



eBPM Creation Review

Planned Review Date: [Date]
Communication Channel: [Forum](#)
Project Lead: Andrea Zoppello

Overview



The eBPM aims at the realization of a **complete BPM solution for OSGi**.

Proposed as subproject of SOA TLP

Will cover the “BPM stuff” in SOA TLP

Excellent feedback received in the proposal forum

Project Scope



Runtime

- **eBPM Core Framework** - framework to define orchestrable OSGi services
- **BPM Engines** on top of core framework (as OSGi bundle / external)
- Provides a set of **standard services and connectors** on top of eBPM Core Framework
- **Management and monitoring** model

UI Tools (based on Eclipse IDE)

- **Service Configuration Editor / Deployment Tools**
(NO PDE overlap, different levels of abstraction)
- **Managing and Monitoring UI Perspective**
- Extension to **existing process modelling tools** (BPMN Modeler, BPEL Designer)

Proposed Components (1)



eBPM Core Framework: Set of bundles, that leverage the capabilities of OSGi Declarative Services, to provide a common way to define/develop OSGi services. eBPM core framework will provide a way to realize OSGi Services based on the messaging paradigm that use the OSGi Event Admin Service to communicate with each others.

eBPM Deployment Service: Bundle that will provide an OSGi Service, that will look for service configuration file and will publish OSGi services using facilities offered by DS and DS Component Factories.

BPM Gateway API: The bundle defining the interface for BPM Gateway implementations.

BPM Gateway Process Engine: The bundle containing an implementation of the API based on an open source process engine.

Standard Connectors and Services: Bundles that provide common connectivity, and business services (SOAP, JMS, File System, FTP, TCP with pluggable applicative protocol implementations, Quartz, XSLT Transformation service, Groovy, JDBC, XSD Validation)

eBPM Monitoring Services: The set of bundle that will leverage information from runtime and store into a relational database.

Metadb-Model: The relational model used by eBPM to store monitoring information. Basically this will be based on the STP Intermediate Model.

Proposed Components (2)



eBPM Project Wizard Tool: Eclipse Plugin to create an eBPM project.

eBPM Service Editor: visual tool for the configurations of OSGi service descriptor. This service descriptor file will be used by eBPM Core Service Framework, to register OSGi services into the runtime. This will not replace PDE tool, as it's intended as a tool to create eBPM service descriptor to be used by eBPM Deployment Service.

BPMN to BPM Gateway Process Engine Bundle: Based on Eclipse STP IM (Mangrove) it produces process definitions to be used by BPM Gateway default implementation (jbpmm) bundle.

BPMN to BPEL: Based on Eclipse STP IM (Mangrove), it produces BPEL processes with using the models realized by means of Eclipse BPMN Designer. The processes are then completed through the BPEL Designer.

Deploy Tools: Utility Tools integrated in Eclipse, that allow to deploy an eBPM project to the runtimes.

Proposed Components (3)



Eclipse BPMN Editor Extension: Extension to the eclipse bpmn editor to support the drag and drop of OSGi service definitions into the diagram.

Eclipse BPEL Editor Extension: Extension to the eclipse bpel editor to support the drag and drop of OSGi service definitions into bpel diagram.

Eclipse eBPM Support Tools: Some support plugins (menu contributions, preference page), to help to work with eBPM.

Monitor and Management Perspective/View: Eclipse plugins to provide a view with the list of processes, process instances, and status information

Extension to BPEL Runtime (eBPM is used in an OSGi runtime embedded in BPEL runtime)

OSGi Extension for the BPEL Runtime: extensions enabling the BPEL runtime to orchestrate OSGi services. It will be initially released for Apache ODE.

Relationship to other Eclipse Projects



Eclipse STP IM (Future Mangrove) – Will be used by eBPM as the internal model to achieve separation between modelling tools and eBPM code generators.

Eclipse BPEL Designer – Will be used and extended by eBPM to allow to model BPEL Processes, able to use eBPM OSGi services/connectors

Eclipse BPMN Modeler - Used and extended by eBPM to allow to model BPMN Processes, able to use eBPM OSGi services/connectors

Eclipse Equinox – eBPM runtime is built on top of OSGi, and will use Equinox as its runtime

Eclipse Swordfish – eBPM runtime can work on Sworfish as OSGi runtime, and benefits of Swordfish features.

Eclipse ECF – eBPM will use the Distributed EventAdmin service provided by ECF and plan to move some of its connectors on top of ECF

EclipseLink – eBPM plans to use Eclipse link as its persistence framework

Eclipse Gemini – eBPM plans to use some components that will be provided by Gemini

Initial Committers (1)



Andrea Zoppello, Engineering

Andrea is an IT Architect with 10 years experience in web application design and development, J2EE and EAI open source frameworks. He's actually active in Spagic design and Development, and on projects focused on EAI, Business Process Management, Workflow, and Data Integration. Since November 2007 has been nominated commiter on Eclipse STP project for his contribution of the STP Intermediate Model.

He'll be the eBPM Project Leader, and he will work on the eBPM Core Frameworks Architecture and on the definitions of the BPM Gateway API.

Gianfranco Boccalon, Engineering

Gianfranco, serving as Spagic and Spago Project Leader, is a Senior Consultant at Engineering Ingegneria Informatica with long experience in the IT field. He has valuable experience in J2EE architectures and frameworks, in the SOA environment and in software development methods, with a specific focus on performance testing and measurements. He is currently working on Spagic Evolution in the SOA domain. He has been contributing to the definition of IM Runtime for both JBI and BPEL and to plug-in development For IM integration with the BPMN editor. In the eBPM project he will work on the eBPM Metadb model definition and on the architecture and development of eBPM monitoring and management features.

Luca Rossato, Engineering

Luca is an IT Developer with 3 year experience in web application, J2EE and SOA. On March 2009 he joined Spagic group and in the last year he participated in the development of the Spagic platform. He has been contributing to the definition of IM Runtime for both JBI and BPEL and to plugins development for Im integration with BPMN editor. In the eBPM project he will work on the development of eBPM standard connectors and services.

Initial Committers (2)



Luca Barozzi, Engineering

Luca is an IT Architect and Developer. Since 2000 he's been designing and developing J2EE and EAI open source frameworks based web applications.

He contributed to Spagic 3 Studio the eclipse form toolkit based Service Editor.

On STP project he contributed to definition of JBI IM Runtime and IM Runtime integration with jBPM.

In eBPM project he'll work on the UI toolset, in particular on Service Configuration Editor and deployment wizard stuff.

Antonietta Miele, Engineering

Antonietta is an IT Developer with experience in J2EE , SOA .She has acquired knowledge on business process modelling and mapping, working as a consultant in some firms in the services and telecommunications markets, before joining the Spagic group in May 2009. She has been contributing to the Spagic development.

In eBPM project she will develop services and connectors, and will be responsible for the project wiki, forums, and documentation.

Daniela Butano, Independent

Daniela is an IT Solution Architect and Java Software Developer in Cambridge, UK.

She has worked at Engineering Ingegneria Informatica and in the last two years she has been exposed to the current architectures SOA and ESB.

At present, as team member of Spagic project, she has taken part in the design and development of various system components. In particular, she has contributed to develop the eclipse plug-ins to generate BPEL processes from STP intermediate models, the eclipse plug-in to generate an STP intermediate model from BPEL process, a listener for monitoring BPEL activities executed by Apache ODE, some extensions to the BPEL designer. In the eBPM project she will work on eBPM BPEL integrations.

Mentors



Oliver Wolf – [SOPERA](#)

Dave Carver – [Intalio](#)

Oisin Hurley - Independent



Code Contribution

The initial eBPM code contribution will be based on Spagic project codebase (www.spagic.org)

Well known and accepted BPM/SOA solution

Proven and used in real projects

Based on OSGi (from version 3)

Moving to eclipse is a natural evolution

Spagic is already powered by Eclipse projects

- STP – Intermediate Model (will be Mangrove)

eBPM will use Parallel IP Process

Need to solve some IP Issues in the actual Spagic codebase

IP Issues resolution strategy

1. Replaces the actual dependencies with ORBIT bundles
2. Submit CQs for dependencies that are not available in ORBIT bundles
3. Introduce Abstraction levels
 - API Bundles (+ pluggable implementations)

Plan



EclipseCON 2010

Project Presentation

June 2010

Planned Release of the first version (0.7)

eBPM aim to participate at the annual eclipse release train as part of the SOA TLP Project.