

**Nova Bonita**

**User Guide**

# Table of contents

LIST OF FIGURES .....	3
1. INTRODUCTION.....	4
2. REQUIREMENTS .....	5
2.1. HARDWARE REQUIREMENTS .....	5
2.2. SOFTWARE REQUIREMENTS.....	5
3. INSTALLING BONITA BPM TOOLS .....	6
3.1. DOWNLOADING THE PRODUCTS.....	6
3.2. INSTALLING AND LAUNCHING THE DESIGNER.....	6
3.3. INSTALLING AND LAUNCHING THE CONSOLE .....	7
4. CREATING A BPM ECLIPSE PROJECT .....	9
4.1. ADDING PARTICIPANTS.....	11
4.2. ACTIVITIES .....	12
4.3. ADDING TRANSITIONS.....	14
4.4. WORKING WITH VARIABLES .....	15
4.5. CONDITIONS IN TRANSITIONS.....	15
4.6. ITERATIONS/LOOPS/CYCLES .....	17
4.7. WORKING WITH CONNECTORS/HOOKS.....	18
5. RUNNING A PROCESS .....	21
5.1. DEPLOYING A PROCESS .....	21
5.2. CREATING AN INSTANCE .....	22
6. MORE EXAMPLES.....	23
6.1. BPMN MODEL FOR A POLICY ISSUANCE.....	23
6.1.1. <i>Process description</i> .....	23
6.1.2. <i>Class Diagram</i> .....	23
6.1.3. <i>Writing the Workflow Process in Bonita</i> .....	24

## List of Figures

Bonita BPM wizard.....	6
Login web 2.0 BPM console .....	7
Web 2.0 BPM environment .....	8
Creating a New BPM Project .....	9
Creating a New BPM Project .....	9
Editing Java settings .....	10
BPM project structure .....	10
New BPM process view .....	11
Participants definition in the designer .....	11
Activity Dialog .....	12
Manual activities .....	13
Hook definition dialog.....	13
Activities creation before arrangement .....	14
WebSale process activities and transitions.....	14
Variable definition dialog.....	15
Decision variable in a condition between two activities .....	16
ExpressDelivery variable in a condition between two activities .....	16
WebSale process view including conditions .....	17
Adding points to transitions.....	17
Designer view using points .....	18
Changing the default activity type .....	18
Hook creation Wizard.....	19
Hook/Connector name dialog .....	19
Developers Hook view .....	20
Web Sale process deployment .....	21
Users worklist application .....	22
Web forms creation in the console.....	22
Workflow emission process. ....	23
Class diagram of emission process .....	23
Activities creation before arrangement .....	25
Process activities and transitions.....	26
Local variable definition dialog.....	26
Condition between Welcome page and Acquista activities.....	27
Condition between Welcome page and Chiamata call center activities.....	28
Condition between Welcome page and Home page callcenter activities .....	28
Condition between Welcome page and Apertura pratica direzione activities .....	28
Condition between Welcome page and Compila preventivo Internet activities .....	29
Condition between Compila Preventivo Internet and Emissione Preventivo Internet activities .....	29
Condition between Compila Preventivo Internet and Chiama call center? activities .....	29
Condition between Acquista and Invio doc activities .....	30
Condition between Chiama call center? and End activities .....	30
Condition between Chiama call center? and Chiamata call center activities.....	30
Condition between Home Page Call Center and Apertura pratica direzione activities .....	31
Condition between Home Page Call Center and Emissione Preventivo Call Center activities .....	31
Condition between Esame Pratica and Ricontattare Cliente activities.....	31
Condition between Esame Pratica and Emissione Preventivo Call Center activities.....	32
Condition between Ricezione ricevuta pagamento and Esame pratica Finance activities .....	32
Condition between Ricezione ricevuta pagamento and Invio tagliando & Doc. activities .....	32

# 1. Introduction

This documentation provides some useful installation and configuration instructions of Nova Bonita (Nova Bonita is the name of Bonita v4 the new version of the **Bonita** Workflow/BPM open source) and will help you to get started right away.

Nova Bonita is composed by three different components that can be used separately or in common. Multiple configurations of those components are allowed: JSE (standard) vs JEE (enterprise). In this guide we will use them in a standard configuration (default):

**Nova Bonita runtime:** this is the Nova Bonita process engine. Processes can be deployed, executed and monitored through a rich API providing BPM services. This component is targeted to be used separately from the console one in distributed environments (client-server installation) or embedded in custom web applications. These use cases are out of the scope of this guide.

**Nova Bonita console:** this is the web 2.0 graphical interface fostering the user experience during BPM deployment, execution and monitoring phases. Nova Bonita console distribution embeds a runtime component. This BPM runtime is already configured to be deployed locally in the same server in which the web 2.0 console will be running.

In Nova Bonita console we have the next available features:

- Web 2.0 console supporting both desktop and traditional portal layout modes
- Monitoring vs Worklist applications (portlets)
- Internal user repository handling access rights to applications
- Automatic generation of forms vs customized forms
- Console customization capabilities: on the fly page creation, add/remove applications and widgets, look and feel...
- Applications (portlets) and widgets support

**Nova Bonita designer:** this is the BPM development environment allowing to graphically define your processes as well as BPM connectors to your information system.

In Nova Bonita designer (Process Editor) we have the next available features:

- Eclipse and Desktop versions
- "Easy BPM project" creation wizard available in Eclipse version
- Graphical support for advanced Nova Bonita entities: hooks, mappers, performers and instantiators
- Support for multiple variables types
- "Smart" conditions editor: graphical definition of complex conditions based on multiple operators and variables types
- Automatic generation of start and end BPM steps

These three components can be downloaded from the Nova Bonita download forge: bonitaruntime-4.1.1, bonita-console-4.1.1, bonita-designer-4.1.1 respectively

## 2. REQUIREMENTS

### 2.1. Hardware requirements

A 1GHz processor is recommended, with a minimum of 512 Mb of RAM. Windows users can avoid swap file adjustments and get improved performance by using 1Gb or more of RAM

### 2.2. Software requirements

Nova Bonita has been successfully tested in the following environments (should work in others but those ones are part of Nova Bonita continuous integration infrastructure):

Operating Systems:

- Solaris-10 (SunOS 5.10) x86
- GNU/Linux kernel 2.6.25-2 x86 Debian
- Windows XP

Java Virtual Machines (jdk 1.5 and 1.6):

- Sun-jdk1.5.0\_13
- Jrockit-R27.1.0-jdk1.5.0\_08
- Ibm-java2-i386-50
- Jrockit-R27.2.0-jdk1.6.0
- Sun-jdk1.6.0\_06

Relational Databases:

- Mysql-server 5.0.51a-6
- Postgresql 8.3.3-1
- Oracle 11.1.0
- H2 1.0.76
- HSQL 1.8.0.7

Application Servers:

- Tomcat 5.5.26
- OnAS 5.10.3
- Jboss-4.2.2.GA
- JOnAS 5.1.0 RC2
- Jboss-5.0.1.GA
- Easybeans 1.0.1

Nova Bonita requires Apache Ant 1.7 or higher. Apache ant will allow users to deal with configuration and administration operations. It can be downloaded from [apache site](#)

This quick guide is based on:

- Eclipse 3.4.2
- Java Development Kit (JDK) 1.6
- Ant 1.7
- Windows XP e Ubuntu 8.10

### 3. Installing Bonita BPM tools

#### 3.1. Downloading the products

In order to install Bonita BPM tools you need to download and extract to your system the next products:

- [Eclipse Release 3.4 SDK](#)
- [bonita-eclipseDesigner-4.1.1.zip](#)
- [bonita-console-4.1.1](#)
- [apache Ant 1.7](#)

#### 3.2. Installing and launching the designer

Unzip bonita-eclipseDesigner-4.1.1.zip file into your favorite Eclipse environment (up to 3.4 version). Previous file already contains a “plugins” directory so please unzip this file into your root Eclipse installation directory.

Once unzipped, you'll need to launch eclipse with the -clean flag to allow the environment to detect the new plugins. Subsequent launching of eclipse won't require the -clean flag.

Windows users  
eclipse -clean

Linux users  
./eclipse -clean

After restart your Eclipse go to “File” menu and then check that the “Bonita BPM” wizard is available in the list:

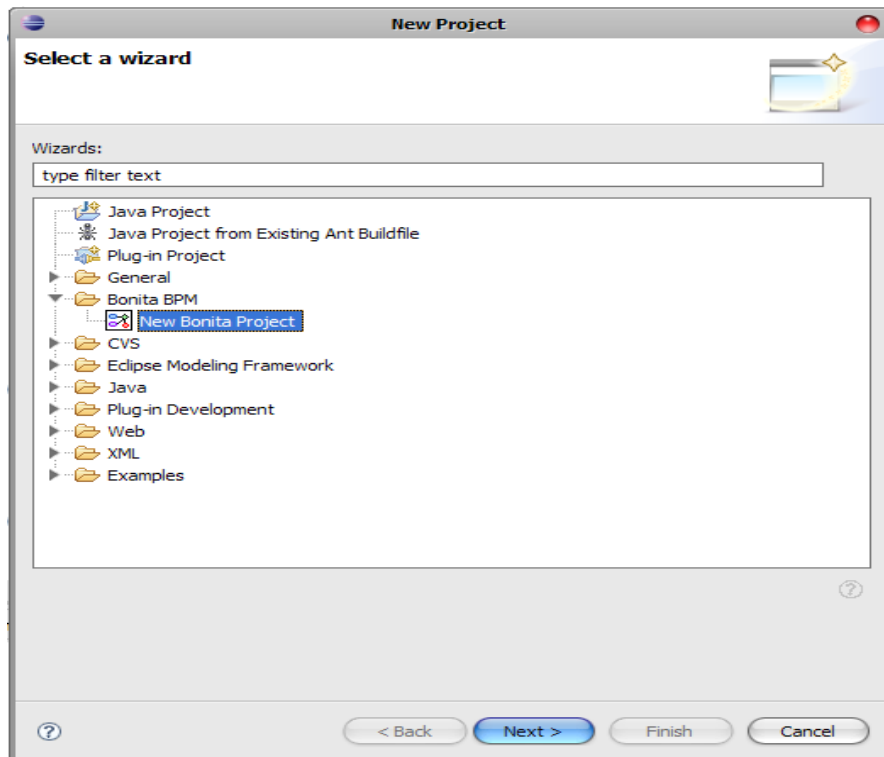


Figure 1 Bonita BPM wizard

### 3.3. Installing and launching the console

In this guide we are going to leverage the integrated version of the runtime and the console components. This component/distribution is in fact the bonita-console-4.1.1.zip file.

Set JAVA\_HOME or JRE\_HOME to point to an appropriate java version recommended in the software prerequisites

Unzip bonita-console-4.1.1.zip file into your file system and a new directory called bonita-console-4.1.1 will be created.

To start the Nova Bonita Console:

#### Windows

Open a command window (cmd.exe), go to the “bin” directory and launch “bpm run” command. This will automatically start the runtime and will deploy the web console application in a custom version of a Tomcat server.

```
bpm run
```

#### Linux

Access the Linux terminal, go to the “bin” directory and launch “bpm run” command. This will automatically start the runtime and will deploy the web console application in a custom version of a Tomcat server.

```
./bpm.sh run
```

Let's now check that Nova Bonita was successfully installed. For that just open your favorite web browser (does not work in google Chrome) and connect to the following URL: <http://localhost:8080/portal/>



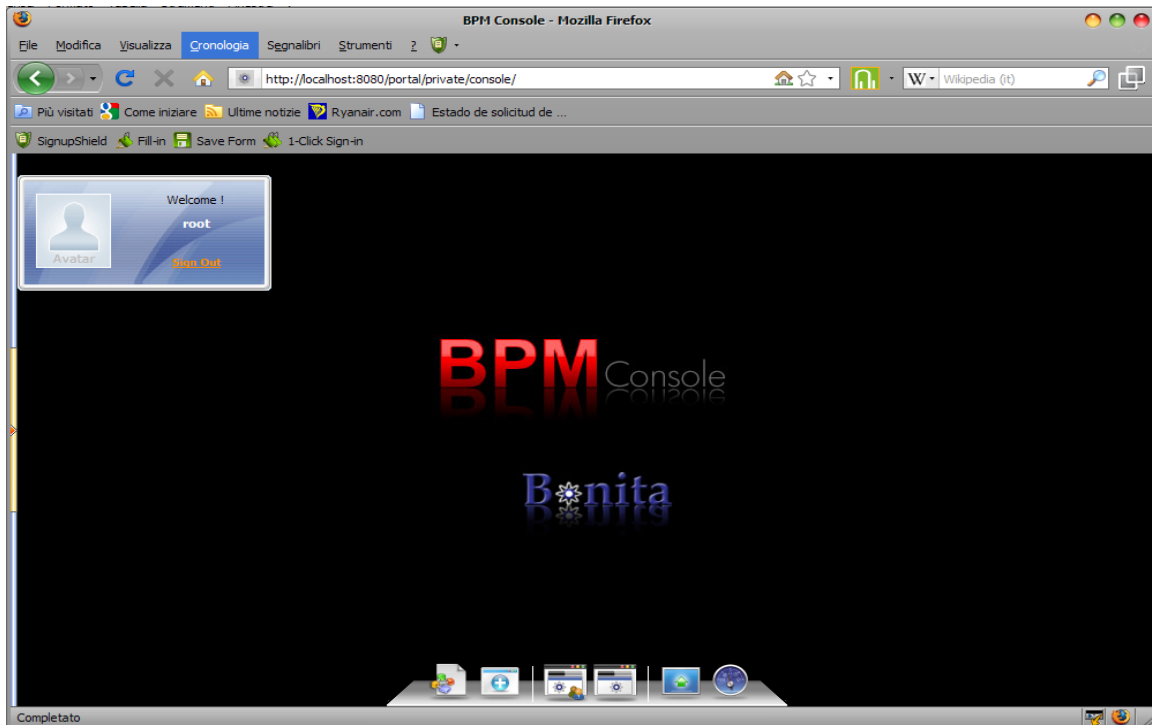
Figure 2 Login web 2.0 BPM console

Login into the web 2.0 BPM console as default admin user. Connect with:

User name: root

Password: bpm

After logged as admin user the following screen will appear:



**Figure 3 Web 2.0 BPM environment**

Nova Bonita console distribution embeds a Tomcat server which is properly configured to deploy Nova Bonita runtime and web console components.

## 4. Creating a BPM Eclipse project

To create a new BPM process with ProEd just go to Eclipse File menu and select “New → Project” feature. If ProEd plugin was successfully installed in your Eclipse environment, the following dialog should appear:

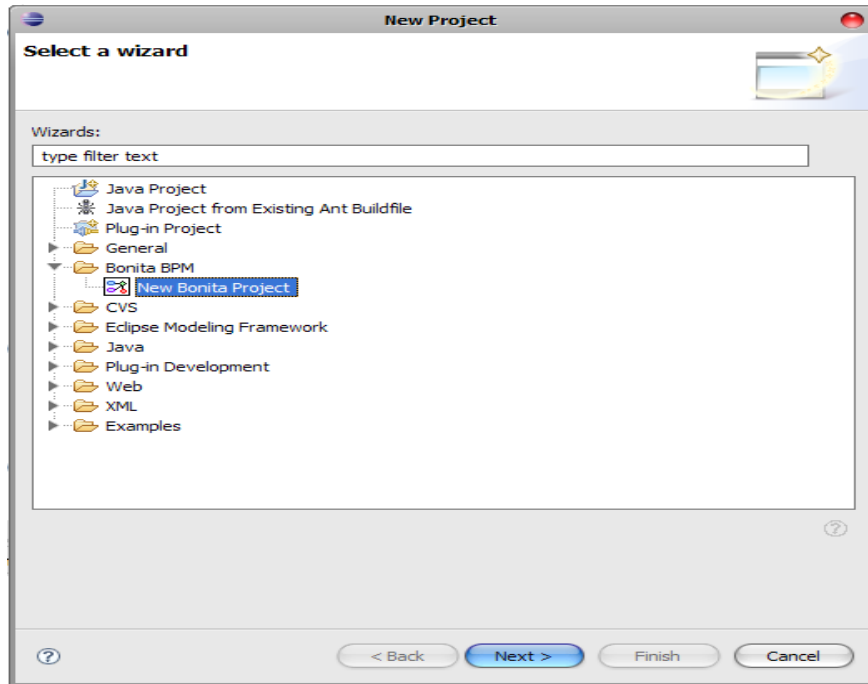


Figure 4 Creating a New BPM Project

Then select the “New Bonita Project” feature and click “Next”. In the next dialog you are allowed to introduce the name of your BPM project as well as the directory name in which the .bar file will be generated.

Just after that, enter the name of your process (i.e SimpleWebSale) as well as a description (if required):

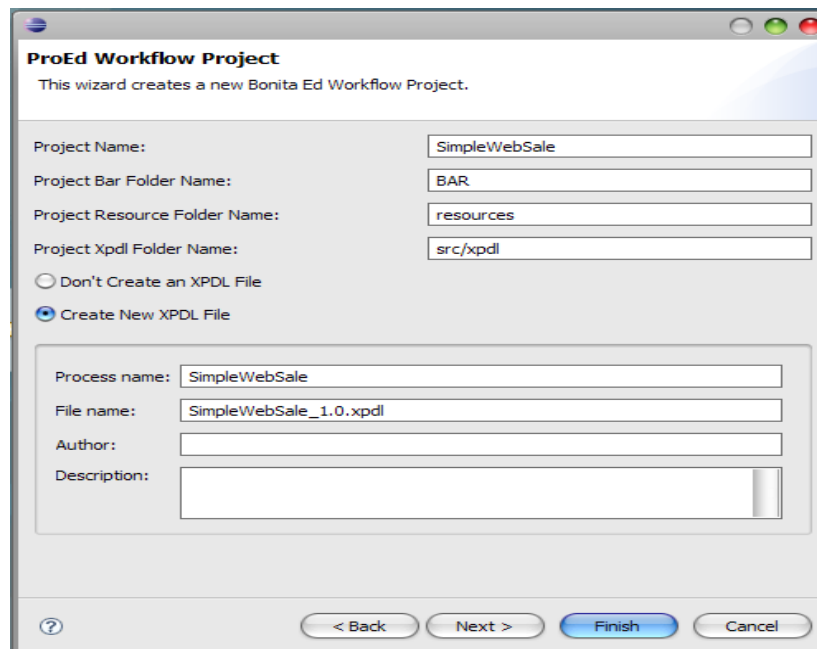


Figure 5 Creating a New BPM Project

Once finished click on “Next” and you will be redirected to the Java Settings dialog. In this dialog, the wizard will suggest you a default directory structure for your BPM project composed by:

- forms directory: default directory in which forms associated to manual activities will be stored
- java directory: default directory in which you could add java related BPM entities such hooks, mappers, performers assignments and Multi-instantiators.
- xpdL directory: default directory to store xpdL files

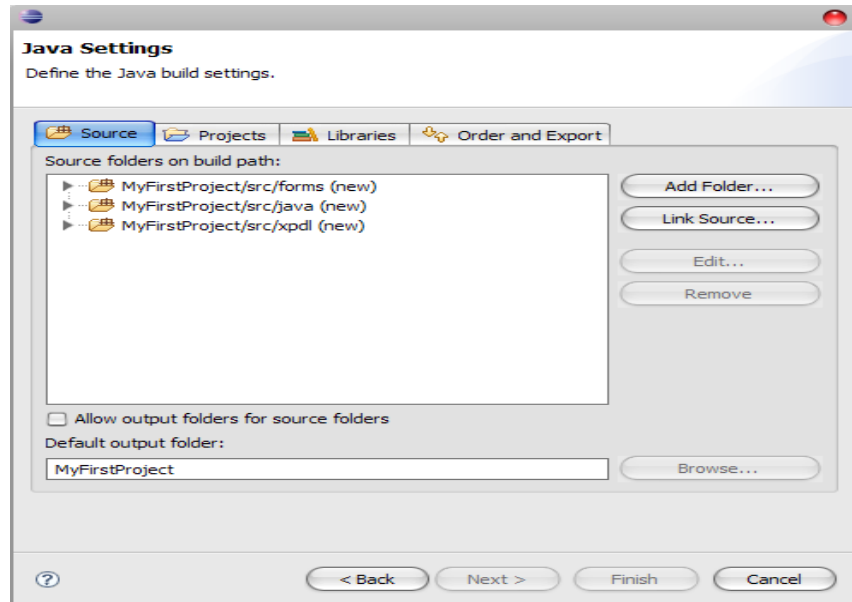


Figure 6 Editing Java settings

Click on the “Finish” button when you are done and this wizard will create a new project in Eclipse with the following structure:

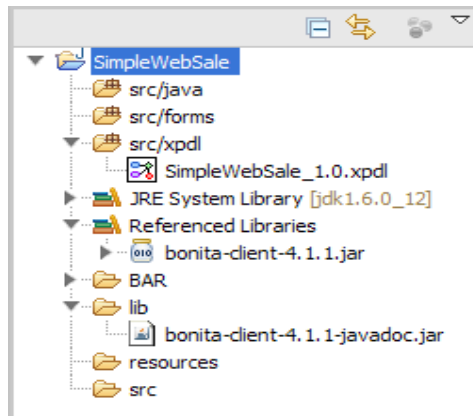
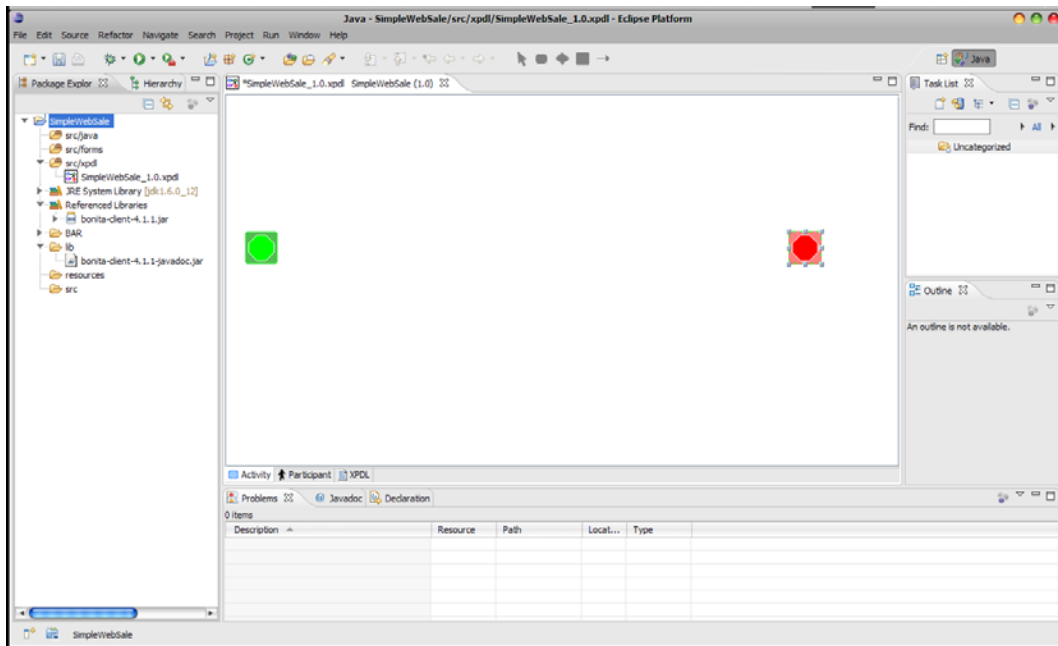


Figure 7 BPM project structure

By default, a process with “start” and “end” activities is created. Time now to add activities in-between corresponding to the business process.



**Figure 8 New BPM process view**

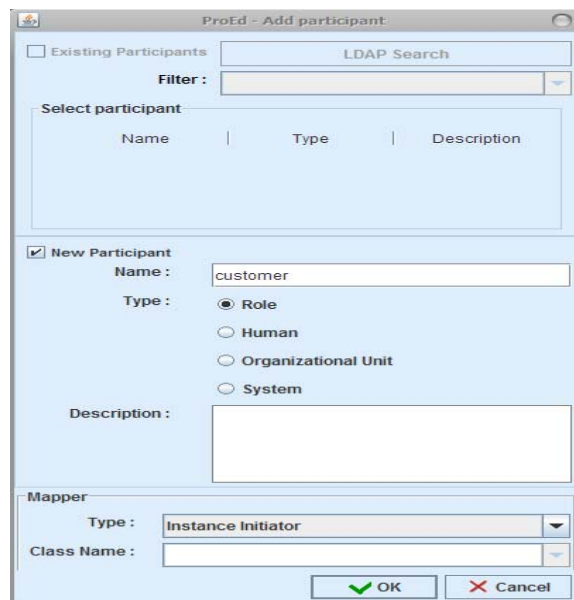
Each “Save” operation in the designer will automatically generate a “.bar” file (SimpleWebSale.bar in the example). This .bar file is the one that you will deploy into the runtime when the definition phase is completed.

By default; the .bar file will be located under the BAR directory of your project.

You are now ready to graphically define a BPM process.

#### 4.1. Adding participants

We will start by creating the actors/participants of this process. Those are a “customer” and an “employee”. Click right over the designer and select “properties” view. Then create the previous participants.




**Figure 9 Participants definition in the designer**

Both participants will be added with type “Role” (meaning potentially more than one actor could be part of them) and using the default mapper operation: Instance Initiator.

## 4.2. Activities

Time now to create the main activities of the WebSale BPM process. The following activities will be created:

- “Request”, “Pay” and “MoreInfo” activities will be created as manual activities (aka user task) and so assigned to the “customer” role previously created.
- “SalesReview” activity is also a user activity that will be assigned to the “employee” role.
- “Reject”, “ExpressDelivery” and “Archive” activities are automatic activities that will interact with the information system (i.e sending a notification, updating a row in the database...)

In order to create those activities we will leverage an activity type called “basic activity”. To add one basic activity just click over the  “Add Activity” icon available in the toolbar:

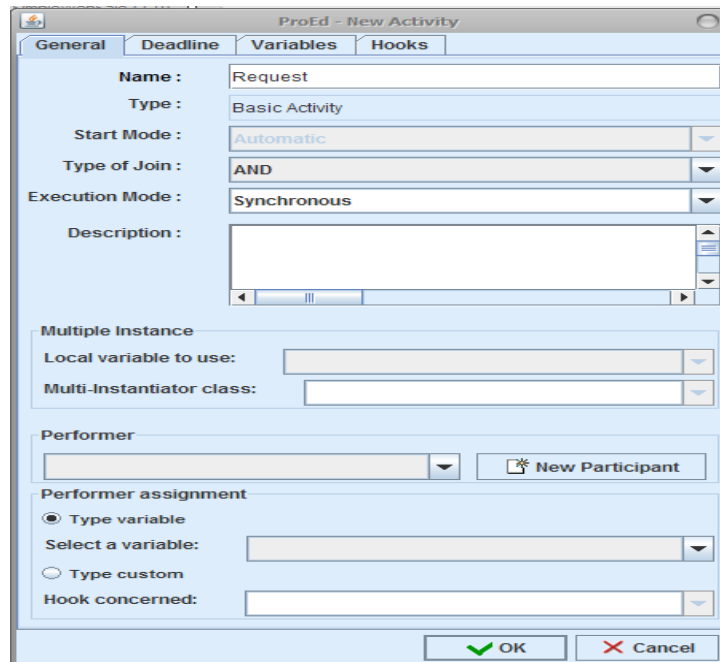


Figure 10 Activity Dialog

By default a basic activity is consider to be automatic. To change it to manual just select one of the existing participants/actors available. Those are called activity performers

After the creation of previous four manual activities your designer should looks like as follows:

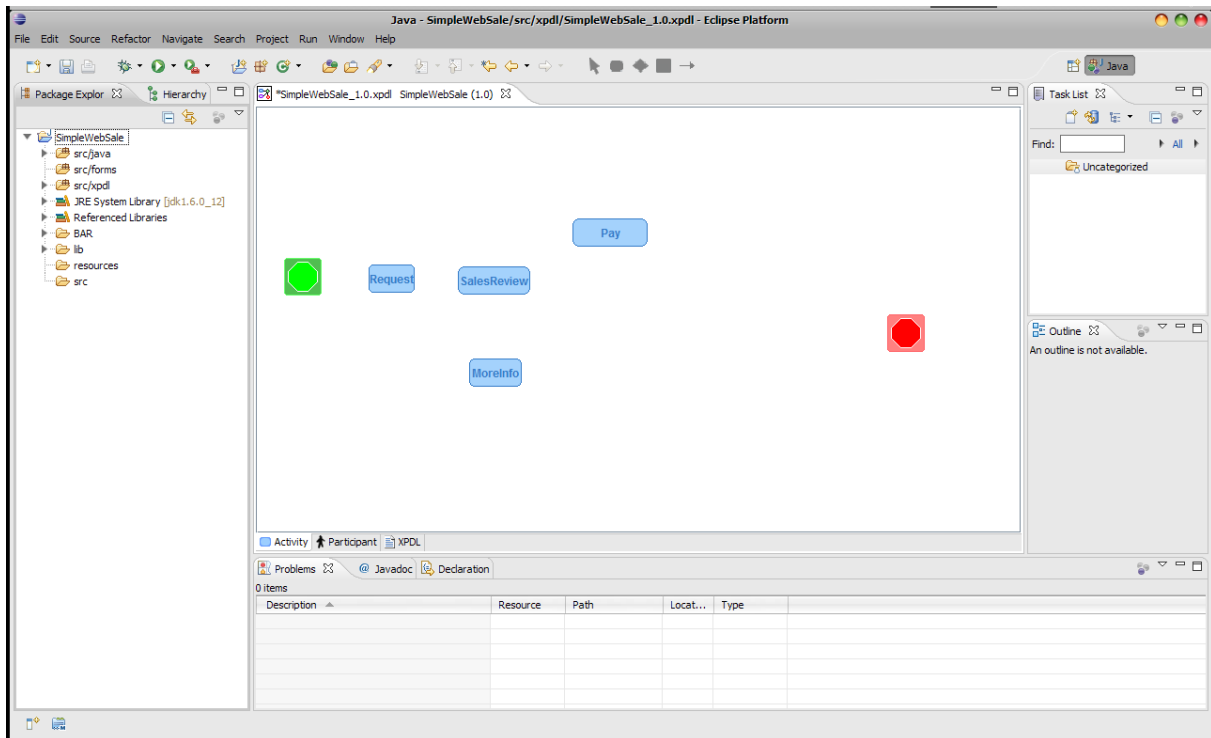


Figure 11 Manual activities

Add now three new “basic activities” to the WebSale process (all three automatic activities) and assign a hook called “Message” to each one of them:

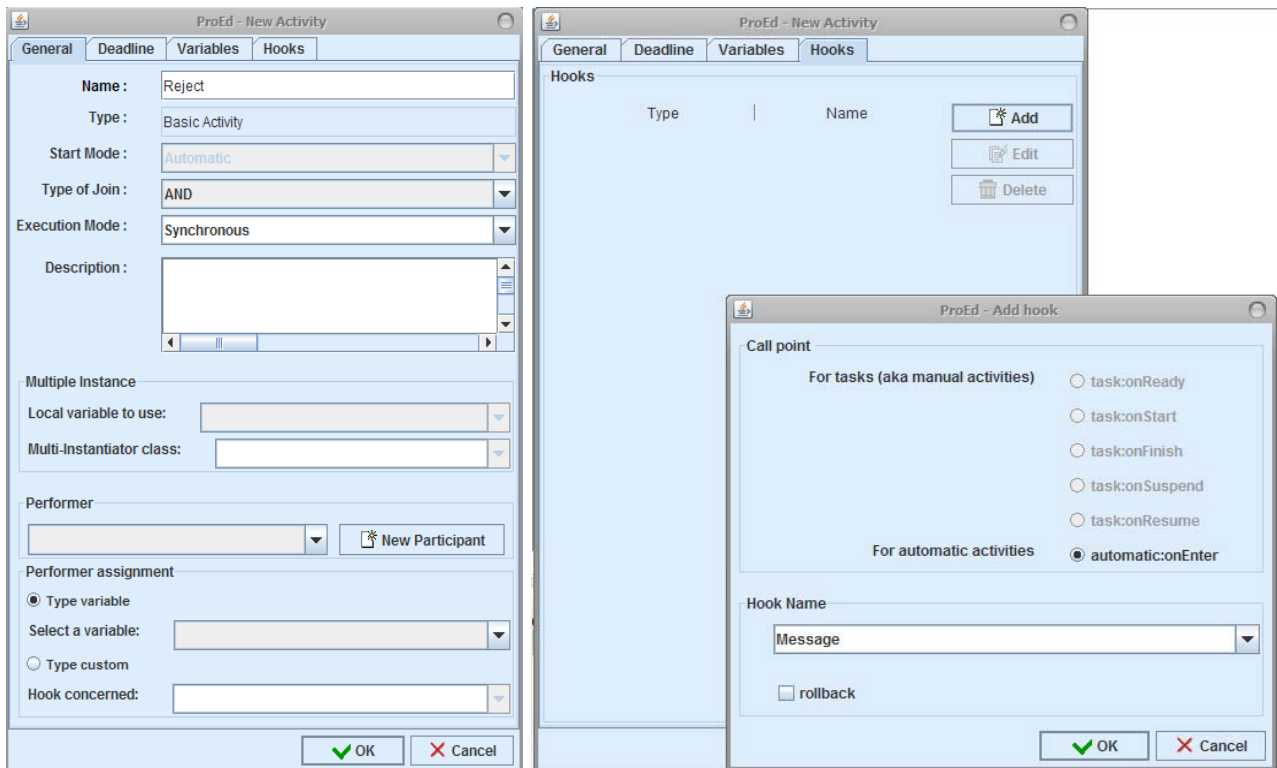


Figure 12 Hook definition dialog

Once those three activities are created your BPM process appearance should be:

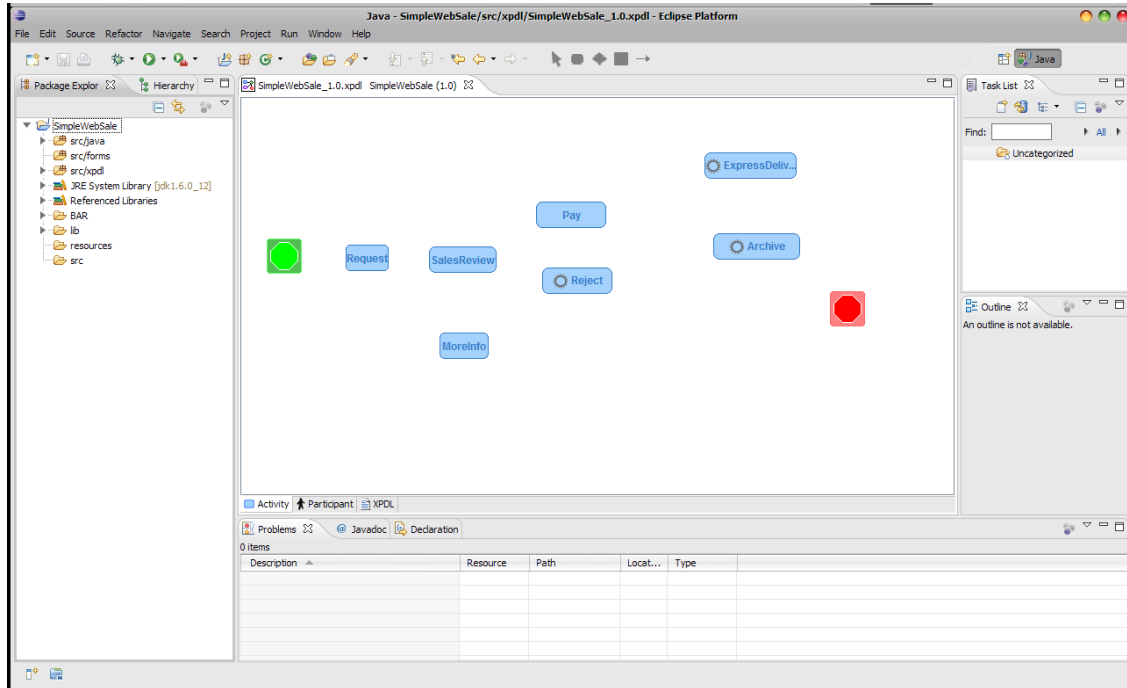



Figure 13 Activities creation before arrangement

### 4.3. Adding transitions

Resize now those activities and add transitions in-between. For that, please use the “transition” icon  and follow the activity sequence illustrated in the diagram below:

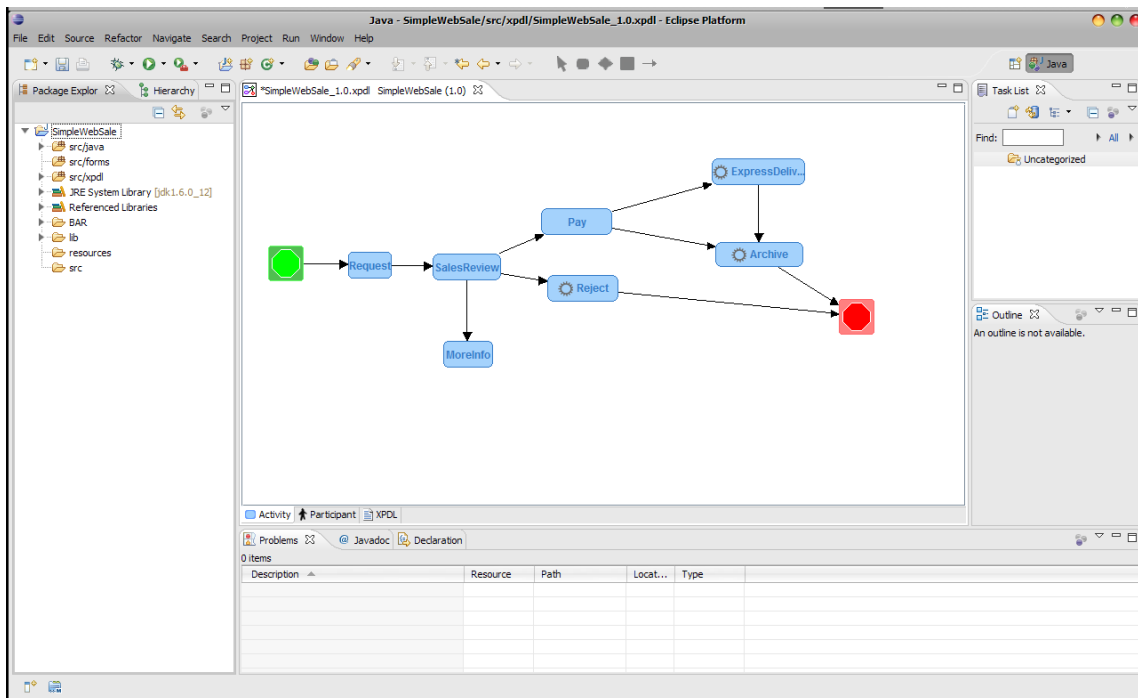


Figure 14 WebSale process activities and transitions

## 4.4. Working with variables

Lets' now focus on the variables definition for the Web Sale process. In this example we will add both global and local variables. Global variables can be added by clicking right on the designer and going over the variables tab. Local variables (aka activity variables) are accessible by either double clicking on a particular activity or clicking right mouse button over it.

In both cases, the following dialog will allow you to add new variables:

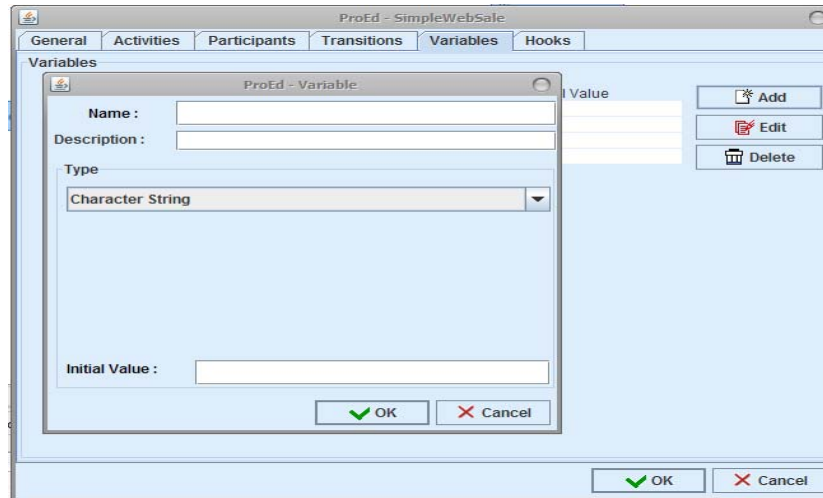


Figure 15 Variable definition dialog

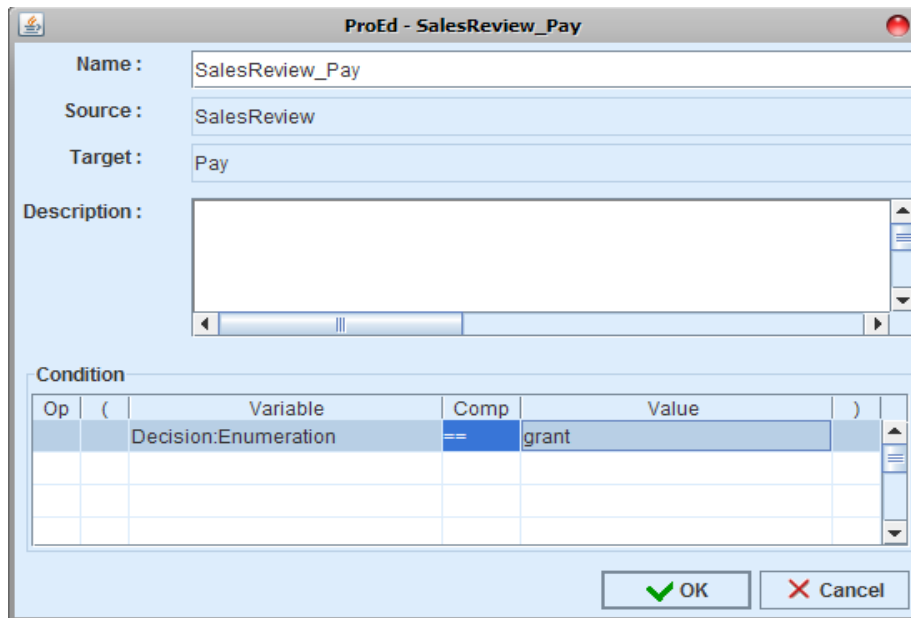
The following variables are required for the Web Sales sample:

- Global variables: "UserName" type String, "UserPhone" type String, "UserEmail" type String and "Products" type Enumeration with the following possible values: TV, Cellphone and Laptop
- Local variables:
  - Activity "SalesReview": variable "Decision" type Enumeration with possible values grant, reject and moreinfo.
  - Activity "Pay": variable "CreditCardNumber" type Integer, "CreditCardExpirationDate" type Date and "ExpressDelivery" type Boolean.
  - Activity "MoreInfo": variable "Comment" type String.

## 4.5. Conditions in transitions

We are about to finish the Web Sale design. Before that we will leverage some of the variables we have just created in transitions:

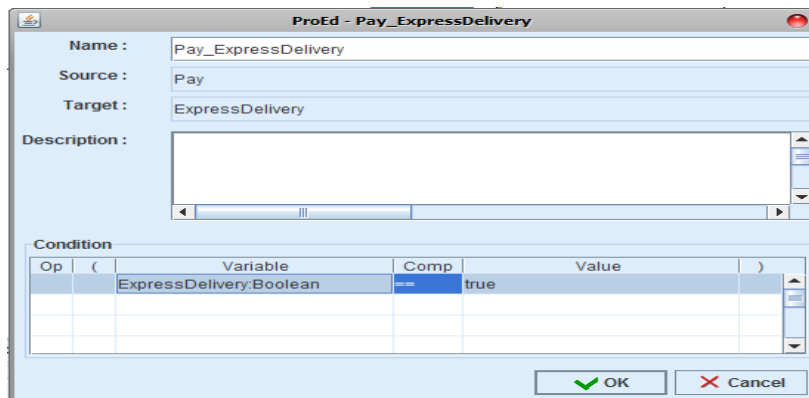
- "SalesReview" activity: click on each one of the outgoing transitions starting from "SalesReview" and set a condition by using the "Decision" local variable with values grant, reject and moreinfo respectively:



**Figure 16 Decision variable in a condition between two activities**

In the dialog above, you can easily use the smart conditions editor to set up conditions in transitions. The editor provides the list of variables accessible from a particular activity as well as corresponding operators.

“Pay” activity: set a condition in both outgoing transitions using the “ExpressDelivery” variable in conditions, use true and false values respectively:



**Figure 17 ExpressDelivery variable in a condition between two activities**

Edit now the “Archive” activity and change the default join mode from AND to XOR, otherwise the process will not be finished as Bonita will wait for both incoming transitions.

The designer will change the look and feel of transitions for which conditions were defined:

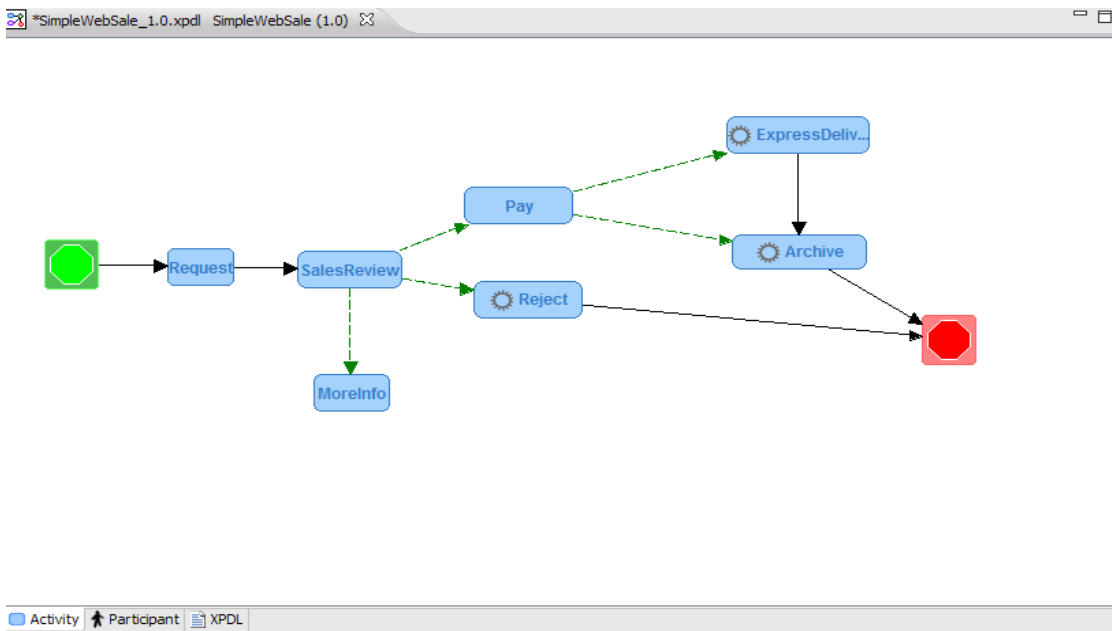


Figure 18 WebSale process view including conditions

#### 4.6. Iterations/loops/cycles

As you can see in the current process definition, there is a missing return path between “MoreInfo” and “ReviewSales” activities. This connection will allow to retry the sales review after some additional information was provided by the customer.

Let’s so add the missing transition between those activities. This transition will create an iteration or a loop between previous activities:

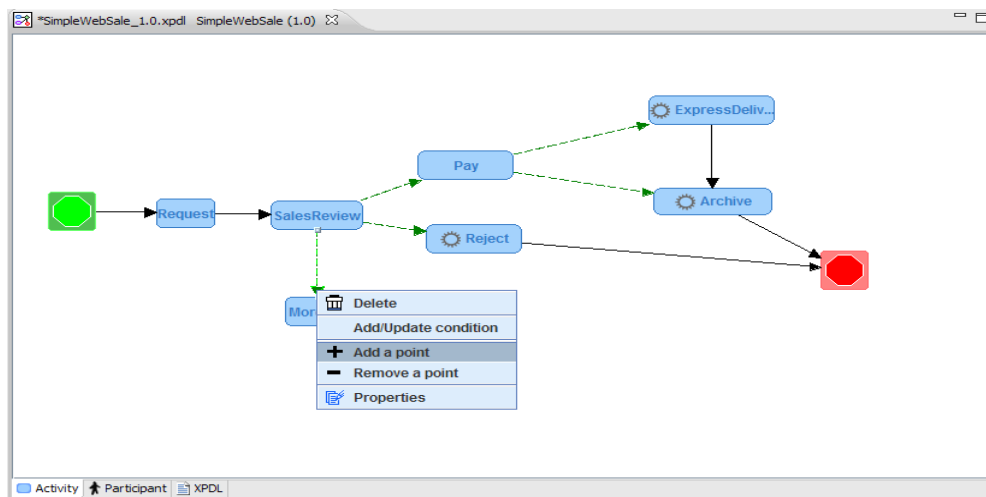


Figure 19 Adding points to transitions

In those use cases we suggest you to “add points” (as shown in the picture above) on one of the transitions of an iteration between activities. Points will allow to properly adjusting the view:

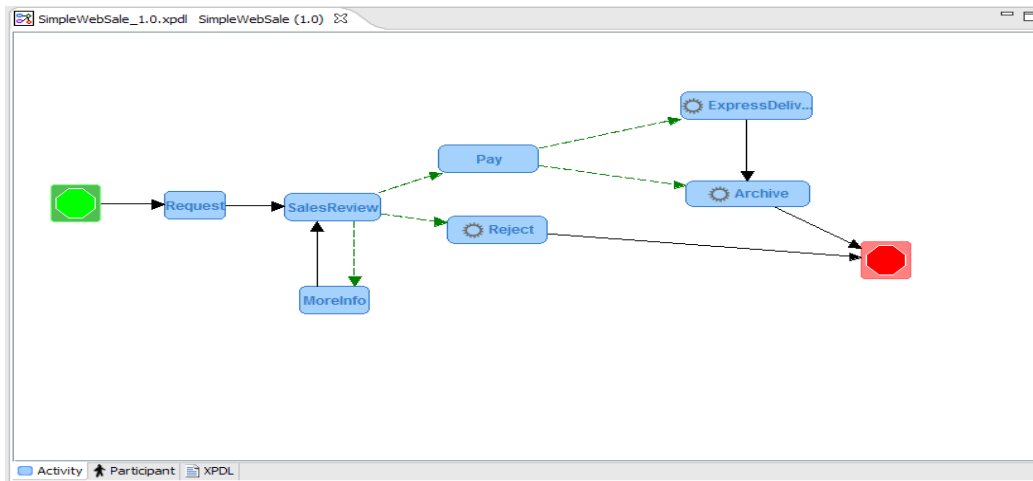


Figure 20 Designer view using points

Edit also the “SalesReview” activity to allow it to handle this iteration. To do that just double clicks on this activity and set the activity type to XOR rather than AND by default.

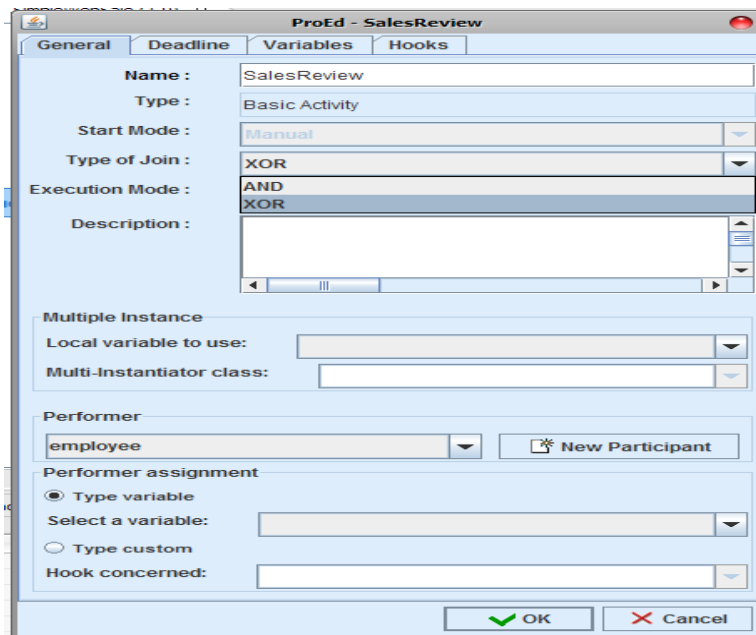
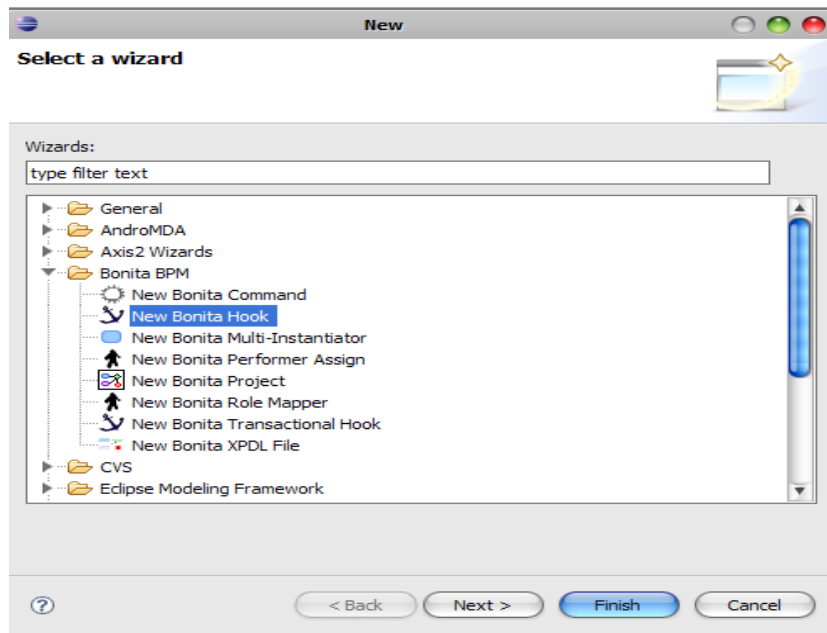


Figure 21 Changing the default activity type

#### 4.7. Working with connectors/hooks

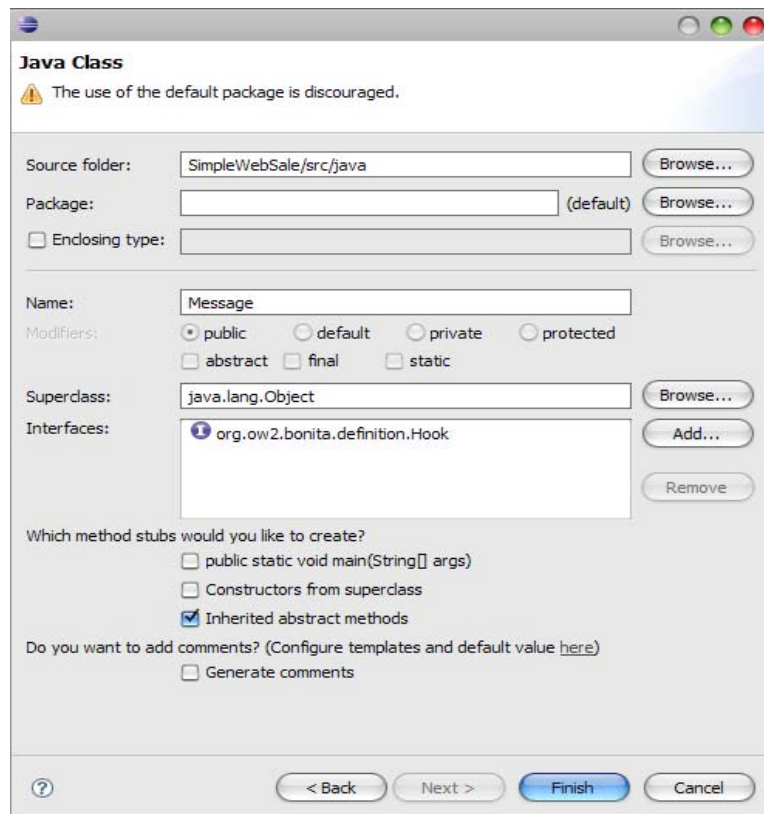
Let's finish the WebSale process definition by adding a new connector to the Web Sale Eclipse project.

Remember that you referenced one connector called “Message” during the automatic activities creation. Let's so create a new connector/hook using the Bonita designer wizard. Select your directory called “java” in your Web Sale Eclipse project and click right button -> “New” -> “Other” -> “Bonita BPM” -> “New Bonita Hook”:



**Figure 22 Hook creation Wizard**

Insert "Message" as Hook name in the following dialog:



**Figure 23 Hook/Connector name dialog**

The Bonita designer wizard will automatically create a Java class with the default structure of a Hook. The wizard will also compile this class automatically each time it's saved. Lets' now add our custom implementation of this connector:

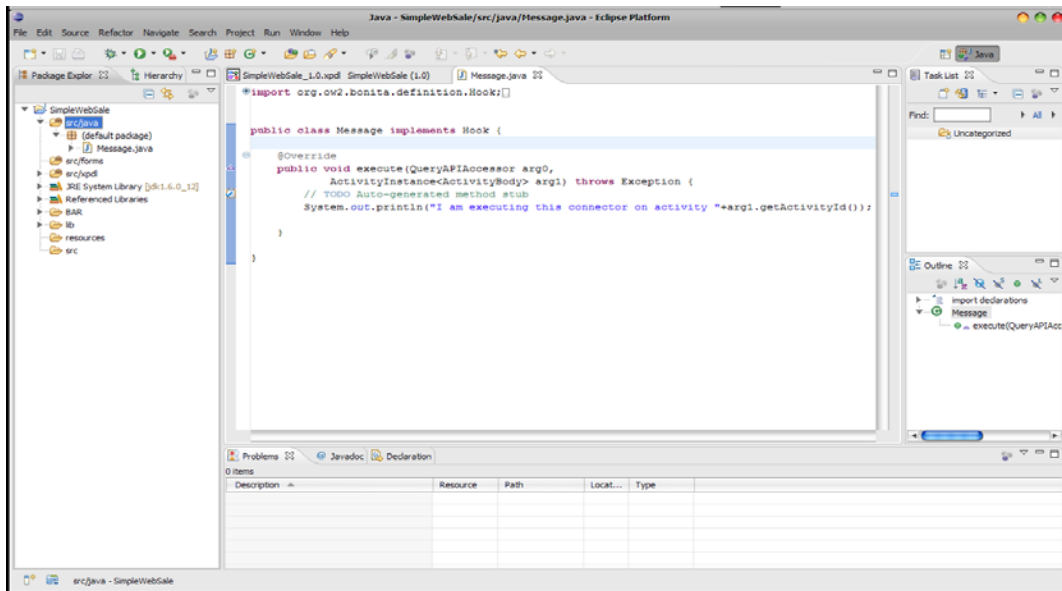


Figure 24 Developers Hook view

This connector will be executed at runtime by each automatic activity in which you referenced it.

You have just finished the definition of your first BPM process with Nova Bonita. You are now ready to deploy it into the runtime

## 5. Running a process


Time now to deploy and run the Web Sale process you have just designed and to start playing with it. Open your favorite web browser and connect to <http://localhost:8080/portal> URL (remember first to start tomcat with bpm run command) to start using the Nova Bonita Web 2.0 console.

Login into the console using root/bpm as user/password. This user will give you access to whole list of features available in the console

Once done, please go to the dynamic toolbar and open the BPM Management application in your default BPM desktop.

This application is in fact the key entry point for processes deployment and monitoring operations.

### 5.1. Deploying a process

Nothing easier than just browse the SimpleWebSale.bar file that was automatically generated by the designer and clicks on the upload icon  to deploy this process and related connectors into the Nova Bonita runtime.

If no definition errors were detected, the Web Sale application will be deployed and so available into the runtime

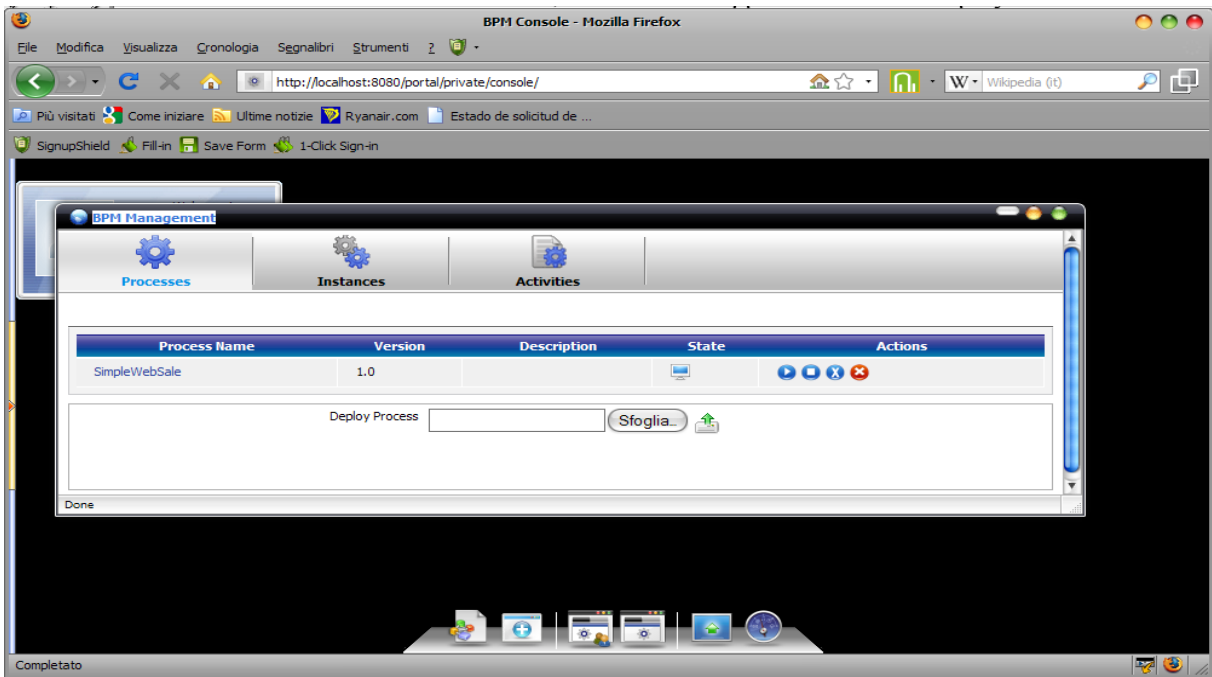


Figure 25 Web Sale process deployment

## 5.2. Creating an instance

Let's now create an instance of the Web Sale process. For that, just open the BPM Users application (also available in the dynamic bar):

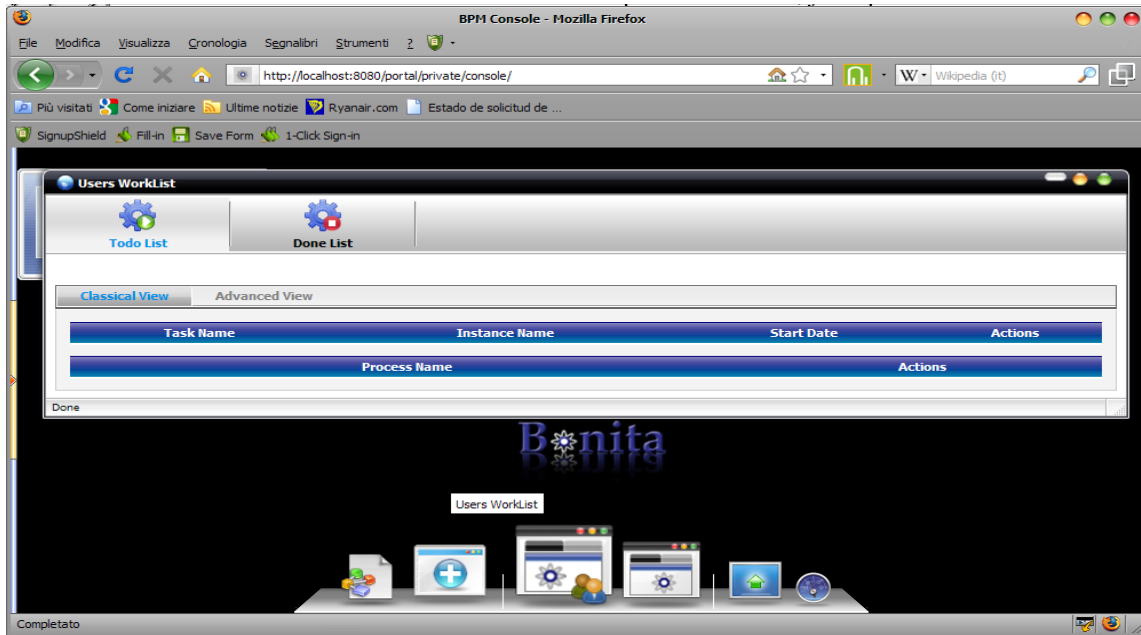


Figure 26 Users worklist application

This application allows users to create new instances of existing processes and take over activities assigned to them. The BPM Users application is the users "worklist". By default, any process deployed into the runtime is available in users worklist.

Let's create a new instance of the Web Sale process. By just clicking on the creation icon :

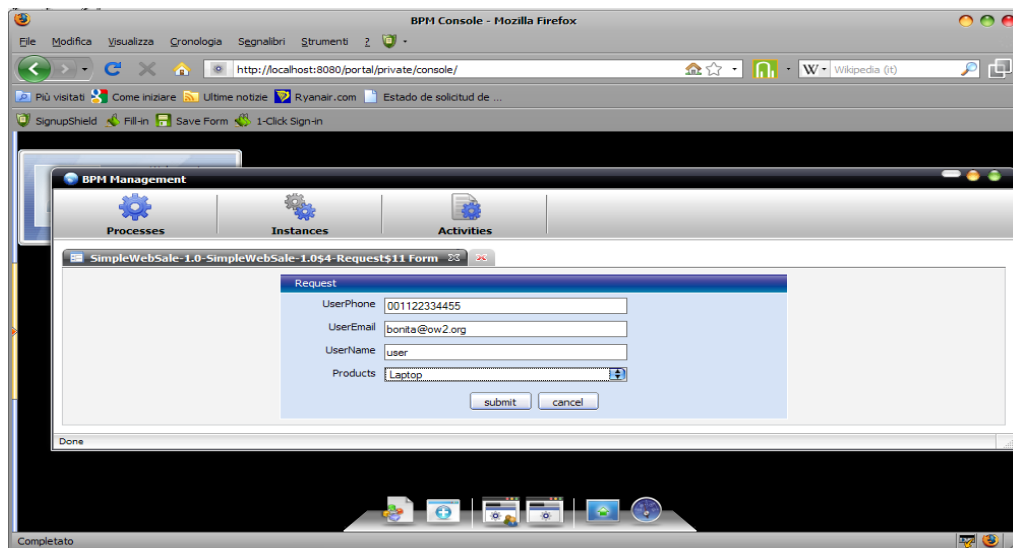


Figure 27 Web forms creation in the console

The console will automatically create a web form composed by the list of variables defined in the Web Sale process. Submitting this form will create a new process instance and will execute the first activity called "Request".

The user worklist contains now the next activity corresponding to the Web Sale instance we have just created.

## 6. More examples

### 6.1. BPMN model for a policy issuance

#### 6.1.1. Process description

This paragraph is intended to illustrate the flow of the policy emission process. It is useful to identify the users involved in the process. They are:

- Cliente: Someone who contacts (via Call Center or by connecting to an Internet portal) the company to stipulate a policy.
- Users of CallCenter: Employees of the company that answer the call of the customer who wants to issue a policy without using the Internet portal or blocked customers by anomalies in the portal.
- CustomerCare: Employees of the company in charge of customer services (i.e. receipt and dispatch of documents and vouchers of insurance)
- Finance: Employees of the company in charge of accounts management
- Direzione: Administrative Employees of the company

#### Describing the workflow scenario

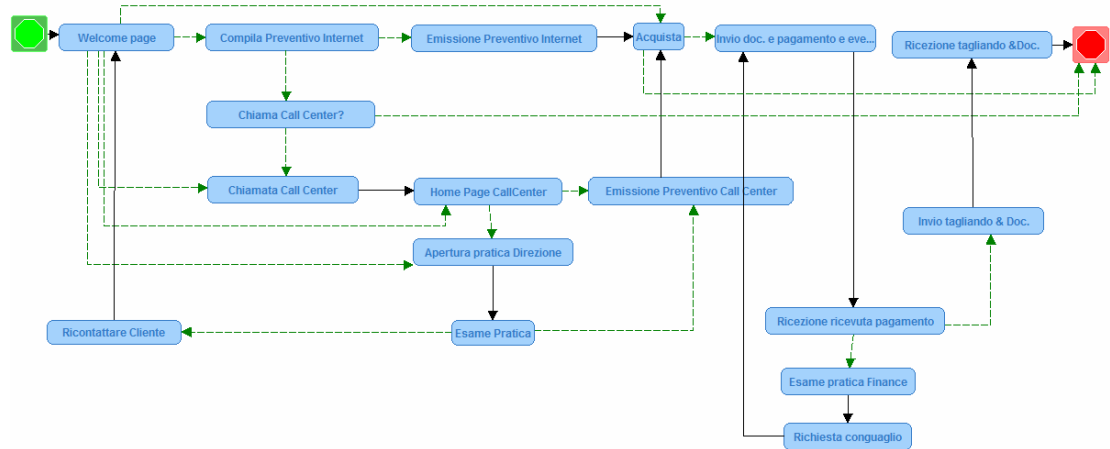


Figure 28 Workflow emission process.

#### 6.1.2. Class Diagram

The class diagram summarizes the structure of the overall operation of the company.

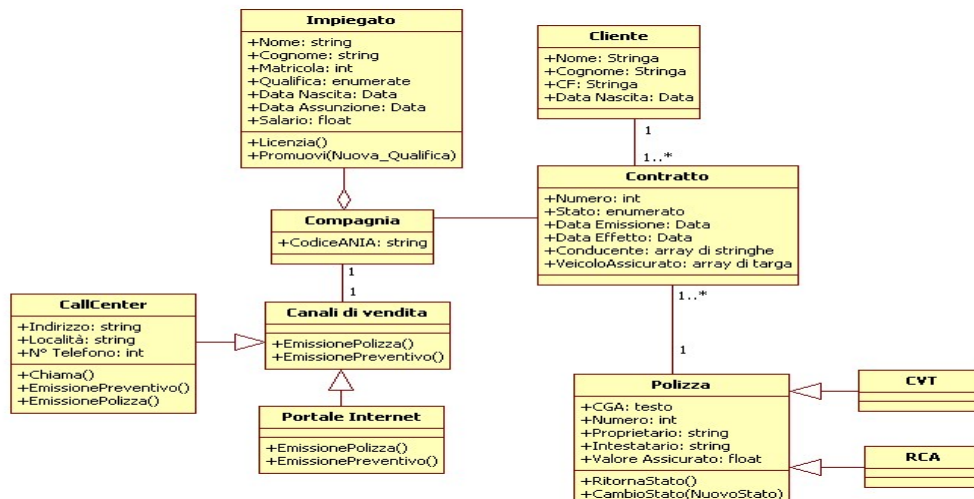


Figure 29 Class diagram of emission process

**Cliente:** contains biographical data and general contractor of a Policy (polizza)

**Contratto:** it is a legal contract between the customer and the company. A customer may sign n contracts

**Polizza:** represents a single insurance policy. Generally a policy is tied to a single contract, but in special cases a customer may conclude m policies with only one contract (collective policy).

**Compagnia:** the company.

**Impiegato:** represents the generic employee of the company.

**Canali di Vendita:** represents the generic sales channel. There is Internet and call center.

Now we can create the Bonita project

### 6.1.3. Writing the Workflow Process in Bonita

To create a new BPM process with ProEd just go to Eclipse File menu and select “New → Project” feature.

Then select the “New Bonita Project” feature and click “Next”. In the next dialog you are allowed to introduce the name of your BPM project as well as the directory name in which the .bar file will be generated.

Just after that, enter the name of your process, in this case *Emissione portale*

Once finished click on “Next” and you will be redirected to the Java Settings dialog. In this dialog, the wizard will suggest you a default directory structure for your BPM project composed by forms directory, java directory and xpdL directory. Just click on the button “finish”

We will start by creating the actors/participants of this process. Click right over the designer and select “properties” view. Then create the participants:

Name	Type
Cliente	Human
Operatore Call Center	Human
Customer Care	Human
Finance	Human
Direzione	Human

Time now to create the workflow activities. The following activities will be created.

Name	Type of Join	Exec Mode	Performer
Welcome page	XOR	Synchronous	Cliente
Compila Preventivo Internet	XOR	Synchronous	Cliente
Emissione preventivo Internet	XOR	Synchronous	Cliente
Acquista	XOR	Synchronous	Cliente
Invio doc. e pagamento e eventuale conguaglio	XOR	Synchronous	Cliente
Ricezione del tagliando e Doc.	XOR	Synchronous	Cliente
Chiama CallCenter?	XOR	Synchronous	Cliente
Chiamata CallCenter	XOR	Synchronous	Cliente
Home page CallCenter	XOR	Synchronous	Operatore Call Center
Emissione Preventivo CallCenter	XOR	Synchronous	Operatore Call Center
Ricontattare Cliente	XOR	Synchronous	Operatore Call Center
Apertura Pratica Direzione	XOR	Synchronous	Direzione
Esame Pratica (Direzione)	XOR	Synchronous	Direzione
Esame pratica finance	XOR	Synchronous	Finance
Richiesta conguaglio	XOR	Synchronous	Finance
Ricezione ricevuta pagamento	XOR	Synchronous	Customer Care
Invio Tagliando e Doc	XOR	Synchronous	Customer Care

Then assign a hook needed:

Activity	Call point	Hook name	Roolback
Compila Preventivo Internet	task:onFinish	Numero_preventivo	X
Acquista	task:onFinish	Numero_polizza	

After the creation of previous manual activities your designer should looks like as follows:

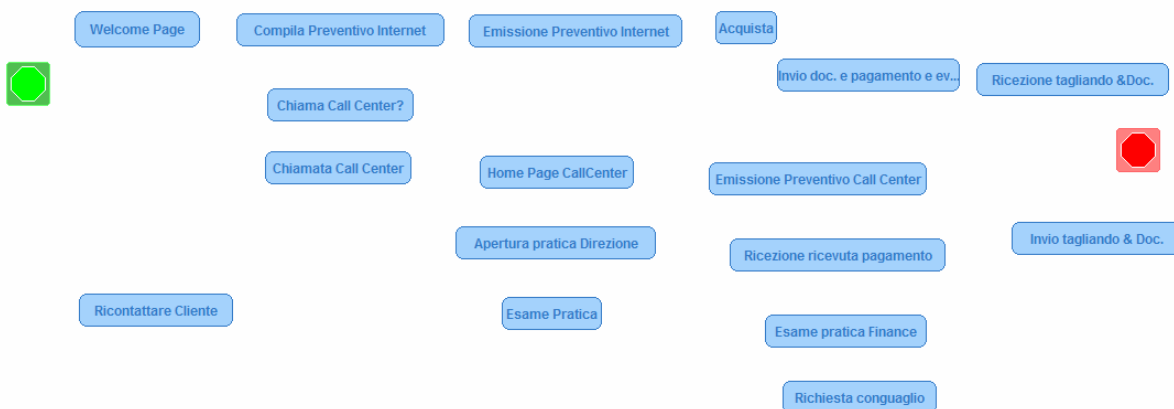


Figure 30 Activities creation before arrangement

Now add transitions in-between the activities. For that, please use the “transition” icon and follow the activity sequence illustrated in the diagram below

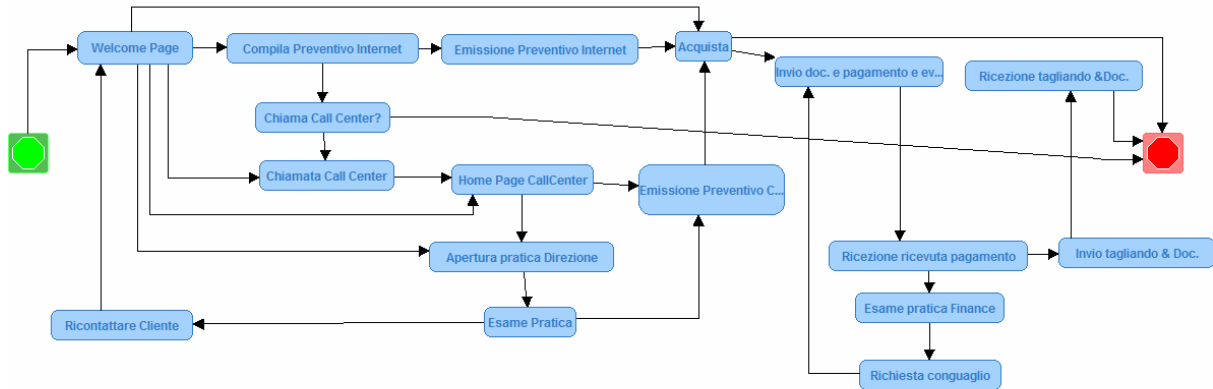


Figure 31 Process activities and transitions

Lets' now focus on the variables definition for the Emission proposal process. In this example we will add local variables. Local variables are accessible by either double clicking on a particular activity or clicking right mouse button over it.

The following dialog will allow you to add new variables

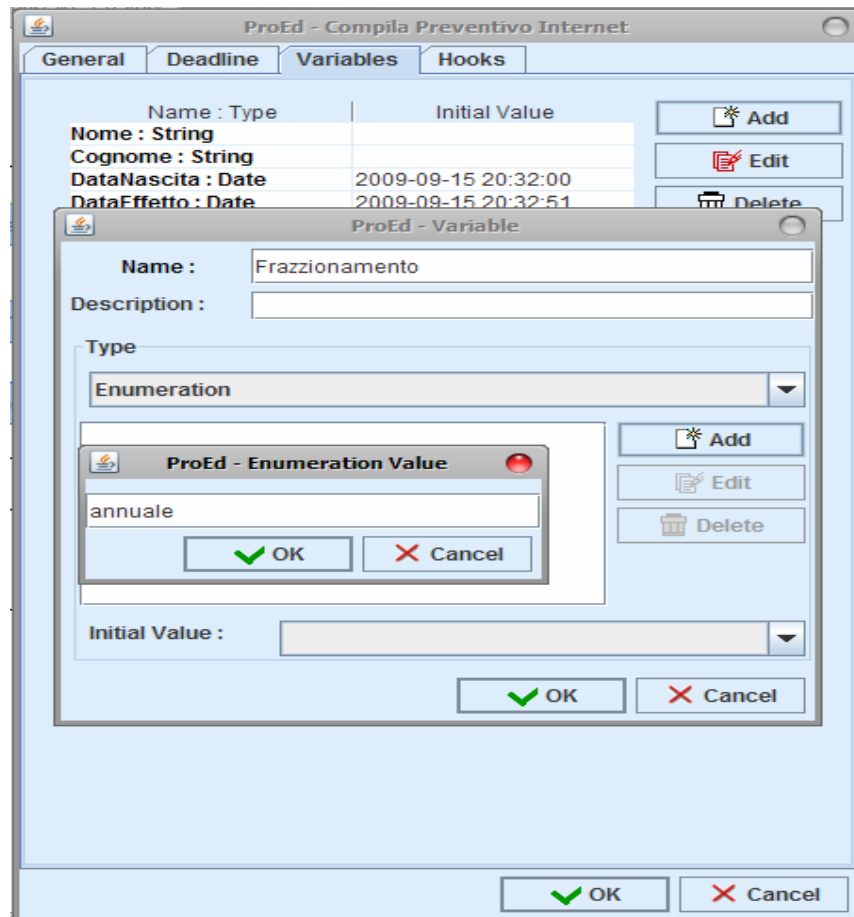


Figure 32 Local variable definition dialog

Activity	Name:type	Initial value	Values
<b>Welcome Page</b>	User:String Password:String PreventivoEsistente:String TipoEmissione:Enumeration	Internet	Internet, CallCenter
<b>Compila Preventivo internet</b>	Nome:String Cognome:String DataNascita:Date DataEffetto:Date Frazzionameto:Enumeration  TipoVeicolo:Enumeration  ValoreDichiarato	Current date Current date Annuale  Auto  Float	Annuale, semestrale, quadrimestrale, trimestrale, temporaneo Auto, moto, autocarro, altro
<b>Acquista</b>	Acquista:Boolean	true	
<b>Chiama Call Center?</b>	Chiamo:Boolean	true	
<b>Home page Callcenter</b>	LivelloDeroga:Enumeration TipoOperazione:Enumeration	sufficiente Nuovo_preventivo	Sufficiente, insufficiente Nuovo_preventivo, Apertura_preventivo_esistente
<b>Esame Pratica</b>	Accetta:Boolean	true	True, false
<b>Ricezione ricevuta pagamento</b>	Pagamento_ok:Boolean	true	True, false

The next step is to add conditions in transitions. click on each one of the outgoing transitions starting from each activity below described and set a condition by using the local variables

### Welcome pages:

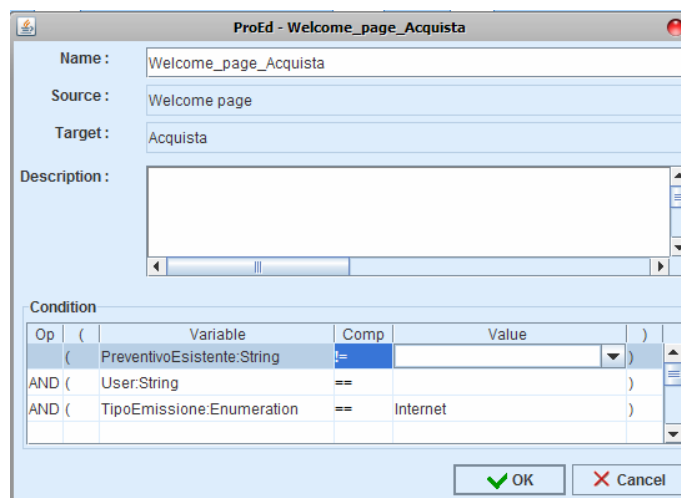


Figure 33 Condition between Welcome page and Acquista activities

ProEd - Welcome page\_Chiamata Call Center

Name : Welcome page\_Chiamata Call Center

Source : Welcome page

Target : Chiamata Call Center

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	TipoEmissione:Enumeration	==	CallCenter	)
AND	(	User:String	==		)
AND	(	PreventivoEsistente:String	==		)

OK Cancel

Figure 34 Condition between Welcome page and Chiamata call center activities

ProEd - Welcome page\_Apertura Pratica CallCenter

Name : Welcome page\_Apertura Pratica CallCenter

Source : Welcome page

Target : Home Page CallCenter

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	User:String	==	Call	)
AND	(	Password:String	==	Call	)

OK Cancel

Figure 35 Condition between Welcome page and Home page callcenter activities

ProEd - Welcome page\_Apertura pratica Direzione

Name : Welcome page\_Apertura pratica Direzione

Source : Welcome page

Target : Apertura pratica Direzione

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	User:String	==	Dir	)
AND	(	Password:String	==	Dir	)

OK Cancel

Figure 36 Condition between Welcome page and Apertura pratica direzione activities

ProEd - Welcome page\_Compila Preventivo Internet

Name : Welcome page\_Compila Preventivo Internet

Source : Welcome page

Target : Compila Preventivo Internet

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	PreventivoEsistente:String	==		)
AND	(	User:String	==		)
AND	(	TipoEmissione:Enumeration	==	Internet	)

OK Cancel

Figure 37 Condition between Welcome page and Compila preventivo Internet activities

### Compila Preventivo Internet

ProEd - Compila Preventivo Internet\_Emissione Preventivo Internet

Name : Compila Preventivo Internet\_Emissione Preventivo Internet

Source : Compila Preventivo Internet

Target : Emissione Preventivo Internet

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	DataNascita:Date	<	1990-01-01 11:57:13	)
AND	(	Frazionamento:Enumeration	!=	Temporaneo	)
AND	(	TipoVeicolo:Enumeration	!=	Altro	)
AND	(	ValoreDichiarato:Float	<	100000	)

OK Cancel

Figure 38 Condition between Compila Preventivo Internet and Emissione Preventivo Internet activities

ProEd - Compila Preventivo Internet\_Chiamata Call Center?

Name : Compila Preventivo Internet\_Chiamata Call Center?

Source : Compila Preventivo Internet

Target : Chiamata Call Center?

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	DataNascita:Date	>	1990-01-01 11:57:13	)
OR	(	Frazionamento:Enumeration	==	Temporaneo	)
OR	(	TipoVeicolo:Enumeration	==	Altro	)
OR	(	ValoreDichiarato:Float	>	100000	)

OK Cancel

Figure 39 Condition between Compila Preventivo Internet and Chiamata call center? activities

**Acquista:**

The screenshot shows a dialog box titled "ProEd - Acquista\_Invio doc. e pagamento". It contains the following fields:

- Name: Acquista\_Invio doc. e pagamento
- Source: Acquista
- Target: Invio doc. e pagamento e eventuale conguaglio
- Description: (empty text area)
- Condition table:

Op	(	Variable	Comp	Value	)
(		Acquista:Boolean	==	true	)

Buttons: OK, Cancel

**Figure 40** Condition between Acquista and Invio doc activities

**Chiama Call Center? :**

The screenshot shows a dialog box titled "ProEd - Chiama Call Center?\_End". It contains the following fields:

- Name: Chiama Call Center?\_End
- Source: Chiama Call Center?
- Target: End
- Description: (empty text area)
- Condition table:

Op	(	Variable	Comp	Value	)
(		Chiamo:Boolean	==	false	)

Buttons: OK, Cancel

**Figure 41** Condition between Chiama call center? and End activities

The screenshot shows a dialog box titled "ProEd - Chiama Call Center?\_Chiamata Call Center". It contains the following fields:

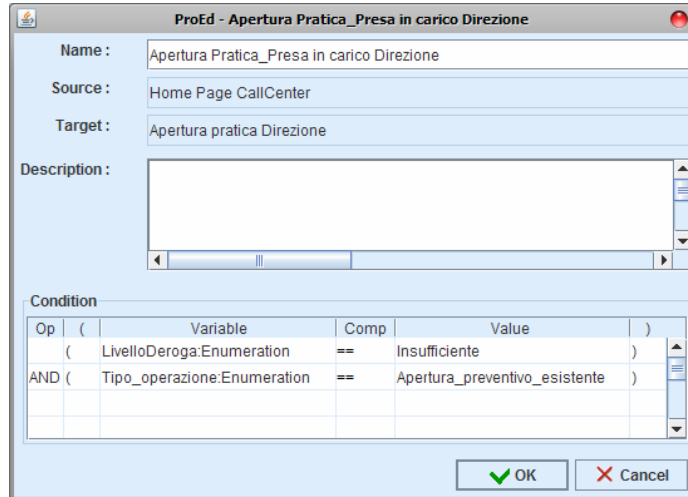
- Name: Chiama Call Center?\_Chiamata Call Center
- Source: Chiama Call Center?
- Target: Chiamata Call Center
- Description: (empty text area)
- Condition table:

Op	(	Variable	Comp	Value	)
(		Chiamo:Boolean	==	true	)

Buttons: OK, Cancel

**Figure 42** Condition between Chiama call center? and Chiamata call center activities

## Home Page Call Center:



ProEd - Apertura Pratica\_Presa in carico Direzione

Name : Apertura Pratica\_Presa in carico Direzione

Source : Home Page CallCenter

Target : Apertura pratica Direzione

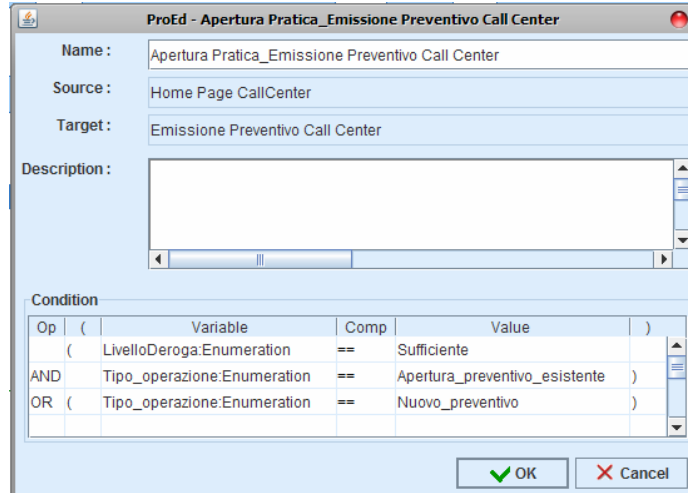
Description :

Condition

Op	(	Variable	Comp	Value	)
	(	LivelloDeroga:Enumeration	==	Insufficiente	)
AND	(	Tipo_operazione:Enumeration	==	Apertura_preventivo_esistente	)

OK Cancel

Figure 43 Condition between Home Page Call Center and Apertura pratica direzione activities



ProEd - Apertura Pratica\_Emissione Preventivo Call Center

Name : Apertura Pratica\_Emissione Preventivo Call Center

Source : Home Page CallCenter

Target : Emissione Preventivo Call Center

Description :

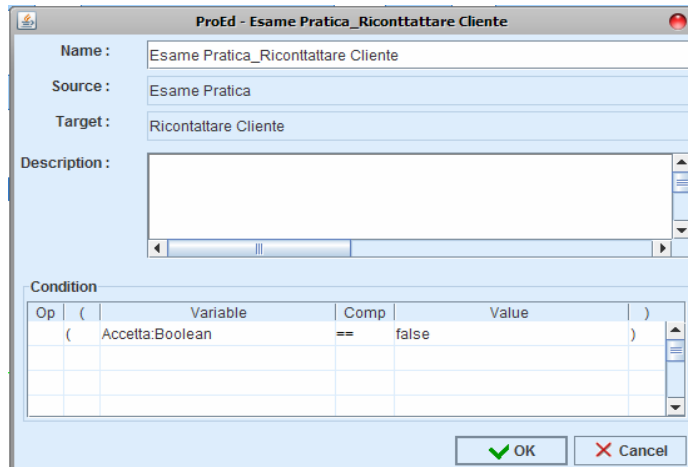
Condition

Op	(	Variable	Comp	Value	)
	(	LivelloDeroga:Enumeration	==	Sufficiente	)
AND	(	Tipo_operazione:Enumeration	==	Apertura_preventivo_esistente	)
OR	(	Tipo_operazione:Enumeration	==	Nuovo_preventivo	)

OK Cancel

Figure 44 Condition between Home Page Call Center and Emissione Preventivo Call Center activities

## Esame Pratica:



ProEd - Esame Pratica\_Ricontattare Cliente

Name : Esame Pratica\_Ricontattare Cliente

Source : Esame Pratica

Target : Ricontattare Cliente

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	Accetta:Boolean	==	false	)

OK Cancel

Figure 45 Condition between Esame Pratica and Ricontattare Cliente activities

ProEd - Esame Pratica\_Emissione Preventivo Call Center

Name : Esame Pratica\_Emissione Preventivo Call Center

Source : Esame Pratica

Target : Emissione Preventivo Call Center

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	Accetta:Boolean	==	true	)

OK Cancel

Figure 46 Condition between Esame Pratica and Emissione Preventivo Call Center activities

Ricezione ricevuta pagamento:

ProEd - Ricezione ricevuta pagamento\_Esame pratica Finance

Name : Ricezione ricevuta pagamento\_Esame pratica Finance

Source : Ricezione ricevuta pagamento

Target : Esame pratica Finance

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	pagamento_ok:Boolean	==	false	)

OK Cancel

Figure 47 Condition between Ricezione ricevuta pagamento and Esame pratica Finance activities

ProEd - Ricezione ricevuta pagamento\_Invio tagliando & Doc.

Name : Ricezione ricevuta pagamento\_Invio tagliando & Doc.

Source : Ricezione ricevuta pagamento

Target : Invio tagliando & Doc.

Description :

Condition

Op	(	Variable	Comp	Value	)
	(	pagamento_ok:Boolean	==	true	)

OK Cancel

Figure 48 Condition between Ricezione ricevuta pagamento and Invio tagliando & Doc. activities

Let's finish the Emission proposal process definition by adding a new connector to the Web Sale Eclipse project

Remember that you referenced three connectors called "Message", "Numero\_Preventivo" and "Numero\_Polizza" during the manual activities creation. Let's so create a new connector/hook using the Bonita designer wizard. Select your directory called "java" in your Emission proposal Eclipse project and click right button -> "New" → "Other" -> "Bonita BPM" -> "New Bonita Hook". Type the name "Message" and click on finish.

Repeat the procedure to create "Numero\_Polizza" hook.

In Message.java just put this code line `System.out.println("PRPxxxxxxxx")`  
In Numero\_Poliza.java add `System.out.println("POLxxxxxxxx")`

Select your directory called "java" in your Emission proposal Eclipse project and click right button → "New" → "Other" → "Bonita BPM" → "New Bonita Transactional Hook". Type "Numero\_Preventivo" on the textbox name and click on finish.

In the parameters of the *execute* method replace `arg0` by `APIAcc` and `arg1` by `actInstance`

Lets' now add our custom implementation of this connector:

```
FileOutputStream out;  
PrintStream p;  
try{  
    Date d=null;  
    String variableData=new String("Nome");  
    QueryRuntimeAPI queryRuntimeAPI = APIAcc.getQueryRuntimeAPI();  
    try{  
        d = (Date)  
queryRuntimeAPI.getProcessInstanceVariable(actInstance.getProcessInsta  
nceUUID(), variableData);  
    }catch(Exception e){  
System.out.println("Istanza non valida:  
"+actInstance.getProcessDefinitionUUID()+"\nOppure\nNome variabile  
errata: "+variableData);  
        return;  
    }  
    System.out.println(d);  
}catch (Exception e) {  
    System.err.println ("Error writing to file");  
}
```

You have just finished the definition of your first BPM process with Nova Bonita. You are now ready to deploy it into the runtime.

To deploy and run the process, please refer to chapter 5 *Running a process*