

**PV207**

# **Business Process Management**

**BPEL**

26. 3. 2009

Petr Vašíček

# Obsah přednášky

- Úvod do WS-BPEL
- Verze BPEL
- Orchestrace vs. Choreografie
- Struktura BPEL dokumentu
- BPEL aktivity
- NetBeans IDE
- OpenESB
- Příklad – Webová aplikace + BPEL + EJB

# WS-BPEL (1)

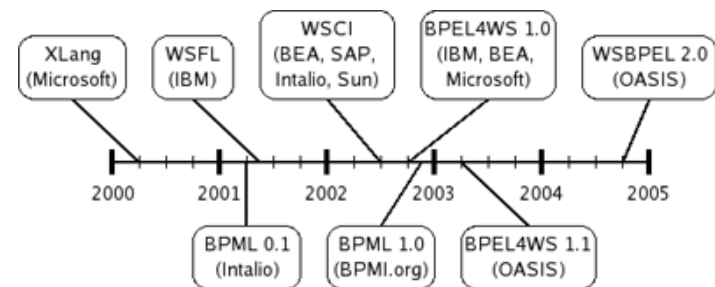
- Web Services – Business Process Executional Language
- Výslovnost „bipple“ nebo „B-pel“
- XML jazyk pro specifikování procesů na vykonatelné úrovni
- Platform-independent
- Silná podpora od velkých hráčů

# WS-BPEL (2)

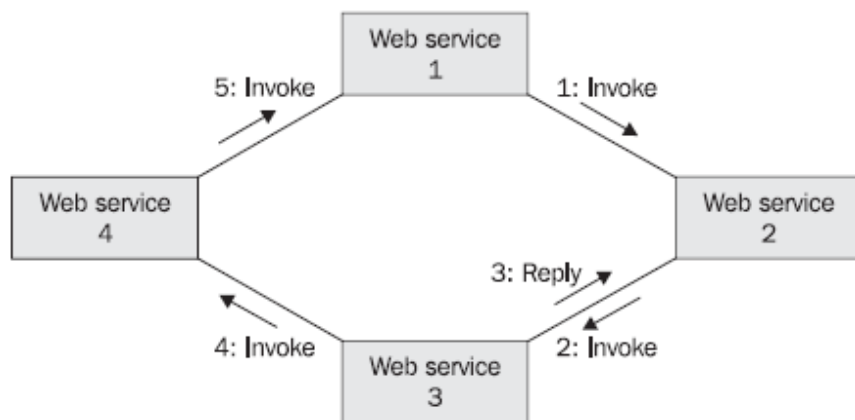
- Orchestrace webových služeb
- BPEL jako webová služba
- Podpora pro dlouho běžící procesy
- Běhové prostředí
- Abstraktní vs. vykonatelné procesy

# Verze BPEL

- BPEL4WS 1.0 (červenec 2002)
  - Původní návrh od BEA, IBM a Microsoft
  - Spojuje myšlenku WSFL (IBM) a XLANG (MS)
- BPEL4WS 1.1 (květen 2003)
  - Revidovaný návrh předložený OASIS
- WS-BPEL 2.0 (prosinec 2005, ratifikace duben 2007)
  - Nové elementy (forEach, repeatUntil...)
  - Přejmenování (switch/case -> if/else, terminate->exit)
  - Inicializace proměnných
  - Několik dalších změn...



# Orchestrace vs. Choreografie

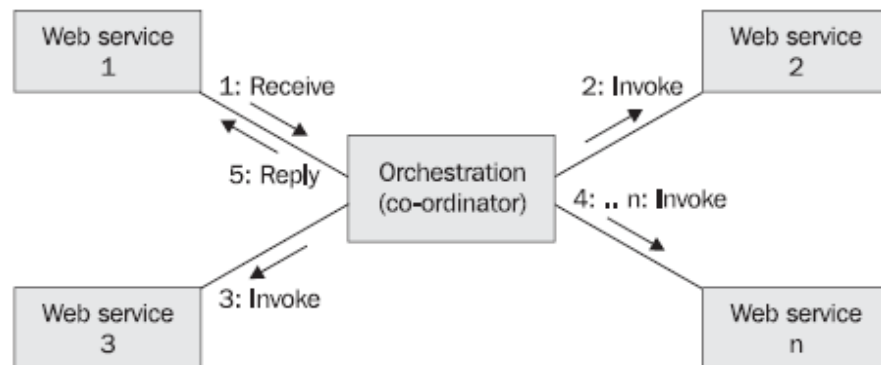


## Choreografie

- Není centrální koordinátor
- Každá služba ví s kým interaguje
- Účastníci znají business proces

## Orchestrace

- Centrální koordinátor (proces)
- Zodpovědnost za proces
- Služby neví, že jsou součástí
- Možnost náhradních scénářů



# Struktura BPEL Dokumentu

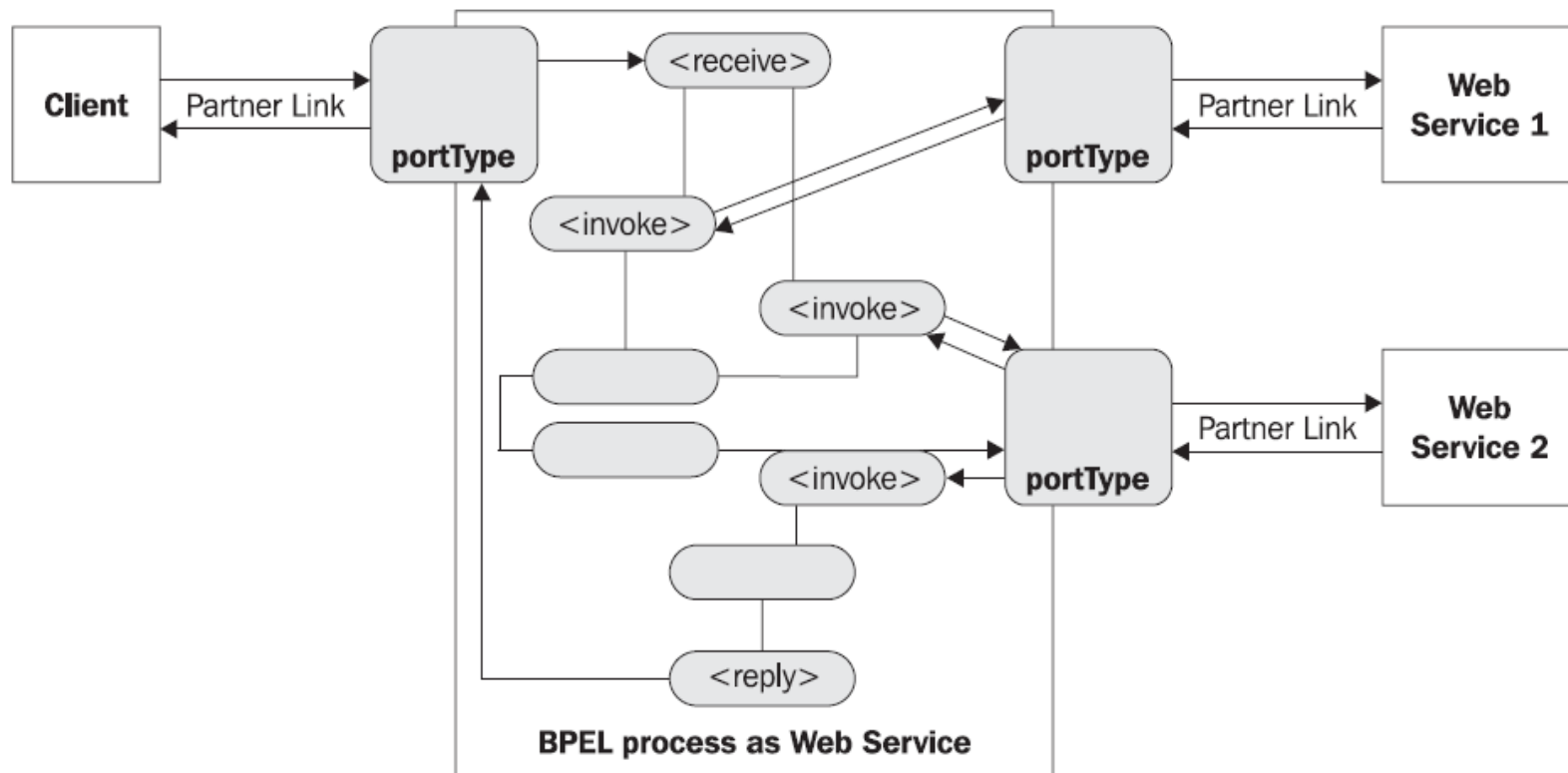
```
<process>  
  <!-- Definition and roles of process participants -->  
  <partnerLinks> ... </partnerLinks>  
  <!-- Data/state used within the process -->  
  <variables> ... </variables>  
  <!-- Properties that enable conversations -->  
  <correlationSets> ... </correlationSets>  
  <!-- Exception handling -->  
  <faultHandlers> ... </faultHandlers>  
  <!-- Error recovery - undoing actions -->  
  <compensationHandlers> ... </compensationHandlers>  
  <!-- Concurrent events with process itself -->  
  <eventHandlers> ... </eventHandlers>  
  <!-- Business process flow -->  
  (activities)*  
</process>
```

# BPEL activity

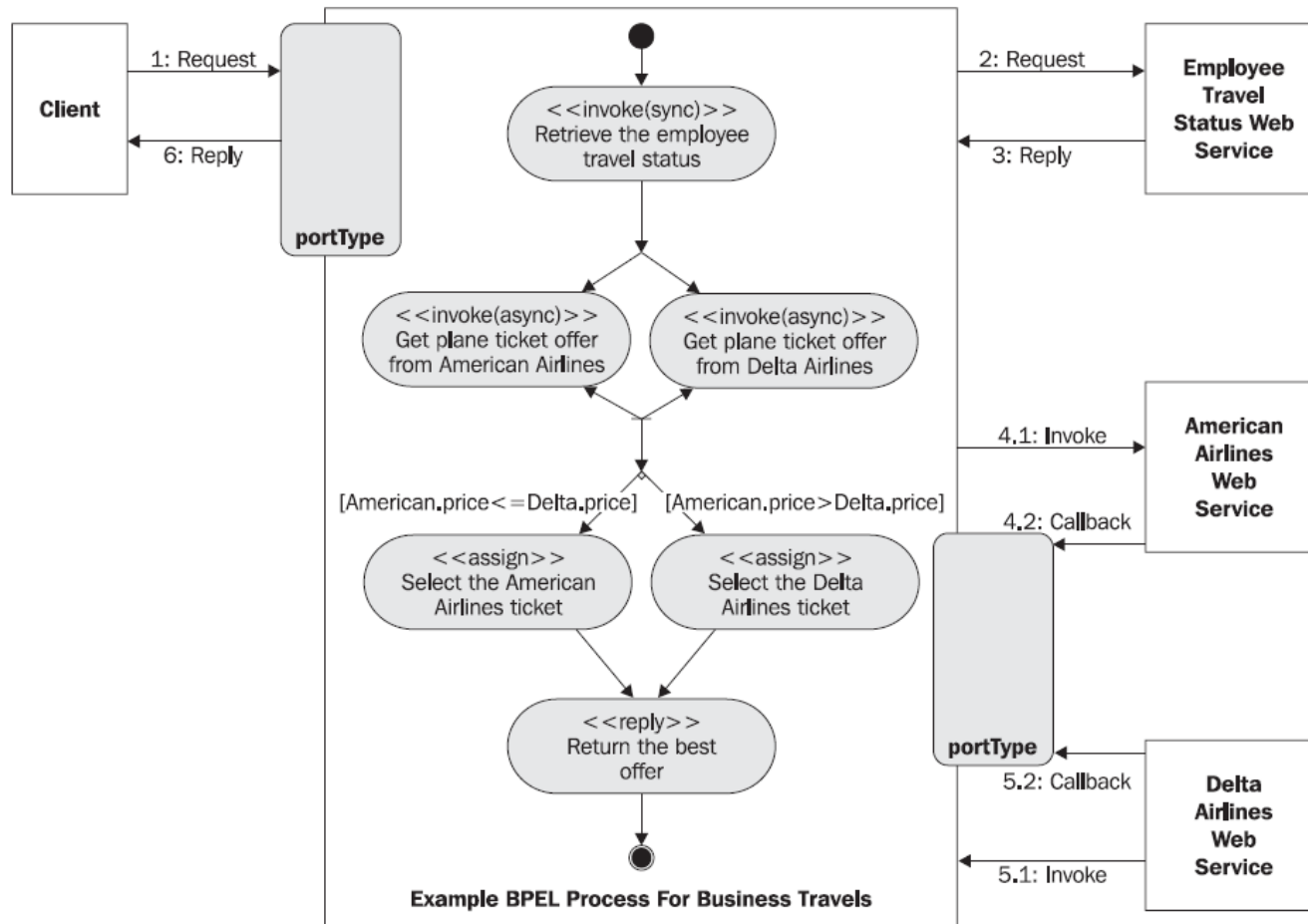
- Basic activities
  - <receive>
  - <reply>
  - <invoke>
  - <assign>
  - <throw>
  - <wait>
  - <empty>
  - <exit>
- Structured activities
  - <if>
  - <while>
  - <repeatUntil>
  - <foreach>
  - <pick>
  - <flow>
  - <sequence>
  - <scope>



# BPEL – Příklad schématu (1)



# BPEL – Příklad schématu (2)



# Základní aktivity

- **Receive**
  - Zastaví proces a čeká na příchod specifikované zprávy
- **Reply**
  - Zašle zprávu jako odpověď na dříve přijatou zprávu
- **Invoke**
  - Vyvolá WS operaci typu one-way nebo request-response
- **Assign**
  - Manipulace s proměnnými
  - Kopírování pomocí elementů <copy>, <from> a <to>

# Základní aktivity

- Wait
  - Zastaví proces na určitý časový interval
  - Je učeno buďto *jak dlouho* má proces čekat nebo *do kdy* (datum)
- Throw
  - Vyvolá výjimku uvnitř business procesu
- Empty
  - Značí, že není definována žádná akce
- Exit
  - Okamžitě ukončí vykonávání instance business procesu

# Strukturované aktivity

- Sequence
  - Obsahuje aktivity, jež jsou spouštěny sekvenčně
- Flow
  - Obsahuje aktivity, jež jsou spouštěny paralelně
- If-else
  - Vybere jednu větev podle pravdivostního výrazu
- Pick
  - Zastaví proces a čeká na příchod první z více definovaných událostí
  - Událostí může být:
    - Příchod zprávy (Receive)
    - Vypršení časového limitu (Wait)

# Strukturované aktivity

- While
  - Obsažené aktivity jsou opakovány, dokud platí určitý predikát
- RepeatUntil
  - Obsažené aktivity jsou opakovány, dokud neplatí určitý predikát
- ForEach
  - Obsažené aktivity jsou vykonány sekvenčně či paralelně
  - Počet opakování kontrolován čítačem
- Scope
  - Obsažené aktivity mají vlastní lokální proměnné, event handlers a fault handlers

# BPEL – Příklad (1)

```
<sequence>
  <receive partnerLink="customer" portType="lns:purchaseOrderPT"
    operation="sendPurchaseOrder" variable="PO"
    createInstance="yes" />
  <flow>
    <invoke partnerLink="inventoryChecker" portType="lns:inventoryPT"
      operation="checkINV" inputVariable="inventoryRequest"
      outputVariable="inventoryResponse" />

    <invoke partnerLink="creditChecker" portType="lns:creditPT"
      operation="checkCRED" inputVariable="creditRequest"
      outputVariable="creditResponse" />
  </flow>
  ...
  <reply partnerLink="customer" portType="lns:purchaseOrderPT"
    operation="sendPurchaseOrder" variable="invoice"/>
</sequence>
```

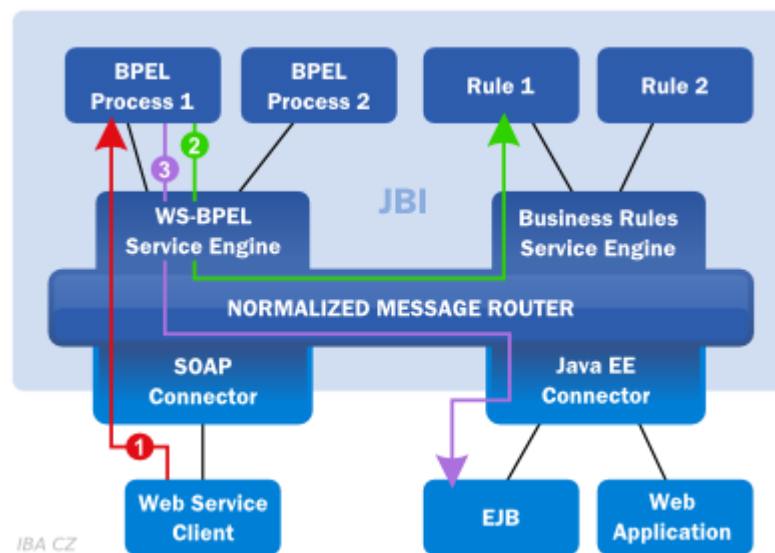
## BPEL – Příklad (2)

```
<receive name="receiveRequest" />
  <assign name="copyRequestToVariable"/>
  <if name="decision">
    <condition> $variable = 1 </condition>
    <sequence>
      <flow>
        <sequence>
          <invoke name="operation1" />
        </sequence>
        <sequence>
          <invoke name="operation2" />
        </sequence>
      </flow>
    </sequence>
    <elseif>
      <condition> $variable = 2 </condition>
      <sequence>
        <invoke name="operation3" />
      </sequence>
    </elseif>
    <else>
      <sequence>
        <invoke name="operation4" />
      </sequence>
    </else>
  </if>
<reply name="replyToUser" />
```



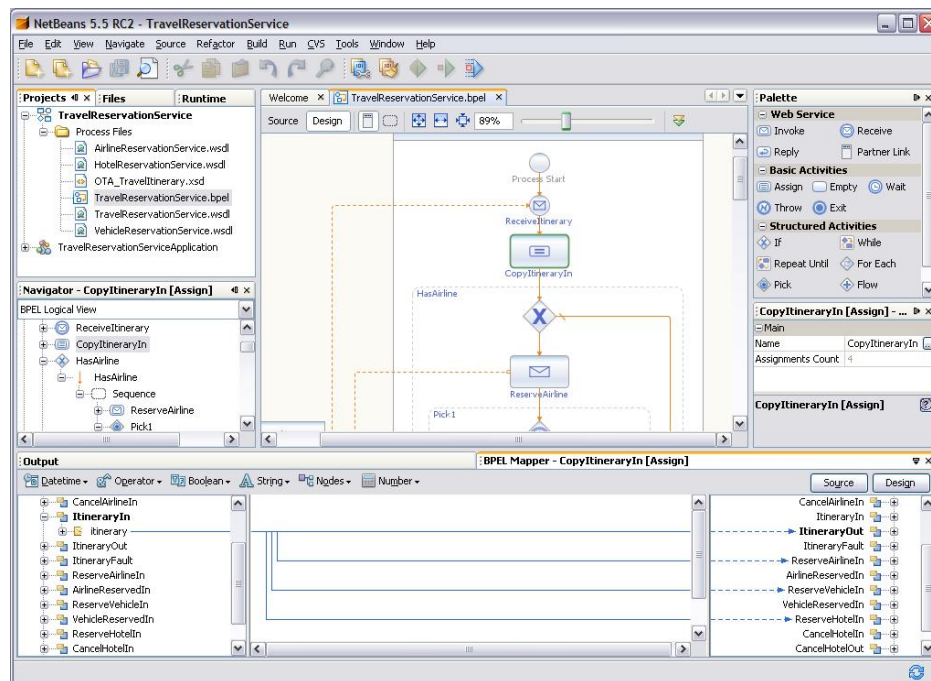
# OpenESB

- Implementace JBI (JSR-208)
- Podniková sběrnice služeb (Enterprise Service Bus)
- Modul v aplikačním serveru Glassfish
- Součástí je běhové prostředí pro BPEL (WS-BPEL Service Engine)



# NetBeans IDE

- Vývojové prostředí od Sun Microsystems
- Obsahuje BPEL Designer pro modelování procesu v jazyce WS-BPEL
- Implementovaný proces lze přímo nasadit do běhového prostředí



# Demo – NetBeans & OpenESB

# Co bude příště...

- Loops
- Correlation sets
- Fault handlers
- Event handlers
- Scopes