschränkungen

- bei sehr kleinen Prozessen (<5 Services) und einfachen Abläufen eventuell zu hoher Aufwand
- standardmäßig keine direkte Benutzerinteraktion vorgesehen
 - ermöglicht durch Erweiterung BPEL4People
- keine Unterstützung von während der Laufzeit entstehenden neuen Aktivitäten

Wie funktioniert BPEL?

BPEL Environment Overview



Wie funktioniert BPEL?

- ein Prozess wird definiert als eine Serie von Aktivitäten, Kontrollflüssen und Servicedefinitionen

Wie funktioniert BPEL?

- ein Prozess wird definiert als eine Serie von Aktivitäten, Kontrollflüssen und Servicedefinitionen
- aus den Definitionen werden ausführbare Werte gebildet

Wie funktioniert BPEL?

- ein Prozess wird definiert als eine Serie von Aktivitäten, Kontrollflüssen und Servicedefinitionen
- aus den Definitionen werden ausführbare Werte gebildet
- der Prozess wird in ein ausführbares Format überführt

Wie funktioniert BPEL?

- ein Prozess wird definiert als eine Serie von Aktivitäten, Kontrollflüssen und Servicedefinitionen
- aus den Definitionen werden ausführbare Werte gebildet
- der Prozess wird in ein ausführbares Format überführt
- Prozess, WSDL Definitionen und Ablaufbeschreibungen werden gepackt und ins Programm eingebunden.

Wie funktioniert BPEL?

- bei einer Kundenanfrage wird auf der ausführenden Maschine eine Prozessinstanz erzeugt und der Ablauf geregelt

Wie funktioniert BPEL?

- bei einer Kundenanfrage wird auf der ausführenden Maschine eine Prozessinstanz erzeugt und der Ablauf geregelt
- erreicht der Prozess sein Ende wird die Instanz zerstört und der Speicher freigegeben

Wie funktioniert BPEL?

- bei einer Kundenanfrage wird auf der ausführenden Maschine eine Prozessinstanz erzeugt und der Ablauf geregelt
- erreicht der Prozess sein Ende wird die Instanz zerstört und der Speicher freigegeben
- ein langlaufender Prozess ohne Aktivität kann ausgesetzt werden.
 - die Daten des Prozesses werden gesichert
 - bei erneuter Aktivität wird der Prozess aufgeweckt und die Daten werden erneut geladen

Wie funktioniert BPEL?

- Konzepte
 - Daten für Prozessinstanzen können referenziert werden
 - Fehler können explizit abgefangen werden
 - Durchgeführte Aktivitäten können kompensiert werden
 - Durch das Beschreiben von Beziehungen zwischen Diensten können diese in Aktivitäten verwendet werden
 - Einzelne Aktivitäten können synchronisiert werden

Wie sieht BPEL aus?

- Da BPEL auf XML aufbaut kann es mit jedem Texteditor bearbeitet werden.

Wie sieht BPEL aus?

- Da BPEL auf XML aufbaut kann es mit jedem Texteditor bearbeitet werden.
- BPEL besteht aus drei verschiedenen Arten von Aktivitäten
 - Basisaktivitäten verrichten die Arbeit im Prozess
 - Strukturierende Aktivitäten leiten den Ablauf
 - Verwaltende Aktivitäten bieten Koordination und Fehlerbehandlung

Wie sieht BPEL aus?

- Deklarationen
 - `<process>` ist das Wurzelement in BPEL
 - `<partnerLink>` beschreibt verknüpfte Dienste
 - `<variable>` enthält die benutzten Daten

Wie sieht BPEL aus?

- Basisaktivitäten
 - <receive> blockiert den Prozess und wartet auf Nachricht
 - <reply> antwortet auf eine Nachricht
 - <invoke> ruft einen Webservice auf
 - kann entweder synchron oder asynchron ablaufen
 - <assign> weist einer Variablen einen neuen Wert zu

Wie sieht BPEL aus?

- Basisaktivitäten
 - <throw> signalisiert eine Fehlermeldung
 - <catch> fängt eine Fehlermeldung auf
 - <wait> setzt den Prozess für eine definierte Zeit aus
 - <empty> ist eine leere Aktivität
 - wird verwendet um Fehler abzufangen oder Prozesse zu synchronisieren

Wie sieht BPEL aus?

- Strukturierende Aktivitäten
 - `<sequence>` führt die nachfolgenden Aktivitäten der Reihe nach aus
 - `<while>` dient als Schleifenkonstrukt
 - `<switch>` dient zur Fallunterscheidung
 - kann eine oder mehrere Bedingungen enthalten: `<case condition="...">`
 - kann einen Alternativzweig enthalten: `<otherwise>`

Wie sieht BPEL aus?

- Strukturierende Aktivitäten
 - <pick> blockiert den Prozess und wartet auf ein Ereignis
 - <eventHandlers> empfängt ein Ereignis
 - <flow> erlaubt das parallele Verarbeiten von Aktivitäten
 - kann mittels <link> synchronisiert werden

Wie sieht BPEL aus?

- **Verwaltende Aktivitäten**
 - `<scope>` ermöglicht die Verbindung einer Aktion mit Variablen, Fehlerbehandlung und Kompensationsbehandlung
 - `<faultHandlers>` gibt eine Ausnahmebehandlung unter bestimmten Bedingungen an
 - `<correlations>` vereinigt Nachrichten mit bestimmten Prozessinstanzen

Wie sieht BPEL aus?

- **Verwaltende Aktivitäten**
 - `<compensate>` erlaubt es bei einem auftretenden Fehler eine ausgeführte Aktion rückgängig zu machen
 - kann nur durch einen `<faultHandler>` oder `<compensationHandler>` aufgerufen werden
 - durch `<scopes>` kann der Umfang begrenzt werden
 - `<terminate>` beendet und zerstört den laufenden Prozess sofort

Wie sieht BPEL aus?

- Beispiel

```
</process>
```

```
...
```

```
<sequence>
```

```
<receive partner="customer"  
  portType="purchaseOrderPT"  
  operation="sendPurchaseOrder"  
  variable="PO"/>
```

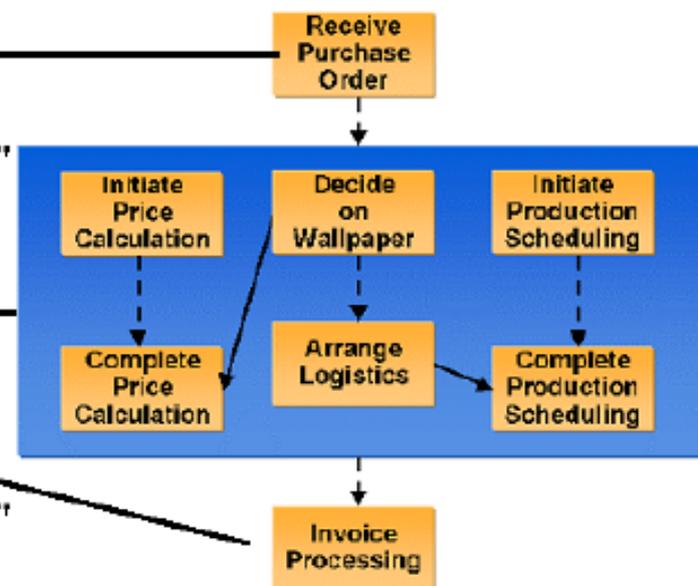
```
<flow>
```

```
</flow>
```

```
<reply partner="customer"  
  portType="purchaseOrderPT"  
  operation="sendPurchaseOrder"  
  variable="Invoice"/>
```

```
</sequence>
```

```
</process>
```



Wie sieht BPEL aus?

- Beispiel 2

```
<process name="purchaseOrderProcess">
  <variables>
    <variable name="PO"
      messageType="POMessage"/>
    ...
  </variables>
  <faultHandlers>
    <catch faultName="cannotCompleteOrder"
      faultVariable="POFault">
      <reply partner="customer"
        portType="purchaseOrderPT"
          operation="sendPurchaseOrder"
            variable="POFault"
              faultName="cannotCompleteOrder"/>
    </catch>
  </faultHandlers>
</process>
```

Literatur

- Brenda M. Michelson: Business Process Execution Language (BPEL) Primer
<<http://www.psgroup.com/detail.aspx?ID=630>>
- Frank Ryan: Web Services Business Process Execution Language - Technical Introduction
<<http://www.oasis-open.org>>
- Andreas Spall: Einführung in BPEL
<<http://www.oio.de/public/xml/einfuehrung-in-bpel4ws.htm>>