



**Project Title:** BIO\_SOS Biodiversity Multisource Monitoring System: from Space TO Species

**Contract No:** FP7-SPA-2010-1-263435

**Instrument:** Collaborative Project

**Thematic Priority:** FP7-SPACE-2010-1

**Start of project:** 1 December 2010

**Duration:** 36 months

Deliverable No: D.3.5

## **AIQ: System Assembly Integration and Qualification Plan**

**Due date of deliverable:** 30/06/2013

**Actual submission date:** 05/07/2013

**Version:** 1st version of D3.5

**Main Authors:** Dimitrios Karachalios (PKH) ; Diomedes Illuzzi (PKI)



<b>Project ref. number</b>	<b>263435</b>
<b>Project title</b>	<b>BIO_SOS: Biodiversity Multisource Monitoring System: from Space to Species</b>

<b>Deliverable title</b>	AIQ – System Assembly Integration and Qualification Plan
<b>Deliverable number</b>	D3.5
<b>Deliverable version</b>	v1
<b>Previous version(s)</b>	
<b>Contractual date of delivery</b>	30/06/2013 (month 31)
<b>Actual date of delivery</b>	05/07/2013
<b>Deliverable filename</b>	BIO_SOS_D3.5_AIQ_v1
<b>Nature of deliverable</b>	R = Report
<b>Dissemination level</b>	PU = Public
<b>Number of pages</b>	20
<b>Workpackage</b>	WP 3
<b>Partner responsible</b>	PKI
<b>Author(s)</b>	Dimitrios Karachalios (PKH) Diomedes Illuzzi (PKI)
<b>Editor</b>	Diomedes Illuzzi (PKI)
<b>EC Project Officer</b>	Florence Beroud

<b>Abstract</b>	The AIQ describes the System Assembly Integration of the system.
<b>Keywords</b>	System, architecture, workflow

## Signatures

Written by	Responsibility- Company	Date	Signature
Diomede Illuzzi	Technical staff WP 3 (PKI)	28/06/2013	
Dimitrios Karachalios	Technical staff WP 3 (PKH)	28/06/2013	
Verified by			
Jens Stutte	WP3 Leader (PKI)	01/07/2013	
Jordi Inglada	WP5 Participant (UPS)	05/07/2013	
Approved by			
Palma Blonda	Project Coordinator, CNR	05/07/2013	
Fifamè Koudogbo	Quality Team, AI	05/07/2013	

## Table of Contents

1.	Scope of the AIQ document.....	5
1.1.	EODHaM Business Process Management Engine .....	5
1.1.1.	Business Process Execution Language (BPEL).....	5
1.1.2.	Business Process Model and Notation (BPMN) .....	5
1.1.3.	Strategic decision for the EODHaM Business Process Management Engine .....	6
1.1.4.	EODHaM Business Process Management implementation with JBPM5 Suite .....	6
1.2.	Setup the EODHaM Process Management Engine .....	7
1.2.1.	Prerequisites.....	7
1.2.2.	Download the installer.....	8
1.2.3.	Setup the application .....	8
1.2.4.	Start the application .....	8
1.2.5.	Import the BIO_SOS Processing Chain .....	9
1.2.6.	Using web management console .....	9
1.2.7.	Development tools for the BIO_SOS Process under EODHaM JBPM Engine .....	10
1.2.7.1.	JBPM Eclipse Plug-in .....	11
1.2.7.2.	Guvnor repository and Designer.....	12
1.2.8.	BPMN2 diagrams of BIO_SOS Process .....	13
1.3.	BIO_SOS Wrapper Module .....	14
1.3.1.	Prerequisites.....	14
1.3.2.	BIO_SOS Processor Wrapper Module Setup.....	15
2.	References .....	17
3.	Appendix I. Acronym and Abbreviation List.....	19

## **1. Scope of the AIQ document**

This document provides detailed information about the EODHaM system structure and composition. The qualified assembly of different building blocks, derived from different applied technologies, characterizes the EODHaM system integration as a complete management solution of the BIO\_SOS work-flow. Detailed information about the installation of each principal building block, as their architectural role in the system is given in the following paragraphs.

### **1.1. EODHaM Business Process Management Engine**

The core of the EODHaM system is based on a Business Process Management Engine. A business process or business method is a collection of related , structured activities or tasks that produce a specific service or product (serve a particular goal) for a particular customer or customers. It often can be visualized with a flowchart as a sequence of activities with interleaving decision points or with a Process Matrix as a sequence of activities with relevance rules based on the data in the process. Two principal standards are used for the dynamic expressive development of a Business Process:

The BPEL standard and the BPMN standard: these two languages affronts the management of Business Process from different point of view, but are used for the same purpose, the efficient and successfully design, management and execution of the Business Process.

#### **1.1.1. Business Process Execution Language (BPEL)**

Business Process Execution Language (BPEL), short for Web Services Business Process Execution Language (WS-BPEL) is an OASIS standard executable language for specifying actions within business processes with web services. Processes in BPEL export and import information by using web service interfaces exclusively. WS-BPEL provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the Web Services interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that should facilitate the expansion of automated process integration both within and between businesses.

#### **1.1.2. Business Process Model and Notation (BPMN)**

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD), based on a flowcharting technique very similar to activity diagrams from Unified Modeling Language (UML).

The objective of BPMN is to support business process management, for both technical users and business users, by providing a notation that is intuitive to business users, yet able to represent complex process semantics. The BPMN specification also provides a mapping between the graphics of the notation and the underlying constructs of execution languages, particularly Business Process Execution Language (BPEL).

The primary goal of BPMN is to provide a standard notation readily understandable by all business stakeholders. These include the business analysts who create and refine the processes, the technical developers responsible for implementing them, and the business managers who monitor and manage them. Consequently, BPMN serves as a common language, bridging the communication gap that frequently occurs between business process design and implementation.

### **1.1.3. Strategic decision for the EODHaM Business Process Management Engine**

As already described, on the ADD, initial, the BPEL standard was selected as the technological choice in which should have built the EODHaM Management Process Engine. The BPEL itself is a language, a standard that fits well the needs of our system. Although the language itself is built well, we cannot express the same satisfaction about the OPENSOURCE BPEL Engines that provide execution functionalities for the BPEL language. During the development with open source BPEL engine solutions, different unusual behaviors due software bugs have been taken place. The conclusion is that a technological language is good enough if the software solutions that use that language, works also well.

The BPMN specifications are compatible with the initial purpose of our goal, the design and execution of BIO\_SOS tasks under an automatic workflow of business process.

Another consideration is that the BPEL requires some amount of coding Knowledge to design a business process, so can be called as “Developer centric”. BPMN in the other hand offers an easiest way to design the business process so it can be done by the user itself, as long as he is aware of BPMN notations. That gives more flexibility and realistic expansion to the EODHaM scientific providers, as they can design easier further BIO\_SOS business process.

For the benefit of the EODHaM system and after ulterior analysis, we took into consideration the alternative solution of the BPMN language and we focused for software solutions that have a mature robust Engine that should permit the development and execution of the Business Process with no problems. That decision does not influence negative the architecture of the initial EODHaM system. The initial architecture design remains invariable to that change.

### **1.1.4. EODHaM Business Process Management implementation with JBPM5 Suite**

jBPM is an extensible and flexible process engine that can run as a standalone server or embedded in any Java application. JBoss JBPM is a sophisticated state machine. JBoss jBPM delivers workflow, business process management (BPM) and service orchestration in a multi-process language platform.

Its light-weight, fully open-source (distributed under Apache license) and written in Java. It can run in any Java environment. It allows the user to model execute and monitor business processes, throughout their life cycle.

Traditional BPM engines have a focus that is limited to non-technical people only. jBPM has a dual focus: it offers process management features in a way that both business users and developers like it. The core of jBPM is a light-weight, extensible workflow engine written in pure Java that allows you to execute business processes using the latest BPMN 2.0 specification. Complex business logic can be modeled as a combination of business processes with business rules and complex event processing.

JBoss Jbpm executes long running processes that can be visualized graphically and enables automation of business processes that coordinate between people, applications and services. jBPM5 is the latest community version of the jBPM project. It is based on the BPMN 2.0 specification and supports the entire life cycle of the business process (from authoring through execution to monitoring and management).

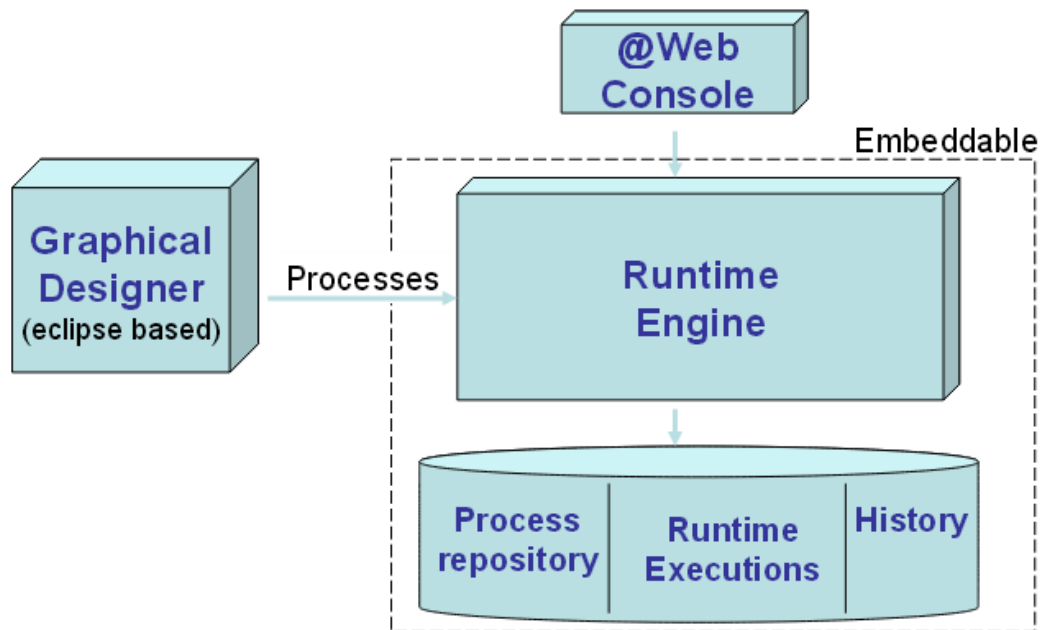


Figure 1 – JBPM typical components

The current jBPM5 snapshot offers open-source business process execution and management, including

- embeddable, lightweight Java process engine, supporting native BPMN 2.0 execution
- BPMN 2.0 process modeling in Eclipse (developers) and the web (business users)
- process collaboration, monitoring and management through the Guvnor repository and the web console
- human interaction using an independent WS-HT task service
- tight, powerful integration with business rules and event processing

jBPM is distributed under the terms of the GNU Lesser General Public License (LGPL) and the JBoss End User License Agreement (EULA).

## 1.2. Setup the EODHaM Process Management Engine

This guide will assist the user in installing and running a setup of the various components of the jBPM project needed for the implementation of the EODHaM Process Management Engine.

### 1.2.1. Prerequisites

This script assumes that Java JDK 1.5+ (set as JAVA\_HOME), and Ant 1.7+ have been installed.

If not, the following links must be used to download and install them:

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

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

### 1.2.2. Download the installer

It is recommend to download the full installer from the sourceforge.net

<http://sourceforge.net/projects/jbpm/files/jBPM%205/jbpm-5.4.0.Final/jbpm-5.4.0.Final-installer-full.zip/download>

### 1.2.3. Setup the application

The easiest way to get started is to simply run the installation script to install the demo setup. Simply go into the install folder and run:

```
ant install.demo
```

This script will make a series of download and installation actions for all the necessary components:

- It will download JBoss AS
- It will download Eclipse
- It will install the Drools Guvnor component into JBoss AS
- It will install the jBPM Designer into JBoss AS
- It will install the jBPM console into JBoss AS
- It will install the jBPM Eclipse plugin
- It will install the Drools Eclipse plugin

That step of the procedure is expected to take a lot of time to accomplish the download and installation of the application server as the Eclipse installation too. During the installation execution the script however shows which files it is downloading.

### 1.2.4. Start the application

Once the setup has finished, the application can be start with the following command:

```
ant start.demo
```

That script enables the execution of the following application components:

- Starts the H2 database
- Starts the JBoss AS
- Starts Eclipse
- Starts the Human Task Service

Once everything is started, the JBPM application is ready for use.

If we want to exclude the use of the Eclipse component, in the setup procedure we can use the alternative commands:

```
ant install.demo.noclipse
```

```
ant start.demo.noecclipse
```



### 1.2.5. Import the BIO\_SOS Processing Chain

Once Eclipse has opened, simple import using File then command Import, and then under the General category, select “Existing Projects into Workspace”.

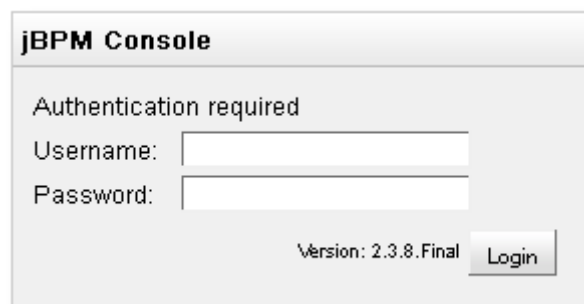
This procedure should add the selected BIO\_SOS Processing Chain process, including a simple BPMN2 process and a Java file to start the process. To execute the process, right click on ProcessMain and select “Run As – Java Application”.

### 1.2.6. Using web management console

Open up the process management console [http:// biosos.planetek.it/ jbpm-console](http://biosos.planetek.it/jbpm-console)

Log in, using the right credential for username and password.

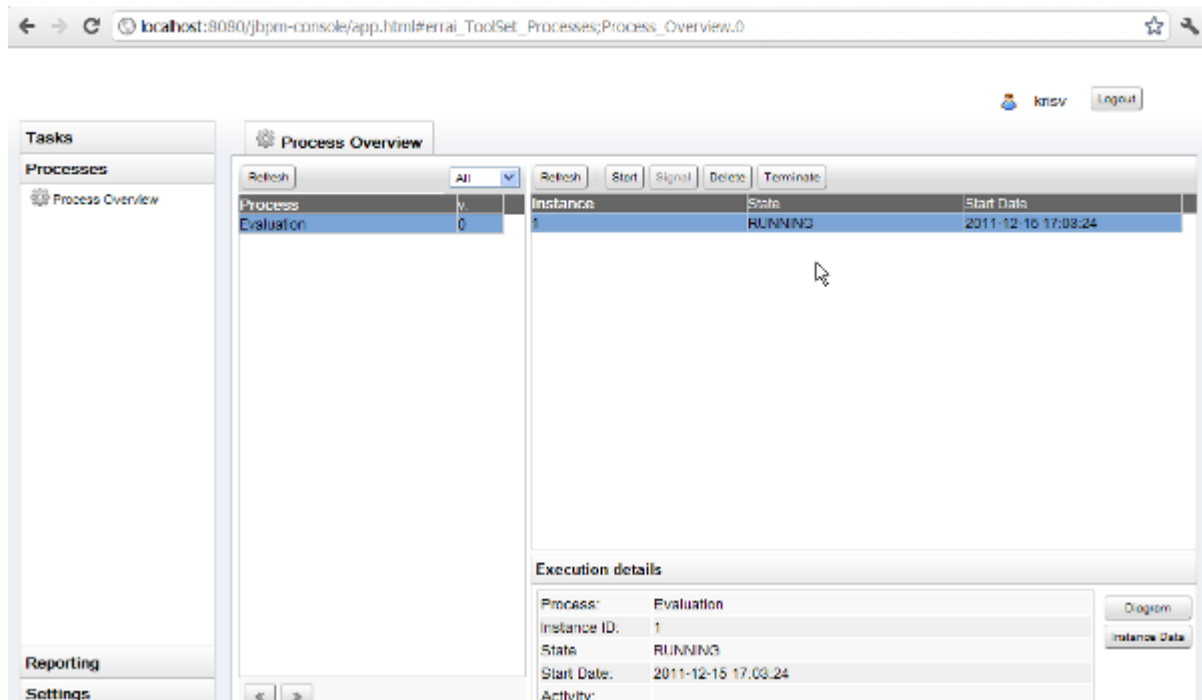
Note: The user name and the password are sensitive information. They have to be requested to the administrator and have to be kept private by the scientific users that have the right to access to the EODHaM system.



**Figure 2 – JBPM Console Log in**

The authorized EODHaM user , after a successful log in , can use the JBPM console to manage the process instance. Through the jbm-console the user can

- Start a new process
- Look up the current status of a running process instance
- Look up tasks
- Complete a task
- Generate reports to monitor the process execution



**Figure 3 – JBPM -console Process Overview**

An overview of the functionalities of the console can be obtained on the following link:

<http://people.redhat.com/kverlaen/install-gwt-console-jbpm.swf>

To manage your process instances, click on the "Processes" tab at the left and select "Process Overview". After a slight delay (if the application opens for the first time, due to session initialization), the "Process" list should show all the known processes. The jBPM currently loads all the process in the "src/main/resources" folder of the Processing Chain process. If the user clicks the process, all current running instances appears.

The user can start a new process instance by click on the "Start" button. After confirming the new execution of the selected process, the user will see a process form where the user needs to fill in the necessary information to start the process. A new instance should show up in the "Instance" table. If the user clicks the process instance, can check its details below and the diagram and instance data by click on the "Diagram" and "Instance Data" buttons respectively. The process instance that the user just started is first requiring a self-evaluation of the user and is waiting until the user has completed this task.

After the starting and/or completing process instances, the user can generate a report of the executed process. Under "Reporting", select "Report Templates". By default, the console has two report templates, one for generating a generic overview for all processes and one for inspecting once specific process definition. For the specific process definition report, an ID related to the selected process has to be provided. The user has to click the "Create Report" button to generate a real-time report of the current status.

### 1.2.7. Development tools for the BIO\_SOS Process under EODHaM JBPM Engine

For the development of the process, jBPM comes with two development tools: the Eclipse tooling and the Guvnor Repository and Designer. Both of them are installed during the full installation procedure.

### 1.2.7.1. JBPM Eclipse Plug-in

The jBPM project has an Eclipse plug-in that simplifies developing applications with JBPM.

The plug-in supports:

- **Design:** This is where the actor creates (or updates) a business process, mostly by using graphical representation. This includes drawing the control flow of the process by dragging and dropping various nodes onto the canvas, filling in the properties of these nodes, etc. Several tools might exist, each targeted to a different type of actor (e.g. business analyst vs. developer)
  - **Domain-specific extensions:** One feature that can prove to be invaluable is the ability to create domain-specific processes. Domain-specific languages are targeted to one particular application domain and therefore can offer constructs that are closely related to the problem the user is trying to solve. The ability to extend the business process with high-level, declarative, domain-specific extensions usually makes processes easier to author, understand and self-documenting.
- **Testing:** Once a process has been defined, it should be possible to test specific process scenarios, and to bundle these as a test suite.
- **Debugging:** To figure out what is going on during the execution of your processes, debugging capabilities allow you to temporally halt the execution, inspect the state of your process instances, what happened so far, etc.
- **Simulation:** This allows you to completely simulate specific use cases (and could thus be seen as an extension of testing and debugging). It includes user-friendly visualization of progress, the ability to manipulate the clock, sending triggers and validating assertions at specific points during the execution, batch processing, etc.

An overview of the Eclipse JBPM functionalities can be obtained on the following link:

<http://people.redhat.com/kverlaen/install-eclipse-jbpm.swf>

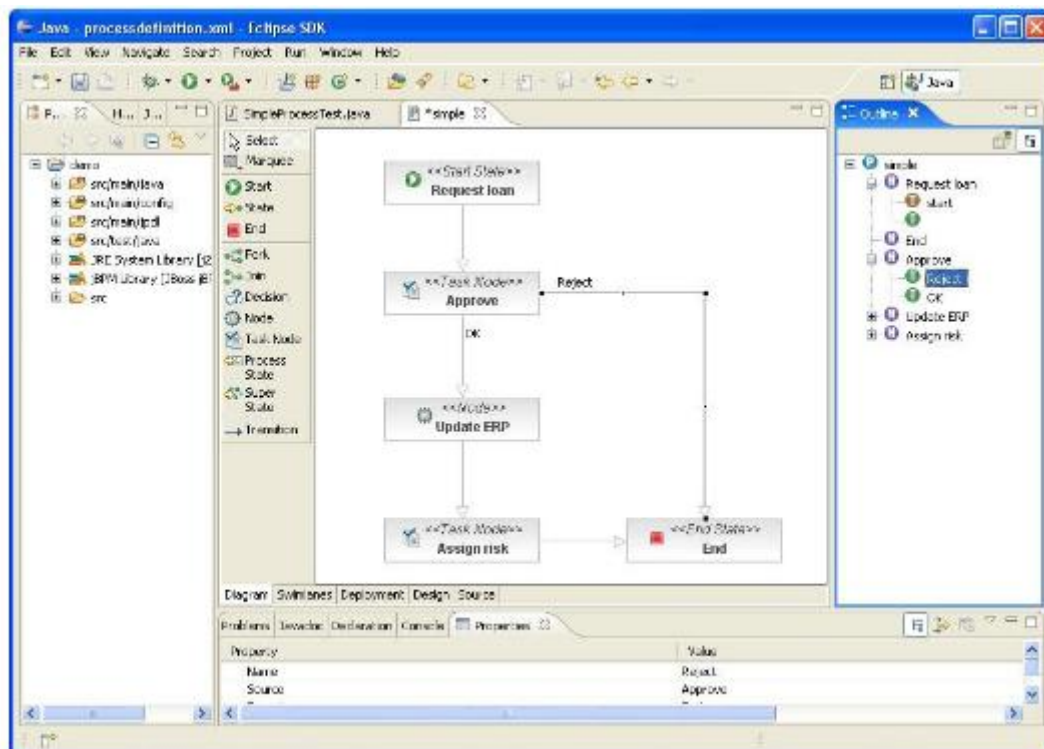


Figure 4 – Eclipse Plug-in

### 1.2.7.2. Guvnor repository and Designer

The Guvnor repository can be used as a process repository to store business processes. Using the installer, the latest version of Guvnor is downloaded and installed as well. It also offers a web-based interface to manage the processes. This includes a web-based editor for viewing and editing processes. For the access to the EODHaM Guvnor web-based interface, the user has to open up the

<http://biosos.planetek.it/drools-guvnor>

The process repository is the location where the user stores and manages business processes. Guvnor acts as a combination of a source code management system, content management system, collaboration suite and development and testing environment. These are the kind of features supported from the process repository:

- Persistent storage of the user processes so the latest version can always easily be accessed from anywhere, including versioning.
- Build and deploy selected processes.
- User-friendly (web-based) interface to manage, update and deploy processes (targeted to business users, not just developers).
- Authentication / authorization to make sure only people that have the right role can see and/or edit your processes.
- Categorization and searching.
- Scenario testing to make sure that the user doesn't break anything when the administrator changes the process.
- Collaboration and other social features like comments, notifications on change, etc.
- Synchronization with the user development environment.

The jBPM-console is configured to get all this information from Guvnor whenever necessary and show them in the console.

Name	Last modified	Status	Categories
Underage	Dec 19, 2008	Draft	Eligibility rules
Bankruptcy history	Oct 1, 2008	Draft	Eligibility rules
No bad credit checks	Oct 1, 2008	Draft	Eligibility rules
no NINJAs No ninjas !	Oct 2, 2008	Draft	Eligibility rules
Pricing loans	Jan 27, 2009	Draft	Pricing rules
CreditApproval	Oct 22, 2008	Draft	Eligibility rules
DateDslRule	Oct 24, 2008	Draft	Technical
CheckBoxDslRule	Oct 23, 2008	Draft	Technical
RegexDslRule	Oct 23, 2008	Draft	Technical
wee	Jan 27, 2009	Draft	Home Mortgage

**Figure 5 – Drools-Guvnor Console**

The following screen cast gives an overview of how to manage the repository:

<http://people.redhat.com/kverlaen/jBPM5-guvnor-integration.swf>

### 1.2.8. BPMN2 diagrams of BIO\_SOS Process

Business Process Model and Notation (BPMN) is a standard for business process modeling that provides a graphical notation for specifying business processes in a Business Process Diagram (BPD), based on a flowcharting technique very similar to activity diagrams from Unified Modeling Language (UML). The objective of BPMN is to support business process management, for both technical users and business users, by providing a notation that is intuitive to business users, yet able to represent complex process semantics.

The last specification is the BPMN 2.0, as defined on the <http://www.omg.org/spec/BPMN/2.0/>

With the Eclipse Plug –in, the BIO\_SOS Workflow Manager (as the Processor Providers) are able to define a new BPMN 2.0 process (under File -> New –Other ...select BPMN –BPMN2 Diagram).

The BIO\_SOS Workflow Manager, using the tools of the JBPM, can create and deploy different Business Process of the Processing Chain activities, taking into consideration the specific observed BIO\_SOS site. For example in the figure 6 and figure 7 we can see, the diagram presentations, using the BPMN language, of the Processing Chain tasks that are defined for the execution of the Business Process for the monitoring areas of Cors Fochno, Wales and the area Le Cesine, Italy.

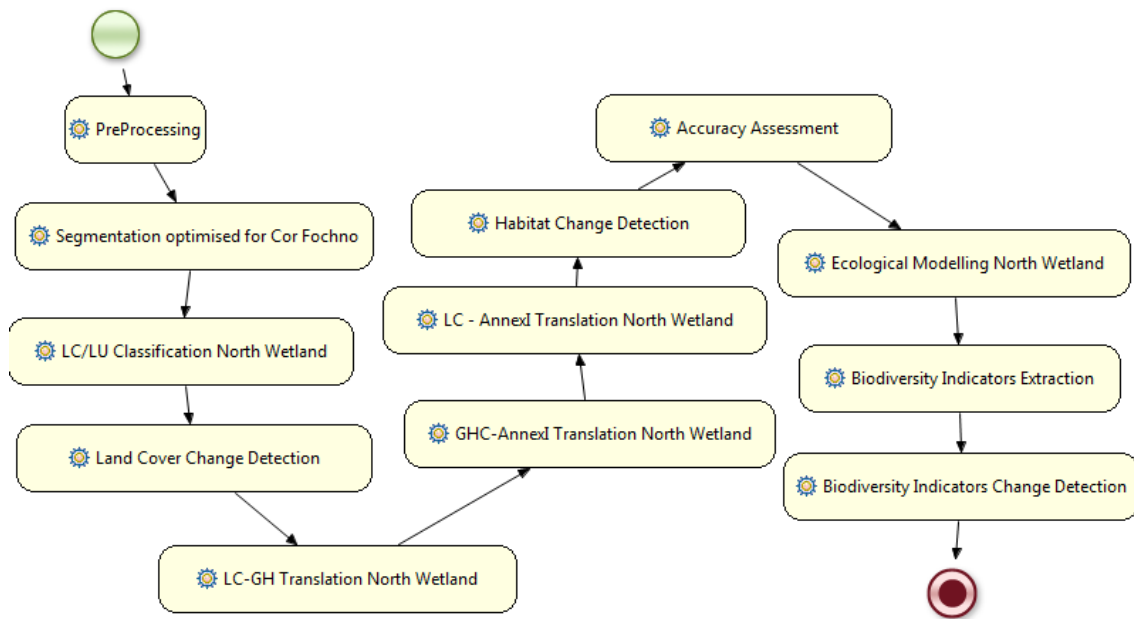
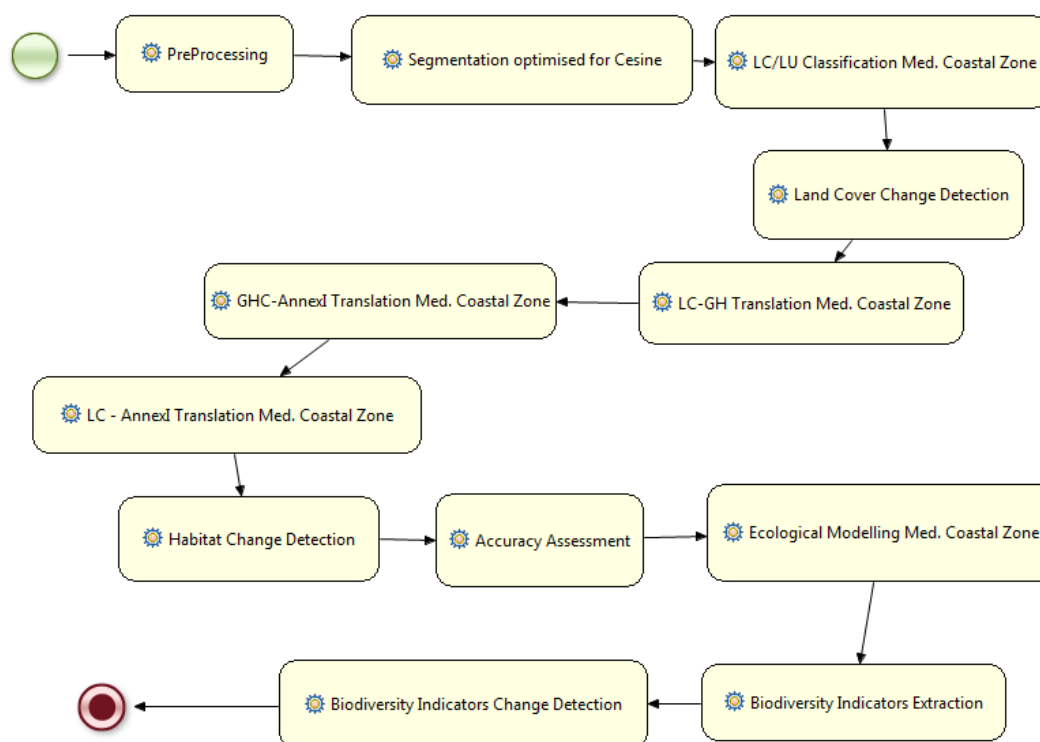


Figure 6 – BPMN diagram- Use case of Nordic Wetland Workflow (Cors Fochno, Wales)



**Figure 7 – BPMN diagram Use case of Mediterranean Wet (Le Cesine, Italy)**

### 1.3. BIO\_SOS Wrapper Module

The wrapper module has the responsibility to expose the BIO\_SOS processor functionality as web services. Wrapper generally refers to a type of packaging. Through the WSDL Interface permits the communication with the other services and activates the internal processing activities in demand of the input messages. After the generation of the output products, replies the response to the requester.

#### 1.3.1. Prerequisites

The BIO\_SOS Wrapper Module has to be deployed to an Apache Tomcat. Apache Tomcat is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run in.

Before installing Tomcat assure that Java JDK 1.6.0\_32 (set as JAVA\_HOME), and Ant 1.7+ have been installed.

If not, the following links must be used to download and install them:

Java: <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u32-oth-JPR>

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

To install Tomcat (the last version available), the official guide "Tomcat Setup" has to be following. The setup has to be done taking in consideration the operating system. For the BIO\_SOS purpose a CentOS machine has been used as host machine. More installation details on <http://tomcat.apache.org/tomcat-7.0-doc/setup.html>

After the installation of the tomcat one more component is required. The required component is the JAX-WS RI 2.2.5. JAX-WS RI 2.2.5 is a Web Services framework that provides tools and infrastructure to develop Web Services solutions for the end users and middleware developers. With JAX-WS RI 2.2.5, clients and web services have a big advantage: the platform independence of the Java programming language.

The next steps have to be done:

- Download the file JAXWS2.2.5-20110729.zip from the <https://jax-ws.java.net/2.2.5/>
- Extract the content to a temporary folder.
- Copy the .jar contents of the sub folder  
`${temp_folder}\JAXWS2.2.5-20110728\jaxws-ri\lib\*.*`  
to the subfolder of the installed tomcat  
`${Path location of installed tomcat}\Apache Software Foundation\Tomcat 7.0\lib\*.*`
- Under the installed JDK folder create a new folder named “endorsed” in the following location  
`${Path location of installed JDK}\jdk1.6.0.32\jre\lib\endorser`
- Copy from the `${temp_folder}\JAXWS2.2.5-20110728\jaxws-ri\lib\*.*` the `jaxb-api.jar` and the `jaxws-api.jar` to the `${Path location of installed JDK}\jdk1.6.0.32\jre\lib\endorser\*.*`
- Under the installed tomcat create a new folder named “endorsed” in the following location  
`${Path location of installed tomcat}\Apache Software Foundation\Tomcat 7.0\endorsed\*.*`
- Copy from the `${temp_folder}\JAXWS2.2.5-20110728\jaxws-ri\lib\*.*` the `jaxb-api.jar` and the `jaxws-api.jar` to the `${Path location of installed tomcat}\Apache Software Foundation\Tomcat 7.0\endorsed\*.*`

### 1.3.2. BIO\_SOS Processor Wrapper Module Setup

A folder called `biosos\conf` that contains the necessary configuration files for the wrapper module; it has to be put in a defined location. The location of the configuration folder is specified in **configuration.properties** as variable **CONFIG\_REPOSITORY**

Customization of the Wrapper Processor can be done, following the next guide instructions:

1) Customize the implementation of: **eu.biosos.processing.server.Processor**, it's the class that includes general behavior of your processor in accordance to the **IProcessor** interface; at least the **'invokeProcessor'** has to be customized, see the javadoc for further details. Put in **processor.config.xml** the configuration generally valid for your processor in all the use cases

2) Define how many use cases your processor needs to implement; for each use case you must do the following:

- define a class extending **DomainProcessor** and implementing the **'IDomainProcessor'** interfaces; you can see `ProcessorUseCase1` as a reference.
- add a new servlet definition in `etc/web.xml`
- add a new endpoint reference in `etc/sun-jaxws.xml`, with an url-pattern matching the newly defined servlet matching the serviceName defined into the `@WebService` annotation of the implementing class (you'll be able to access then the wsdl at `http://<your-host>:<your-port>/biosos/<your-service-name>?wsdl`).
- add a configuration file with same name as the implementing class (no matter the case) including the parameters specific for this use case (see **processor.config.xml**; so if you implement the class `usecasex` you have to create a `usecasex.config.xml` file containing all the configuration parameters affecting that use case).



Then, the **Biosos\_wrapper.war** file has to be put into the subfolder **\${Path location of installed tomcat}\Apache Software Foundation\Tomcat 7.0\webapps\\*.\*** of the installed tomcat.

After that, start the tomcat using the **\${Path location of installed tomcat}\Apache Software Foundation\Tomcat 7.0\bin\Tomcat7.exe**. The wrapper has to be deployed and the following endpoints are available for the service.

## Web Services

Endpoint	Information
Service Name: {http://server.processing.biosos.eu/} x-usecase1 Port Name: {http://server.processing.biosos.eu/} ProcessorUseCase1Port	Address: <a href="http://localhost:8088/jaxws-biosos/x-usecase1">http://localhost:8088/jaxws-biosos/x-usecase1</a> WSDL: <a href="http://localhost:8088/jaxws-biosos/x-usecase1?wsdl">http://localhost:8088/jaxws-biosos/x-usecase1?wsdl</a> Implementation class: eu.biosos.processing.server.ProcessorUseCase1
Service Name: {http://server.processing.biosos.eu/} x-usecase2 Port Name: {http://server.processing.biosos.eu/} ProcessorUseCase2Port	Address: <a href="http://localhost:8088/jaxws-biosos/x-usecase2">http://localhost:8088/jaxws-biosos/x-usecase2</a> WSDL: <a href="http://localhost:8088/jaxws-biosos/x-usecase2?wsdl">http://localhost:8088/jaxws-biosos/x-usecase2?wsdl</a> Implementation class: eu.biosos.processing.server.ProcessorUseCase2

**Figure 8 – Wrapper Biosos Processor web service endpoints**

After that the web service of the BIO\_SOS Processor wrapper is successfully installed and ready to be invoked by the workflow of a executed Business Process provided from the JBPM.



## 2. References

1. Web Service,
  - [http://en.wikipedia.org/wiki/Web\\_service](http://en.wikipedia.org/wiki/Web_service)
  - [http://www.w3schools.com/webservices/ws\\_intro.asp](http://www.w3schools.com/webservices/ws_intro.asp)
2. Web Services Business Process Execution Language (WS-BPEL) ,
  - <http://docs.oasis-open.org/wsbpel/2.0/varprop>
  - <http://www.information-management.com/infodirect/20060602/1055720-1.html?zkPrintable=1&nopagination=1>
3. Business Process Execution Language for Web Services (BPEL4WS),
  - <http://xml.coverpages.org/bpel4ws.html>
4. Web Services Description Language (WSDL) 1.1 ,
  - <http://www.w3.org/TR/wsd/>
  - <http://www.w3.org/TR/ws-desc-usecases/>
  - <http://www.w3.org/TR/ws-arch/>
5. Web Services Description Language (WSDL) Version 2.0,
  - <http://www.w3.org/TR/wsd20/>
6. Simple Object Access Protocol,
  - <http://en.wikipedia.org/wiki/SOAP>
  - [http://www.w3schools.com/soap/soap\\_intro.asp](http://www.w3schools.com/soap/soap_intro.asp)
  - <http://www.w3.org/TR/ws-addr-soap/>
7. Service-Oriented Architecture,
  - [http://en.wikipedia.org/wiki/Service-oriented\\_architecture](http://en.wikipedia.org/wiki/Service-oriented_architecture)
  - <http://msdn.microsoft.com/en-us/library/aa480021.aspx>
8. Apache ODE, BPEL, Language Guide, WS-BPEL 2.0 Specification Compliance ,
  - <http://ode.apache.org/ws-bpel-20-specification-compliance.html>
9. Virtual Machine,
  - [http://en.wikipedia.org/wiki/Virtual\\_machine](http://en.wikipedia.org/wiki/Virtual_machine)

10. JBPM

- [http://docs.huihoo.com/jboss/jbpm/5.4.0/html\\_single/#d0e481](http://docs.huihoo.com/jboss/jbpm/5.4.0/html_single/#d0e481)

11. BPMN

- [http://en.wikipedia.org/wiki/Business\\_Process\\_Model\\_and\\_Notation](http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation)
- <http://www.omg.org/spec/BPMN/2.0/>
- <http://www.omg.org/spec/BPMN/2.0/PDF>

12. Guvnor

- <http://www.jboss.org/drools/drools-guvnor.html>

### 3. Appendix I. Acronym and Abbreviation List

ADD	Architecture Design Document
API	Application Programming Interface
AIQ	System Assembly Integration and Qualification Plan
BIO_SOS	BIOdiversity multi-SORce monitoring System: from Space TO Species
BPEL	Business Process Execution Language
BPEL4WS	Business Process Execution Language for Web Services
BPM	Business Process Management
BPMN	Business Process Model and Notation
BIO_SOS	Biodiversity Multi-Source Monitoring System: From Space To Species
CEOS	Committee of Earth Observations
CERTH	Informatics And Telematics Institute Of The Centre For Research And Technology – Greece
CIBIO	Research Center in Biodiversity and Genetic Resources (Portugal)
CNR	Consiglio Nazionale delle Ricerche
DN	Digital Number
EO	Earth Observation
EODHaM	EO Data for Habitat Monitoring
EU	European Union
FP7	Seventh Framework Program
FTP	File Transfer Protocol
INSPIRE	Infrastructure for Spatial Information in Europe
ISO	International Organization for Standardization
JBoss	JavaBeans Open Source Software Application Server
JBPM	Jboss Business Process Management Suite
ODE	Orchestration Director Engine
PKH	Planetek Hellas
PKI	Planetek Italia
QA4EO	Quality Assurance Framework for Earth Observation
QAP	Quality Assurance Plan
QI	Quality Indicator
RDS	Relational Database Service
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SDD	Service Design Document

SLA	Service Level Agreement
UDDI	Universal Description Discovery and Integration
WP	Work Package
WS	Web Service
WSDL	Web Service Description Language
XML	eXtensible Markup Language