

jBPM####

.....	V
1. ##	1
1.1. ##	1
1.1.1. ##Drools Flow Eclipse #####BPMN2##	1
1.1.2. jBPM5 Eclipse #####BPMN2##	2
1.1.3. ##web#####-Oryx###	3
1.2. ##	4
1.3. ##	4
1.3.1. #####	4
1.3.2. #####	6
1.4. ##	7
1.4.1. ##web####	8
2. ###	11
2.1. ##	11
2.2. #####	11
2.3. ##Demo	11
2.4. ##Eclipse##	12
2.5. ##web#####	12
2.6. ##Guvnor##	13
2.7. #####?	13
3. BPMN 2.0	15
3.1. #####2.0##	15
3.2. ##	19
3.3. #####/##	19
4. API	23
4.1. jBPM API	24
4.1.1. Knowledge Base	24
4.1.2. Session	24
4.1.3. ##	26
4.2. Knowledge-based API	28
5. #####	29
5.1. #####	29
5.1.1. ##	31
5.2. #####	32
5.2.1. #####	32
5.2.2. #####jBPM##	33
5.2.3. #####	34
5.2.4. #####	35
5.3. #####	36
5.3.1. Eclipse##	36
5.3.2. ##web#####	36
6. #####	37
6.1. Introduction	37
6.2. #####	38

6.2.1. #####	38
6.2.2. #####	38
6.2.3. #####	39
6.2.4. #####	40
7. ###	43
7.1. #####	43
7.1.1. #####	43
7.1.2. ###	43
7.1.3. #####	43
7.1.4. ##	46
7.2. #####	47
7.3. #####	47
7.3.1. #####	47
8. ###	49
8.1. ##	49
8.2. #####	49
8.2.1. #####	50
8.2.2. #####	58
8.2.3. ##	60
8.3. #####/#####	64
##	67



1 #

####jBPM#####

1.1.

jBPM#####Eclipse#####web#####3#BPMN2#
####

1.1.1. ##Drools Flow Eclipse ####BPMN2##

Eclipse#####BPMN2###

1 #

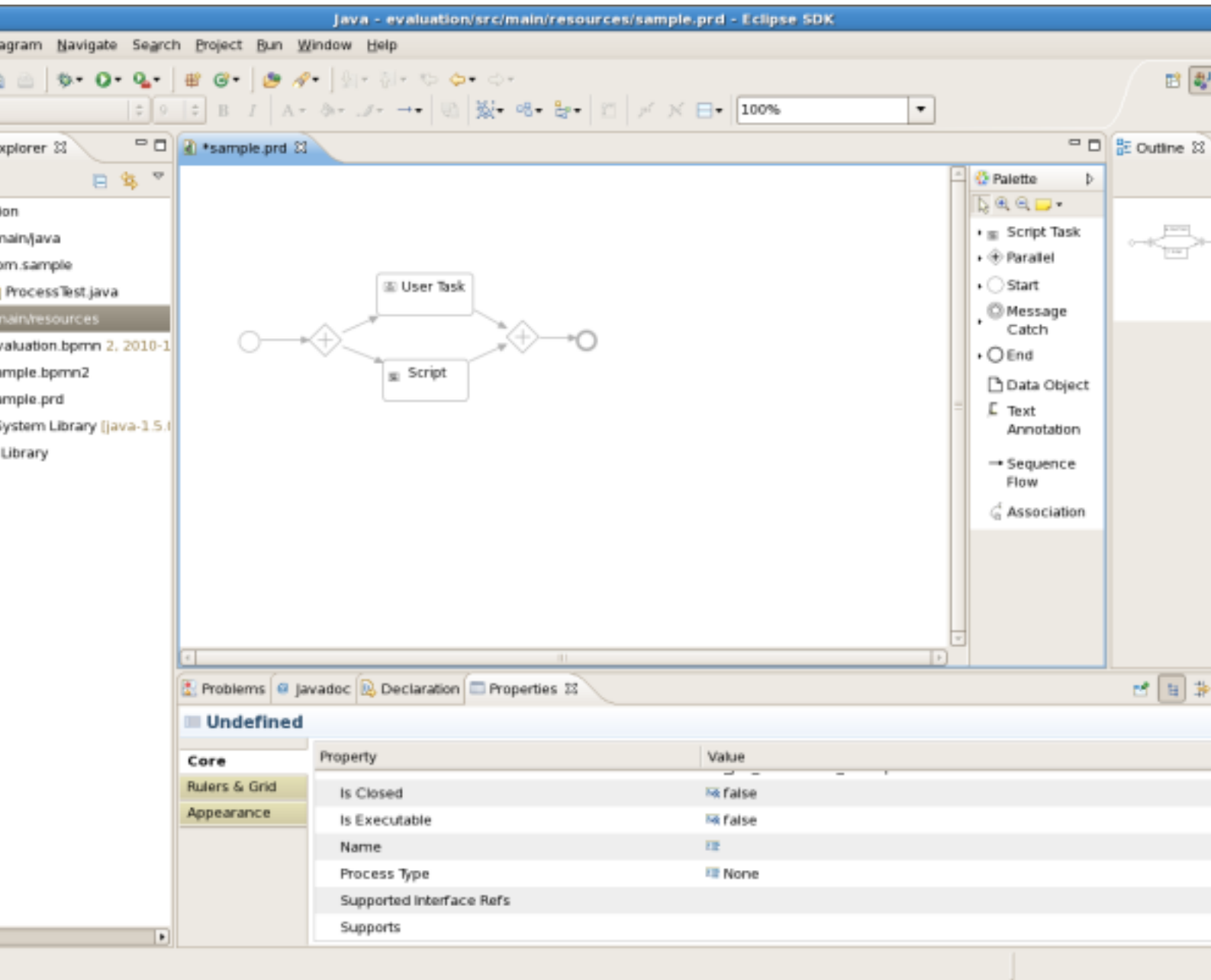
The screenshot displays the Eclipse IDE interface with the 'Evaluation.bpmn' file open. The main canvas shows a BPMN process flow: a green start event leads to a 'Self Evaluation' task, which then branches into two parallel tasks, 'HR Evaluation' and 'PM Evaluation'. These two tasks merge at a converge gateway, which then leads to a red end event. The left sidebar shows the 'Properties' window for the selected 'Self Evaluation' task.

Property	Value
ActorId	#{employee}
Comment	Please perform a self-evaluation.
Content	
Id	2
MetaData	{width=135, height=48, y=56, UniqueId=_2, x=96}
Name	Self Evaluation
On Entry Actions	
On Exit Actions	

1.1. Drools Flow #####BPMN2##

1.1.2. jBPM5 Eclipse #####BPMN2##

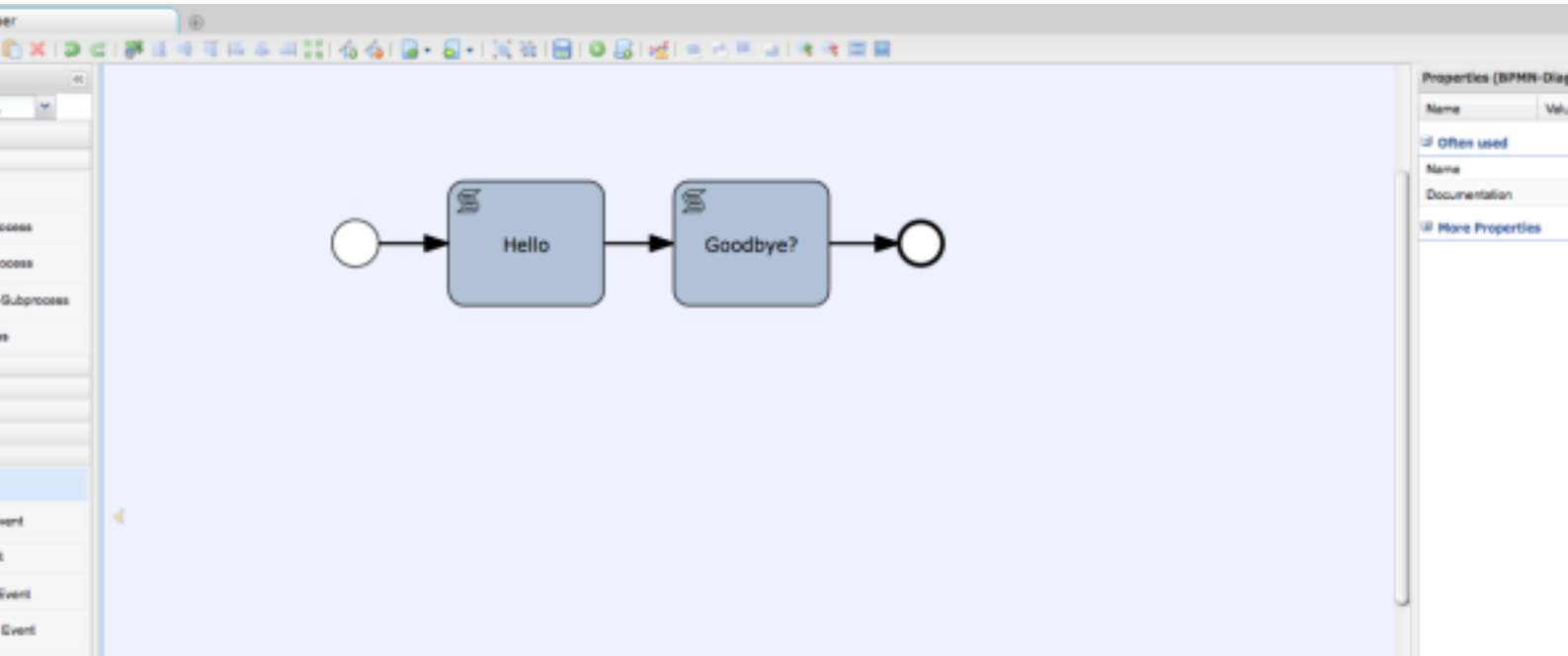
##Eclipse#####BPMN2#####BPMN2###



1.2. jBPM#####BPMN2##

1.1.3. ##web#####-Oryx###

Oryx#####web#####Guvnor#####BPMN2###



1.3. ##web#Oryx #####BPMN2##

1.2.

Guvnor###knowledge#####Guvnor#####web#####Oryx###
###Guvnor ####Guvnor#####

1.3.

1.3.1.

#####

- # BPMN2 ####
- #####
- #####
- #####

#####BPMN 2.0#####BPMN 2.0###
###:

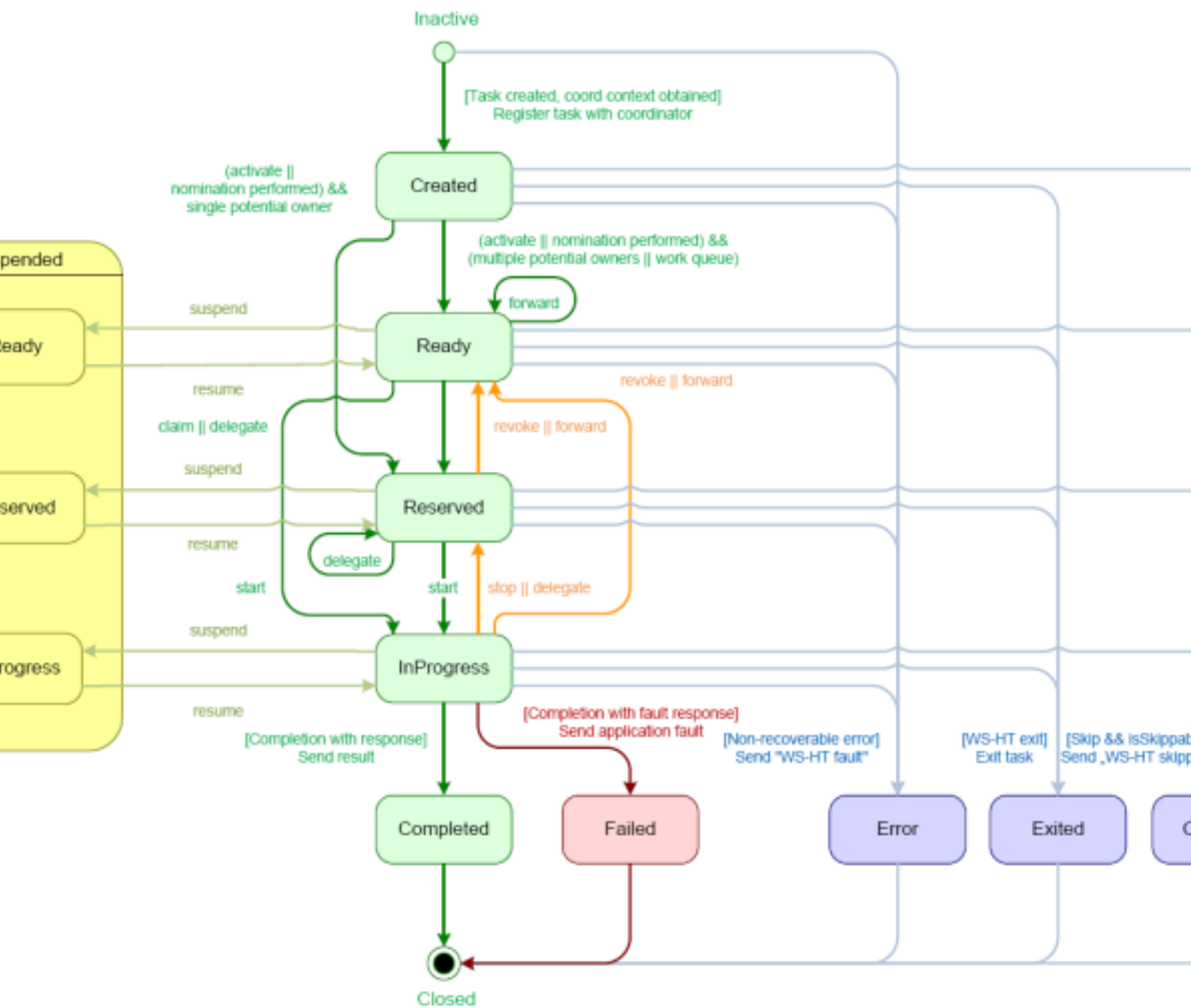
- ###
- Events
 - Start Event (None, Conditional, Signal, Message, Timer)
 - End Event (None, Terminate, Error, Escalation, Signal, Message, Compensation)

- Intermediate Catch Event (Signal, Timer, Conditional, Message)
- Intermediate Throw Event (None, Signal, Escalation, Message, Compensation)
- Non-interrupting Boundary Event (Escalation, Timer)
- Interrupting Boundary Event (Escalation, Error, Timer, Compensation)
- ##
 - Script Task (Java or MVEL expression language)
 - Task
 - Service Task
 - User Task
 - Business Rule Task
 - Manual Task
 - Send Task
 - Receive Task
 - Reusable Sub-Process (Call Activity)
 - Embedded Sub-Process
 - Ad-Hoc Sub-Process
 - Data-Object
- Gateways
 - Diverging
 - Exclusive (Java, MVEL or XPath expression language)
 - Inclusive (Java, MVEL or XPath expression language)
 - Parallel
 - Event-Based
 - Converging
 - Exclusive
 - Parallel

- Lanes
- *Data*
 - Java type language
 - Process properties
 - Embedded Sub-Process properties
 - Activity properties
- *Connecting objects*
 - Sequence flow

1.3.2.

#####BPM#####/#####
#####jBPM## WS-HT##. #####WS-HT#####



1.4. WS-HT#####

#####.

1.4. ##

#####

#####

1 #

1.4.1. ##web####

###web#####/###/####

The screenshot displays a BPM system interface. At the top left, the text "BPM" is visible. Below it, there are navigation tabs for "Process Definitions" and "Process Instances". The "Process Instances" tab is active, showing a table with the following data:

Instance ID	State	Start Date
1	RUNNING	2009-09-11 18:23:37

Below the table, there is a "Process Instance Activity" window for instance 1. The activity diagram shows a flow starting with a green circle, followed by a red triangle, then a yellow rounded rectangle labeled "Self Evaluation". This leads to a yellow diamond with a plus sign, which branches into two paths: "Project Manager Evaluation" and "HR Manager Evaluation". Both paths converge at another yellow diamond with a plus sign, which then leads to a red circle with a white center, representing the end of the process.

At the bottom of the interface, there is a summary box for the instance:

State: RUNNING
Start Date: 2009-09-11 18:23:37
Activity:

1.5.

... ##### ...

ks


Management


sk Lists

Processes

orting

ings

 Group Tasks

 Personal Tasks

Refresh | View | Release

Priority	Process	Task Name	Due Date
0		Performance Evaluation	

Task Interface

Process: , Task: Performance Evaluation

Employee evaluation

As part of your performance evaluation, you have to do a self-assessment.

Please fill in the following evaluation form:

Rate the overall performance:

Outstanding

Check any that apply:

- Displaying initiative
- Thriving on change
- Good communication skills

Task details

ID: 1

Process:

Name: Performance Evaluation

Assignee: krisv

Description:

1.6.

... ####

jbpm

Processes

Monitoring

Process Reports

Processes

gs

Process Reports

Available Reports ▾



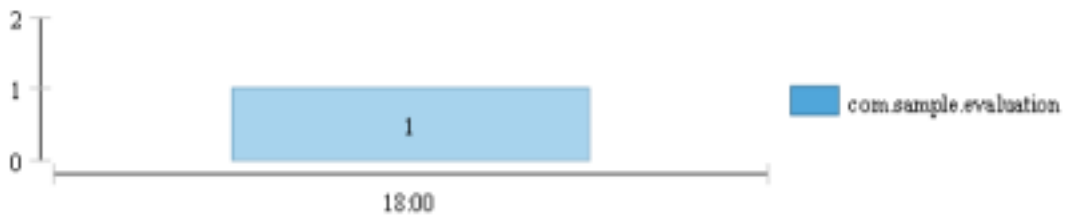
11 september

Business Activity Monitoring

Process com.sample.evaluation

Process Definition Id: com.sample.evaluation
Total number of instances: 1
Number of instances last 24h: 1
Number of active instances: 1

Start Process Instances



Id	Start	End
1	11 september 2009 18:23:37	

2 #

#####jBPM#####demo#####jBPM###

2.1.

#####Java JDK 1.5+ (set as JAVA_HOME), Ant 1.7+ . #####, #####:

Java: <http://java.sun.com/javase/downloads/index.jsp>

Ant: <http://ant.apache.org/bindownload.cgi>

2.2.

#####: jBPM-{version}-install.zip

#####.

<http://hudson.jboss.org/hudson/job/jBPM5/lastSuccessfulBuild/artifact/target/>

2.3. ##Demo

#####:

```
ant install.demo
```

##:

- ## JBoss AS
- ## Eclipse
- ## Drools Guvnor # JBoss AS
- ## Oryx Designer # JBoss AS
- ## jBPM gwt-console # JBoss AS
- ##jBPM Eclipse plugin
- ##Drools Eclipse plugin

#####(#####Eclipse). #####
##

##demo setup #####, #####:

```
ant start.demo
```

##:

- ##H2 ###
- ## JBoss AS
- ## Eclipse
- #####

#####Eclipse###Guvnor # gwt-console.

2.4. ##Eclipse##

[http://people.redhat.com/kverlaen/install-eclipse-jbpm.swf]#####demo#####

- #####
 - #####BPMN2##
 - #####Java#
- #####

##Eclipse#, ##### ("File -> Import ..." ##### "Existing Projects into Workspace")(#jbpm-installer/sample/evaluation ###). #####BPMN2#####Java#####eclipse###BPMN2#####
#####com.sample package##ProcessTest.java##### "Run As - Java Application"#####. #####
#####"krisv"#####jbpm#####

#####jBMP#####HelloWorld #BPMN2#####Java#####
#"File - New ... - Project ..." #";jBPM" ###, ## "jBPM project" ### "Next". #####"Finish".
#####"sample.bpmn"#####"com.sample.ProcessTest" Java #####. #####BPMN2##. ##
ProcessTest.java ## "Run As - Java Application"#####. #####"Hello World" .

2.5. ##web#####

#####

<http://localhost:8080/jbpm-console>

##krisv / krisv ## ###/## ##. [jbpm console demo](http://people.redhat.com/kverlaen/install-gwt-console-jbpm.swf) [http://people.redhat.com/kverlaen/install-gwt-console-jbpm.swf] #####. ###:

- #####

- #####
- #####
- #####
- #####

2.6. ##Guvnor##

#####Guvnor#####. ##### ###

- #####Guvnor#####BPMN2#####"Create New - New File" #####bpmn#####.
- #####Google Chrome#.
- #####package ##jBPM5#####
- #####BPMN2#####jBPM###

##Drools Guvnor:

<http://localhost:8080/drools-guvnor>

#####(#####). [Guvnor demo\[TODO\]](#) [TODO] #####.###:

- #####
- ###web#####
- #####session#package#

#####Drools Guvnor##.

#####

```
ant stop.demo
```

#####.

2.7. #####?

#####jBPM##.

Email: jbpm-dev@jboss.org

IRC: #jbpm at irc.codehaus.org

[jBPM#####](http://community.jboss.org/en/jbpm?view=discussions) [<http://community.jboss.org/en/jbpm?view=discussions>]

3 # BPMN 2.0

3.1. #####2.0##

BPMN#####
##

#####(BPMN)2.0#####OMG#####(# BPMN 1.x)#####
#####XML###

jBPM5##BPMN 2.0#####jBPM5#####BPMN 2.0#####BPMN 2.0##
#####choreographies#collaboration#####jBPM#####

jBPM5####BPMN 2.0#####BPMN 2.0##"Common Executable"#####

- ###
 - Events
 - Start Event (None, Conditional, Signal, Message, Timer)
 - End Event (None, Terminate, Error, Escalation, Signal, Message, Compensation)
 - Intermediate Catch Event (Signal, Timer, Conditional, Message)
 - Intermediate Throw Event (None, Signal, Escalation, Message, Compensation)
 - Non-interrupting Boundary Event (Escalation, Timer)
 - Interrupting Boundary Event (Escalation, Error, Timer, Compensation)
 - Activities
 - Script Task
 - Task
 - Service Task
 - User Task
 - Business Rule Task
 - Manual Task
 - Send Task

3 # BPMN 2.0

- Receive Task
- Reusable Sub-Process (Call Activity)
- Embedded Sub-Process
- Ad-Hoc Sub-Process
- Data-Object
- Gateways
 - Diverging
 - Exclusive
 - Inclusive
 - Parallel
 - Event-Based
 - Converging
 - Exclusive
 - Parallel
- Lanes
- *Data*
 - Java type language
 - Process properties
 - Embedded Sub-Process properties
 - Activity properties
- *Connecting objects*
 - Sequence flow

#####"Hello World"BPMN 2.0#####"Hello World"##



##BPMN 2.0#####

```

<?xml version="1.0" encoding="UTF-8"?>
<definitions id="Definition"
  targetNamespace="http://www.example.org/MinimalExample"
  typeLanguage="http://www.java.com/javaTypes"
  expressionLanguage="http://www.mvel.org/2.0"
  xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL"
  xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
  xs:schemaLocation="http://www.omg.org/spec/BPMN/20100524/MODEL BPMN20.xsd"
  xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI"
  xmlns:dc="http://www.omg.org/spec/DD/20100524/DC"
  xmlns:di="http://www.omg.org/spec/DD/20100524/DI"
  xmlns:tns="http://www.jboss.org/drools">

<process processType="Private" isExecutable="true" id="com.sample.HelloWorld" name="Hello
World" >

  <!-- nodes -->
  <startEvent id="_1" name="StartProcess" />
  <scriptTask id="_2" name="Hello" >
    <script>System.out.println("Hello World");</script>
  </scriptTask>
  <endEvent id="_3" name="EndProcess" >
    <terminateEventDefinition/>
  </endEvent>

  <!-- connections -->
  <sequenceFlow id="_1-2" sourceRef="_1" targetRef="_2" />
  <sequenceFlow id="_2-3" sourceRef="_2" targetRef="_3" />

</process>

<bpmndi:BPMNDiagram>
  <bpmndi:BPMNPlane bpmnElement="Minimal" >
    <bpmndi:BPMNShape bpmnElement="_1" >
      <dc:Bounds x="15" y="91" width="48" height="48" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="_2" >
      <dc:Bounds x="95" y="88" width="83" height="48" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="_3" >
      <dc:Bounds x="258" y="86" width="48" height="48" />
    </bpmndi:BPMNShape>
    <bpmndi:BPMNEdge bpmnElement="_1-2" >

```

3 # BPMN 2.0

```
<di:waypoint x="39" y="115" />
<di:waypoint x="75" y="46" />
<di:waypoint x="136" y="112" />
</bpmndi:BPMNEdge>
<bpmndi:BPMNEdge bpmnElement="_2_3" >
  <di:waypoint x="136" y="112" />
  <di:waypoint x="240" y="240" />
  <di:waypoint x="282" y="110" />
</bpmndi:BPMNEdge>
</bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>

</definitions>
```

#####BPMN 2.0#####

- ##Eclipse Drools#####Drools5.1#####BPMN 2.0#####
#####BPMN 2.0#####BPMN 2.0#####
#####
- Oryx#####web#####BPMN 2.0#####Guvnor#####Oryx#####Oryx#####/##BPMN
2.0#####Oryx##BPMN 2.0#####
- ##Eclipse BPMN2#####BPMN2.0#####BPMN2.0#
#####BPMN2#####.bpmn2#####.prd#####.prd#####
- #####XML####BPMN2.0#####

#####BPMN2#####knowledge base ...

```
private static KnowledgeBase createKnowledgeBase() throws Exception {
    KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();
    kbuilder.add(ResourceFactory.newClassPathResource("sample.bpmn2"), ResourceType.BPMN2);
    return kbuilder.newKnowledgeBase();
}
```

... ##### ...

```
KnowledgeBase kbase = createKnowledgeBase();
StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();
ksession.startProcess("com.sample.HelloWorld");
```


3.2.

BPMN 2.0#####

jbpm-bpmn2#####junit tests#####
#####

3.3. #####/##

3.1.

##	#####	#####	####
definitions		rootElement BPMNDiagram	
process	processType isExecutable name id	property laneSet flowElement	packageName
sequenceFlow	sourceRef targetRef isImmediate name id	conditionExpression	endpoints
interface	name id	operation	
operation	name id	inMessageRef	
laneSet		lane	
lane	name id	flowNodeRef	
Events			
startEvent	name id	dataOutput dataOutputAssociation outputSet eventDefinition	x y width height
endEvent	name id	dataInput dataInputAssociation inputSet eventDefinition	x y width height
intermediateCatchEvent	name id	dataOutput dataOutputAssociation outputSet eventDefinition	x y width height
intermediateThrowEvent	name id	dataInput dataInputAssociation inputSet eventDefinition	x y width height

3 # BPMN 2.0

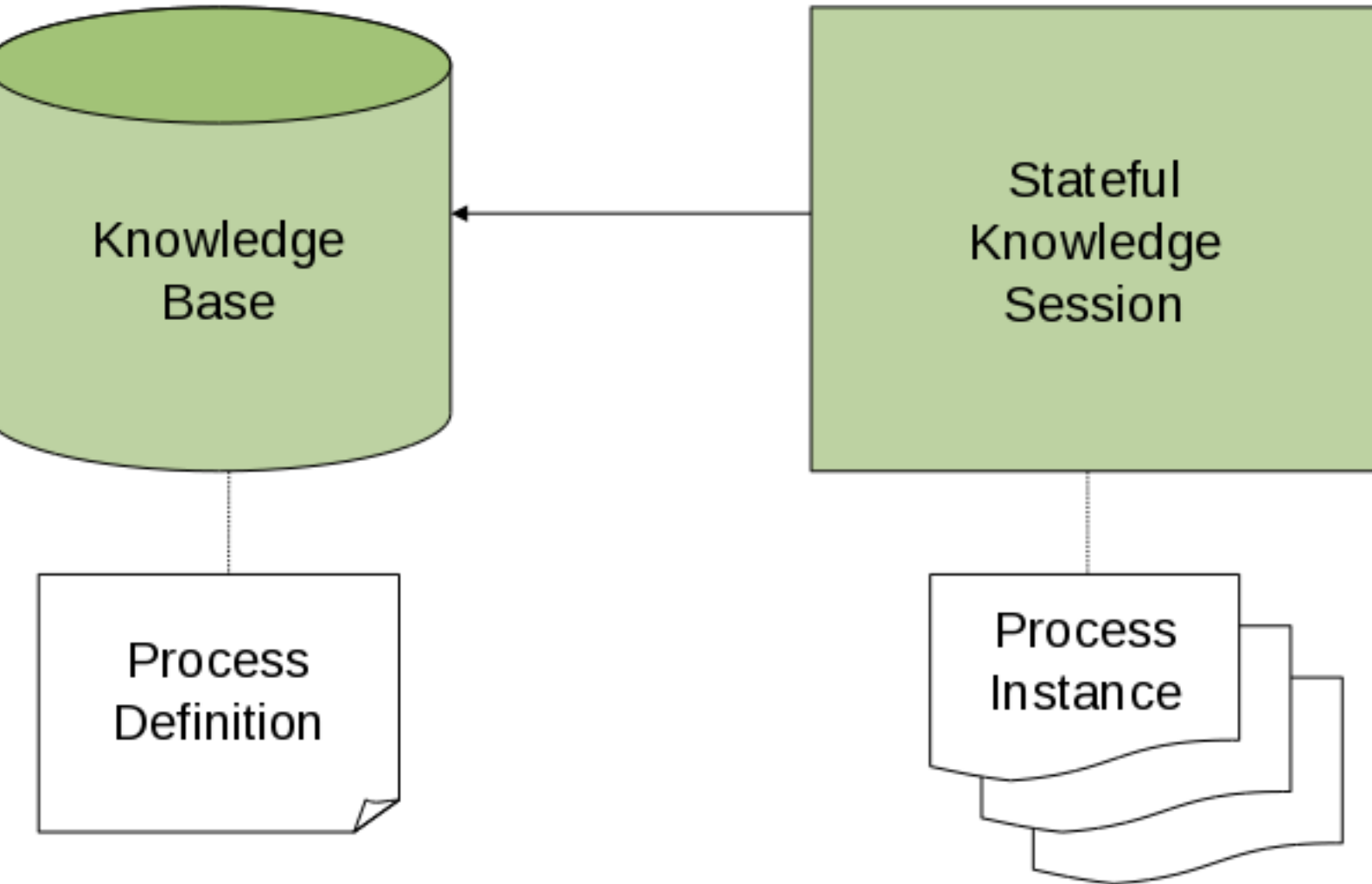
##	#####	#####	####
boundaryEvent	cancelActivity attachedToRef name id	eventDefinition	x y width height
terminateEventDefinition			
cancelEventDefinition			
compensateEventDefinition	activityRef	documentation extensionElements	
conditionalEventDefinition		condition	
errorEventDefinition	errorRef		
error	errorCode id		
escalationEventDefinition	escalationRef		
escalation	escalationCode id		
messageEventDefinition	messageRef		
message	itemRef id		
signalEventDefinition	signalRef		
timerEventDefinition		timeCycle	
Activities			
task	name id	ioSpecification dataInputAssociation dataOutputAssociation	taskName x y width height
scriptTask	scriptFormat name id	script	x y width height
script		text[mixed content]	
userTask	name id	ioSpecification dataInputAssociation dataOutputAssociation resourceRole	x y width height
potentialOwner		resourceAssignmentExpression	
resourceAssignmentExpression		expression	
businessRuleTask	name id		x y width height ruleFlowGroup
manualTask	name id		x y width height
sendTask	messageRef name id	ioSpecification dataInputAssociation	x y width height
receiveTask	messageRef name id	ioSpecification dataOutputAssociation	x y width height

##	####	####	####
serviceTask	operationRef name id	ioSpecification dataInputAssociation dataOutputAssociation	x y width height
subProcess	name id	flowElement property loopCharacteristics	x y width height
adHocSubProcess	cancelRemainingInstances name id	completionCondition flowElement property	x y width height
callActivity	calledElement name id	ioSpecification dataInputAssociation dataOutputAssociation	x y width height waitForCompletion independent
multiInstanceLoopCharacteristics		loopDataInputRef inputDataItem	
Gateways			
parallelGateway	gatewayDirection name id		x y width height
eventBasedGateway	gatewayDirection name id		x y width height
exclusiveGateway	default gatewayDirection name id		x y width height
inclusiveGateway	default gatewayDirection name id		x y width height
Data			
property	itemSubjectRef id		
dataObject	itemSubjectRef id		
itemDefinition	structureRef id		
ioSpecification		dataInput dataOutput inputSet outputSet	
dataInput	name id		
dataInputAssociation		sourceRef targetRef assignment	
dataOutput	name id		
dataOutputAssociation		sourceRef targetRef assignment	

##	####	####	####
inputSet		dataInputRefs	
outputSet		dataOutputRefs	
assignment		from to	
formalExpression	language	text[mixed content]	
BPMNDI			
BPMNDiagram		BPMNPlane	
BPMNPlane	bpmnElement	BPMNEdge BPMNShape	
BPMNShape	bpmnElement	Bounds	
BPMNEdge	bpmnElement	waypoint	
Bounds	x y width height		
waypoint	x y		

4 # API

```
#####(#####)#####(session)#####knowledge base##
##knowledge base #####
```



```
#####
knowledge base#####knowledge base## ##session#####
#####
```

```
knowledge base####session#####knowledge base(knowledge base#### #
#####)#Knowledge bases#####
```

```
##knowledge base##Sessions#Sessions#####session, session#####
#####session#####session#####session. #####
session(##### ######session)#####
session##### ##knowledge base#####session#####
```

4.1. jBPM API

```
#####API#####API#####'###'##### #####API#####  
#####API#####
```

```
#####jBPM API#####(1)## knowledge base##### (2) ##session#####  
##
```

4.1.1. Knowledge Base

```
jBPM API#####knowledge base###knowledge base#####  
###knowledge base###knowledge base#####(##knowledge builder##knowledge base###  
#####
```

```
KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();  
kbuilder.add(ResourceFactory.newClassPathResource("MyProcess.bpmn"), ResourceType.BPMN2);  
KnowledgeBase kbase = kbuilder.newKnowledgeBase();
```

4.1.2. Session

```
#####session#####session####id#####.
```

```
StatefulKnowledgeSession ksession = kbase.newStatefulKnowledgeSession();  
ProcessInstance processInstance = ksession.startProcess("com.sample.MyProcess");
```

```
ProcessRuntime #####session#####
```

```
/**  
 * Start a new process instance. The process (definition) that should  
 * be used is referenced by the given process id.  
 *  
 * @param processId The id of the process that should be started  
 * @return the ProcessInstance that represents the instance of the process that was started  
 */  
ProcessInstance startProcess(String processId);  
  
/**  
 * Start a new process instance. The process (definition) that should  
 * be used is referenced by the given process id. Parameters can be passed
```

```

* to the process instance (as name-value pairs), and these will be set
* as variables of the process instance.
*
* @param processId the id of the process that should be started
* @param parameters the process variables that should be set when starting the process instance
* @return the ProcessInstance that represents the instance of the process that was started
*/
ProcessInstance startProcess(String processId,
                             Map<String, Object> parameters);

/**
* Signals the engine that an event has occurred. The type parameter defines
* which type of event and the event parameter can contain additional information
* related to the event. All process instances that are listening to this type
* of (external) event will be notified. For performance reasons, this type of event
* signaling should only be used if one process instance should be able to notify
* other process instances. For internal event within one process instance, use the
* signalEvent method that also include the processInstanceId of the process instance
* in question.
*
* @param type the type of event
* @param event the data associated with this event
*/
void signalEvent(String type,
                 Object event);

/**
* Signals the process instance that an event has occurred. The type parameter defines
* which type of event and the event parameter can contain additional information
* related to the event. All node instances inside the given process instance that
* are listening to this type of (internal) event will be notified. Note that the event
* will only be processed inside the given process instance. All other process instances
* waiting for this type of event will not be notified.
*
* @param type the type of event
* @param event the data associated with this event
* @param processInstanceId the id of the process instance that should be signaled
*/
void signalEvent(String type,
                 Object event,
                 long processInstanceId);

/**
* Returns a collection of currently active process instances. Note that only process

```

```
* instances that are currently loaded and active inside the engine will be returned.
* When using persistence, it is likely not all running process instances will be loaded
* as their state will be stored persistently. It is recommended not to use this
* method to collect information about the state of your process instances but to use
* a history log for that purpose.
*
* @return a collection of process instances currently active in the session
*/
Collection<ProcessInstance> getProcessInstances();

/**
 * Returns the process instance with the given id. Note that only active process instances
 * will be returned. If a process instance has been completed already, this method will return
 * null.
 *
 * @param id the id of the process instance
 * @return the process instance with the given id or null if it cannot be found
 */
ProcessInstance getProcessInstance(long processInstanceId);

/**
 * Aborts the process instance with the given id. If the process instance has been completed
 * (or aborted), or the process instance cannot be found, this method will throw an
 * IllegalArgumentException.
 *
 * @param id the id of the process instance
 */
void abortProcessInstance(long processInstanceId);

/**
 * Returns the WorkItemManager related to this session. This can be used to
 * register new WorkItemHandlers or to complete (or abort) WorkItems.
 *
 * @return the WorkItemManager related to this session
 */
WorkItemManager getWorkItemManager();
```

4.1.3.

```
session#####ProcessEventListener#####
#####ProcessEventListener#####
```



```

public interface ProcessEventListener {

    void beforeProcessStarted( ProcessStartedEvent event );
    void afterProcessStarted( ProcessStartedEvent event );
    void beforeProcessCompleted( ProcessCompletedEvent event );
    void afterProcessCompleted( ProcessCompletedEvent event );
    void beforeNodeTriggered( ProcessNodeTriggeredEvent event );
    void afterNodeTriggered( ProcessNodeTriggeredEvent event );
    void beforeNodeLeft( ProcessNodeLeftEvent event );
    void afterNodeLeft( ProcessNodeLeftEvent event );

}

```

```
#####
```

1. Console logger: #####
2. File logger: #####XML#####IDE#####Log## #####
3. Threaded file logger: ####logger##logger##### #####
#####threaded file logger##### #####

```

KnowledgeRuntimeLoggerFactory####session####logger#####console   logger##knowledge
session #####file logger#####threaded file logger#####.

```

```






KnowledgeRuntimeLogger logger =
    KnowledgeRuntimeLoggerFactory.newFileLogger( ksession, "test" );
// add invocations to the process engine here,
// e.g. ksession.startProcess(processId);
...
logger.close();

```

```

#####Eclipse Drools## #Audit View##, #####
##action#####

```

- ▼  RuleFlow started: ruleflow[com.sample.ruleflow]
- ▼  RuleFlow node triggered: Start in process ruleflow[com.sample.ruleflow]
- ▼  RuleFlow node triggered: Hello in process ruleflow[com.sample.ruleflow]
- ▼  RuleFlow node triggered: End in process ruleflow[com.sample.ruleflow]
-  RuleFlow completed: ruleflow[com.sample.ruleflow]

4.2. Knowledge-based API

```
#####jBPM#####API###knowledge    API##### #knowledge#  
#####KnowledgeBase    #KnowledgeSession###    ProcessBase#  
ProcessSession.
```

```
#####knowledge-based API#####  
#####Drools Expert ## Drools Fusion###API#####
```

5 #

####BPM##### jBPM#####

#####

#####(1)##### (2) #####(### WS-HT) ## (3) #####
#####

5.1.



jBPM##### ##jBPM#####

#####

- Id: ##id(#####)
- Name: #####.
- TaskName: #####.
- Priority: #####.
- Comment: #####.
- ActorId: #####id#####.
- Skippable: #####
- Content: #####.
- Swimlane: #####
- Wait for completion: #####true#####
- On-entry and on-exit actions: #####Actions
- Parameter mapping: #####
- Result mapping: ##### "Wait for completion" # true##### "Result"###"ActorId" #####id#

5 #

- Timers: #####(#####'timers').
- ParentId: #####id, #####(##### #'sub task')

#####

Property	Value
ActorId	Sales Representative
Comment	You should call #{customer.name} to confirm the order.
Content	
Id	4
Name	Human Task
On Entry Actions	
On Exit Actions	
Parameter Mapping	{}
Priority	3
Result Mapping	{}
Skippable	true
Swimlane	
TaskName	Call customer
Timers	
Wait for completion	true

Human Task Editor

Name: Call customer

Actor(s): Sales Representative

Comment: You should call #{customer.name} to confirm the order.

Priority: 3

Skippable

Content:

OK Cancel

```
#####(#####)# #####
#{expression}#### #####MVEL####{person.name.firstname} #
##### "Dear #{customer.name}, ..."#####
```

5.1.1.

```
#####
#id#####id actorId ##### actorId#####actorId#
#####actorId#####actorId#####
#####
#####"Swimlane"#####
#"Swimlanes"#####
```

5.2.

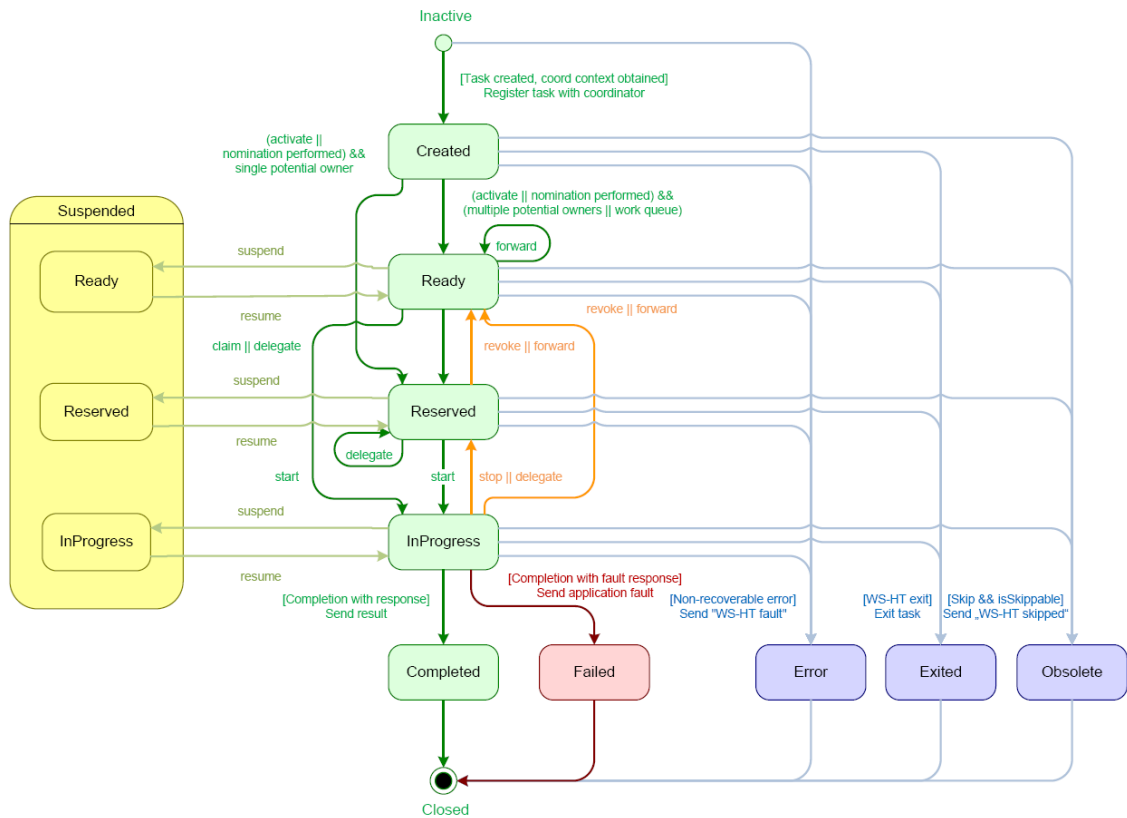
#jBPM#####
#####

#####WS-HumanTask#####(#####
#####)##### .

#####WS-HumanTask#####
[\[http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf\]](http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf).

5.2.1.

#####WS-HumanTask###



#####"Created"##. ##### "Ready" ##, #####
#####"Reserved". #####
#####"InProgress". ##### "Completed". #####
"Failed".

#####

- #####

- #####
- #####
- #####
- #####

5.2.2. #####jBPM##

```
#####      jBPM#####      #####
(org.jboss.process.workitem.wsht.WSHumanTaskHandler)#####
```

```
StatefulKnowledgeSession ksession = ...;
ksession.getWorkItemManager().registerWorkItemHandler("Human
Task", new WSHumanTaskHandler());
```

```
#####(handler)#####9123#####WSHumanTaskHandler      #
setConnection(ipAddress, port)#####
```

```
####session#####
org.jboss.process.workitem.wsht.CommandBasedWSHumanTaskHandler#####
##
```

```
#####Mina (http://mina.apache.org/) [http://mina.apache.org/] . #####HornetQ#
```

```
#####
```

```
public void start( long taskId, String userId, TaskOperationResponseHandler responseHandler )
public void stop( long taskId, String userId, TaskOperationResponseHandler responseHandler )
public void release( long taskId, String userId, TaskOperationResponseHandler responseHandler )
public void suspend( long taskId, String userId, TaskOperationResponseHandler responseHandler )
public void resume( long taskId, String userId, TaskOperationResponseHandler responseHandler )
public void skip( long taskId, String userId, TaskOperationResponseHandler responseHandler )
public void delegate( long taskId, String userId, String targetUserId,
                    TaskOperationResponseHandler responseHandler )
public void complete( long taskId, String userId, ContentData outputData,
                    TaskOperationResponseHandler responseHandler )
...

```

```
#####GUI#####
```

- **taskId:** ##id. #####

- **userId:** ##action###id#####id.
- **responseHandler:** #####

#####

5.2.3.

#####

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("org.jbpm.task");
TaskService taskService = new TaskService(emf, SystemEventListenerFactory.getSystemEventListener());
MinaTaskServer server = new MinaTaskServer( taskService );
Thread thread = new Thread( server );
thread.start();
```

#####JPA#####persistence.xml## #####hibernate#####H2:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence
  version="1.0"
  xsi:schemaLocation=
    "http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd
    http://java.sun.com/xml/ns/persistence/orm
    http://java.sun.com/xml/ns/persistence/orm_1_0.xsd"
  xmlns:orm="http://java.sun.com/xml/ns/persistence/orm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/persistence">

  <persistence-unit name="org.drools.task">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <class>org.jbpm.task.Attachment</class>
    <class>org.jbpm.task.Content</class>
    <class>org.jbpm.task.BooleanExpression</class>
    <class>org.jbpm.task.Comment</class>
    <class>org.jbpm.task.Deadline</class>
    <class>org.jbpm.task.Comment</class>
    <class>org.jbpm.task.Deadline</class>
    <class>org.jbpm.task.Delegation</class>
    <class>org.jbpm.task.Escalation</class>
    <class>org.jbpm.task.Group</class>
    <class>org.jbpm.task.I18NText</class>
    <class>org.jbpm.task.Notification</class>
```



```

<class>org.jbpm.task.EmailNotification</class>
<class>org.jbpm.task.EmailNotificationHeader</class>
<class>org.jbpm.task.PeopleAssignments</class>
<class>org.jbpm.task.Reassignment</class>
<class>org.jbpm.task.Status</class>
<class>org.jbpm.task.Task</class>
<class>org.jbpm.task.TaskData</class>
<class>org.jbpm.task.SubTasksStrategy</class>
<class>org.jbpm.task.OnParentAbortAllSubTasksEndStrategy</class>
<class>org.jbpm.task.OnAllSubTasksEndParentEndStrategy</class>
<class>org.jbpm.task.User</class>

<properties>
  <property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>
  <property name="hibernate.connection.driver_class" value="org.h2.Driver"/>
  <property name="hibernate.connection.url" value="jdbc:h2:mem:mydb" />
  <property name="hibernate.connection.username" value="sa"/>
  <property name="hibernate.connection.password" value="sasa"/>
  <property name="hibernate.connection.autocommit" value="false" />
  <property name="hibernate.max_fetch_depth" value="3"/>
  <property name="hibernate.hbm2ddl.auto" value="create" />
  <property name="hibernate.show_sql" value="true" />
</properties>
</persistence-unit>
</persistence>

```

```

##### #
##taskSession.addUser(user) # taskSession.addGroup(group) ##### #
#"Administrator"#####"Administrator"###

```

```

jbpm-human-task#####src/test/java#####org.jbpm.task.RunTaskService #####
###LoadUsers.mvel#LoadGroups.mvel#####

```

5.2.4.

```

#####API#####API#####
###API#####

```

```

TaskClient client = new TaskClient(new MinaTaskClientConnector("client 1",
    new MinaTaskClientHandler(SystemEventListenerFactory.getSystemEventListener()));
client.connect("127.0.0.1", 9123);

// adding a task
BlockingAddTaskResponseHandler addTaskResponseHandler = new BlockingAddTaskResponseHandler();

```

5 #

```
Task task = ...;
client.addTask( task, null, addTaskResponseHandler );
long taskId = addTaskResponseHandler.getTaskId();

// getting tasks for user "bobba"
BlockingTaskSummaryResponseHandler taskSummaryResponseHandler =
    new BlockingTaskSummaryResponseHandler();
client.getTasksAssignedAsPotentialOwner("bobba", "en-UK", taskSummaryResponseHandler);
List<TaskSummary> tasks = taskSummaryResponseHandler.getResults();

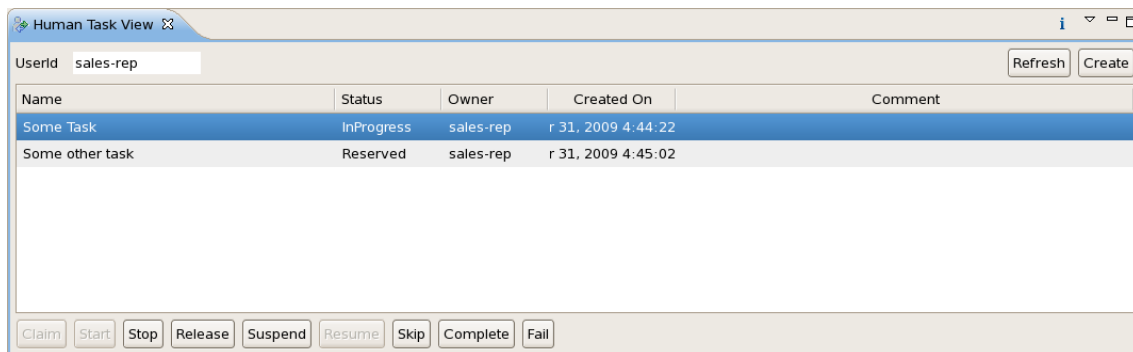
// starting a task
BlockingTaskOperationResponseHandler responseHandler =
    new BlockingTaskOperationResponseHandler();
client.start( taskId, "bobba", responseHandler );

// completing a task
responseHandler = new BlockingTaskOperationResponseHandler();
client.complete( taskId, users.get( "bobba" ).getId(), null, responseHandler );
```

5.3.

5.3.1. Eclipse##

Drools IDE #### org.drools.eclipse.task #####
#####(select
Window -> Preferences #### Drools Task).#####url####(default = 127.0.0.1:9123).



5.3.2. ##web#####

jBPM#####jBPM###.

6 #

6.1. Introduction

jBPM#####
#####jBPM####
#####jBPM#####

#####action#####action#####action# #####action#####
#####

#####

#####

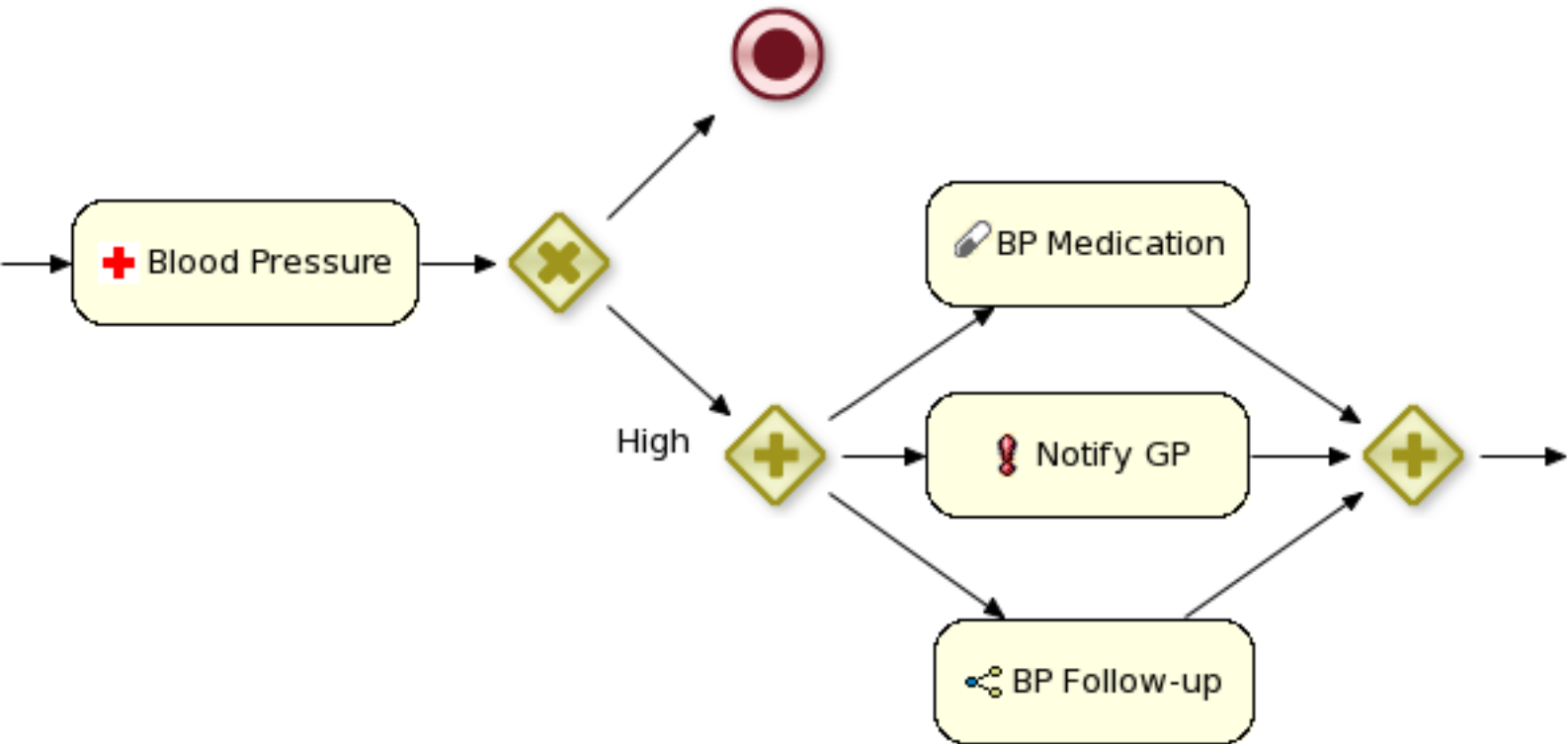
1. #####

2. #####

3. #####

4. #####

#####



6.2.

###

```
Name: "Notification"  
Parameters  
From [String]  
To [String]  
Message [String]  
Priority [String]
```

6.2.1.

#####map# #####
###MyWorkDefinitions.conf#####

```
import org.drools.process.core.datatype.impl.type.StringDataType;  
[  
  // the Notification work item  
  [  
    "name" : "Notification",  
    "parameters" : [  
      "Message" : new StringDataType(),  
      "From" : new StringDataType(),  
      "To" : new StringDataType(),  
      "Priority" : new StringDataType(),  
    ],  
    "displayName" : "Notification",  
    "icon" : "icons/notification.gif"  
  ]  
]
```

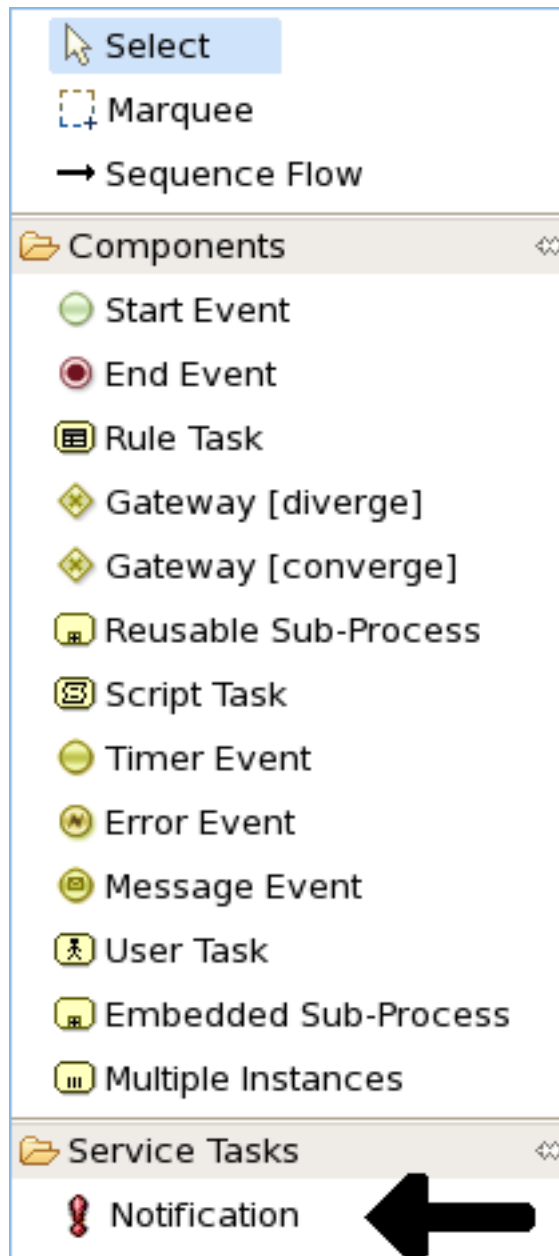
6.2.2.

##API#####drools.workDefinitions, #####META-INF##
drools.rulebase.conf#####

drools.workDefinitions = MyWorkDefinitions.conf

6.2.3.

#####



#####

#####

1. Parameter Mapping: ##### (#####)

2. Result Mapping: #####

3. Wait for completion: ##### "wait for completion" ####false#####.

6.2.4.

Drools#####WorkItemManager#####WorkItemManager##### WorkItemHandlers#
WorkItemHandlers#####WorkItemManager#####
NotificationWorkItemHandler###(##WorkItemHandler##):

```
package com.sample;

import org.drools.runtime.process.WorkItem;
import org.drools.runtime.process.WorkItemHandler;
import org.drools.runtime.process.WorkItemManager;

public class NotificationWorkItemHandler implements WorkItemHandler {

    public void executeWorkItem(WorkItem workItem, WorkItemManager manager) {
        // extract parameters
        String from = (String) workItem.getParameter("From");
        String to = (String) workItem.getParameter("To");
        String message = (String) workItem.getParameter("Message");
        String priority = (String) workItem.getParameter("Priority");
        // send email
        EmailService service = ServiceRegistry.getInstance().getEmailService();
        service.sendEmail(from, to, "Notification", message);
        // notify manager that work item has been completed
        manager.completeWorkItem(workItem.getId(), null);
    }

    public void abortWorkItem(WorkItem workItem, WorkItemManager manager) {
        // Do nothing, notifications cannot be aborted
    }

}
```

WorkItemHandler#email#####WorkItemManager#####
#####manager###
#####

WorkItemHandlers##WorkItemManager###, ####API:

```
ksession.getWorkItemManager().registerWorkItemHandler(
```

```
"Notification", new NotificationWorkItemHandler());
```

#####

1. #####

2. #####

3. #####.

4. #####

7 #

jBPM#####

7.1.

#####

#####jBPM#####
#####

7.1.1.

jBPM#####
#####

7.1.2.

#####'###'#####
#####action#####
#####

7.1.3.

#####JPA#####
#####Hibernate#####H2.#####

#####EclipseIDE#####Drools runtime####jar#####jar#####
jar##jbpm-persistence-jpa.jar,#####jar#####
##hibernate##JPA ###provider#H2#####JTA#####Bitronix#####jar###

1. jbpm-persistence-jpa (org.jbpm)
2. drools-persistence-jpa (org.drools)
3. persistence-api (javax.persistence)
4. hibernate-entitymanager (org.hibernate)
5. hibernate-annotations (org.hibernate)
6. hibernate-commons-annotations (org.hibernate)
7. hibernate-core (org.hibernate)
8. dom4j (dom4j)
9. jta (javax.transaction)
10. btm (org.codehaus.btm)

7 #

11javassist (javassist)

12slf4j-api (org.slf4j)

13slf4j-jdk14 (org.slf4j)

14h2 (com.h2database)

15commons-collections (commons-collections)

```
#####jbpm##### JPAKnowledgeService##knowledge session##### knowledge
base, ## knowledge session ##### Entity Manager Factory#####
```

```
// create the entity manager factory and register it in the environment
EntityManagerFactory emf =
    Persistence.createEntityManagerFactory( "org.jbpm.persistence.jpa" );
Environment env = KnowledgeBaseFactory.newEnvironment();
env.set( EnvironmentName.ENTITY_MANAGER_FACTORY, emf );

// create a new knowledge session that uses JPA to store the runtime state
StatefulKnowledgeSession ksession =
    JPAKnowledgeService.newStatefulKnowledgeSession( kbase, null, env );
int sessionId = ksession.getId();

// invoke methods on your method here
ksession.startProcess( "MyProcess" );
ksession.dispose();
```

```
#####session id##JPAKnowledgeService####session:
```

```
// recreate the session from database using the sessionId
ksession = JPAKnowledgeService.loadStatefulKnowledgeSession( sessionId, kbase, null, env );
```

```
#####
#####history log#####
```

```
##### persistence.xml#####META-INF##### JPA#
Hibernate###
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence
  version="1.0"
  xsi:schemaLocation=
```

```

"http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd
http://java.sun.com/xml/ns/persistence/orm
http://java.sun.com/xml/ns/persistence/orm_1_0.xsd"
xmlns:orm="http://java.sun.com/xml/ns/persistence/orm"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/persistence">

<persistence-unit name="org.jbpm.persistence.jpa">
  <provider>org.hibernate.ejb.HibernatePersistence</provider>
  <jta-data-source>jdbc/processInstanceDS</jta-data-source>
  <class>org.drools.persistence.session.SessionInfo</class>
  <class>org.jbpm.persistence.processinstance.ProcessInstanceInfo</class>
  <class>org.jbpm.persistence.processinstance.WorkItemInfo</class>

  <properties>
    <property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>
    <property name="hibernate.max_fetch_depth" value="3"/>
    <property name="hibernate.hbm2ddl.auto" value="update"/>
    <property name="hibernate.show_sql" value="true"/>
    <property name="hibernate.transaction.manager_lookup_class"
      value="org.hibernate.transaction.BTMTransactionManagerLookup"/>
  </properties>
</persistence-unit>
</persistence>

```

```
#####"jdbc/processInstanceDS".###java#####H2####
```

```

PoolingDataSource ds = new PoolingDataSource();
ds.setUniqueName("jdbc/testDS1");
ds.setClassName("org.h2.jdbcx.JdbcDataSource");
ds.setMaxPoolSize(3);
ds.setAllowLocalTransactions(true);
ds.getDriverProperties().put("user", "sa");
ds.getDriverProperties().put("password", "sasa");
ds.getDriverProperties().put("URL", "jdbc:h2:file:/NotBackedUp/data/process-instance-db");
ds.init();

```

```
#####deploy#####:
```

```

<?xml version="1.0" encoding="UTF-8"?>
<datasources>

```

```
<local-tx-datasource>
  <jndi-name>jdbc/testDS1</jndi-name>
  <connection-url>jdbc:h2:file:/NotBackedUp/data/process-instance-db</connection-url>
  <driver-class>org.h2.jdbcx.JdbcDataSource</driver-class>
  <user-name>sa</user-name>
  <password>sasa</password>
</local-tx-datasource>
</datasources>
```

7.1.4.

```
#####
#####
```

```
#####Bitronix #####Java Transaction API (JTA)#
#####
```

```
// create the entity manager factory and register it in the environment
EntityManagerFactory emf =
    Persistence.createEntityManagerFactory( "org.jbpm.persistence.jpa" );
Environment env = KnowledgeBaseFactory.newEnvironment();
env.set( EnvironmentName.ENTITY_MANAGER_FACTORY, emf );
env.set( EnvironmentName.TRANSACTION_MANAGER,
    TransactionManagerServices.getTransactionManager() );

// create a new knowledge session that uses JPA to store the runtime state
StatefulKnowledgeSession ksession =
    JPAKnowledgeService.newStatefulKnowledgeSession( kbase, null, env );

// start the transaction
UserTransaction ut =
    (UserTransaction) new InitialContext().lookup( "java:comp/UserTransaction" );
ut.begin();

// perform multiple commands inside one transaction
ksession.insert( new Person( "John Doe" ) );
ksession.startProcess( "MyProcess" );

// commit the transaction
ut.commit();
```

7.2.

```
#####XML#####
knowledge repository####knowledge#
```

```
Guvnor###Drools#####repository#####webDAV####
#####knowledge #####web#####Drools Guvnor##
```

7.3.

```
#####action### #####
#####
```

```
#####Drools #####
#####
```

7.3.1.

```
jbpm-bam#####:
```

1. *ProcessInstanceLog*: #####id###id#####
2. *NodeInstanceLog*: #####id###id#####id###

id#####

ID	PROCESSINSTANCEID	PROCESSID	START_DATE
----	-------------------	-----------	------------

ID	TYPE	NODEINSTANCEID	NODEID	PROCESSINSTANCEID	PROCESSID	LOG_DATE
----	------	----------------	--------	-------------------	-----------	----------

```
#####session###logger###:
```

```
StatefulKnowledgeSession ksession = ...;
JPAWorkingMemoryDbLogger logger = new JPAWorkingMemoryDbLogger(ksession);

// invoke methods one your session here

logger.dispose();
```

```
#####logger#####logger##### addFilter#####
```

```
###persistence.xml#####
```

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<persistence
  version="1.0"
  xsi:schemaLocation=
    "http://java.sun.com/xml/ns/persistence
     http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd
     http://java.sun.com/xml/ns/persistence/orm
     http://java.sun.com/xml/ns/persistence/orm_1_0.xsd"
  xmlns:orm="http://java.sun.com/xml/ns/persistence/orm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/persistence">

  <persistence-unit name="org.jbpm.persistence.jpa">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <jta-data-source>jdbc/processInstanceDS</jta-data-source>
    <class>org.drools.persistence.session.SessionInfo</class>
    <class>org.jbpm.persistence.processinstance.ProcessInstanceInfo</class>
    <class>org.jbpm.persistence.processinstance.WorkItemInfo</class>
    <class>org.jbpm.process.audit.ProcessInstanceLog</class>
    <class>org.jbpm.process.audit.NodeInstanceLog</class>
    <class>org.jbpm.process.audit.VariableInstanceLog</class>

    <properties>
      <property name="hibernate.dialect" value="org.hibernate.dialect.H2Dialect"/>
      <property name="hibernate.max_fetch_depth" value="3"/>
      <property name="hibernate.hbm2ddl.auto" value="update"/>
      <property name="hibernate.show_sql" value="true"/>
      <property name="hibernate.transaction.manager_lookup_class"
        value="org.hibernate.transaction.BTMTTransactionManagerLookup"/>
    </properties>
  </persistence-unit>
</persistence>
```

#####

##log#####
##

8 #

###web#####shili(##/##/##),#####.

jBPM#####war#####

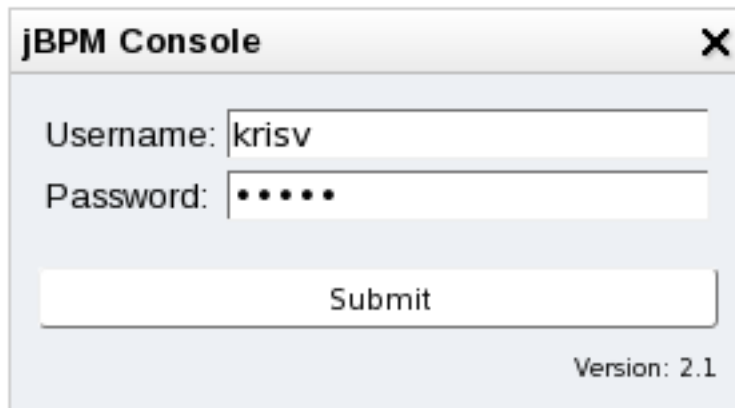
8.1.

#####installer##### ##installer##

8.2.

#####url#(#####): <http://localhost:8080/jbpm-console>

krisv/krisv, admin/admin, john/john # mary/mary.



The image shows a web browser window titled "jBPM Console". Inside the window, there is a login form with two input fields: "Username:" containing the text "krisv" and "Password:" containing five dots. Below the password field is a "Submit" button. In the bottom right corner of the window, it says "Version: 2.1".

#####tab#####



Personal Tasks

h View Release

Process

Task Name



Details

ss:

:

nee:

ipion:

8.2.1. #####

"Processes"#####

8.2.1.1.

#####

+

Process Overview

All [dropdown]

Start Signal Delete Terminate

Instance	State
on	0

Execution details

Process:	
Instance ID:	
Key:	
State	
Start Date:	
Activity:	

>>

8.2.1.2.

#####"Start" #####

Process Overview

All [dropdown] [Start] [Signal] [Delete] [Terminate]

Instance	State
on	0

New Process Instance: com.sample.evaluation

Start Performance Evaluation

Please fill in your username:

8.2.1.3.

#####



Process Overview

Refresh All 

Start Signal Delete Terminate

Process	v.
Evaluation	0

Instance	State
1	RUNNING

Execution details

Process:	Evaluation
Instance ID:	1
Key:	
State	RUNNING
Start Date:	2010-11-22 16:46:59
Activity:	

« »

8.2.1.4.

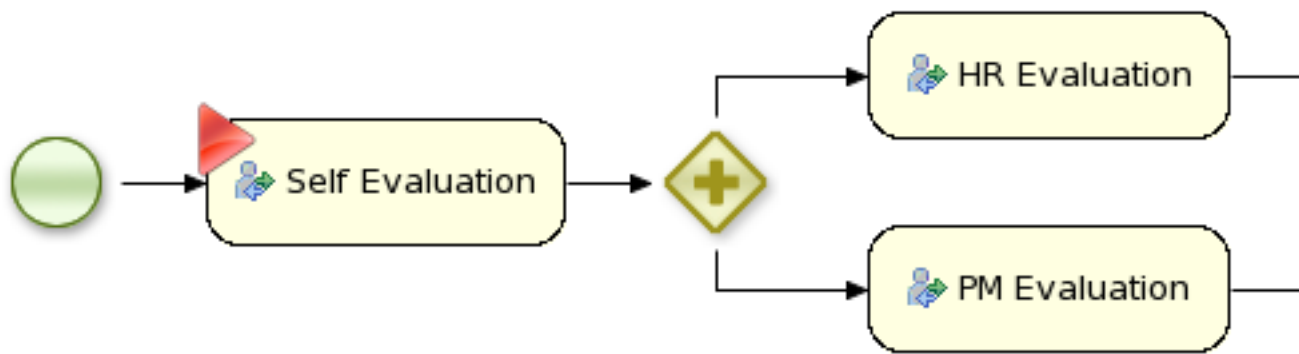
##"Diagram"#####[#####
#####]

All Start Signal Delete Terminate

v.	Instance	State
0	1	RUNNING

Process Instance Activity

Instance: 1



8 #

8.2.1.5.

##"Instance Data"#####

All

Start

Signal

Delete

Terminate

v.	Instance	State
0	1	RUNNING

Instance Data: 1

XSD Type	Java Type
xs:string	java.lang.String

8.2.2.

#####"View"#####. #####
#####

	Process	Task Name
		Performance Evaluation

Performance Evaluation

evaluation

a self-evaluation.

following evaluation form:

performance:

apply:
initiative
change
communication skills

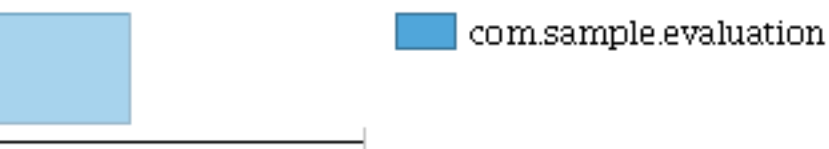
8.2.3.

#####

This report doesn't require any parameters.

Business Activity Monitori

Instances / Hour



8 #

#####

Please enter a process definition id

com.sample


Business Activity Monitoring

Process `com.sample.evaluation`

com.sample.evaluation

1
1
1

Process Instances

 com.sample.evaluation

jBPM#####

8.3. #####/#####

#####(1)#####(2)#####freemarker#####form## #### {processId}.ftl##
##. #####HTML#####evaluation#####

```
<html>
<body>
<h2>Start Performance Evaluation</h2>
<hr>
<form action="complete" method="POST" enctype="multipart/form-data">
Please fill in your username: <input type="text" name="employee" /></BR>
<input type="submit" value="Complete">
</form>
</body>
</html>
```

#####{taskName}.ftl##. #####
drools-process-task#####evaluation#####

```
<html>
<body>
<h2>Employee evaluation</h2>
<hr>
${task.descriptions[0].text}<br/>
<br/>
Please fill in the following evaluation form:
<form action="complete" method="POST" enctype="multipart/form-data">
Rate the overall performance: <select name="performance">
<option value="outstanding">Outstanding</option>
<option value="exceeding">Exceeding expectations</option>
<option value="acceptable">Acceptable</option>
<option value="below">Below average</option>
</select><br/>
<br/>
Check any that apply:<br/>
<input type="checkbox" name="initiative" value="initiative">Displaying initiative<br/>
<input type="checkbox" name="change" value="change">Thriving on change<br/>
<input type="checkbox" name="communication" value="communication">Good communication
skills<br/>
```

```
<br/>
<input type="submit" value="Complete">
</form>
</body>
</html>
```

```
##### "performance", "initiative", "change" #
"communication". ##### ###
```

```
#####jbpm-gwt-form.jar#.
```

##

