The PVM: A revolutionary technology for BPM

Miguel Valdes Faura
BPM Manager
Bull

Bonita
Because BPM Matters!

Learn how an open source technology can create mindshare around BPM and how it can be used in your Java applications.
Speaker’s qualifications

- Miguel Valdes is the Latin Lover of BPM
- Miguel is a BPM/Workflow open source evangelist world wide
- Miguel is the founder of Bonita Open Source BPM solution hosted in OW2 Consortium
- BPM Manager at Bull R&D
Life is all about processes!

Processes are everywhere and are part of our day to day life! Processes are around us, we play with them every day…soon inside us 😊
Processes misconceptions and misunderstandings leads to users frustration every day!
Processes misconceptions and misunderstandings leads to users frustration every day!

I’m sure you have already experimented frustration:

As a citizen: taxes payment, building permit…

As a consumer: online orders, products and services…
BPM is great because they allow:

- To control the organization of work
- To drive efficiently your processes
- To optimize and share existing processes
- To line up the Information System on the business objectives
BPM is great because they allow:

- To control the organization of work
- To drive efficiently your processes
- To optimize and share existing processes
- To line up the Information System on the business objectives

and what about adding: because it will simplify the development of your Java applications!
Looking for a BPM Solution?
Looking for a BPM Solution?

- Commercial vendors:
  - BPM pure players: Tibco, Savvion, Lombardi, PegaSystems, Appian, Global360
  - Big vendors: Oracle, IBM, Fujitsu, Adobe…
Looking for a BPM Solution?

- Commercial vendors:
  - BPM pure players: Tibco, Savvion, Lombardi, PegaSystems, Appian, Global360
  - Big vendors: Oracle, IBM, Fujitsu, Adobe…

- Adhoc vendors (seems to be open source but they are not!)
  - Intalio, Active Enpoints
Looking for a BPM Solution?

Commercial vendors:

- BPM pure players: Tibco, Savvion, Lombardi, PegaSystems, Appian, Global360
- Big vendors: Oracle, IBM, Fujitsu, Adobe…

Adhoc vendors (seems to be open source but they are not!)

- Intalio, Active Enpoints

Open Source vendors

- Jbpm, Bonita, Orchestra
- ODE, OpenWFE

PVM based vendors
The Process Virtual Machine, yeah baby!

- History
- Architecture
- Main concepts
- How to use it?

The BONITA way of live…
The Process Virtual Machine

- Redhat/Jboss and Bull/OW2 common effort around BPM
- Underlying technology for BPM solutions: jBPM, Bonita and Orchestra
The Process Virtual Machine

- Redhat/Jboss and Bull/OW2 common effort around BPM
- Underlying technology for BPM solutions: jBPM, Bonita and Orchestra

- Generic: processes based on a diagram structure (graph based DSL’s)
Redhat/Jboss and Bull/OW2 common effort around BPM

Underlying technology for BPM solutions: jBPM, Bonita and Orchestra

Generic: processes based on a diagram structure (graph based DSL’s)

Extensible: support for multiple process languages (JPDL, XPDL, BPEL…) and services
Redhat/Jboss and Bull/OW2 common effort around BPM

Underlying technology for BPM solutions: jBPM, Bonita and Orchestra

Generic: processes based on a diagram structure (graph based DSL’s)

Extensible: support for multiple process languages (JPDL, XPDL, BPEL…) and services

Embeddable: running in your application (as a library)
The Process Virtual Machine

Pluggable Process Languages

- BPEL
- XPDL
- JPDL
- ...

The Process Virtual Machine

PVM Core
- Process
- Variable
- Transition
- Execution
- Event

Extensible infrastructure

Services Container (environment)

Notifications
Persistence
History
Security
Human Task
...

Pluggable Services
The PVM, main concepts
The PVM, main concepts

- Two main process constructs: nodes and transitions
The PVM, main concepts

- Two main process constructs: nodes and transitions

- Two forms of process languages: graph vs composite based
The PVM, main concepts

- An execution is a pointer to the current Node
The PVM, main concepts

- An execution is a pointer to the current Node

Process Definition
The PVM, main concepts

- An execution is a pointer to the current Node

Process Definition

Process Execution
The PVM, main concepts

- Activity: interface to implement the runtime behaviour of process constructs in plain Java

```java
public interface Activity extends Serializable {
    void execute(Execution execution) throws Exception;
}
```

- Activity's can be used:
  - As node behaviours
  - As listeners to process events
The PVM, main concepts

- External Activity: activities for which the responsibility for proceeding the execution is transferred externally

```java
public interface ExternalActivity extends Activity {
    void signal(Execution execution, String signal, Map<String, Object> parameters) throws Exception;
    Set<SignalDefinition> getSignals(Execution execution) throws Exception;
}
```

- The execution will wait until an external trigger is given
The PVM, Configuration

- An IoC container is used for PVM configuration and extensibility
- XML descriptor: `environment.xml`
- Container for PVM execution:
  - Getters for service implementations
  - Services can be accessed by:
    - Name (e.g. `environment.get("MyDBService")`)
    - Type (e.g. `environment.get(DBService.class)`)
- Designed as a pluggable module: others IoC containers could be leveraged: i.e. Spring
The PVM, How to use it?

Just open your favorite IDE and code in plain Java!
The PVM, How to use it?

Just open your favorite IDE and code in plain Java!

```java
ProcessDefinition processDefinition = ProcessFactory.build()
    .node("a").initial().behaviour(new Display("Spain"))
    .transition().to("b")
    .node("b").behaviour(new Display(" is different !"))
    .done();
```

Process Definition
The PVM, How to use it?

Just open your favorite IDE and code in plain Java!

```java
ProcessDefinition processDefinition = ProcessFactory.build()
    .node("a").initial().behaviour(new Display("Spain"))
    .transition().to("b")
    .node("b").behaviour(new Display("is different!"))
    .done();
```

```java
public class Display implements Activity {
    String message;
    public Display(String message) {
        this.message = message;
    }
    public void execute(Execution execution) {
        System.out.println(message);
    }
}
```

Process Definition

Activity behaviour
The PVM, How to use it?

- A more complex example… but easy for you guys!

- “Evaluate” would be an External Activity

- Others would be Wait States
The PVM, How to use it ?
public class Decision implements ExternalActivity {
    public void execute(Execution execution) {
        execution.waitForSignal();
    }
    public void signal(Execution execution, String signalName, Map<String, Object> parameters) throws Exception{
        execution.take(signalName);
    }
    public Set<SignalDefinition> getSignals(Execution execution) throws Exception {
        return null;
    }
}
ProcessDefinition processDefinition = ProcessFactory.build()
    .node("Request").initial().behaviour(new WaitState())
    .transition().to("Evaluation")

    .node("Evaluation").behaviour(new Decision())
    .transition("more").to("More Info")
    .transition("grant").to("Pay")
    .transition("reject").to("Cancel")

    .node("More Info").behaviour(new WaitState())
    .node("Pay").behaviour(new WaitState())
    .node("Chncel").behaviour(new WaitState())
    .done();
The PVM, How to use it?

```java
ProcessDefinition processDefinition = ProcessFactory.build()
    .node("Request").initial().behaviour(new WaitState())
    .transition().to("Evaluation")

    .node("Evaluation").behaviour(new Decision())
    .transition("more").to("More Info")
    .transition("grant").to("Pay")
    .transition("reject").to("Cancel")

    .node("More Info").behaviour(new WaitState())
    .node("Pay").behaviour(new WaitState())
    .node("Cancel").behaviour(new WaitState())
    .done();

Execution execution = processDefinition.startExecution();
execution.signal();
execution.signal("grant");
```
The Process Virtual Machine, yeah baby!

The BONITA way of live...

- Bonita BPM solution in a nutshell!
- How Bonita is leveraging the PVM?
- 10 reasons to use Bonita!
Bonita in a nutshell!

Three components, one solution!

BONITA 4.0

Designer

Runtime

Console
Bonita in a nutshell !

- Bonita: “à la carte” BPM Open Source Solution
  - Bonita runs wherever Java runs !
  - BPM “as a server” vs “as a library” solution
  - Develop once, deploy it in different environments !

Don’t write code, think process !
Bonita Object Model

- PVM Extension adding support for XPDL

Diagram showing PVM and BONITA with nodes for Process, Activity, Transition, Variable, Execution, External Activity, Sub Process, Route, Task, and Automatic Activity.
Bonita Environment

- Extends PVM Environment
- Adds specific XML tags for Bonita Services
- Entry point for Bonita configuration:
  - Database configuration
Bonita BPM APIs

- Rich Java APIs to deal with processes

CUSTOM APPLICATION

- Management
- Runtime
- Definition
- Queries
- Command

Bonita Object Model (XPDL Support)

PVM Object Model

BPM Services

Bonita BPM Solution
10 Reasons to use Bonita!
10 Reasons to use Bonita !
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
4.- Bonita can be embedded in your Java application
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
4.- Bonita can be embedded in your Java application
5.- Bonita can be deployed as a BPM Server
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
4.- Bonita can be embedded in your Java application
5.- Bonita can be deployed as a BPM Server
6.- Support for XPDL standard
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊

2.- Bonita is a truly open source alternative to commercial vendors

3.- You are going to save time developing and maintaining Java applications

4.- Bonita can be embedded in your Java application

5.- Bonita can be deployed as a BPM Server

6.- Support for XPDL standard

7.- Even Orchestra has decided to join Bonita to brings BPEL support!
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
4.- Bonita can be embedded in your Java application
5.- Bonita can be deployed as a BPM Server
6.- Support for XPDL standard
7.- Even Orchestra has decided to join Bonita to brings BPEL support!
8.- Eclipse based BPM designer
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
4.- Bonita can be embedded in your Java application
5.- Bonita can be deployed as a BPM Server
6.- Support for XPDL standard
7.- Even Orchestra has decided to join Bonita to brings BPEL support!
8.- Eclipse based BPM designer
9.- Web 2.0 based BPM Console: including web forms generation
10 Reasons to use Bonita!

1.- Jboss guys have not yet released jBPM 4 😊
2.- Bonita is a truly open source alternative to commercial vendors
3.- You are going to save time developing and maintaining Java applications
4.- Bonita can be embedded in your Java application
5.- Bonita can be deployed as a BPM Server
6.- Support for XPDL standard
7.- Even Orchestra has decided to join Bonita to brings BPEL support!
8.- Eclipse based BPM designer
9.- Web 2.0 based BPM Console: including web forms generation
10.- Because there is a lot more coming: ETL connectors, BI/BAM, ESB, Portal and ECM, BPM Studio
Processes and BPM matters!

PVM as a technical solution for working with processes in Java

How the PVM is leveraged by a BPM Solution: Bonita
Q&A

Need to talk?

bonita@ow2.org