

**PV207**

# **Business Process Management**

Pokročilé modelování v BPEL

2. 4. 2009

Petr Vašíček

# Obsah přednášky

- Opakování BPEL
- Příklad – proces žádostí
- Correlation sets
- Pick
- Scopes
- Event handlers
- Fault handlers
- Shrnutí & domácí úloha

# Náplň následujících přednášek

- 9. 4. - Modelování v BPEL
- 16. 4. - IBM Websphere
- 23. 4. - IBM Websphere
- 30. 4. - Intalio | BPMS
- 7. 5. - Projekt
- 14. 5. - Projekt
- 21. 5. - Projekt

# Opakování - BPEL

- Co je to WS-BPEL?
- Jak BPEL využívá webové služby?
- Orchestrace vs. Choreografie
- Dva typy BPEL aktivit
- Jak vypadá BPEL projekt (např. v NetBeans)?

# Opakování - BPEL aktivity

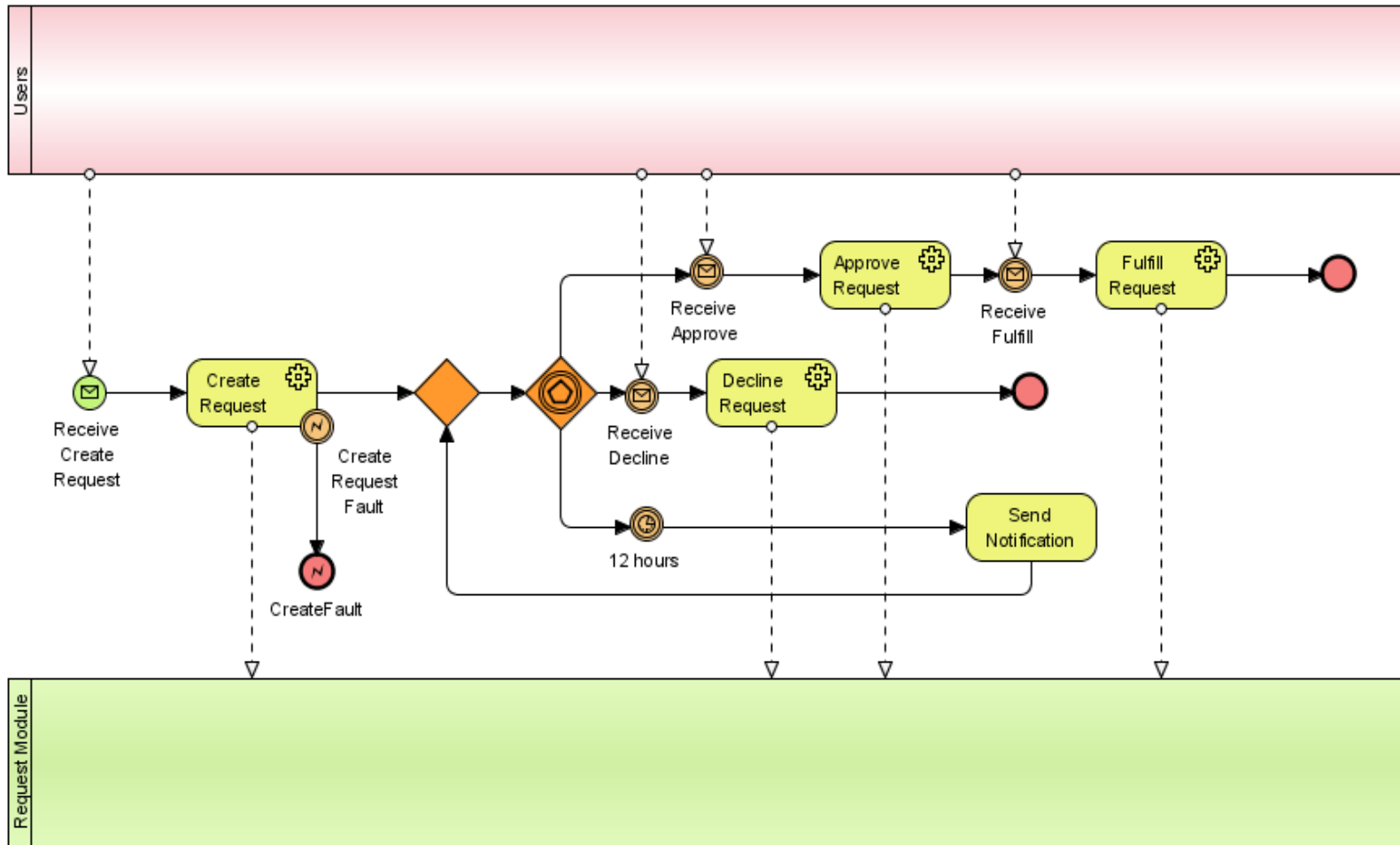
- Basic activities

- <receive>
- <reply>
- <invoke>
- <assign>
- <throw>
- <wait>
- <empty>
- <exit>

- Structured activities

- <if>
- <while>
- <repeatUntil>
- <foreach>
- <pick>
- <flow>
- <sequence>
- <scope>

# Příklad – Proces žádostí (BPMN)



# Demo – NetBeans & OpenESB

<http://www.fi.muni.cz/~xvasice5/RequestExample.zip>

# Correlation set

- Dlouho trvající procesy = více vstupů
- Synchronizace příchozí zprávy s konkrétní instancí procesu
- WSDL
  - Property
    - `<vprop:property name="requestId" type="xsd:long"/>`
  - PropertyAlias
    - `<vprop:propertyAlias propertyName="tns:requestId" messageType="tns:approveRequest" part="part1">`
    - `<vprop:query>/ns:approveRequest/ns:requestId</vprop:query>`
    - `</vprop:propertyAlias>`
- BPEL
  - CorrelationSet
    - `<correlationSet name="requestId" properties="ns1:requestId"/>`
- Inicializace – Receive, Invoke, Reply
- Napojení - Receive, OnMessage, OnEvent



# Pick

```
<pick name="ApproveOrDecline" xmlns:tns="http://services.request.fi.muni.cz/">
  <onMessage partnerLink="Client" operation="approveRequest"
    portType="tns:processRequest.wsdlPortType" variable="ReceiveApprove">
    <correlations>
      <correlation set="requestId" initiate="no"/>
    </correlations>
    <sequence>

```

# **Demo – NetBeans & OpenESB**

## **Correlations & Pick**

# Scopes

- `<scope>`
- Rozdělení procesu do více částí
- Obsahuje aktivity
- Hlavní scope = process scope
- V rámci jednoho scope:
  - Lokální proměnné
  - Lokální fault handlers
  - Lokální event handlers
  - Lokální compensation handlers

# Event handlers

- Událost může přijít kdykoli v daném scope
- OnEvent / OnAlarm

```
<eventHandlers>
  <onEvent partnerLink="Client" operation="updateRequestName"
    portType="ns1:processRequest.wsdlPortType" variable="ReceiveUpdateName" messageType="ns1:updateRequestName">
    <correlations>
      <correlation set="requestId" initiate="no"/>
    </correlations>
    <scope name="ScopeUpdateName">
      <variables>
        <variable name="UpdateNameIn" xmlns:tns="http://services.request.fi.muni.cz/" messageType="tns:changeName"/>
        <variable name="ReplyUpdateName" messageType="ns1:requestActionResponse"/>
      </variables>
      <sequence>
        <!-- Content of the sequence -->
      </sequence>
    </scope>
  </onEvent>
</eventHandlers>
```

# **Demo – NetBeans & OpenESB**

## **Scopes & Event handler**

# Fault handlers

- Odchytí výjimku v daném scope
- Pro dokočení sekvence ve fault handleru pokračuje proces do následujícího scope

```

<!-- process scope -->
<faultHandlers>
  <catchAll>
    <sequence name="Sequence8">
      <empty name="LogFault"/>
    </sequence>
  </catchAll>
</faultHandlers>
<sequence>
  <scope name="Creation">
    <variables>
    <faultHandlers>
      <catchAll>
        <scope name="Scope1">
          <sequence name="Sequence7">
            <assign>
              <reply name="ReplyCreateFault" partnerLink="Client" operation="createRequest"
                portType="ns1:processRequest.wsdlPortType" faultName="ns1:createRequestFault"
                variable="CreateRequestFault"/>
              <throw name="ThrowFault" faultName="ns1:createRequestFault"/>
            </sequence>
          </scope>
        </catchAll>
      </faultHandlers>
    </sequence>
  </scope>
  ...
</sequence>

```

# **Demo – NetBeans & OpenESB**

## **Fault handler**

# Co už jsme v příkladech použili?

- Receive, Invoke, Reply
- If, RepeatUntil
- Pick (OnMessage & OnAlarm)
- Correlation set
- Fault handlers, Event handlers (OnMessage)
- Scopes, Throw, Exit
- BPEL Variables
- Debugging
- Test cases



# Domácí úloha - zadání

- Namodelujte libovolný (jednoduchý) proces v BPEL, včetně backendu, jehož funkcionality bude přístupná přes webovou službu. Proces musí splňovat následující:
  - Musí mít alespoň dva vstupy (použití Correlation Set)
  - Musí v něm být použity elementy If a Pick
  - Na každý vstup do procesu bude v kompozitní aplikaci vytvořen Test Case
  - Použití klientské aplikace je volitelné
- Úkol bude kontrolován na příští přednášce a bude započítán do konečného hodnocení
- Při neúčasti uložit všechny části aplikace v ZIP do odevzdáárny nejpozději v úterý 7.4.
- Zdroje pro inspiraci – NetBeans Dev Guide, příklady z přednášek ve stud. materiálech