BPM beyond Web Services

+ Activiti

+ Apache Camel
Main Tasks
Evaluate Technologies and Products
Requirements Engineering
Enterprise Architecture Management
Business Process Management
Architecture and Development of Applications
Planning and Introduction of SOA
Integration of Legacy Applications
Cloud Computing

Contact
Email: kai.waehner@mwea.de
Blog: www.kai-waehner.de/blog
Twitter: @KaiWaehner
Social Networks: Xing, LinkedIn

Consulting
Developing
Speaking
Coaching
Writing
What is the Key Message?
BPM should be used for optimizing business processes!
BPM should be used for optimizing business processes!

BPM should NOT be used for systems integration!
BPM should be used for optimizing business processes!

BPM should NOT be used for systems integration!

Activiti and Apache Camel are a perfect combination!
1) Business Process Management (BPM)

1) Activiti

1) Apache Camel

2) Combination of Activiti and Apache Camel
1) Business Process Management (BPM)

1) Activiti

1) Apache Camel

2) Combination of Activiti and Apache Camel
BPM attempts to **improve processes continuously.**

*It can therefore be described as a "process optimization process."*  

---

Wikipedia
Why BPM?

Business-IT-Alignment

- increase efficiency
- better quality
- reduce costs
- enable new business models
How to do BPM?
Standards

BPM

BPEL

jPDL

BPMN

WF-XML

XPDL

ARIS EPC

BPEL4People
**Business Process Model and Notation (BPMN)** is a **graphical representation** for specifying business processes in a business process model.

Wikipedia
BPMN 2.0

• BPMN is a standard notation for designing business processes
  (versus: UML is a standard modeling language best suited for designing and implementing software)

• not just flow charts! sufficient restrictions / constraints => executable!

• standardized XML format

• orchestration and choreography

• extension points => add specific needs without breaking interoperability

• optional mapping of a BPMN subset to BPEL
  (restricted to block-structured flows without cycles)
BPMN 2.0 - Business Process Model and Notation

Activities
- Task
- Transaction
- Task Flow
- Call activity
- Call gateway
- Gateway

Conversations
- Conversation Diagram

Choreographies
- Choreography Diagram

Collaboration Diagram
- Flow (Black Box)

Gateways
- Exclusive Gateway
- Parallel Gateway
- Complex Gateway

Swimlanes
- Process Lane
-泳游的类型

Data
- Collection Data Object
- Source Data Object

Events
- Start Event
- Intermediate Event
- Ending Event
- Association Role

Activities
- Task
- Transaction
- Task Flow
- Call activity
- Call gateway
- Gateway

Conversations
- Conversation Diagram

Choreographies
- Choreography Diagram

Collaboration Diagram
- Flow (Black Box)

Gateways
- Exclusive Gateway
- Parallel Gateway
- Complex Gateway

Swimlanes
- Process Lane
-泳游的类型

Data
- Collection Data Object
- Source Data Object

Events
- Start Event
- Intermediate Event
- Ending Event
- Association Role
20-80 rule
When to use BPM?

- long-running stateful workflows
- frequently changing processes
- human interaction
The Four Myths of BPM Projects

1) Business analysts will create executable process models
2) Business analysts can create executable process models
3) Business analysts want to create executable process models
4) IT wants business analysts to create executable process models

http://www.activevos.com/blog(soa/the-four-myths-of-bpm-projects-what-it-project-teams-need-to-know/2011/01/18/
1) Business Process Management (BPM)

1) Activiti

1) Apache Camel

2) Combination of Activiti and Apache Camel
Alternatives for BPM

- No Tool
- BPM Framework
- BPM Suite

Complexity of Orchestration

Low

High
Alternatives for BPM

BPM Framework

No Tool

BPM Suite

Low

High

Complexity of Orchestration

Activiti vs. JBoss jBPM vs. Bonita vs. ProcessMaker
What is Activiti?

- open source
- BPMN 2.0 process engine
- lightweight
- easy to use
- Java API
- developer-focused
- embeddable
Tool Stack

Activiti Modeler

Process Engine

Activiti Designer

camunda fox style

Activiti Explorer
How does Activiti work?

- BPMN 2.0 process engine
- state machine with one active state
- execution progresses via transitions
- most BPMN 2.0 elements are implemented as a state
- states are connected with leaving and arriving transitions (called sequence flows)
- every state (i.e. its corresponding BPMN 2.0 element) can have a piece of logic attached (executed when the process instance enters the state)
**Services**

- RuntimeService
- TaskService
- FormService
- HistoryService
- IdentityService
- ManagementService
- RepositoryService

**Service Tasks (BPMN Standard)**

- Web Service Task
- Script Task (e.g. Groovy or JavaScript)
- User Task
- Business Rule Task

**Service Tasks (Activiti Extensions)**

- Java Tasks
- Spring Service Task
- ... more
<process id="bookorder" name="Order book">
  <startEvent id="startevent1" name="Start"/>
  <sequenceFlow id="sequenceflow1" name="Validate order"
              sourceRef="startevent1" targetRef="scripttask1"/>
  <scriptTask id="scripttask1"
              name="Validate order" scriptFormat="groovy">
    <script>out:println "validating order for isbn "+ isbn;</script>
  </scriptTask>
  <sequenceFlow id="sequenceflow2" name="Ending process"
              sourceRef="usertask1" targetRef="endevent1"/>
  <endEvent id="endevent1" name="End"/>
</process>
ProcessEngine processEngine = ProcessEngineConfiguration
    .createStandaloneInMemProcessEngineConfiguration()
    .buildProcessEngine();

RuntimeService runtimeService =
    processEngine.getRuntimeService();

RepositoryService repositoryService =
    processEngine.getRepositoryService();

repositoryService.createDeployment()
    .addClasspathResource("bookorder.simple.bpmn20.xml")
    .deploy();

ProcessInstance processInstance =
    runtimeService.startProcessInstanceByKey("simplebookorder");

assertNotNull(processInstance.getId());

System.out.println("id " + processInstance.getId() + " " +
    processInstance.getProcessDefinitionId());
Use Case

Start Event

Script Task

Service Task

End Event

Log Order
Automatic

Transform Order
Automatic

Confirm Order
Manual
Activiti in Action
1) Business Process Management (BPM)

1) Activiti

1) Apache Camel

2) Combination of Activiti and Apache Camel
Come on, guys! We are all CamelOne!
Alternatives for Systems Integration

- **No Tool**
- Integration-Framework
- Enterprise Service Bus (ESB)

**Complexity of Integration**

- **Low**
- **High**
Alternatives for Systems Integration

Apache Camel vs. Spring Integration vs. Mule ESB

http://www.kai-waehner.de/blog/2012/01/10/spoilt-for-choice-which-integration-framework-to-use-spring-integration-mule-esb-or-apache-camel
Why Apache Camel?

- Standardized Modeling
- Efficient Realization
- Developer-focused
- Automatic Testing
- Many Components
- Several DSLs
- Awesome Community
1) Business Process Management (BPM)

1) Activiti

1) Apache Camel

2) Combination of Activiti and Apache Camel
Apache Camel + Activiti
BPMN Integration

- Script Task
  - Log Order
  - Transform Order
  - Confirm Order

- Service Task

- Groovy
- JavaScript etc.

- Java SOAP Web Service

- Everything (from Cobol to Ruby)
Reinventing the Wheel ...

All Roads lead to Rome!
Separation of Concerns
Both can realize processes. Both can integrate services.
Both can realize processes. Both can integrate services.

- Support for **long running stateful processes**
- **Human workflow integration**
Overlappings?

Both can realize processes. Both can integrate services.

- Support for long running stateful processes
- Human workflow integration

- Connectivity / Adaptors to connect to external systems using a variety of different protocols
- Predefined EIP for message routing

http://www.pleus.net/blog/?p=1028
Both ...

- ... are lightweight
- ... are open source
- ... are developer-focused
- ... offer combination out-of-the-box
Connecting BPMN, Activiti and Camel

**BPMN - Java Service Task**

```xml
<serviceTask id="myServiceTask"
    activiti:delegateExpression="${camelBehaviour}" />
<sequenceFlow sourceRef="myServiceTask" targetRef="myUserTask" />
```

**BPMN Process Instance**

**Spring Bean**

```xml
<bean id="camelBehaviour" class="org.activiti.camel.CamelBehaviour">
    <constructor-arg index="0">
        <list>
            <bean class="org.activiti.camel.SimpleContextProvider">
                <constructor-arg index="0" value="activitiCamelProcess" />
                <constructor-arg index="1" ref="camelContext" />
            </bean>
        </list>
    </constructor-arg>
</bean>
```

**CamelContext**
// Producer => Call Activiti process from Camel
from("direct:start")
 .to("activiti:myProcess");

// Consumer => Get called from Activiti process
from("activiti:myProcess:myServiceTask")
 .log(LoggingLevel.INFO, "Received message on service task ${property.var1}")
 .setProperty("var2").constant("world")
 .setBody().properties();
Activiti and Apache Camel combined ...
Did you get the Key Message?
BPM should be used for optimizing business process
BPM should NOT be used for systems integration
Activiti and Apache Camel are a perfect combination
Did you get the Key Message?
Whet your appetite?
Become a Part of the Open Source Community!
Thank you for your Attention. Any Questions?

Kai Wähner

MaibornWolff et al:
www.mwea.de
Email: kai.waehner@mwea.de
Twitter: @KaiWaehner
Blog: www.kai-waehner.de/blog

Activiti + Apache Camel = BPM beyond Web Services