



JBossBPM Product Roadmap (Draft)

thomas.diesler@jboss.com
18-Jul-2008

Confidential



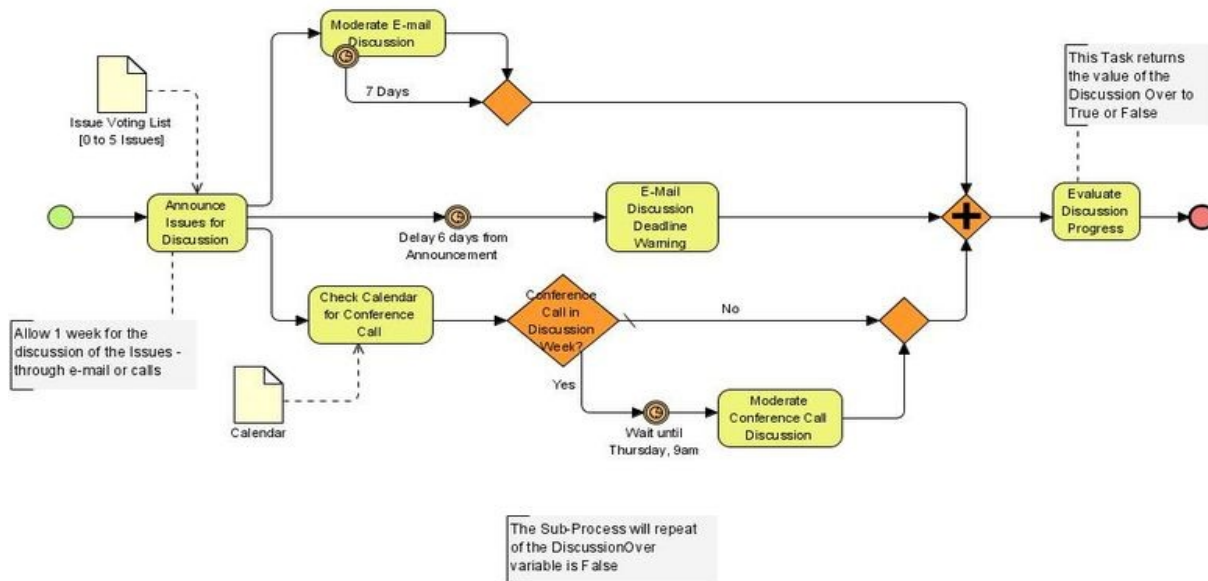
JBossBPM Product Roadmap

The Mission of the JBossBPM Product is to improve the current solution in the following areas

- Modeling
- Functional Completeness
- API Stability
- Reliable Release Cycles
- Target Environment Integration
- Automated Test Coverage
- BI/BAM capability

Modeling

Users will be able to use a BPMN compliant editor to model their processes



BPMN Modelling

The Business Process Modeling Notation (BPMN) is a standardized graphical notation for drawing business processes in a workflow

- BPMN is the most widely accepted standard for process modeling
- Output from BPMN editors may currently not be portable nor directly executable
- BPMN-2.0 is supposed to address the above weaknesses

<http://en.wikipedia.org/wiki/BPMN>

The executable Model

The internal JBossBPM domain model is conceptually close to BPMN

- BPMN models should be transferable to the internal JBossBPM data structures
- All JBossBPM functional tests start from the BPMN model
- The JBossBPM descriptor is easy to read but does not compromise completeness
- Runtime aspects are merged with the BPMN model to get an executable model
- XPD-2.1 is expected to be one of the dialects supported by the executable model



Functional Completeness

Support for every BPMN construct

- We exercise every BPMN construct in automated QA

Support for the established BPM patterns

- Basic Control Patterns
- Advanced Branching and Synchronization Patterns
- Structural Patterns
- Patterns Involving Multiple Instances
- State-based patterns
- Cancellation Patterns

<http://jbpm.dyndns.org/jbpmwiki/index.php?title=BPMPatterns>



API Stability

API model foundation is BPMN

- Leverage an established standard to describe Business Processes.
- Ensure the API model is complete in the sense that it can hold every BPMN model use case

API client/developer view

- JBossBPM clients interact with the API
- Extensions to JBossBPM (i.e. custom Tasks, Events, Gateways) also interact with the API
- A BPM engine might have dependencies on the Implementation



API Stability Promise

We promise stability in these areas

- Application code that uses the API will not be effected by jBPM updates.
- Minor versions of the API are backward compatible.
- The API has support for a number of process descriptor formats, which are also backward compatible between minor versions.
- Database layout is considered implementation detail that might change between minor versions



Reliable Release Cycles

JBossBPM follows these rules

- We have a release cycle of **eight weeks** per minor release
- After two weeks we have **Jira Freeze**
- After six weeks we have **Code Freeze**
- The remaining two weeks are for **QA/Docs**
- All components align with the JBossBPM release cycle

JBossBPM components ownership

- There is an owner for every JBossBPM component
- The owner is responsible that issues associated with that component are resolved in time



Target Environment Integration

Support for target containers

- Support for the last three stable JBossAS releases
- Support target containers are covered by Hudson QA

Support for target databases

- Support for a number of target DBs
- Supported databases are covered by Hudson QA

Support for JDKs

- Support JDK1.5 & JDK6
- Supported JDKs are covered by Hudson QA

IzPack Installer Based Distribution

- The installer based distribution deals with differences of the target environments



Automated Test Coverage

Hudson Test Environment

- The Hudson test environment is an integral part of the JBossBPM code base
- For every release it is possible to restore the QA environment used by that release
- Every component has Hudson test coverage
- The installer based distribution is automatically verified
- There is public access to the Hudson test environment



BI/BAM Capability

JBossBPM provides BI/BAM

- In parallel to the API effort we bring BI/BAM functionality to the product
- Heiko Braun leads the effort
- Scope of initial implementation to be defined



Roadmap Overview

Modeling

- Simple BPMN use cases starting from jpdI-3.2.4 (Q3/2008)
- Following releases build on it

Functional Completeness

- Basic client/runtime API bpm-api-1.0.0.Alpha (Q3/2008)
- More use cases/patterns through API in (Q4/2008)
- Persistence and Transactions through API (Q1/2009)
- Functional completeness (Q2/2009)

API Stability

- bpm-api-1.0.0.GA with completeness (Q2/2009)

Reliable Release Cycles

- starting from jpdI-3.2.4 (Q3/2008)



Roadmap Overview cont.

Target Environment Integration

- IzPack Installer starting from jpdI-3.2.4 (Q3/2008)

Automated Test Coverage

- Integrated Hudson QA starting from jpdI-3.2.4 (Q3/2008)

BI/BAM capability

- Initial implementation (Q4/2008)