jBPM 4

Jiří Pechanec
JBoss QE
Supervisor, Red Hat
Sep 11th, 2009

Agenda

- Introduction into jBPM
- jPDL Activities
- API
- Tools
- Demonstration
- Questions

What is BPM

- Business process modeling (BPM) in systems engineering and software engineering is the activity of representing processes of an enterprise, so that the current process may be analyzed and improved in future. BPM is typically performed by business analysts and managers who are seeking to improve process efficiency and quality. - Wikipedia.com

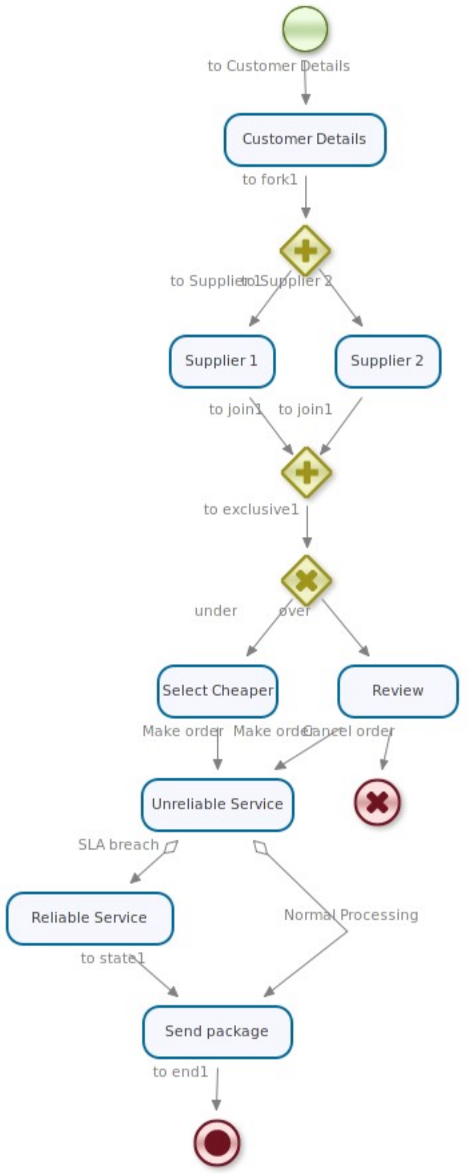
What is jBPM

- jBPM is an extensible and flexible process engine that can run as a standalone server or embedded in any Java application
- Helps to bridge the gap between developer and business analyst
- Promotes graph-oriented programming

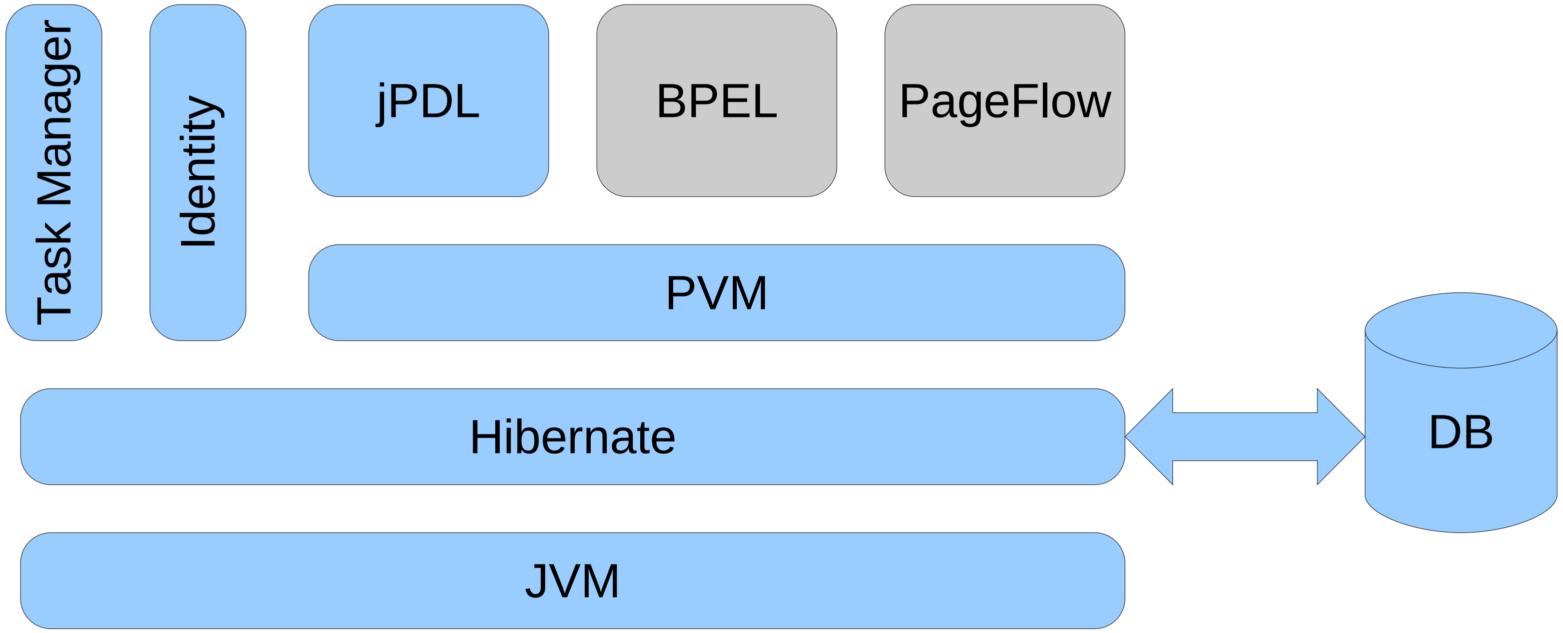
Main features

- Embeddable
  - Can be used as library in any Java application
- Extensible
  - New languages can be developed
  - Out-of-box languages can be enriched
- Accompanied by useful tools
  - Web console
  - Eclipse designer
  - Web designer (new in 4.1)

Example of business process



Overall Architecture



Process

- Process definition
  - Recipe or template expressed in XML with corresponding graphical representation
  - Versioned
- Process instance
  - An execution of the given process definition
  - Contains pointer to current activity and set of process variables
- Wait state
  - A state in which the process is persisted into the database

jPDL Activities (1/2)

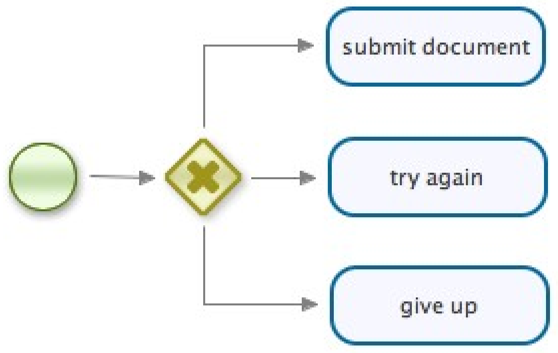
- Control Flow
  - start
  - end
  - state
  - decision
  - fork/join
  - task
  - sub-process
  - custom
- Automatic
  - java
  - script
  - hql
  - sql
  - mail

jPDL Activities (2/2)

- Incubating
  - timer
  - group

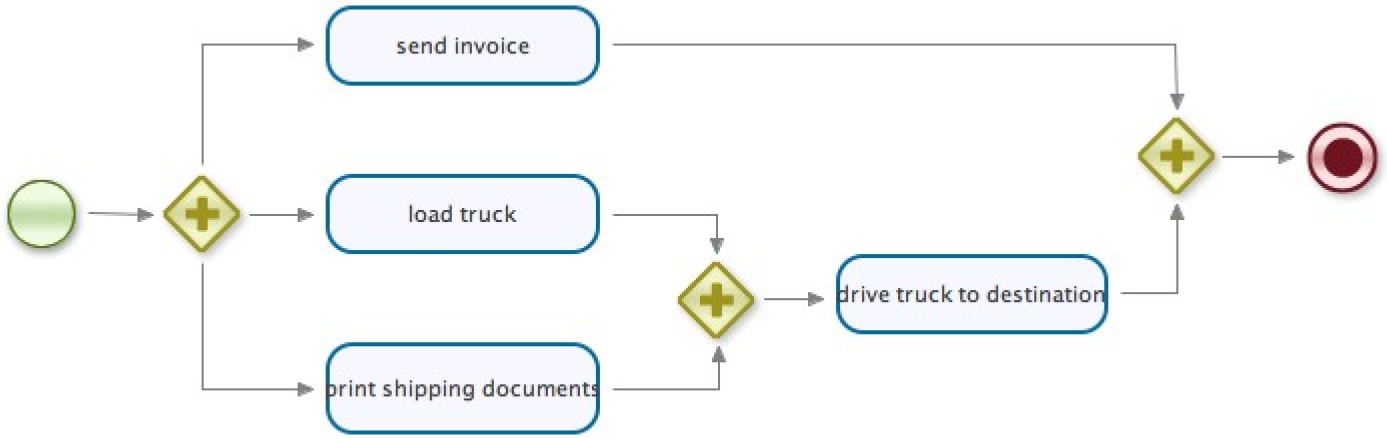
Decision

- Directs process based on predefined condition
- The condition can be expressed either in process definition or can be coded into Java class



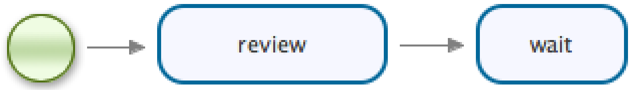
Concurrency

- Parts of the process can be executed in parallel
- The execution does not need to be done concurrently
- The process do not need to wait for all brnaches to be completed



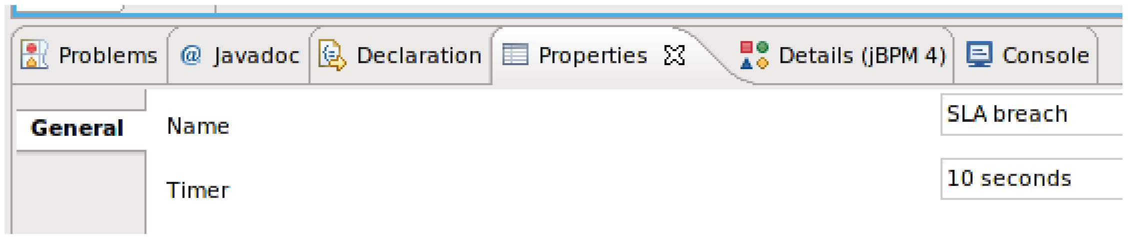
Task

- Process describes work to be done in cooperation between machines and humans
- Task is a unit of work that is executed by human
- Task is assigned to an user and can have associated a set of variables that can be modified by the assignee
- The task is visible inside web console and assignee can be notified by an email



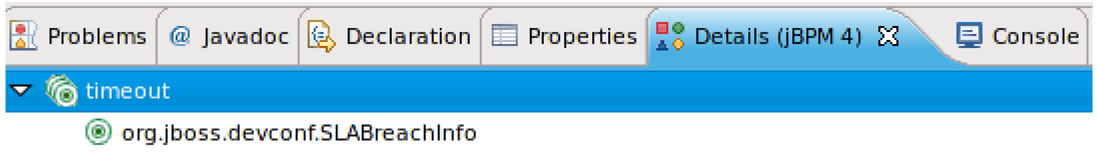
Timer

- Still under redesign
- Associated with outgoing node from activity – will continue the process if activity processing is stuck
- Associated with event on wait state
- Timer can be fired repeatedly



Name: SLA breach

Timer: 10 seconds



timeout

org.jboss.devconf.SLABreachInfo

Asynchronous execution

- Default behavior – execute process in caller thread
- Activity can have attribute continue="async"
- An activity and the rest of he process is executed in special thread dedicated for the execution (Job Executor)
- With default behavior the process is executed inside one transaction
- The asynchronous execution demarcates the transaction boundaries

Client API (1/2)

- Based on Service pattern
  - Repository Service
  - Execution Service
  - Task Service
  - History Service
  - Management Service

Client API (2/2)

```java
Configuration jbpmConfiguration = new Configuration();

ProcessEngine engine = jbpmConfiguration.buildProcessEngine();

NewDeployment deployment = engine.getRepositoryService().createDeployment();
deployment.addResourceFromClasspath("OrderProcess.jpdl.xml");
deployment.addResourceFromClasspath("OrderProcess.png");
deployment.addResourceFromClasspath("OrderProcess.ftl");
deployment.deploy();
```

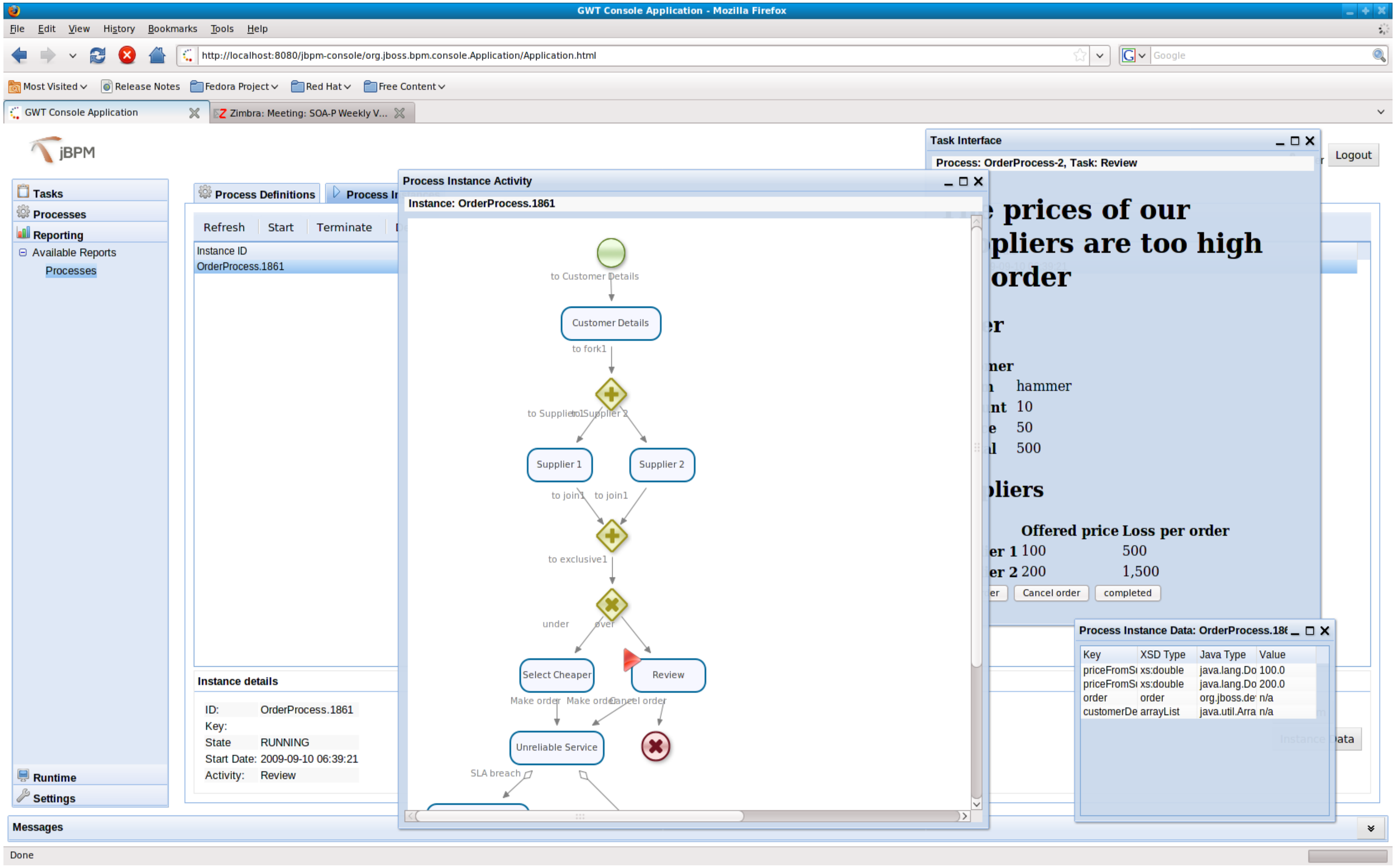
```java
Configuration jbpmConfiguration = new Configuration();

ProcessEngine engine = jbpmConfiguration.buildProcessEngine();

Map<String, Object> processVars = new HashMap<String, Object>();
Order order = new Order("ACME", "hammer", 10, 150);
processVars.put("order", order);
ProcessInstance process = engine.getExecutionService()
        .startProcessInstanceByKey("OrderProcess", processVars);
System.out.println("Executing process id=" + process.getId());
```

```java
Configuration jbpmConfiguration = new Configuration();

ProcessEngine engine = jbpmConfiguration.buildProcessEngine();

HistoryProcessInstance processHistory = engine.getHistoryService()
        .createHistoryProcessInstanceQuery().processInstanceId(
                processId).uniqueResult();
```

Console (1/2)

- Process management
- Process tracking and debugging
- Task management
- Business Activity Monitoring

Console (2/2)



Instance details

ID:	OrderProcess.1861
Key:	
State	RUNNING
Start Date:	2009-09-10 06:39:21
Activity:	Review

Process: OrderProcess-2, Task: Review

e prices of our
ppliers are too high
order

er

ner

n	hammer
nt	10
e	50
l	500

pliers

	Offered price	Loss per order
er 1	100	500
er 2	200	1,500

er | Cancel order | completed

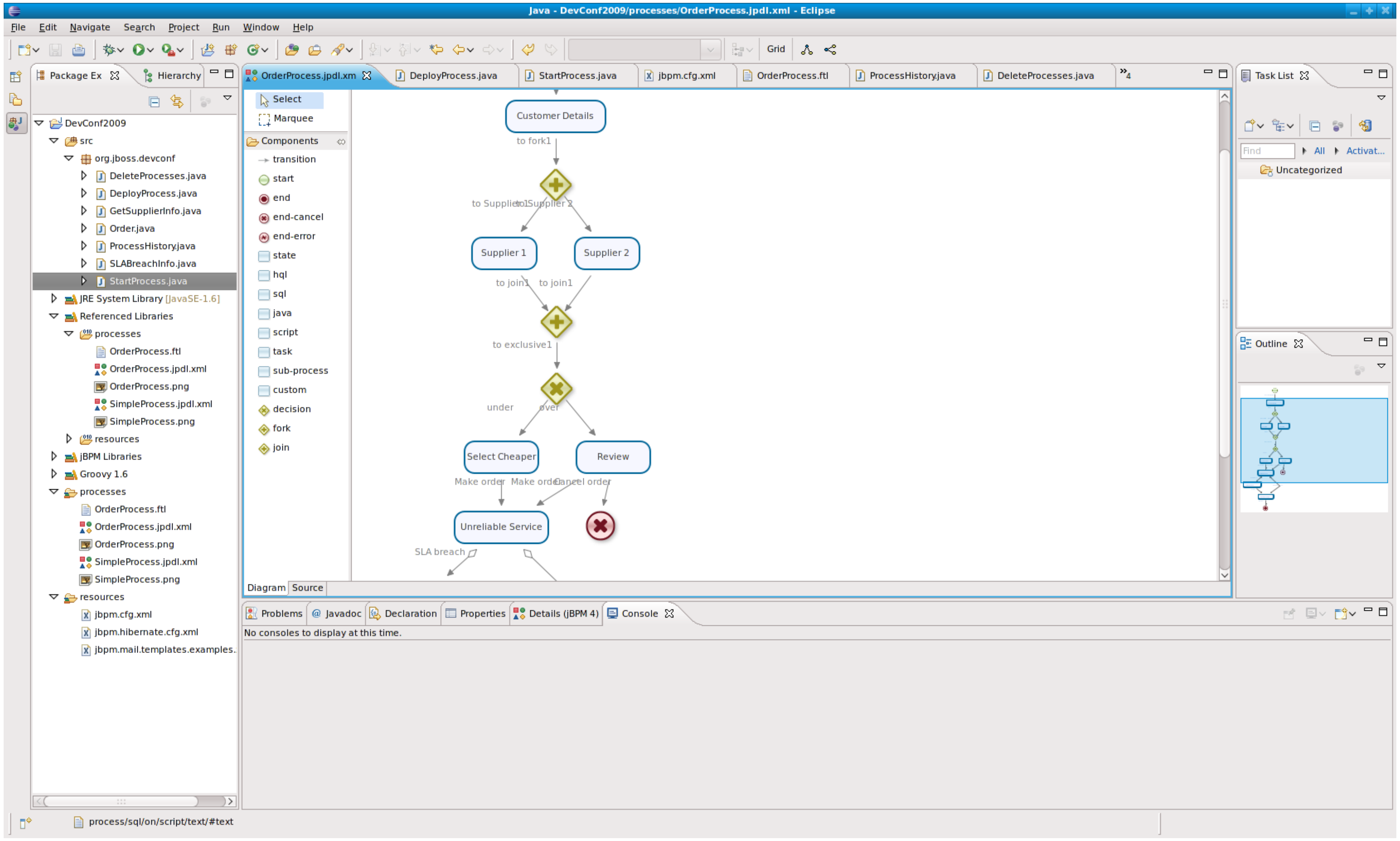
Process Instance Data: OrderProcess.186

Key	XSD Type	Java Type	Value
priceFromSu	xs:double	java.lang.Do	100.0
priceFromSu	xs:double	java.lang.Do	200.0
order	order	org.jboss.de	n/a
customerDe	arrayList	java.util.Arra	n/a

Eclipse Designer (1/2)

- Graphical process editor
  - BPMN 2.0 notation
  - Auto-arrange
- Text process editor
- Process deployer

Eclipse Designer (2/2)

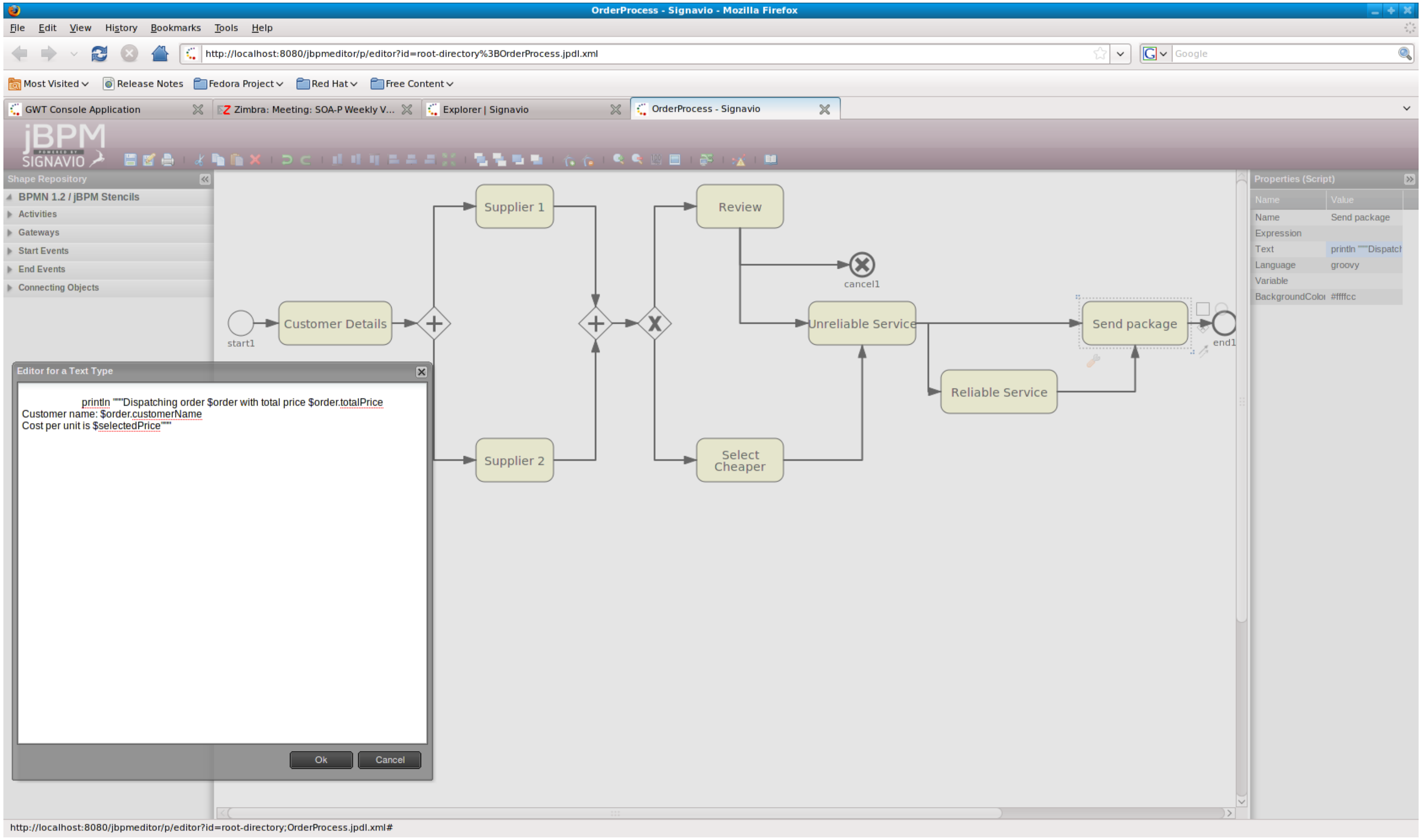


No consoles to display at this time.

Web Process Designer (1/2)

- Since jBPM 4.1
- Rich Internet Application
- Graphical process editor
- BPMN 1.2 notation
- Allows to remotely edit process definitions
- Only sub-set ob jPDL activities supported

Web Process Designer (2/2)


OrderProcess - Signavio - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/jbpmeditor/p/editor?id=root-directory%3BOrderProcess.jpdl.xml
Most Visited Release Notes Fedora Project Red Hat Free Content
GWT Console Application
Zimbra: Meeting: SOA-P Weekly V...
Explorer | Signavio
OrderProcess - Signavio
jBPM SIGNAVIO
Shape Repository
BPMN 1.2 / jBPM Stencils
Activities
Gateways
Start Events
End Events
Connecting Objects
start1
Customer Details
Supplier 1
Supplier 2
Review
cancel1
Unreliable Service
Select Cheaper
Reliable Service
Send package
end1
Ok
Cancel
http://localhost:8080/jbpmeditor/p/editor?id=root-directory;OrderProcess.jpdl.xml#

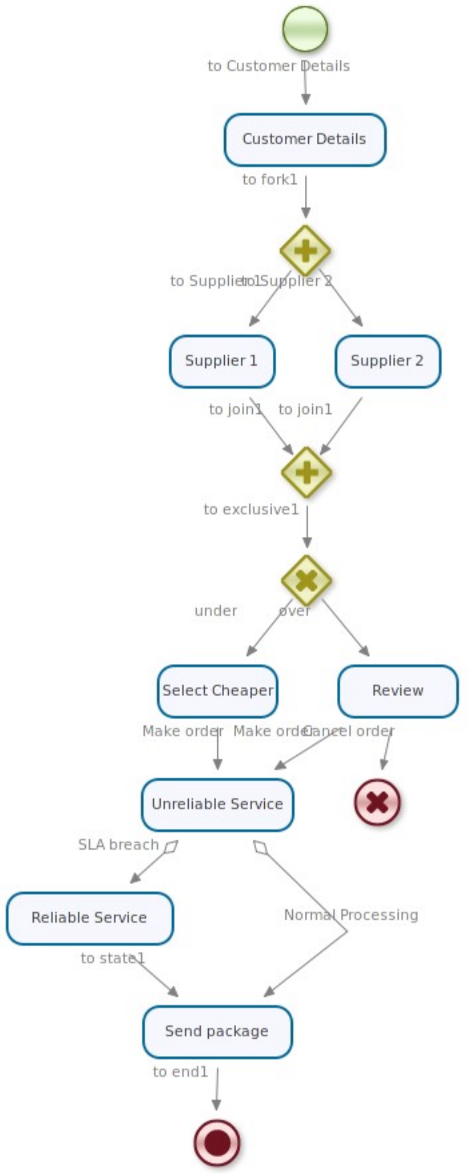
Editor for a Text Type

```
            println """Dispatching order $order with total price $order.totalPrice
Customer name: $order.customerName
Cost per unit is $selectedPrice"""
```

Properties (Script)

Name	Value
Name	Send package
Expression	
Text	println """Dispatch
Language	groovy
Variable	
BackgroundColor	#ffffcc

Putting it all-together



Resources

- Project page - http://jboss.org/jbossjbpm
- Tom Bayens's blog - http://processdevelopments.blogspot.com/
- Seven Forms of Business Process Management With JBoss jBPM - http://java.dzone.com/articles/seven-forms-business-process-m