Business Process addon (par)

Business Process addon (par)

- Introduction
- Task user interface
- ui.xml descriptor

Introduction

Soffid Business Processes use JBoss JBPM engine. You can find the original JBPM documentation here.

In order to develop a new business process addon you need Eclipse Java EE IDE for Web Developers and three plugins installed on it:

- JBPM plugin. You can find it on http://fr.sourceforge.jp/projects/sfnet_jbpm/downloads/jBPM%20Process%20Designer/jbpm-jpdl-designer-3.1.7/jbpm-jpdl-designer-site-3.1.7.zip/ and you can install it from **Help/Install New Software/Add** and select the file you have downloaded with the **Archive** button.
- ZK Studio. Follow this instructions in order to install http://books.zkoss.org/wiki/ZK_Studio_Essentials/Installation#Install_ZK_Studio.
- maven plugin (m2e-Maven Integration for Eclipse). Go to Help/Install New Software/Work with: "All Available Sites" and select the pluguin.

The business process addon can be build using the com.soffid.tools:maven-par-plugin addon.

They follow the JBPM conventions for .par files:

- **/processdefinition.xml** contains the business process definition that you define.
- **/processimage.jpg** contains a bitmap representation of the definition you have created.
- **/gpd.xml** contains the position of business process elements on processimage.jpb bitmap.
- **/classes directory** contains the java compiled classes that support the business process that you implement.
- **/ui.xml** file describes some aspects not covered by processdefinition.xml, just like permissions, user interface and process versioning and upgrade.
- **/ui directory** contains the user interface (ZK) components that are needed to render the workflow tasks.

Examples

/processdefinition.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<process-definition xmlns="urn:jbpm.org:jpdl-3.3"
   name="Recertification process">
```

```
    <description>
        Process to execute recertification tasks.
    </description>

    <start-state name="Start">
        <description>
            Starts a recertification process
        </description>
        <transition to="Select groups"></transition>
        <event type="before-signal">
            <action name="Create recertification process"
class="com.soffid.iam.recertification.bpm.CreateRecertificationProcessHandler"></action>
        </event>
    </start-state>

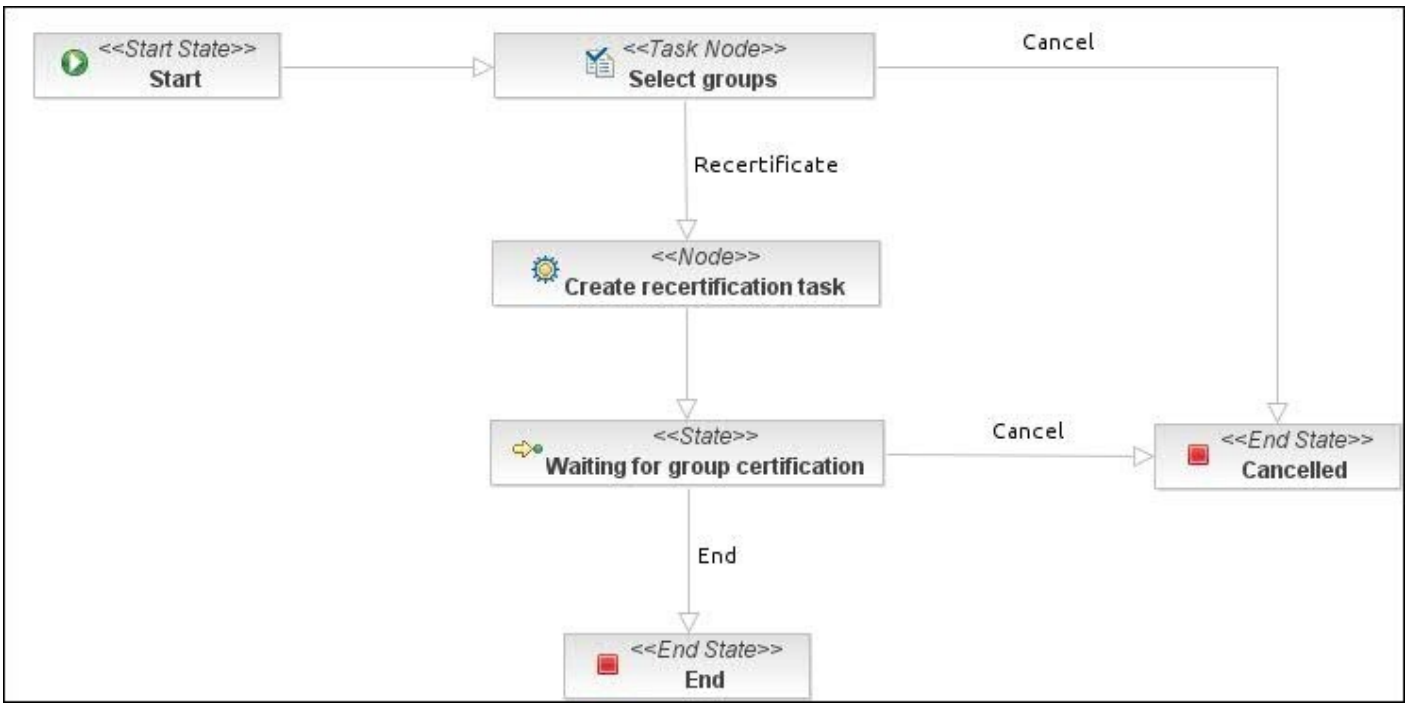
    <task-node name="Select groups">
        <description>
            Selects one or more business units.
            ...
        </description>
        <task name="Select business units">
            <assignment pooled-actors="SEU_ADMIN, SEU_ADMINISTRADOR, SOFFID_ADMIN"></assignment>
        </task>
        <transition to="Cancelled" name="Cancel"></transition>
        <transition to="Create recertification task" name="Recertificate"></transition>
    </task-node>
    <node name="Create recertification task">
        <action name="Create recertification info"
class="com.soffid.iam.recertification.bpm.CreateRecertificationTaskHandler"></action>
        <transition to="Waiting for group certification"></transition>
    </node>
    <state name="Waiting for group certification" async="true">
        <transition to="Cancelled" name="Cancel"></transition>
        <transition to="End" name="End"></transition>
    </state>

    <end-state name="Cancelled"></end-state>
    <end-state name="End"></end-state>

    <event type="process-end">
```

```
        <action class="com.soffid.iam.recertification.bpm.ProcessFinishedHandler"></action>
    </event>
</process-definition>
```

/processimage.jpg



/gpd.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<root-container name="Recertification process" width="743" height="369">
  <node name="Start" x="16" y="14" width="132" height="36">
    <edge>
      <label x="5" y="-10"/>
    </edge>
  </node>
  <node name="Select groups" x="260" y="14" width="203" height="36">
    <edge>
      <label x="-135" y="-21"/>
      <bendpoint w1="314" h1="0" w2="-1" h2="-207"/>
    </edge>
    <edge>
      <label x="5" y="-10"/>
    </edge>
  </node>
  <node name="Create recertification task" x="259" y="124" width="207" height="36">
```

```
    <edge>
      <label x="5" y="-10"/>
    </edge>
  </node>
  <node name="Waiting for group certification" x="258" y="219" width="210" height="36">
    <edge>
      <label x="-14" y="-20"/>
    </edge>
    <edge>
      <label x="5" y="-10"/>
    </edge>
  </node>
  <node name="Cancelled" x="610" y="221" width="132" height="36"/>
  <node name="End" x="297" y="332" width="132" height="36"/>
</root-container>
```

/classes directory

Some additional files must be included.

/ui.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<process>
    <tag>${pom.version}</tag>
    <roles>
        <initiator>
            <role name="SEU_ADMIN" />
            <role name="SOFFID_ADMIN" />
        </initiator>
        <supervisor>
            <role name="SEU_ADMIN" />
            <role name="SOFFID_ADMIN" />
        </supervisor>
        <observer>
            <role name="SEU_ADMIN" />
            <role name="SOFFID_ADMIN" />
        </observer>
    </roles>
    <task name="Select business units">
        <file path="ui/recertificateBUform.zul" />
```

```
    </task>

    <upgrade>
        <process>
            <tag>${project.version}</tag>
        </process>
    </upgrade>
</process>
```

Task user interface

Introduction

To define the user interface for a task, a .zul file must be located at the ui directory, and the corresponding task tag should be located at ui.xml file.

The user interface must be a ZK .zul page. This zul page must contain a task tag. This custom ZK tag is needed to properly manage the user interface behaviour.

Attributes

The task component extends es.caib.bpm.ui.WorkflowWindow and it has the following custom attributes:

Attribute	Description
canAddAttachments	a true value enables the user to upload files attached to the process instance
canDeleteAttachments	a true value enables the user to delete attached files from the process instance
task	read only attribute that exposes current task information
processInstance	read only attribute that exposes current process information
signatureHandler	read only handler responsible for recognized signature generation.
engine	read only BPM engine handler

Events

The task component will receive some custom events:

Event	Description

onLoad	The task information has been loaded on the task component
onSave	The task information is going to be persisted
onPrepareTransition	A transition is going to be performed. The transition name is sent as event data If the event launches a UserWorkflowException, the transition will be cancelled
onCompleteTransition	A transition is performed. The transition name is sent as event data
onTabSelected	The task tab has been selected or deselected. The tab id is sent as event data

In order to show or modify task attributes, the task tag is included in a data model context that eases components data binding.

> See ZK Data Binding for details.

Example

Here is a sample zul page used to authorize synchronization servers connection.

```
<zk xmlns="http://www.zkoss.org/2005/zul"
    xmlns:h="http://www.w3.org/1999/xhtml"
    xmlns:zk="http://www.zkoss.org/2005/zk"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.zkoss.org/2005/zul http://www.zkoss.org/2005/zul/zul.xsd">
    <task id="window" width="600px">
        <attribute name="onPrepareTransition">
            item = lb.selectedItem;
            if (item == null || item.value.length() == 0) {
                throw new es.caib.bpm.toolkit.exception.UserWorkflowException("Request must be approved or
denied");
            }
        </attribute>
        <grid fixedLayout="true">
            <columns visible="false">
                <column width="180px">Attribute</column>
                <column>Value</column>
            </columns>
```

```
<rows>
   <row>
      <label value="User" />
      <hbox width="100%">
         <!-- Display task's attribute user -->
         <textbox width="98%"
            readonly="true" bind="/user" />
      </hbox>
   </row>
   <row>
      <label value="Name" />
      <hbox width="100%">
         <!-- Display task's attribute name -->
         <textbox width="98%"
            readonly="true" bind="name" />
      </hbox>
   </row>
   <row>
      <label value="Surname" />
      <hbox width="100%">
         <!-- Display task's attribute surname -->
         <textbox width="98%"
            readonly="true" bind="/surname" />
      </hbox>
   </row>
   <row>
      <label value="Server name" />
      <hbox width="100%">
         <!-- Display task's attribute hostname -->
         <textbox width="98%"
            bind="/hostname" readonly="true" />
      </hbox>
   </row> <row>
      <label value="Approve" />
      <hbox width="100%">
         <!-- Listbox bound to task's attribute aprove.
            Any change will be stored on the task -->
         <listbox id="lb" width="98%" mold="select" bind="/approve">
            <listitem value="">
               <listcell label="Select an action to do"/>
```

```
                    </listitem>
                    <listitem value="yes">
                        <listcell label="Approve"/>
                    </listitem>
                    <listitem value="no">
                        <listcell label="Reject"/>
                    </listitem>
                </listbox>
            </hbox>
        </row>
    </rows>
</grid>
</task>
</zk>
```

ui.xml descriptor

Introduction

The user interface descriptor follows the attached schema: ui.xsd

Versioning

JBPM denotes business process definition using a sequential number. This approach is not practical for business process management. So, every ui.xml file should have a **tag** tag, identifying the actual version of the process definition.

Permissions

The business process permissions are enclosed in a **roles** tag. This tag must have three inner tags:

Tag	Description
initiator	Contains the users and roles that area authorized to start a new process instance.
supervisor	Contains the users and roles that can act as this process supervisor. A supervisor can take ownership of any task of the process at any type, as well as cancel any active process. Process supervisors are also enabled to upgrade or disable the process definition
observers	Observers can query the process status at any time. Any user that has been assigned a task (despite taken their ownership or not) within a process instance is automatically given the observer permission on this particular process instance.

Process definition upgrade

Soffid allows the process definition developer to decide whether existing but not yet finished process instances should be upgraded to the new process definition when it is uploaded on Soffid console.

To do this, the ui.xml should have an **upgrade** tag. Within there can be one or more **process** tags. Each process must have a **tag** tag that identifies previous business definition versions that should be upgraded to this one. The upgrade will be performed as long as the current node and current task of the process instance token exists on this new version.

If no node and tasks with such name exists on the new process definition, the process instance will remain on its elder version.

To circumvent this limitation, specially when nodes or tasks have been renamed, the developer can specify the target node or task for some current tasks. This mission is accomplished using one or more **task** tags. Each task tag should have a **source** and a **target** attribute. So, when the process upgrade is performed, the first step will be to search for a task tag with a source attribute matching the current task. If it exists, the process will be upgraded and the new task will be that one with the target attribute name.

User Interface resources

Soffid will try to guess the user interface page based on the task name. If no user interface exists with the same name as the task name, a ui/default.zul will be used. Anyway, it's a good practice to specifiy which user interface page will be used for each business process task. This is accomplished by using the **task** tag. Following the **task** tag, a **file** tag with a path attribute will tell Soffid which resouce to use with such a task.

Example

Sample ui.xml file

```
<?xml version="1.0" encoding="UTF-8"?>
<process>
   <!-- Current work in progress version -->
   <tag>1.0.3-SNAPSHOT</tag>
   <roles>
      <!-- Everyone can start this process -->
      <initiator>
         <user name="*"/>
      </initiator>
```

```
        <!-- Only SOFFID_ADMIN can manage this process -->
        <supervisor>
            <role name="SOFFID_ADMIN"/>
        </supervisor>
        <!-- HR group members will be allowed to quey processes status -->
        <observer>
            <role name="HR"/>
        </observer>
    </roles>
    <!-- This specify the resource for New sync server request -->
    <task name="New sync server request">
        <file path="ui/request.zul"/>
    </task>
    <upgrade>
        <!-- Upgrade 1.0.0 version processes -->
        <process>
            <tag>1.0.0</tag>
        </process>
        <!-- Do not upgrade 1.0.1 or 1.0.2 version processes -->
        <!-- Upgrade prior 1.0.3-SNAPSHOT versions, renaming bad spelled task -->
        <process>
            <tag>1.0.3-SNAPSHOT</tag>
            <task source="New synk server request" target="New sync server request"/>
        </process>
    </upgrade>
</process>
```