



Business Process Digitalization and Cloud Computing

9. jBPM & Exam Roadmap

Andrea Morichetta, Phd

October 18, 2016

Computer Science Division

slides are partially based on an introduction of Kris Verlaenen

Table of contents

1. Presentations
2. jBPM

Presentations

Presentation contents

1. Understand the **possibilities** that the tool provide to the developers.
2. **Study the technology** considering both theory notions and practical one.
3. Keep in mind the **final goal** (automation of a business process).
4. Focus the main effort on the **implementation** and **integration** of custom and external services.
5. **Verification and analysis** aspect should be considered.
6. Give a **presentation** in class explaining what was studied and the possibility that the tool offer for the above issues.
7. Propose a short idea of **case study** to implement with the studied technology

Presentation Date

- Presentations should be between **45-60 minutes**.
- Every candidate of the group must present something.
- Presentations dates:
 - 21/11/16
 - 22/11/16
- The mark is structured according to a 40-60 balance considering 40% for the presentation and 60% for the project.
 - group 1: Samuel & Frederick (jBPM)
 - group 2: Kashif & Ilyas (Bonita)
 - group 3: Ceccotti & Senzacqua (Imix)
 - group 4: Serrani & Vici (Activiti)
 - group 5: Fioravanti (jBPM)

- jBPM (<https://www.jbpm.org/>)
- Activiti (<http://activiti.org/>)
- Imixs (http://www.imixs.org/sub_jee.html)
- Bonita (<http://www.bonitasoft.com/>)

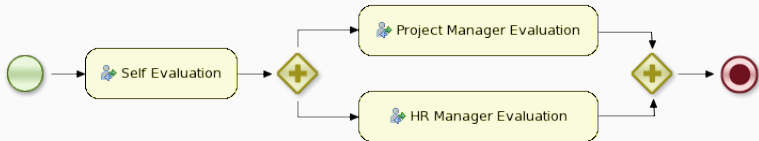
jBPM

What is jBPM?

- jBPM is a **flexible** Business Process Management (BPM) Suite.
- It makes the **bridge** between business analysts and developers.
- Traditional BPM engines have a focus that is limited to non-technical people only.
 - jBPM has a dual focus: it offers **process management features** in a way that both **business users** and **developers** like it.
- It allows you to model, execute, and monitor business processes throughout their life cycle.

Why jBPM?

- Visibility
- Monitoring
- Higher-level
- Continuous improvement
- Speed of development
- Increased agility

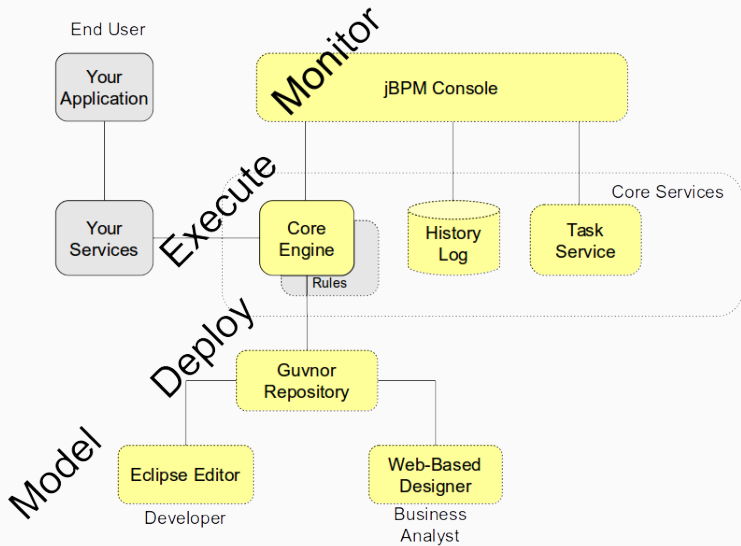


- **Eclipse-based and web-based editor** to support the graphical creation of your business processes (modeling, testing and debugging of processes).
- **Pluggable persistence and transactions based** on JPA JTA.
- **Pluggable human task service** based on WS-HumanTask for including tasks that need to be performed by human actors.
- **Management console** supporting process instance management, task lists and task form management, and reporting.
- Optional **process repository** to deploy your process (and other related knowledge).
- **History logging** (for querying monitoring analysis).
- **Integration** with Seam, Spring, OSGi, etc.
- **Remote API** to process engine as a service (REST, JMS, Remote Java API)



- **Design & Model:** Business Analyst
- **Deploy:** Developer
- **Execute:** End User
- **Monitor:** System Administrator
- **Optimize:** Business Analyst

BPM LifeCycle



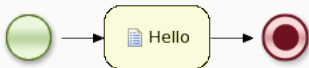
Key Characteristics

- **Open-source**, lightweight
- **Native** BPMN 2.0
- High-level business processes
- Business user & developer collaborate
- Advanced, **adaptive** processes
- **Modularized**, pluggable, standards
- Adaptive, ad-hoc, dynamic processes
- **Integration** and unification with business rules and complex event processing

Core engine

Core engine is a workflow engine in pure Java. Allows to execute business processes in a flexible manner.

- Solid, stable core engine for executing your process instances.
- Native support for the latest BPMN 2.0 specification.
- Strong focus on performance and scalability. Light-weight (can be deployed on a simple Java Runtime Environment).
- (Optional) pluggable persistence with a default JPA implementation.
- Pluggable transaction support with a default JTA implementation.
- Implemented as a generic process engine, so it can be extended to support new node types or other process languages.
- Listeners to be notified of various events.
- Ability to migrate running process instances to a new version of their process definition



```
<definitions ... >
  <process id="com.sample.bpmn.hello" name="Hello World" >
    <startEvent id="_1" name="StartProcess" />
    <sequenceFlow sourceRef="_1" targetRef="_2" />
    <scriptTask id="_2" name="Hello" >
      <script>System.out.println("Hello World");</script>
    </scriptTask>
    <sequenceFlow sourceRef="_2" targetRef="_3" />
    <endEvent id="_3" name="EndProcess" />
  </process>
</definitions>
```

Eclipse plugin

The screenshot shows the Eclipse IDE interface for editing a Drools BPMN file. The main editor displays a BPMN diagram for 'sample.bpmn'. The diagram starts with a Start event (green circle), followed by a 'Self Evaluation' task (yellow rounded rectangle). This is followed by a divergent gateway (yellow diamond with a plus sign). The flow then splits into two parallel paths: one leading to a 'R.Eval' task (yellow rounded rectangle) and another leading to a 'User Task' (yellow rounded rectangle). Both paths converge at a convergent gateway (yellow diamond with a plus sign), which then leads to an End event (red circle).

The Package Explorer on the left shows the project structure:

- Evaluation
 - src/main/java
 - com.sample
 - ProcessTest.java
 - src/main/rules
 - sample.bpmn
 - JRE System Library [jdk1.5.0_16]
 - Drools Library
 - src
 - test.log

The Properties view at the bottom shows the details for the selected 'User Task':

Property	Value
ActorId	
Comment	
Content	
Id	6
Name	User Task
On Entry Actions	
On Exit Actions	
Parameter Mapping	{}

- Developer-focused
- Features:
 - Wizards, runtimes, perspective
 - Graphical editor
 - Testing and debugging
 - Service repository, generators
 - Guvnor integration

Web Based Designer

File Edit Source Status: [Draft]

Shape Repository

- JBPM BPMN2
 - Activities
 - Task
 - Collapsed Subprocess
 - Expanded Subprocess
 - Collapsed Event-Subprocess
 - Event-Subprocess
 - Artifacts
 - Catching Intermediate Event
 - Connecting Objects
 - Data Objects
 - End Events
 - Gateways
 - Start Events
 - Start Event
 - Start Message Event

```
graph LR; Start((Start)) --> G1{XOR}; G1 --> Greetings[Greetings]; G1 --> G2{XOR}; Greetings --> G2; Valedictions[Valedictions] --> G2; G2 --> End((End));
```

Properties (BPMN-Diagram)

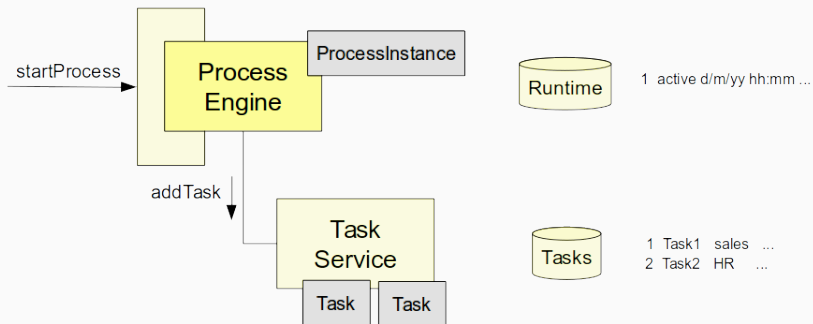
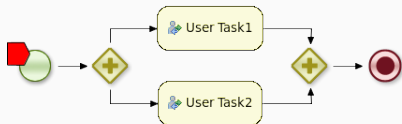
Name	Value
Often used	
Documentation	
Name	Demo2 Process
More Properties	
Author	
CreationDate	
ExpressionLang	http://www.w3.org/
Language	English
ModificationDate	
Namespaces	
Target Namespace	http://www.omg.org/
TypeLanguage	http://www.w3.org/
Variable Definit	
Version	1
executable	true
id	demo2process
package	demo2

Title: [demo2Process]

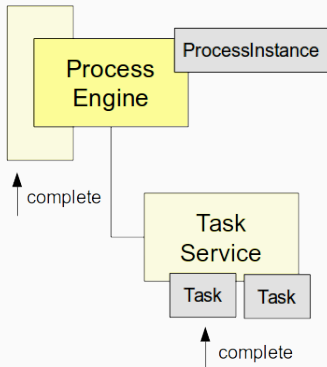
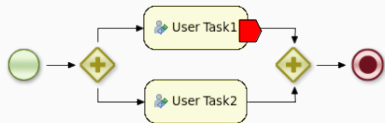
- demo2Process
- Other meta data ...
 - Subject:
 - Type:
 - External link:
 - Source:
- Version history ...

- Business User-focused
- Features
 - Graphical Editor
 - Visual validation
 - Service repository, generators
 - Guvnor integration
 - JavaScript, JSON

Services Implementation



Services Implementation

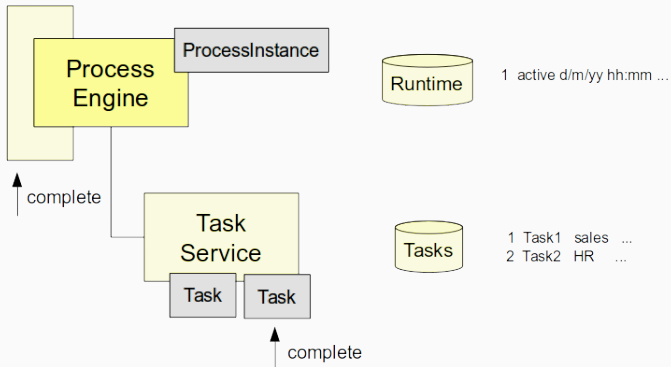
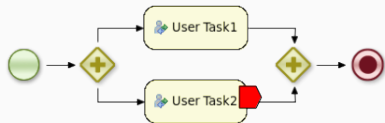


1 active d/m/yy hh:mm ...



1 Task1 sales ...
2 Task2 HR ...

Services Implementation



- Web-based management
- Business user
- Features
 - Process instance management
 - User task lists / forms
 - Reporting
 - History log
 - Management Console
- **Process instance management:** the ability to start new process instances, get a list of running process instances, visually inspect the state of a specific process instances.
- **Human task management:** being able to get a list of all your current tasks, and completing tasks on your task list (using customizable task forms).

Business Activity Monitoring (BAM)

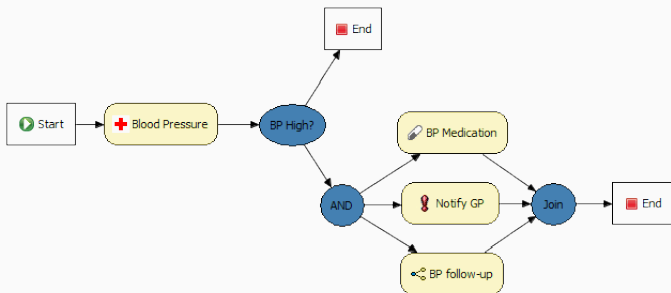
- Key features:
 - Visual configuration of dashboards (Drag'n'drop).
 - Graphical representation of KPIs (Key Performance Indicators).
 - Configuration of interactive report tables.
 - Data export to Excel and CSV format.
 - Filtering and search, both in-memory or SQL based.
 - Data extraction from external systems, through different protocols.
 - Granular access control for different user profiles.
 - Look'n'feel customization tools.
 - Pluggable chart library architecture.
- Target users:
 - Managers / Business owners. Consumer of dashboards and reports.
 - IT / System architects. Connectivity and data extraction.
 - Analysts. Dashboard composition & configuration.

Integration is given by:

- ESB
- Human Task Service
- Domain-specific Process

Domain-specific Processes

- Extend palette with domain-specific, declarative service nodes
 - define input / output parameters
 - runtime binding



- Why?
 - Domain-specific
 - Declarative
 - High-level
 - Context-defined
- Other examples
 - Human task
 - Java method
 - WS, Rest
 - Email, Twitter
 - FTP, RSS, Jabber
 - Finder, Exec, Archive
 - Google Calendar
 - YOUR SERVICES !

- Process instance migration
- JUnit testing
- Spring
- OSGi

Demo

Questions?