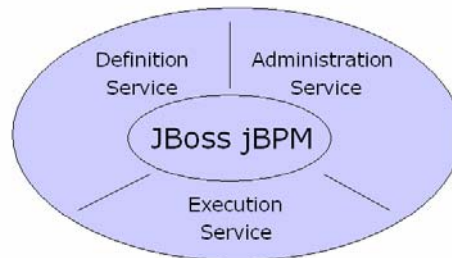


**NOVEMBER 2004**

# **JBoss jBPM**



## **WHITE PAPER**

*by*

**JOHN KOENIG**

[www.riseforth.com](http://www.riseforth.com)

Copyright © 2004 John Koenig  
All Rights Reserved

This document may not be published or reproduced in whole or in part without the prior written approval of John Koenig.

# JBoss jBPM White Paper

## Introduction

BPM (business process management) offers a programmatic structure for designing transactions and executing them using automated decisions, tasks and sequence flows. For example, an insurance company can use BPM to automate the steps involved in processing insurance claims. BPM solutions typically include three components: an engine that executes process definitions, services that allow the engine to interact with the outside world, and tools that aid process development and monitoring.

Although the notion of “workflow” and BPM have promised enterprise application integration for a number of years, their mainstream acceptance has been delayed by the lack of real standards, and more significantly by the enterprise software architectural model. There are literally hundreds of workflow and BPM products in the market typically selling for six-figure license fees, targeted at solving specific problems like document management or web services integration. No inexpensive alternative has been available to IT organizations and software vendors that would allow them to experiment with BPM. With a high cost of entry and the focus on BPM point solutions, developers choose to avoid the complications and risk of BPM and instead elect to code state and business logic directly into enterprise applications as usual.

## Investing in the Future of BPM

Business people recognize the potential benefits in adopting workflow management or BPM in their enterprise architecture. But the distinction between business-level and technical-level product excellence can easily escape a non-programmer. On a business level, deploying a workflow or business process forces an organization to create formal procedural descriptions of

the way an activity works. In many cases this particular exercise reveals obvious business inefficiencies that are of serious consequence, and which additionally constitute an obstacle to process re-engineering.

On a technical level, a capable workflow management or BPM system avoids the error prone task of translating a business software into an executable function. BPM allows a process developer to implement a business process using the same constructs as a business analyst.

In the past 15 years or so, business people have been disappointed in the technical results of workflow and BPM. Until recently, these management systems have introduced more technical issues than they solved. In response, JBoss has focused on making BPM technology more accessible and easier to apply. JBoss jBPM facilitates the natural transition from declarative input by the business analyst to the programming logic needed to implement a business process.

### **BPM and Enterprise Application Co-Existence**

In the future BPM may become a software technology as vital as relational databases. But BPM is unlikely to replace the logic in enterprise applications, and most pragmatists do not hold that out as a goal. Enterprise applications serve an important purpose and BPM is not intended to replace them. A dedicated enterprise application is a repository of acquired knowledge about a target application domain. Companies are strongly accustomed to acquiring or developing domain expertise through the enterprise software applications they purchase, and it is this domain functionality that ordinarily makes an enterprise application so valuable to a customer.

A BPM tool on the other hand contains no intrinsic knowledge about the domain to which it will be applied. As a result, BPM requires process implementation but offers greater flexibility. For highly specialized application domains, dedicated application software is frequently a better value as long as an organization can adapt to the process concepts incorporated in the application. Rather, integration of enterprise applications remains the challenge. Inside each enterprise application the business logic can be hard to find and expose, even by a programmer.

By competitive necessity, companies may frequently need to integrate new application logic. Since this is usually accomplished through high level programming languages and project teams of programmers or consultants, enterprise application modification and integration at most companies can be prone to delays and unplanned costs.

In response to these issues, application vendors have created proprietary BPM tools to allow customers to enhance their enterprise applications or database systems, frequently through visual development tools. This BPM approach however is enterprise application-centric, and therefore cannot satisfy the need for better integration between separate enterprise applications.

BPM vendors have attempted to simplify business process development by offering standalone BPM platforms, allowing even non-programmers to develop business processes again using visual tools. One obstacle however, is that these platform solutions are typically expensive and require a major investment of time and training, without any guarantee of success.

### **JBoss Challenges the Status Quo**

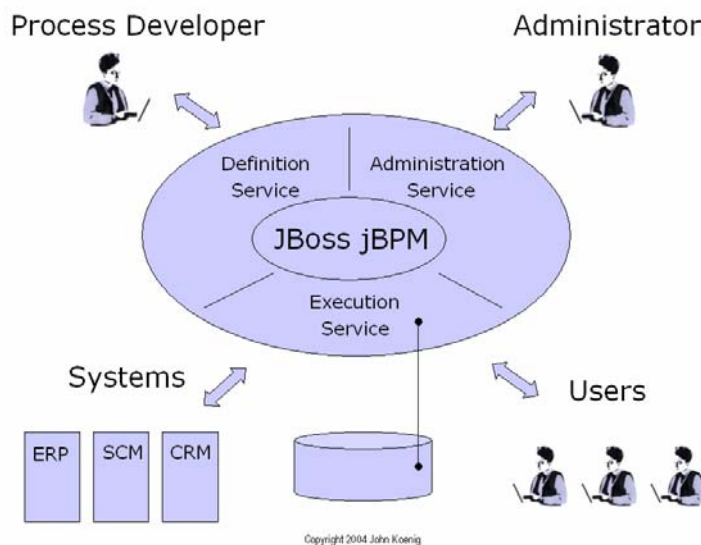
In acquiring jBPM, JBoss executes its mission of delivering professional open source software infrastructure alternatives. With jBPM, developers now have access to BPM tools that they can easily use to experiment and gain confidence. JBoss also believes that open source alternatives complement the state of emerging BPM standards, establishing a community to push BPM standards forward. For example BPEL is an emerging standard, but it has shortcomings. With JBoss jBPM support of BPEL and beyond, community adoption serves to both encourage and improve the BPEL standard.

Whether JBoss jBPM meets the requirements of a particular situation can be evaluated and determined in a relatively short time. JBoss jBPM includes a tutorial that has developers building sample applications in half a day. This allows software vendors, integrators and enterprise IT departments to make a rapid, bottom-up evaluation of jBPM and its potential benefits.

## The JBoss jBPM Solution

JBoss sees BPM as an orchestration engine that sits in the middle of enterprise applications, enabling integration and coordination between different dedicated applications. It is this vision that motivated JBoss to acquire the jBPM project as a complement to the existing JBoss middleware. jBPM is a standard Java application and does not need an application server. Enterprises that are interested in jBPM can use it without adding more complexity. jBPM can also be deployed in a web application or a standalone Java application.

The JBoss vision for jBPM is illustrated below.



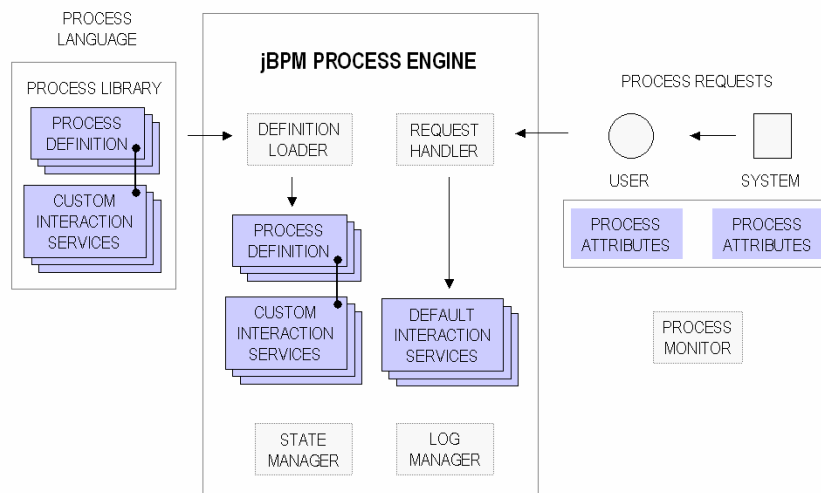
JBoss sees jBPM being applied in three scenarios:

1. In an enterprise application as an application component: A company developing an HR system on a J2EE platform can incorporate JBoss jBPM functionality as easily as including a library.
2. To deliver process based applications: An ERP vendor can include JBoss jBPM in their product and implement their process based software above it. The application can

additionally expose the JBoss jBPM engine to power users of the ERP product. That allows the application to be easily extendible and customizable, a benefit not well-addressed by in current generation ERP packages.

3. As a component in an enterprise architecture: An enterprise can deploy JBoss jBPM as a separate component in the enterprise IT infrastructure. JBoss jBPM provides standardized, reliable business process management that parallel the status of database software in managing corporate data.

To meet each of these various scenarios, JBoss jBPM specifically includes the components illustrated below.



Copyright 2004 John Koenig

1. **Process Engine.** The process engine keeps track of the states and variables of all active processes. It includes:
  - **A Request Handler:** This is the communication infrastructure that forwards tasks to the appropriate process, user or application.
  - **Interaction Services:** These are standard and custom services that expose existing applications as functions or data for use in end-to-end processes.

- A State Manager: This module handles potentially thousands of processes including interlocking records and data, and prepares multi-table databases of record as the outcome of actions.
2. Process Monitor: This module provides visibility into the current end-to-end state of processes with which users and applications are interacting. It enables tracking of the status of users or applications that are performing a process.
  3. Process Language: The core engine is based on a directed graph. JPDL, the current jBPM process language, is a powerful extension. On top of the directed graph core engine, can be build support for other standards like BPEL, BPELJ, BPML, ebXML's BPSS, WSCI and WfMC's XPD.

### **jBPM Focuses on Core Engine Technology**

The primary focus of JBoss jBPM development has been the BPM core engine. Properly executing processes across multiple applications is a complicated task. To accomplish this challenge, a business process engine should have a solid technical basis, such as a mathematical core in pi-calculus. This branch of computer science was initially developed for the mobile market and provides the mathematical foundation for jBPM management of similar process complexity. The jBPM engine is based on leading academic research led by Professor van der Aalst in the Netherlands as well as extensive user feedback.

JBoss jBPM incorporates the vision of its architect Tom Baeyens, that a workflow engine must support a combination of i) declarative specification of the state of a workflow and ii) programming logic. The JBoss jBPM 2.0 engine is designed with two main principles in mind. First, it provides a very simple mechanism for incorporating a basic state machine, making it easy for Java developers to bundle JBoss jBPM into their projects. Second, it is designed to scale to the most complex workflow patterns. In fact, JBoss jBPM is the first BPM engine to

support comprehensively the workflow patterns common across all commercial products, enabling it to be used in the most complex Java applications.

## **The JBoss jBPM Roadmap**

The JBoss roadmap for jBPM focuses on three areas:

- native BPEL support,
- a visual designer to model workflows, and
- process management capabilities enhancement.

As part of the JBoss middleware solution, jBPM integration is planned with Nukes on JBoss, its open source portal and content management system. jBPM is very flexible and can stand alone in a Java VM, inside of any Java application, inside any J2EE application server, or as part of an enterprise service bus (ESB). JBoss plans extensions to jBPM functionality to deliver a ESB capability. An ESB combines 2 primary parts: a workflow engine and a messaging system. JBoss currently offers JBossMQ as a JMS implementation for messaging and JBoss jBPM as a workflow engine. JBoss offers these 2 components separately for better flexibility. While JBoss does not yet have a complete ESB, many of the pieces are in place for delivering an ESB over the next 12 to 24 months.

## **Conclusion**

In summary, JBoss jBPM provides a high level view of applications that accomplishes several things:

- It facilitates more agile implementation of the processes required by business people.
- It describes business processes in a common dialect that lets business people and developers speak the same language.
- It organizes embedded logic of applications into separate and easily changed “state machines” to allow a new level of processes within businesses.

From the JBoss perspective, customers purchasing expensive software from IBM or BEA actually inhibit the BPM market from moving forward. In looking at the problem, JBoss recognizes the importance of offering a professional open source solution to early technology adopters as way to evaluate and understand BPM technology. In this promising but disruptive way, the JBoss open source approach circumvents many traditional limitations to BPM adoption.

No longer does it require an act of corporate heroism to make the decision to evaluate a BPM solution. Instead, it may only take one or two developers with the motivation and time to experiment with freely available JBoss jBPM open source software to bring about major process improvements. Since independent software vendors and IT departments need time to gain BPM experience by trial, it may take several years before the JBoss vision for jBPM takes hold across the software industry.

## **About the Author**

John Koenig is the founder of Riseforth, Inc., management consultants. John has worked in development and marketing at Fortune 500 companies General Electric and Tektronix, and at startups Myrio and Mercado. He served for six years as corporate counsel and an officer of nCUBE. He has a B.S. from the University of San Francisco and a J.D. from University of California, Hastings. He is a member of the Institute of Electrical and Electronic Engineers and the State Bar of California.

For open source and software-on-demand development and marketing strategies for your business, contact us at:

Riseforth, Inc.  
795 Main St.  
Half Moon Bay, CA 94019

Tel: (650) 726-7775  
Fax: (650) 726-7779

[jkoenig@riseforth.com](mailto:jkoenig@riseforth.com)