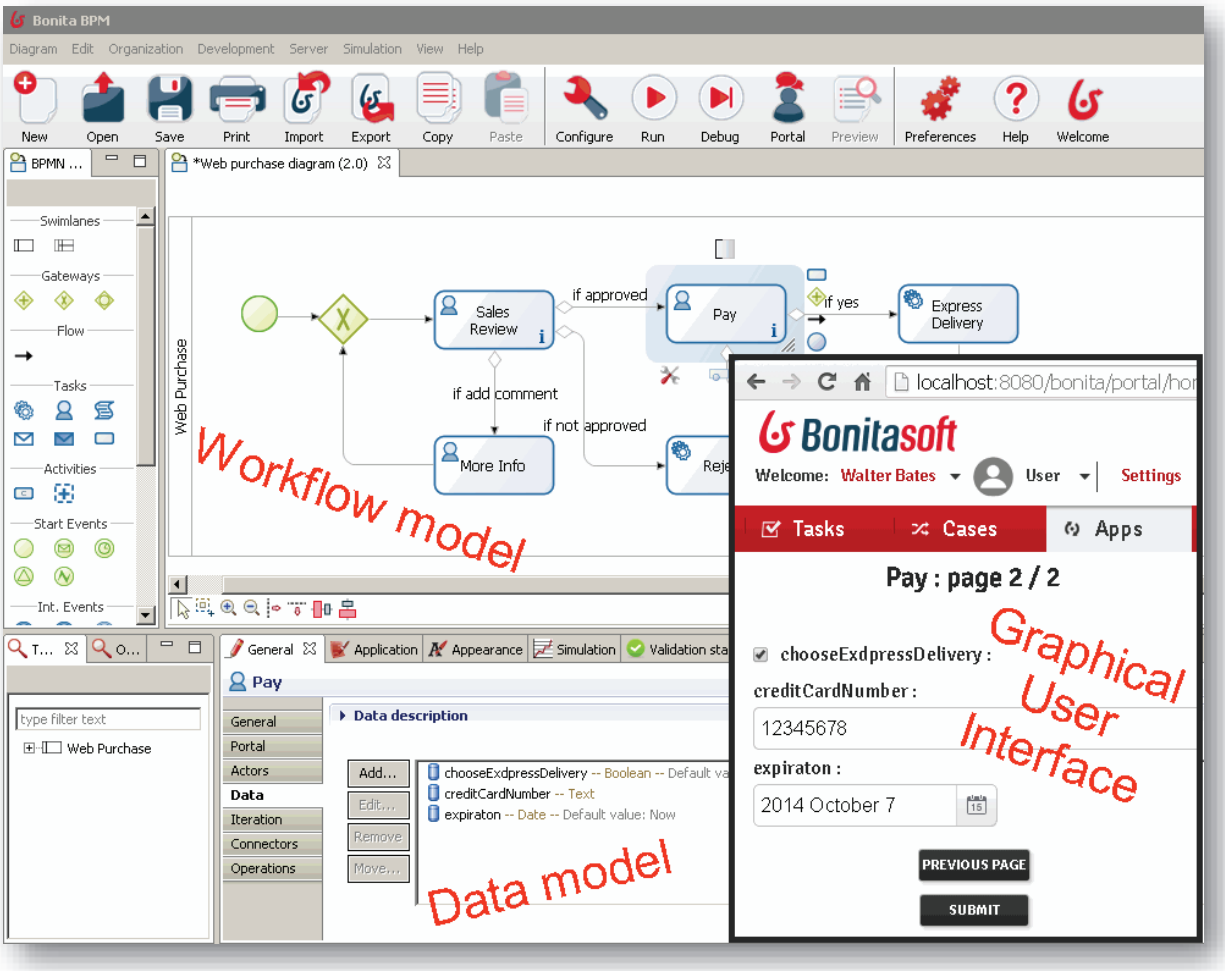
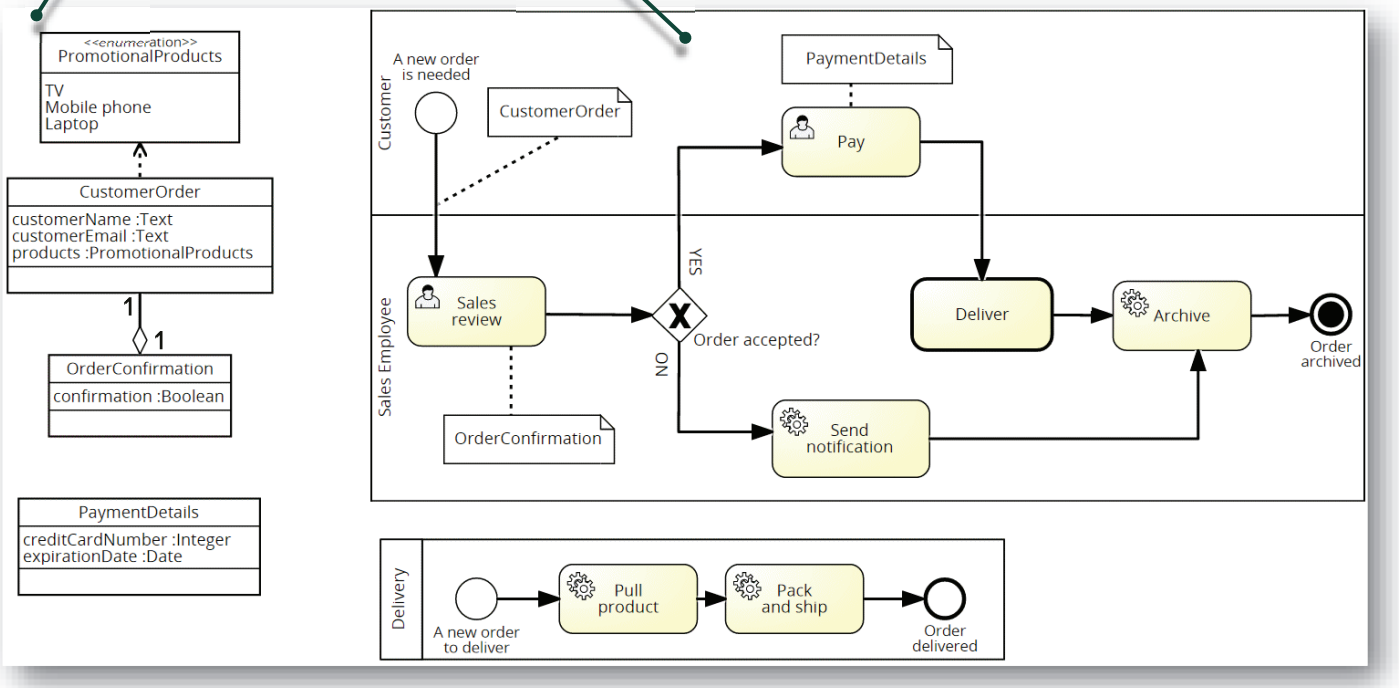
Introduction to Bonita BPM

Mario G.C.A. Cimino, DII, University of Pisa

- ✓ Bonita BPM 7 is a powerful application platform for building personalized, process-based business applications that adapt to your business changes in real time.
- ✓ Bonita BPM has two parts: the development environment, Bonita BPM Studio, and the runtime environment, Bonita BPM Platform.
- ✓ Bonita BPM adopts the **model-driven approach**, a software design methodology for the development of software systems, launched by the Object Management Group (OMG) in 2001.
- ✓ With model-driven engineering, specifications are expressed as models. Models can be expressed with standards, such as the executable Unified Modeling Language (UML), and the BPMN.
- ✓ Models are then processed to automatically generate software. Code generation means that an automated tool derives from the models parts or all of the source code for the software system.



- ✓ Our first model, edited with Signavio
- ✓ **Class diagram** (data model)
- ✓ **Business Process diagram** (workflow model + data objects)



1. Download the Bonita BPMS from

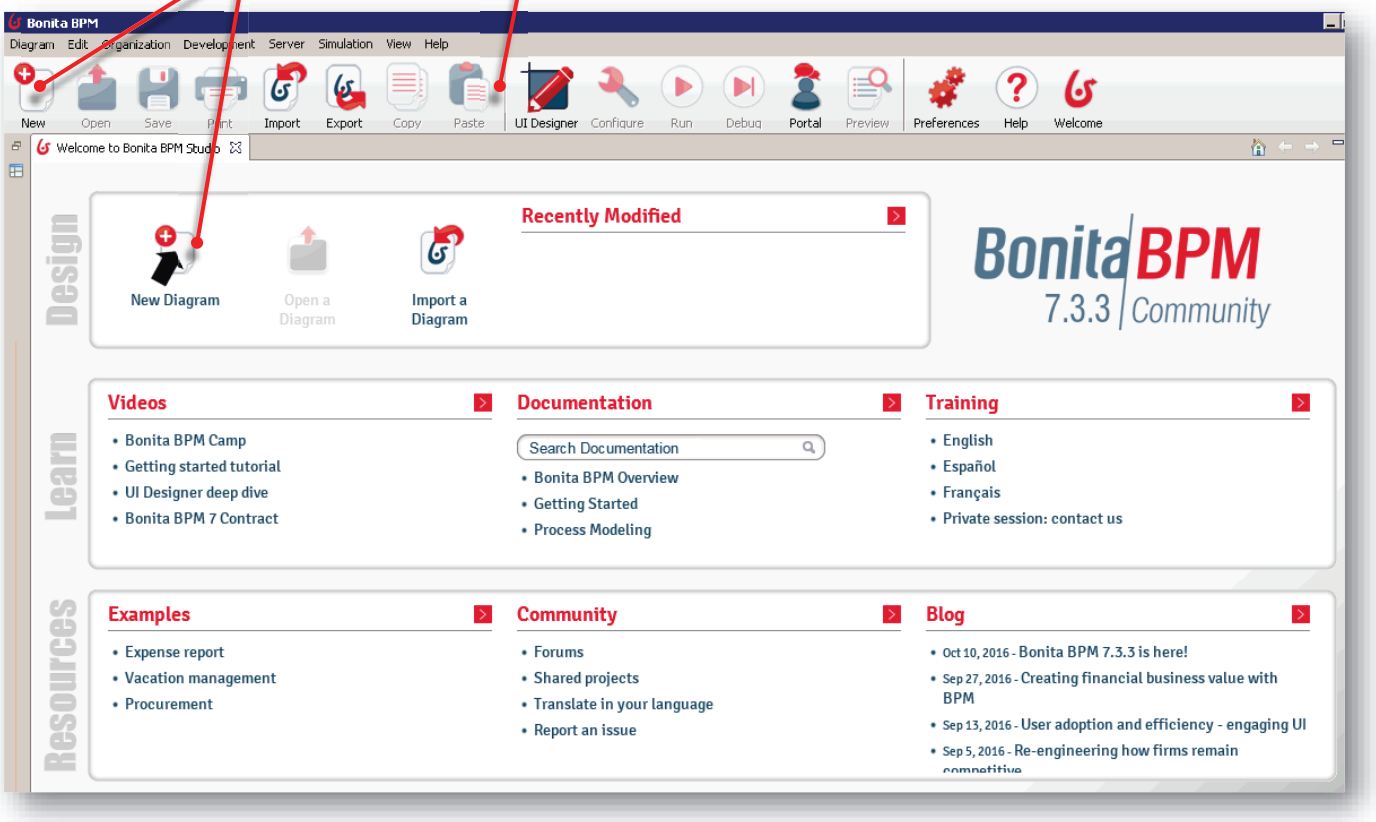
http://docenti.ing.unipi.it/m.cimino/_sw/BonitaBPMCommunity-7.5.4.zip
http://docenti.ing.unipi.it/m.cimino/_sw/jdk8.zip

2. Extract it to c:\pmi

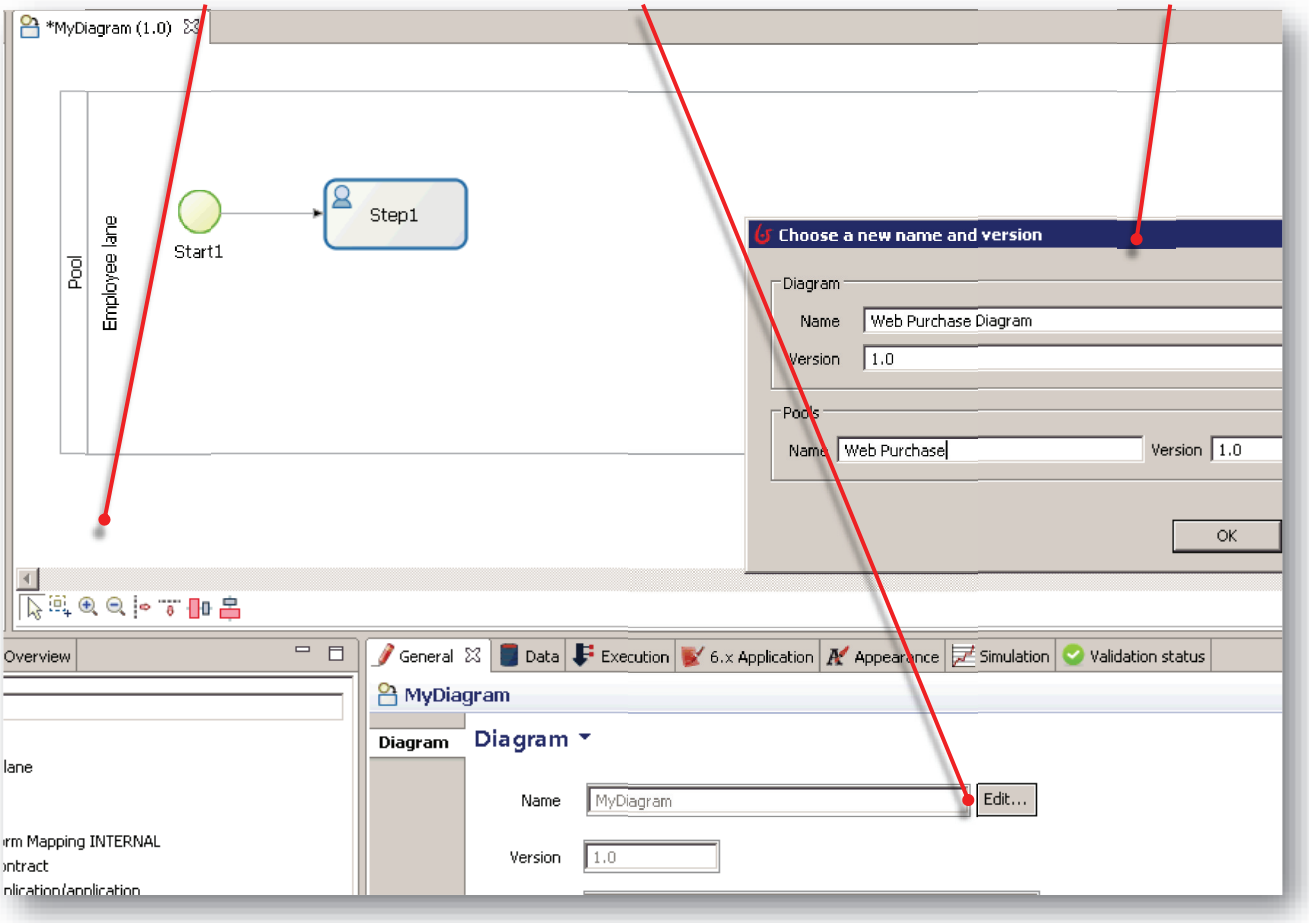
3. If needed, change the JDK: create a batch file (go.bat)

set JAVA_HOME=C:\pmi\jdk8
set PATH=C:\pmi\jdk8\bin;%PATH%
java -version
BonitaBPMCommunity64.exe

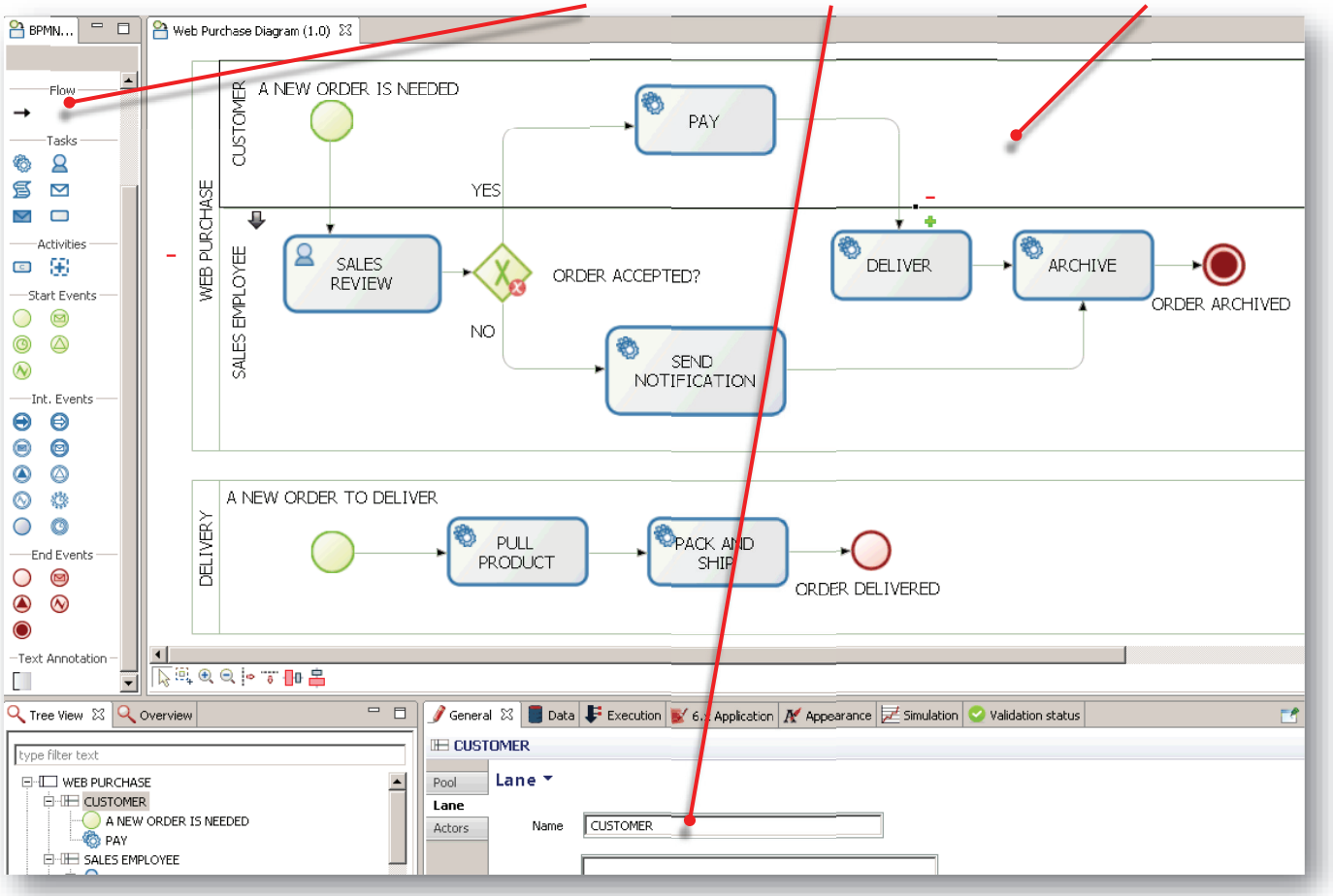
1. Select **New** from the Cool bar to create a new diagram

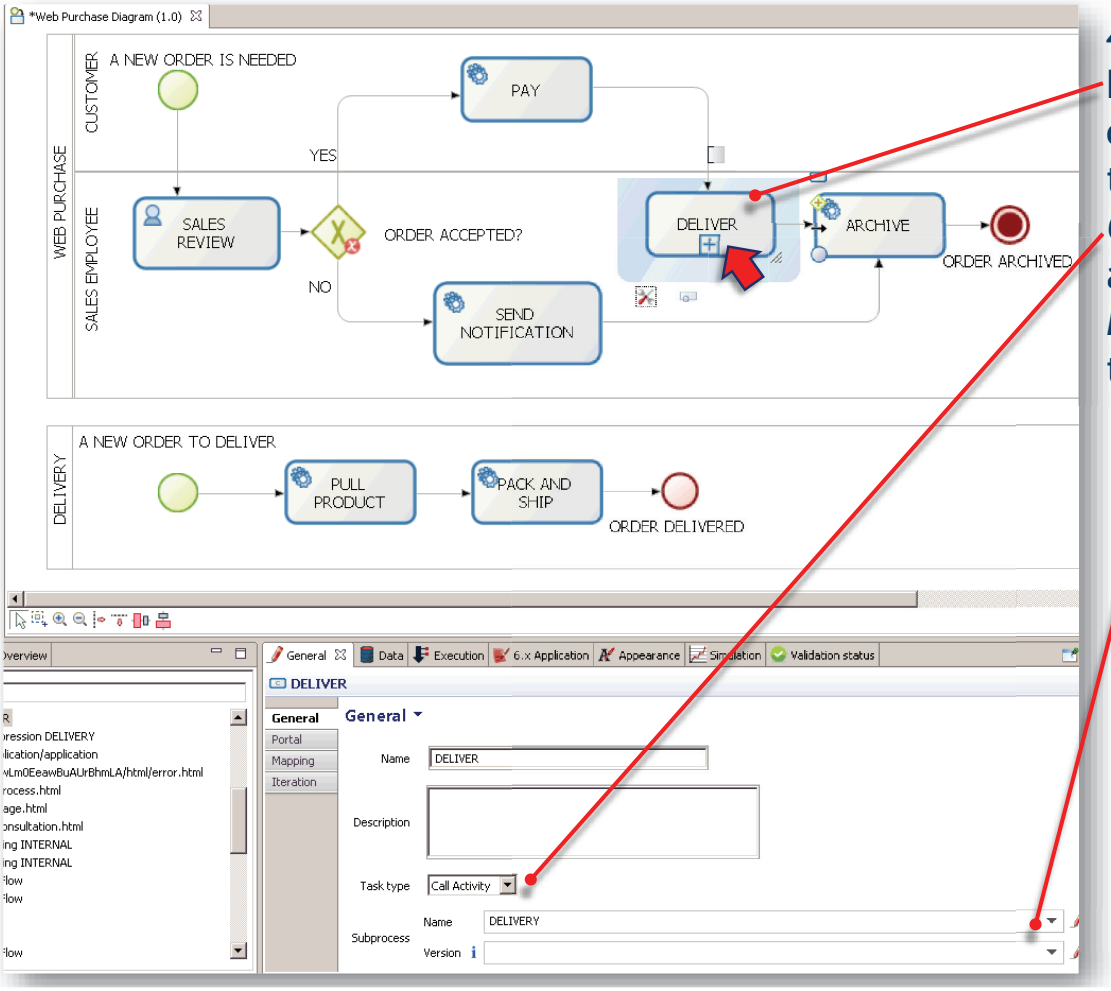


2. Click outside the pool, click on Edit, Enter Diagram and Pool Name



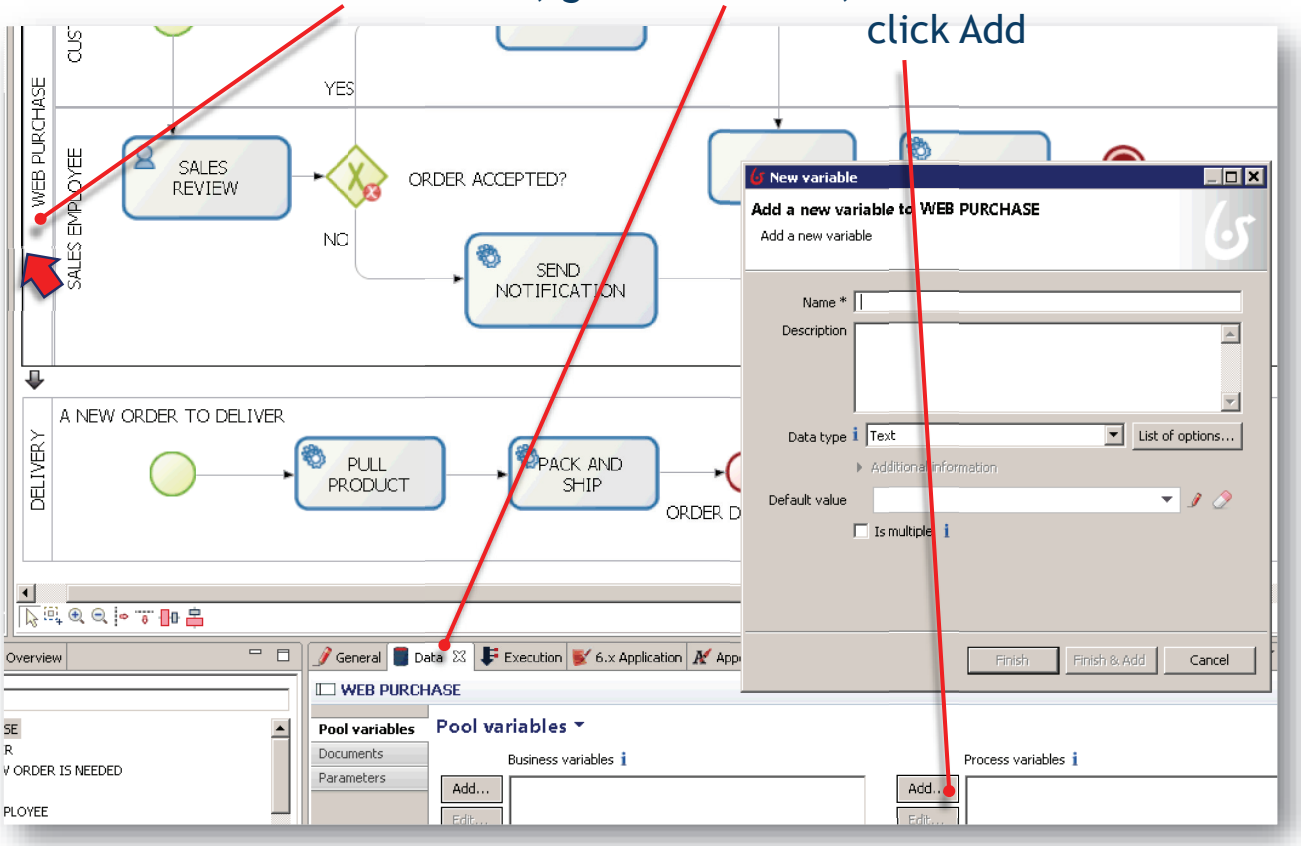
3. Create the diagram using the toolkit, configure the selected element





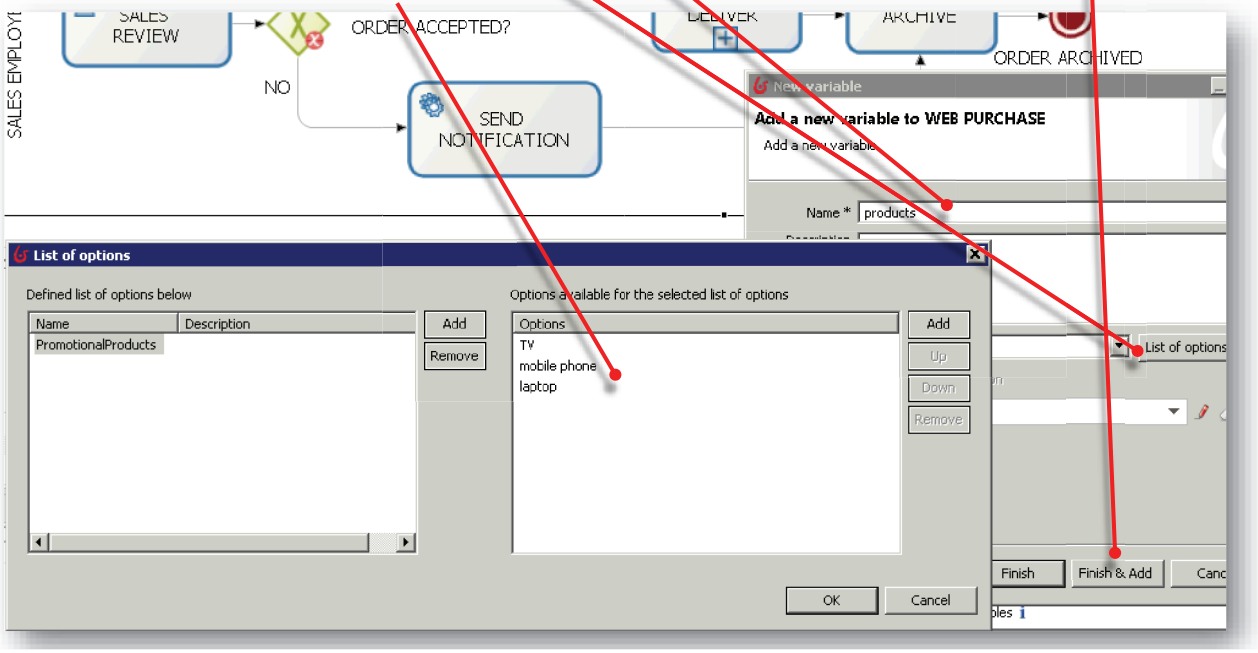
4. Select the Deliver task, choose the task type *Call Activity*, and choose *Delivery* as a target name

5. Select the Web Purchase Pool, go to Data Pane, on Process variable click Add



Process variables: can be used in a process and until the process instance is completed.

6. Enter *customerName*, leave Data type *Text*, and press *Finish&Add*
   enter *customerEmail*, leave Data type *Text*, and press *Finish&Add*
   enter *creditCardNumber*, Data type *Integer,* press *Finish&Add*
   enter *expirationDate,* Data type *Date,* press *Finish&Add*
   enter *confirmation*, Data type Boolean, press *Finish&Add*
   enter *products,* click on *List of options*, Name: *PromotionalProducts*,
   Options: *TV, Mobile phone e laptop*.

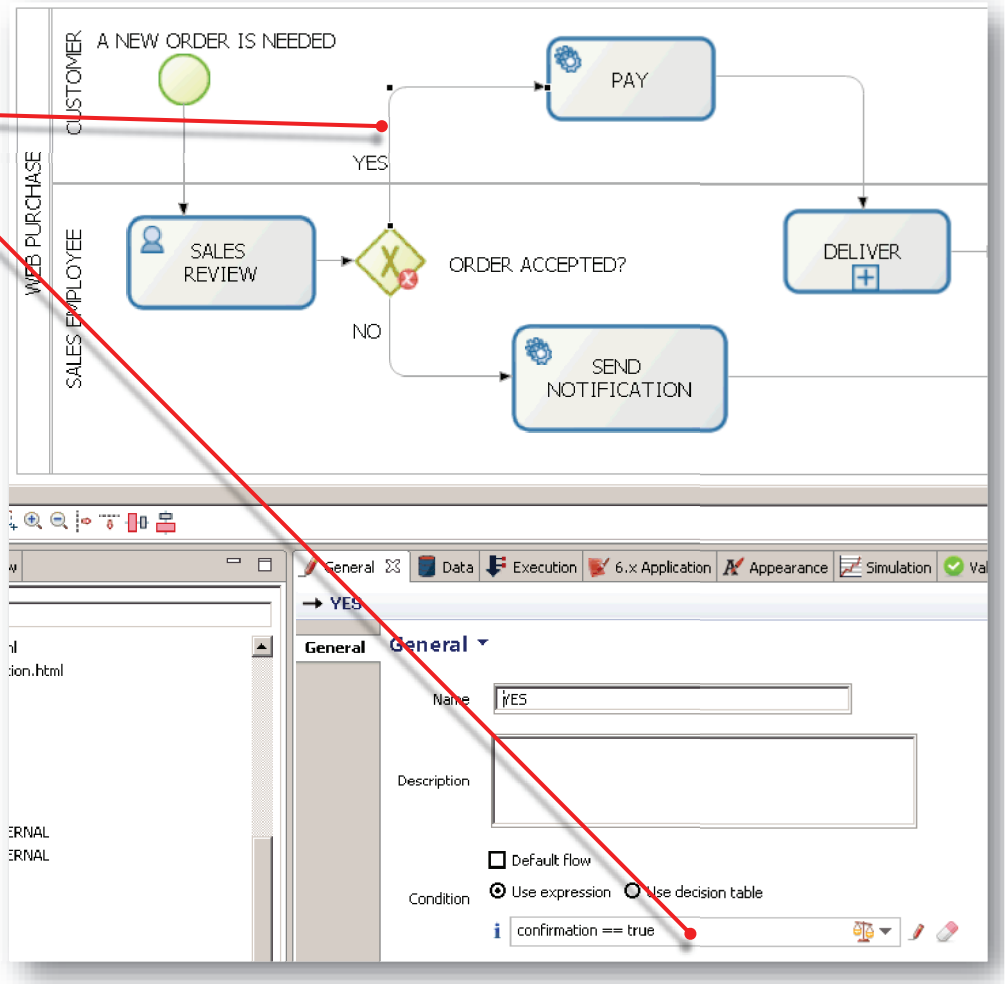


Bonita BPM: set expressions for the exclusive transitions 24 of 85

Transitions:

Select the branch “YES”, enter the expression “confirmation == true”

Select the branch “NO”, anter the expression “confirmation == false”.



Connector:

Select the *Send notification* task,

Select *Execution* tab

Select *Connectors out*

➢ *Add*

In the popup windows Select *Categories Messaging*

➢ Email (SMTP)

Name: *Send notification*

➢ Next

➢ enter gmail account (*)

➢ Next

➢ enter *from, to*

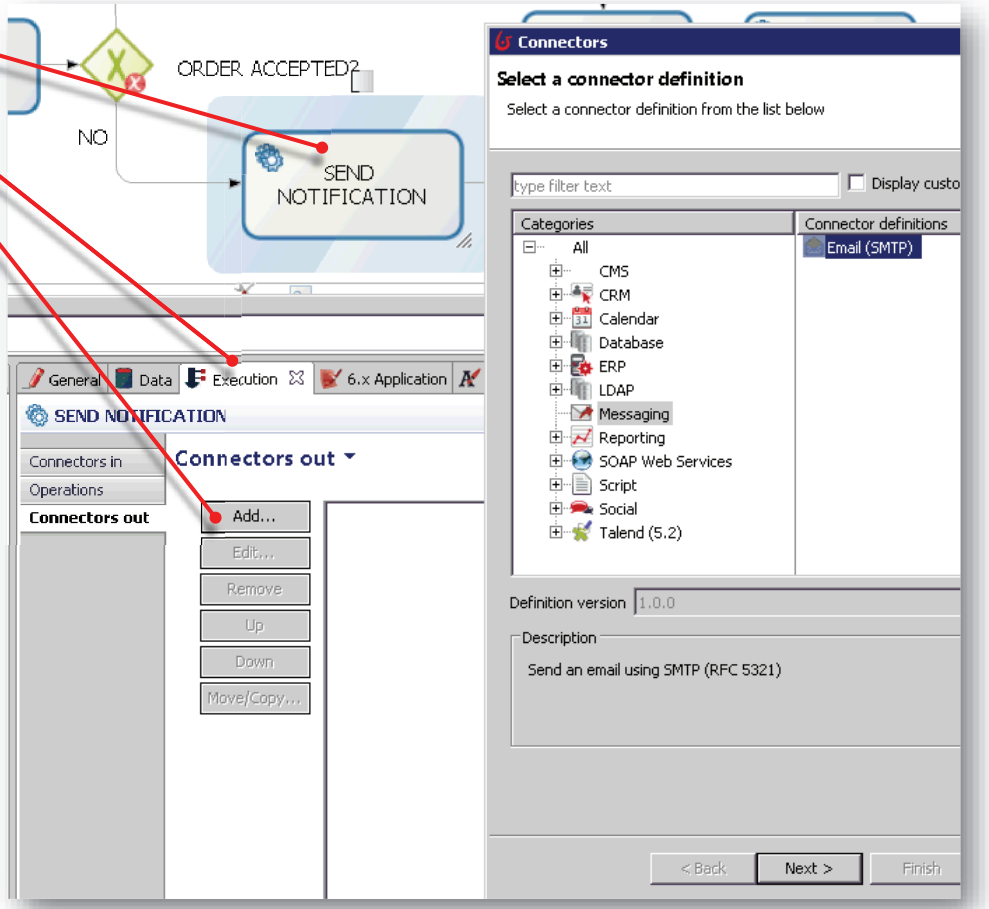
➢ *Next*

➢ enter

Subject: *Order refused*,

Message: *We are unable to fill your order at this time*

➢ Click on test, ignore the popup warning, check your email, > Finish.



(*) to send a test email from google to google account, go to https://myaccount.google.com/lesssecureapps and set "Allow less secure apps: ON".

Connector:

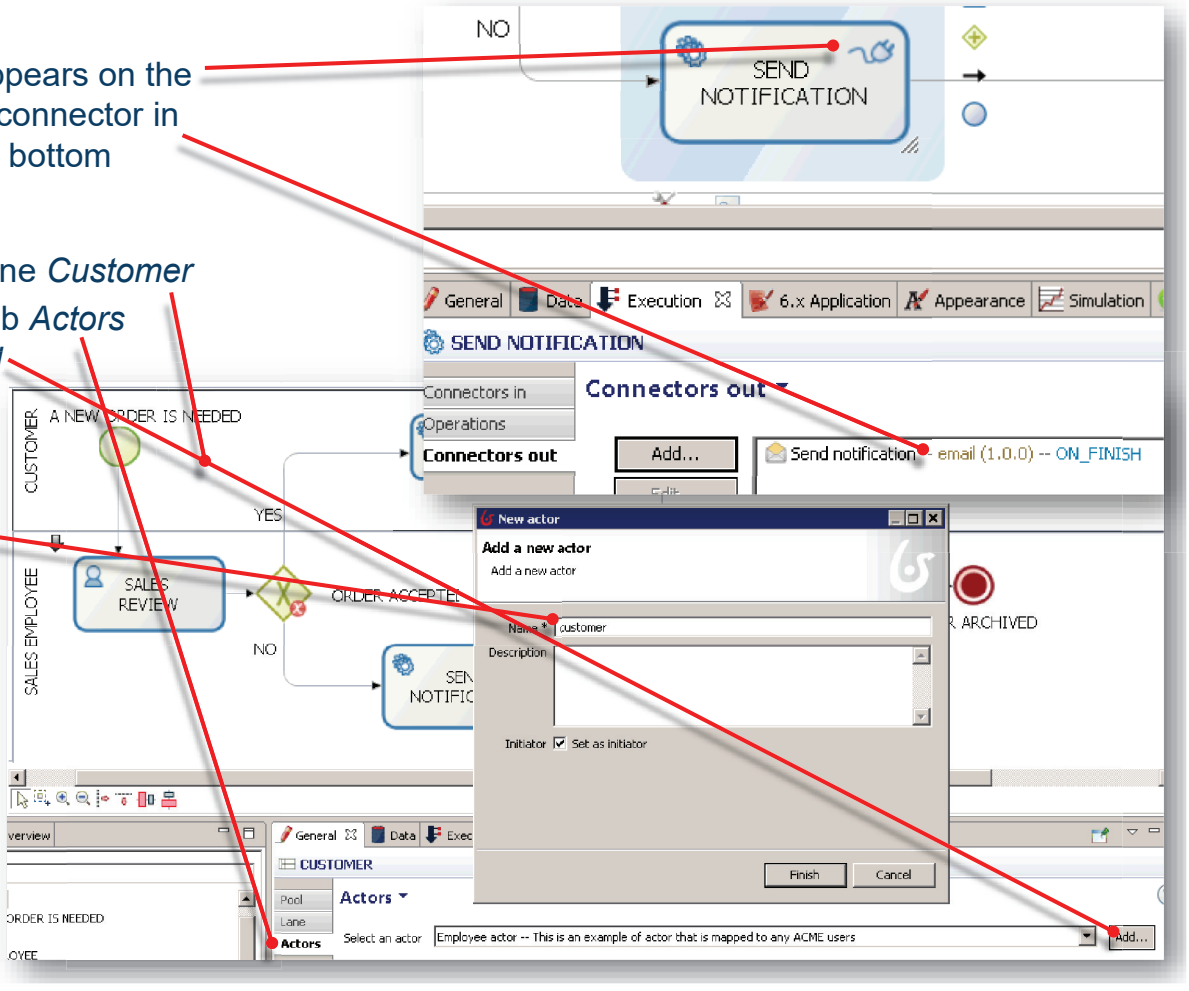
A plug icon appears on the task, and the connector in the tab on the bottom

Actors:

- Select the lane *Customer*
- Select the tab *Actors*
- Click on *Add*
- Name: *customer*
- *Set as initiator*

Check *set as initiator*

- Select the Lane *Sales Employee*
- *Add*
- Name: *employee*
- Finish



Mapping Actors –people

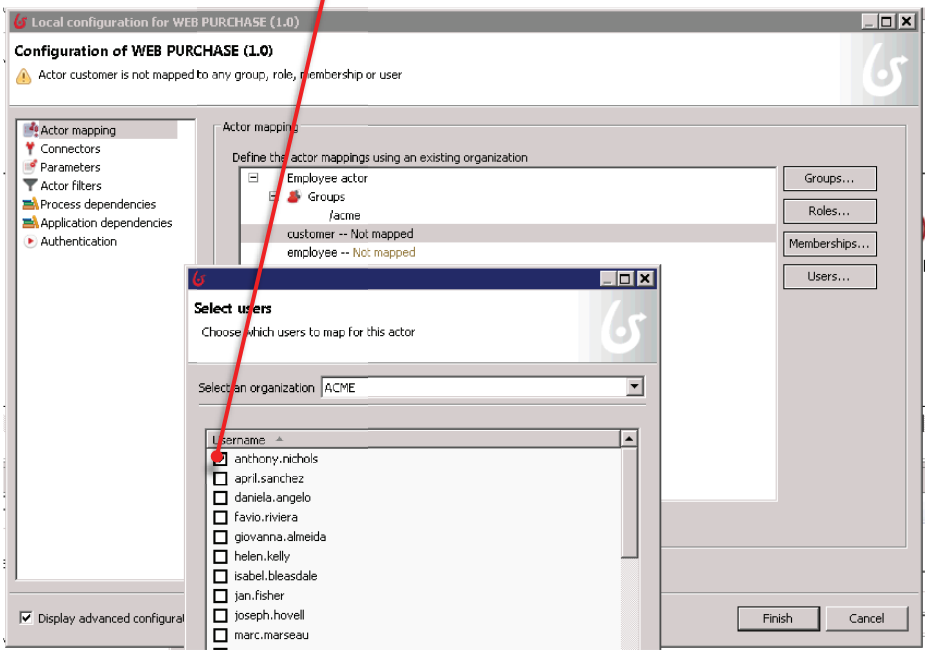
- Click on *Configure* on the cool bar.
- Select *customer* in Actor mapping > *Users* > *anthony.nichols (pwd* bpm)

Similarliy

- Click on *Configure* on the cool bar.
- Select *employee* in Actor mapping > *Users* > *april.sanchez (pwd* bpm)

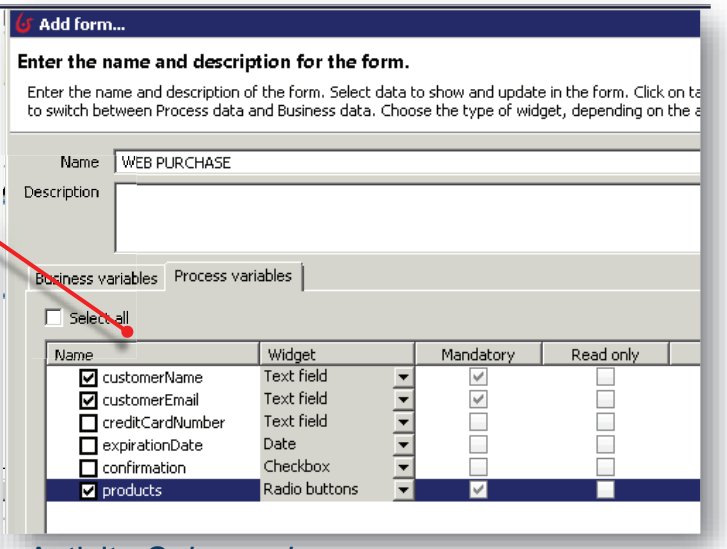
Forms and Data Objects

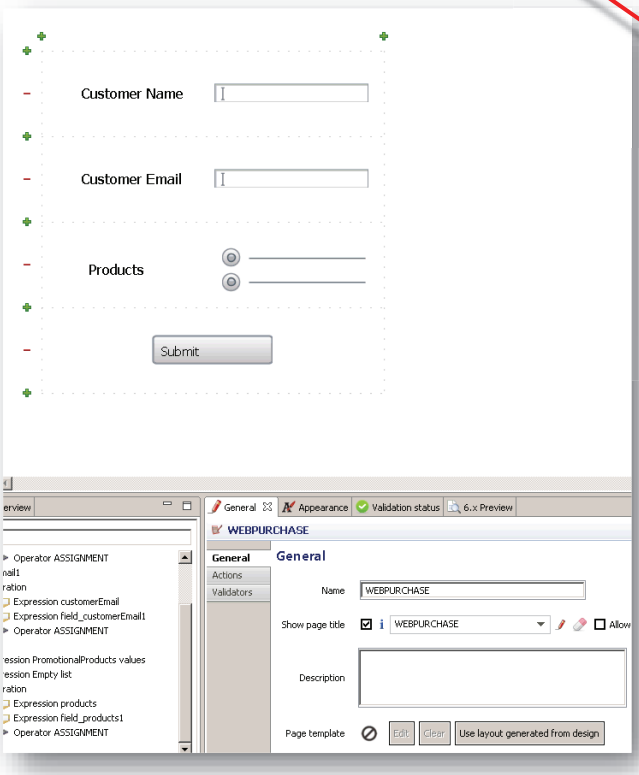
- Select the *Web Purchase Pool* > Tab *Execution* > *Tab Instantiation Form*
  - 6.x
  - 6.x Application
  - Add
  - …



Forms and Data Objects

- Select the Process Variables *customerName, customerEmail*, *products.*
- A default form is created





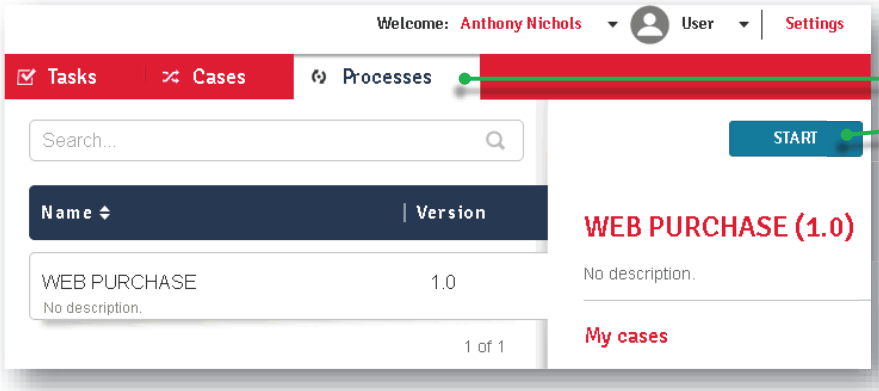
Activity *Sales review*

- Tab *Execution* > *Tab Form* > *6.x* Application > Add Select *customerName*, *customerEmail*, and *products* as read only; finally add *confirmation*

Activity *Pay*

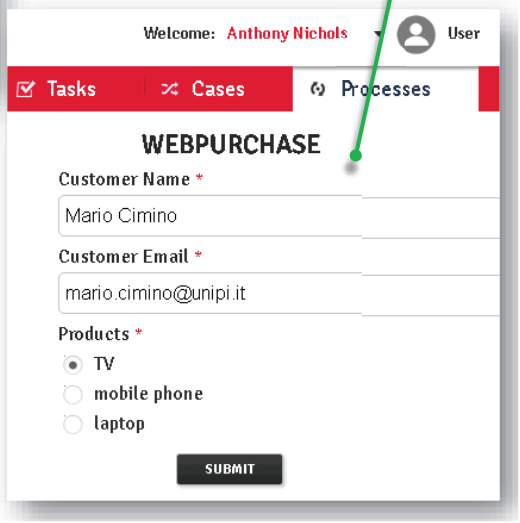
- Tab *General* > *task type: Human*
- Tab *Execution* > *Tab Form* > *6.x Application* > Add > Select *creditCardNumber*, and *expiration Date.*

- Click the Run button in the Cool bar
- Open two different browsers and point to http://localhost:8080/bonita/login.jsp
- First browser > customer login > username: *anthony.nichols* password: *bpm*
- Second browser > employee login > username: *april.sanchez* password: *bpm*



- as a customer, on the tab *processes* click on *start*

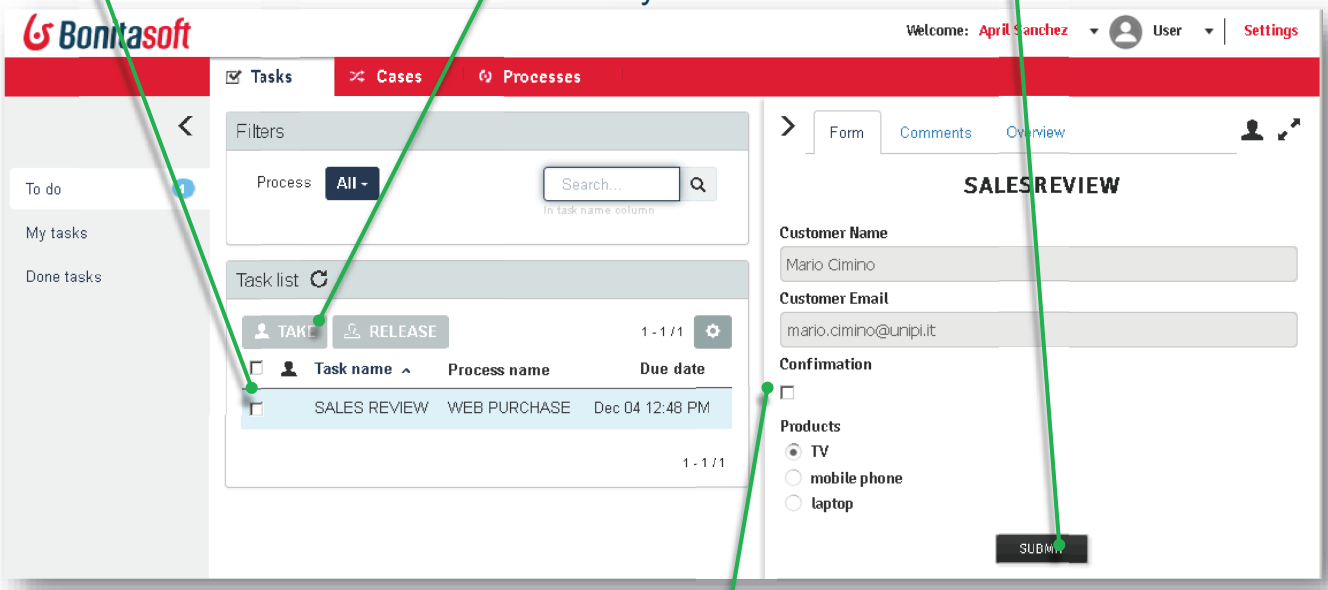
- fill the Web Purchase form and click on *submit*



- In the second browser, as an employee, on the tab tasks there is a task **to do**

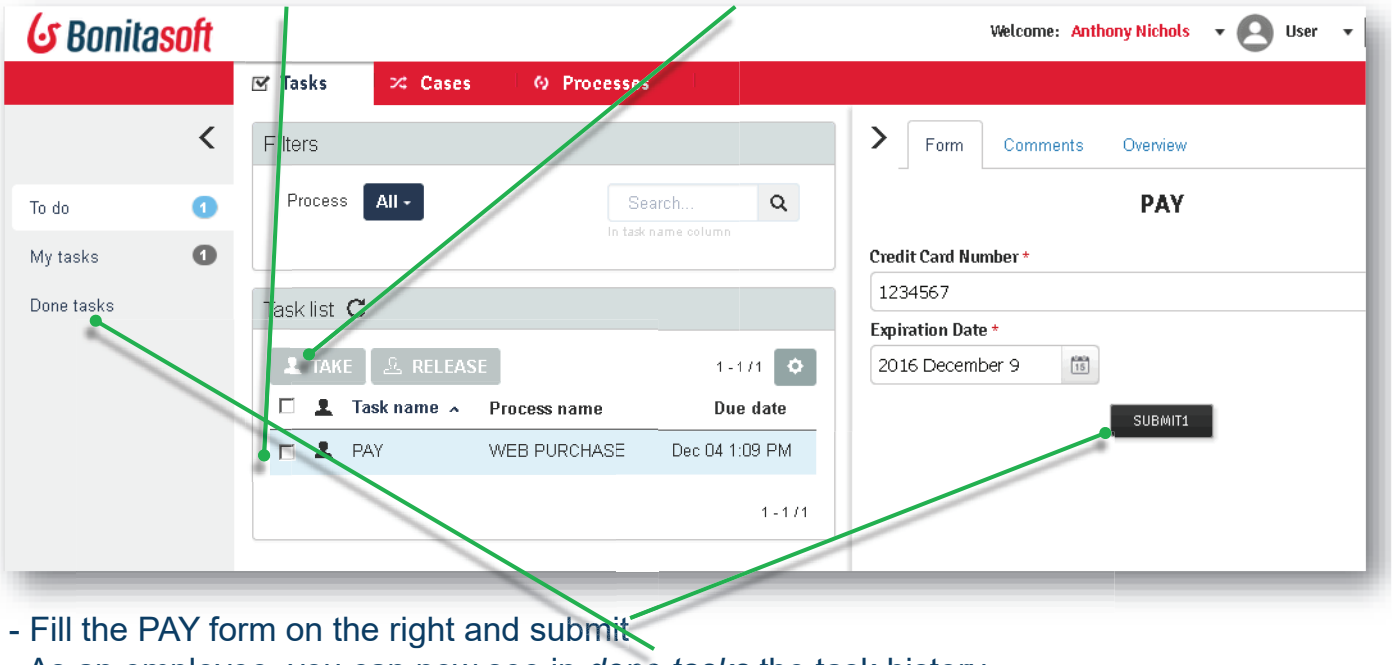
- Select the task and press *take*

- first case: press on SUBMIT (without confirmation) > you will receive an email



- As a customer, start a new process in the first browser
- Fill again the customer form and submit
- As an employee, check the confirmation flag and submit

- As a customer, select the task PAY and press *take*



- Fill the PAY form on the right and submit
- As an employee, you can now see in *done tasks* the task history

1. **Create the diagram above (for detailed steps see the first tutorial):**
2. New Diagram > complete the flow with the toolkit leaving the default task types.
3. Select *Step1* > *General* Tab > *Task type:* Service.
4. Select *Step2* > *General* Tab > *Task type:* Human.
5. Click on *Save* in the cool bar.
6. **Create the process variables:**
7. Select *Pool* > *Data* Tab > *Process Variables: Add* > Name: *customer* > *Finish & Add* > Name: *deposit* > *Finish*
8. ***Create the pool form***
9. Select *Pool* > Tab *Execution* > Instantiation form > 6.x
10. Tab *6.x Application* > *Add* > Select Tab *Process variables* > Select *deposit*, and *mandatory* > *Finish*
11. ***Create the Step2 form***
12. Select *Step2* > Tab *Execution* > form > 6.x
13. Tab *6.x Application* > *Add* > Select Tab *Process variables* > Select *customer*, and *read only* > *Finish*

14. Create the MySQL Database:

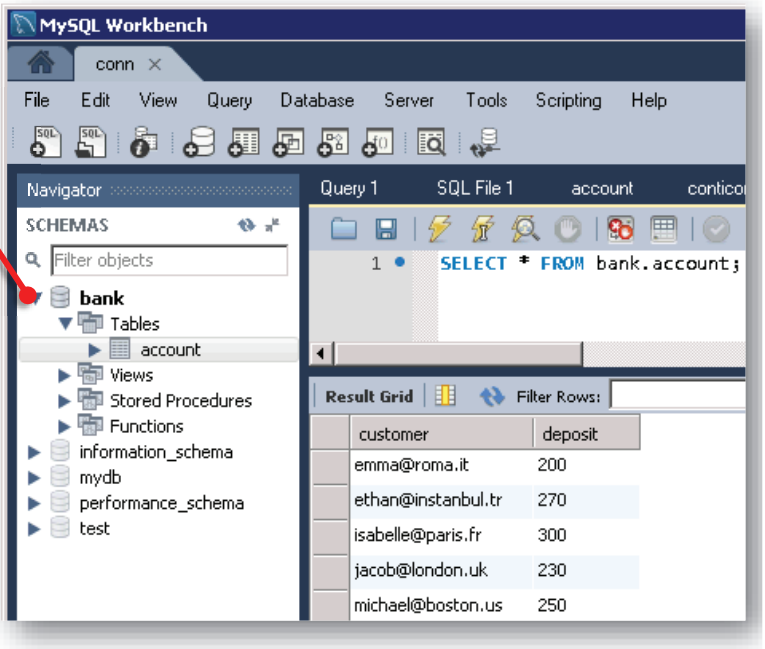
15. 1st method: import the file *bank-dump.sql* into a MySQL server.
16. 2nd method: download the file www.iet.unipi.it/m.cimino/wdis/res/dbms.zip and extract it on C:\wdis. Finally, click on C:\wdis\mysqlStart

17. Access the Database with MySQL client:

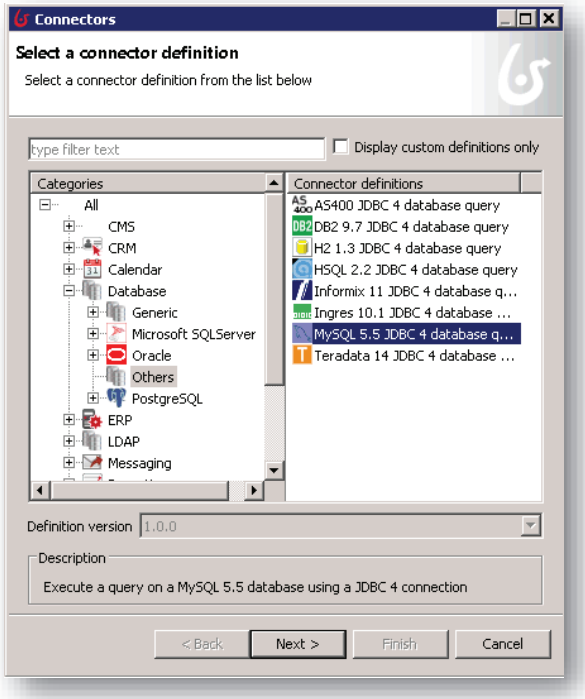
18. Click on C:\wdis\mysqlClient6.1 > Click on the "+" icon close to MySQL connections > enter a name and click OK.
19. Select the *bank* schema > *Tables* > *account* > right click > Select rows.

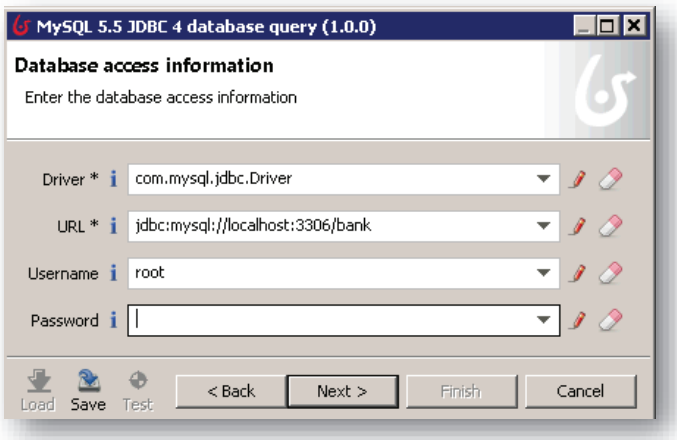
20. Create the DB Connector:

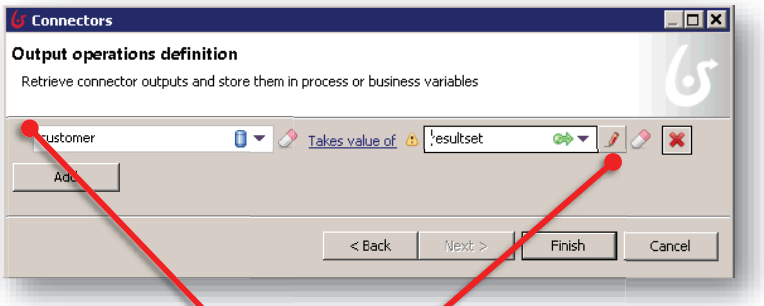
21. On Bonita, select *Step1* > Tab *Execution* > Connectors out (*) > *Add* > Categories: *Database* > *Others* > *Connector definition* > *MySQL 5.5* JDBC 4… > Next
22. Name: *dbconn1* > *Next.* Enter URL: *jdbc:mysql://localhost:3306/bank* Username: *root* Password: *Next*



(*) *Connectors out* are carried out at the end of the step, whereas *Connectors in* at the begin of the step.





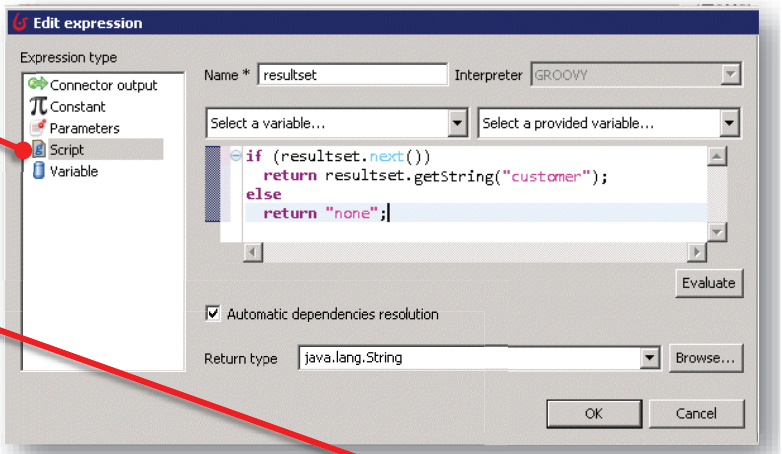


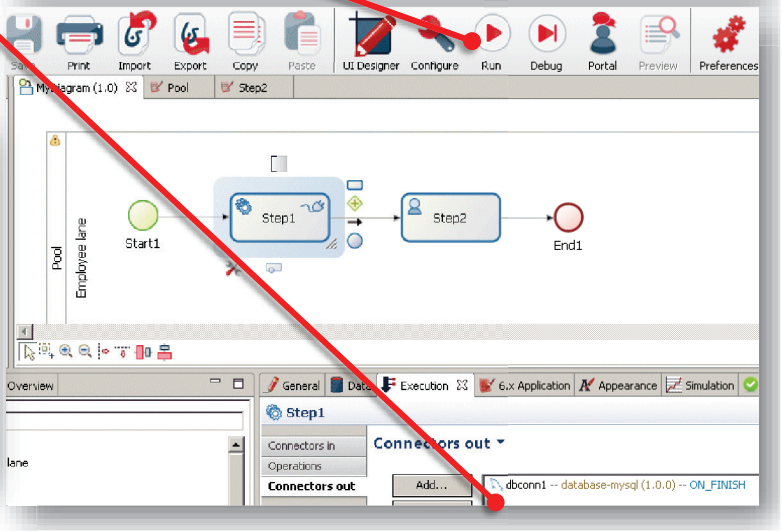
23. Enter the query
24. *SELECT * FROM account WHERE deposit > ${deposit};*
    (for autocompletion of variables press CTRL + SPACE)
23. Select *Next* > *Scripting Mode* > *Next* > Select target: *customer*
24. Click on the pencil icon to open the Groovy editor.

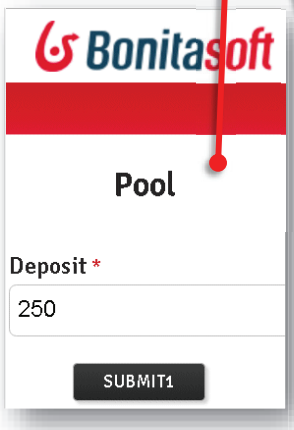
28. Expression type: *Script*
29. In the text area enter

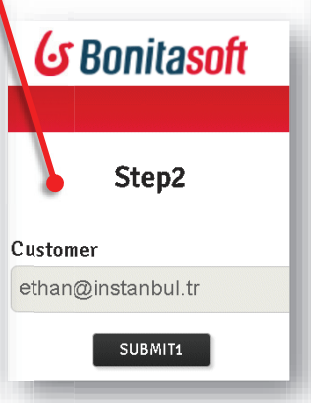
```
if (resultset.next())
 return resultset.getString("customer");
else
 return "none";
```

28. Click on OK > Finish.
29. *Click on Start button in the coolbar*
30. The Bonita launches the browser
31. Enter a deposit and SUBMIT
32. At Step 2, a customer with more than the deposit will be shown



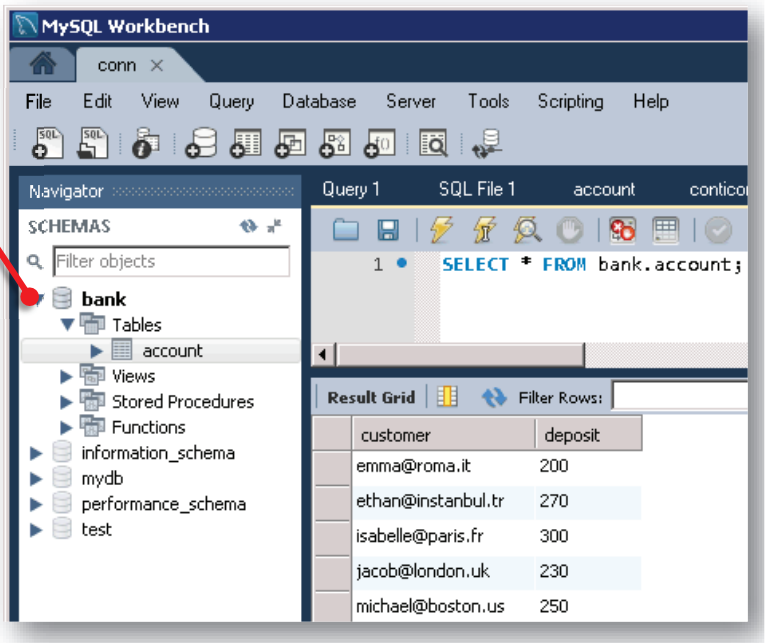




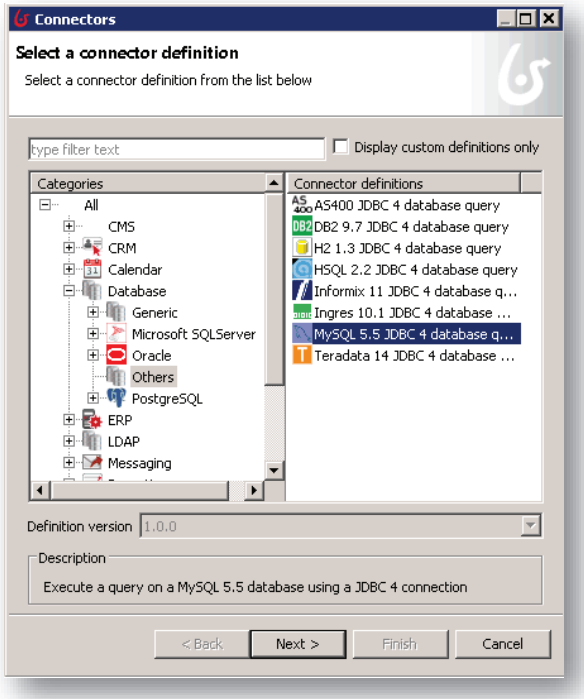


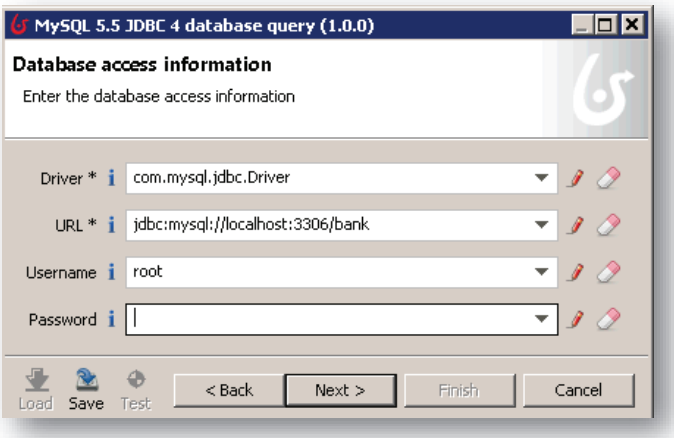
1. **Create the diagram above (for detailed steps see the first tutorial):**
2. New Diagram > complete the flow with the toolkit leaving the default task types.
3. Select *Step1* > *General* Tab > *Task type:* Service.
4. Select *Step2* > *General* Tab > *Task type:* Human.
5. Click on *Save* in the cool bar.
6. **Create the process variables:**
7. Select *Pool* > *Data* Tab > *Process Variables: Add* > Name: *customer* > *Finish & Add* > Name: *deposit* > *Finish*
8. ***Create the pool form***
9. Select *Pool* > Tab *Execution* > Instantiation form > 6.x
10. Tab *6.x Application* > *Add* > Select Tab *Process variables* > Select *deposit*, and *mandatory* > *Finish*
11. ***Create the Step2 form***
12. Select *Step2* > Tab *Execution* > form > 6.x
13. Tab *6.x Application* > *Add* > Select Tab *Process variables* > Select *customer*, and *read only* > *Finish*

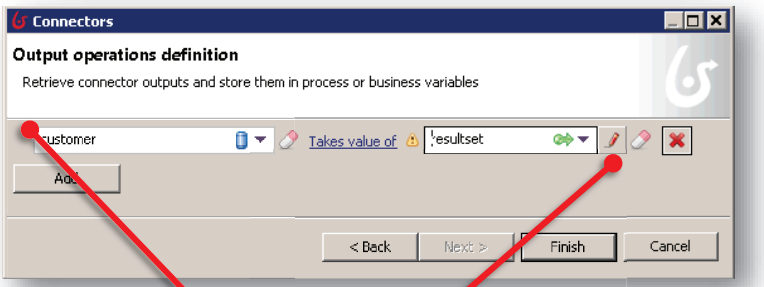
14. **Create the MySQL Database:**
15. 1st method: import the file *bank-dump.sql* into a MySQL server.
16. 2nd method: download the file www.iet.unipi.it/m.cimino/wdis/res/dbms.zip and extract it on C:\wdis. Finally, click on C:\wdis\mysqlStart
17. **Access the Database with MySQL client:**
18. Click on C:\wdis\mysqlClient6.1 > Click on the "+" icon close to MySQL connections > enter a name and click OK.
19. Select the *bank* schema > *Tables* > *account* > right click > Select rows.
20. **Create the DB Connector:**
21. On Bonita, select *Step1* > Tab *Execution* > Connectors out (*) > *Add* > Categories: *Database* > *Others* > *Connector definition* > *MySQL 5.5* JDBC 4… > Next
22. Name: *dbconn1* > *Next.* Enter URL: *jdbc:mysql://localhost:3306/bank* Username: *root* Password: *Next*



(*) *Connectors out* are carried out at the end of the step, whereas *Connectors in* at the begin of the process.





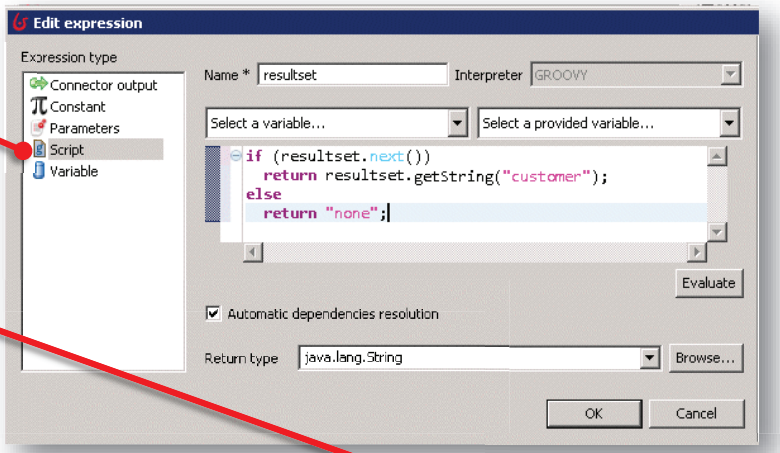


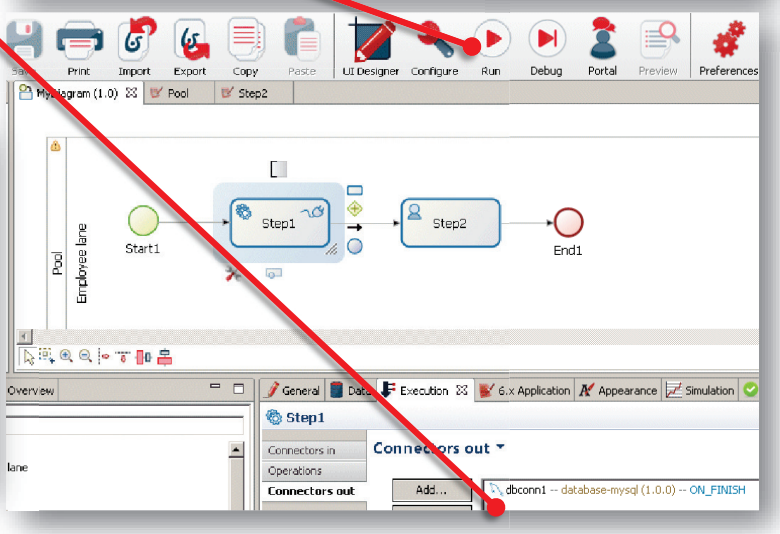
23. Enter the query
24. *SELECT * FROM account WHERE deposit > ${deposit};*
    (for autocompletion of variables press CTRL + SPACE)
23. Select *Next* > *Scripting Mode* > *Next* > Select target: *customer*
24. Click on the pencil icon to open the Groovy editor.

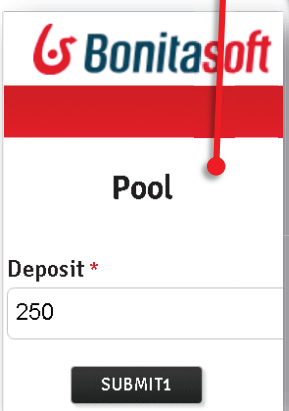
28. Expression type: *Script*
29. In the text area enter

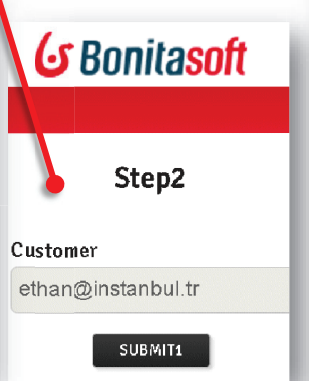
```
if (resultset.next())
 return resultset.getString("customer");
else
 return "none";
```

28. Click on OK > Finish.
29. *Click on Start button in the coolbar*
30. The Bonita launches the browser
31. Enter a deposit and SUBMIT
32. At Step 2, a customer with more than the deposit will be shown



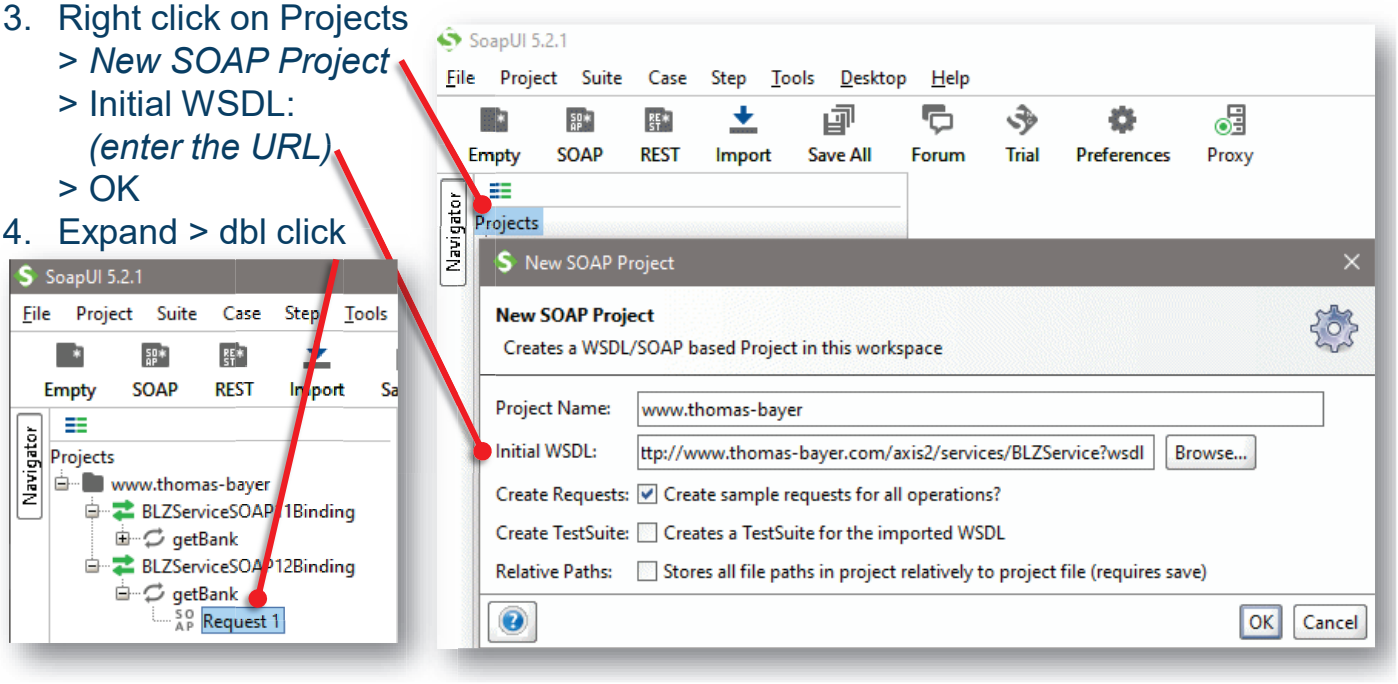




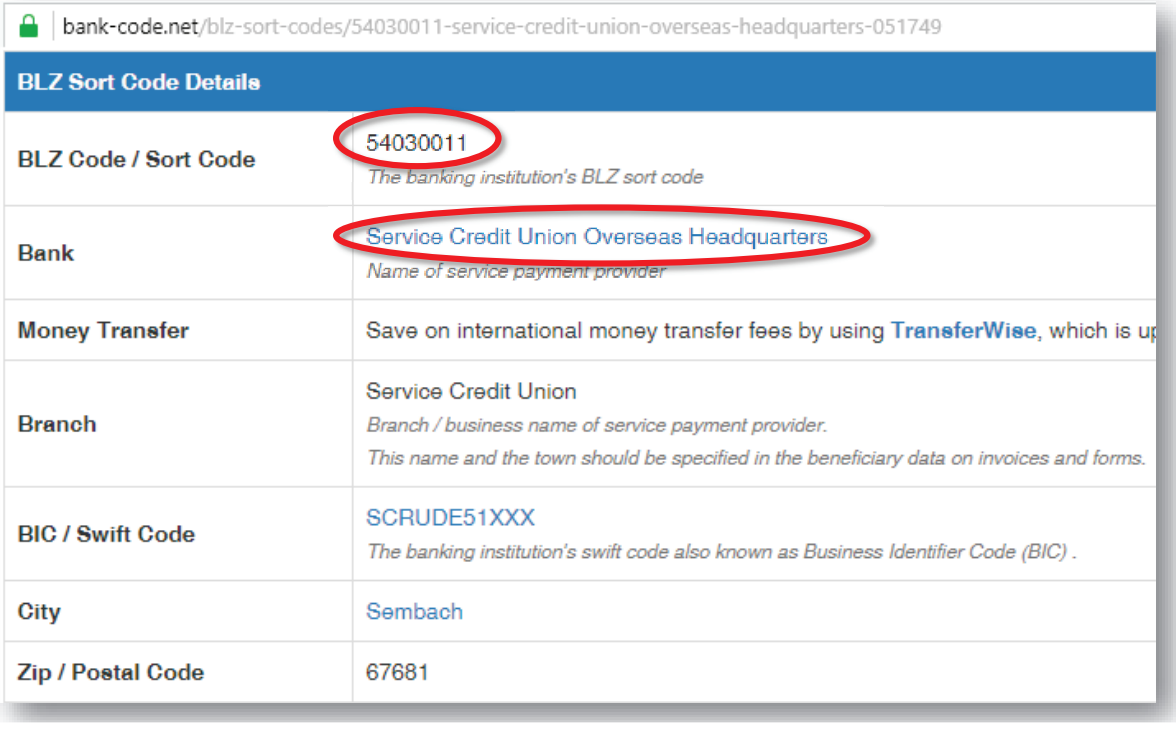


Example of Web service:
http://www.thomas-bayer.com/axis2/services/BLZService?wsdl

1. Install the SOAP UI tool:
2. WIN64: http://www.iet.unipi.it/m.cimino/sse/res/SoapUI-x64-5.2.1.exe
   WIN32: http://www.iet.unipi.it/m.cimino/sse/res/SoapUI-x32-5.2.1.exe
   MACOS: http://www.iet.unipi.it/m.cimino/sse/res/SoapUI-5.2.1.dmg
   LINUX: http://www.iet.unipi.it/m.cimino/sse/res/SoapUI-x64-5.2.1.sh
3. Right click on Projects
   > *New SOAP Project*
   > Initial WSDL:
   (enter the URL)
   > OK
4. Expand > dbl click



- The service takes the BLZ bank code (used in Germany/Austria, ABI+CAB in Italy, incorporated into the IBAN as part of SEPA standardization) as an input
- Example:
  54030011 the BLZ of the Bank *Service Credit Union Overseas Headquarters*
  https://bank-code.net/blz-sort-codes/54030011-service-credit-union-overseas-headquarters-051749



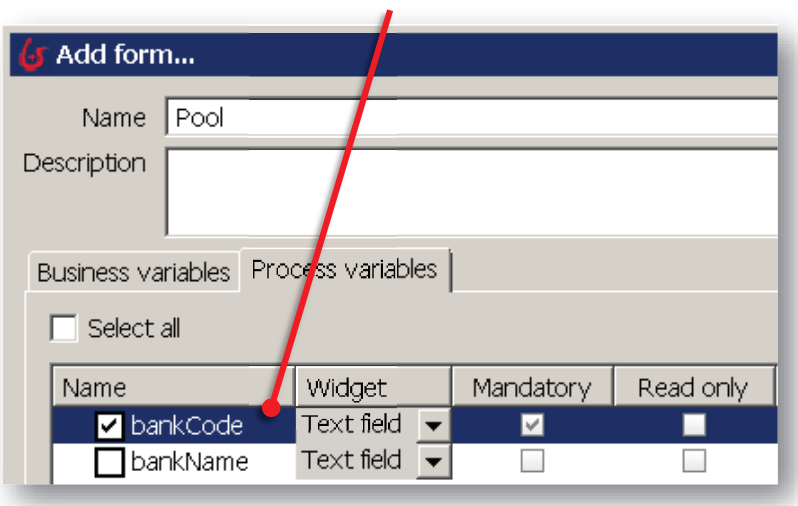
Enter the code and click the play icon (▶)

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
               xmlns:blz="http://thomas-bayer.com/blz/">
   <soap:Header/>
   <soap:Body>
      <blz:getBank>
         <blz:blz>54030011</blz:blz>
      </blz:getBank>
   </soap:Body>
</soap:Envelope>
```

The service provides the following details: bank name (ns1:bezeichnung), BIC code (ns1:bic), place (ns1:ort), and postal code (ns1:plz)

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
   <soapenv:Body>
      <ns1:getBankResponse xmlns:ns1="http://thomas-bayer.com/blz/">
         <ns1:details>
            <ns1:bezeichnung>Service Credit Union Overseas Headquarters</ns1:bezeichnung>
            <ns1:bic>SCRUDE51XXX</ns1:bic>
            <ns1:ort>Kaiserslautern</ns1:ort>
            <ns1:plz>67661</ns1:plz>
         </ns1:details>
      </ns1:getBankResponse>
   </soapenv:Body>
</soapenv:Envelope>
```


1. **Remove the DB connector**
2. Select *Step1* > Tab *Execution* > Connectors out > *Remove*
3. **Remove the Process Variables**
4. Select *Pool* > Tab *Data* > Process variables > select *customer* > Remove > OK, select *deposit* > Remove > OK.
5. **Add the process variables *bankCode (blz), bankName (bezeichnung)***
6. *Add* > Name: *bankCode* > *Finish&Add* > Name: *bankName* > *Finish*
7. **Update the Pool form**
8. Select *Pool* > *6.x Application* > Pageflow > Select *Pool* > *Remove. Add* > *Process variables* > Select *bankCode*. Press *Finish.*



9. **Update the Step2 form**
10. *Select Step2 > 6.x Application > Pageflow > Select Step2 > Remove. Add > Process variables* > Select *bankName* > *Finish*
11. **Add the WS connector**
12. Select *Step1 > Execution > Connectors out > Add.*
13. *Categories: SOAP WebService > Web Service Soap1.2 > Choose the NAME > conn2 > Next*

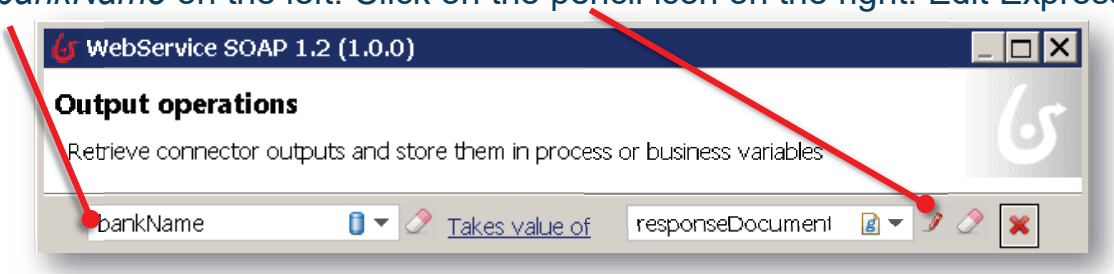
45. Name: *wsconn2* > *Next* > Enter parameters *
    Service NS: *http://thomas-bayer.com/blz/*
    Name: *BLZService*
    Press *Next*
    Port Name: *BLZServiceSOAP12Binding*
    EndPoint: *http://www.thomas-bayer.com/axis2/services/BLZService*
    Binding: *http://www.w3.org/2003/05/soap/bindings/HTTP/*
    Envelope:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
               xmlns:blz="http://thomas-bayer.com/blz/">
   <soap:Header/>
   <soap:Body>
      <blz:getBank>
         <blz:blz>${bankCode}</blz:blz>
      </blz:getBank>
   </soap:Body>
</soap:Envelope>
```

46. Next > Next > Returns body > Next > Output operations:
    (Ctrl + space to find parameters values)

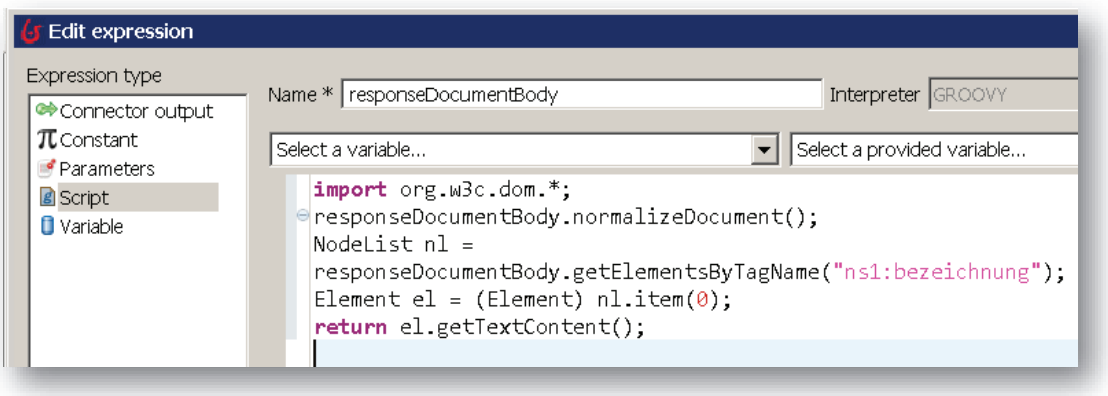
(*) Parameters are extracted by WSDL document
http://www.thomas-bayer.com/axis2/services/BLZService?wsdl
and by using a SOAP client software such as *SoapUI*.

47. Next > Next > Returns body > Next > Output operations:
48. Select *bankName* on the left. Click on the pencil icon on the right. Edit Expression: *Script*.



49. In the text area (Ctrl + space to select parameters values if needed):

```
import org.w3c.dom.*;
responseDocumentBody.normalizeDocument();
NodeList nl = responseDocumentBody.getElementsByTagName("ns1:bezeichnung");
Element el = (Element) nl.item(0);
return el.getTextContent();
```



49. Click on *Start* button in the coolbar
50. The Bonita launches the browser
51. Enter *Bank Code* and SUBMIT
52. At Step 2, the Bank Name is shown
53. Note: The WS may reply with "-1" when the WS is not available (this my occur for free WS)