

Flow of Work

User Documentation

1 Installation

1.1 Set up Eclipse and the FoW Plugins

The Distribution of *Flow of Work* comes as a set of plugins for your eclipse distribution.

1. Download a current **Eclipse Juno** distribution: <http://www.eclipse.org/downloads/>
2. Unpack the distribution to a folder on your harddisk
3. Copy the plugin files (*.jar) of the folder <https://github.com/protossoftware/FlowOfWork/releases/<version>/plugins> to your eclipse dropins folder (e.g. eclipse/dropins)
4. Now you can start Eclipse and select a folder for your workspace

1.2 Set up Graphviz

Graphviz is open source graph visualization software and is used to create the diagrams for the process models. FoW codegenerators generate graphs as ascii .dot files which are converted by the layouter dot into .jpg files.

1. Install the current version of graphviz: <http://www.graphviz.org/>
2. In order to be able to run the layouter, add the path to the program dot.exe (e.g. C:\Program Files (x86)\Graphviz 2.28\bin) to your environment variable "Path"

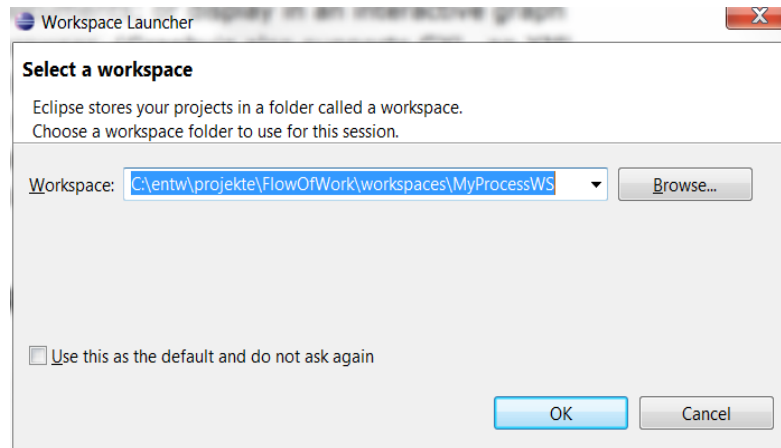
2 Tutorials

2.1 A very simple process model

We start with a minimal process model to introduce the basic features.

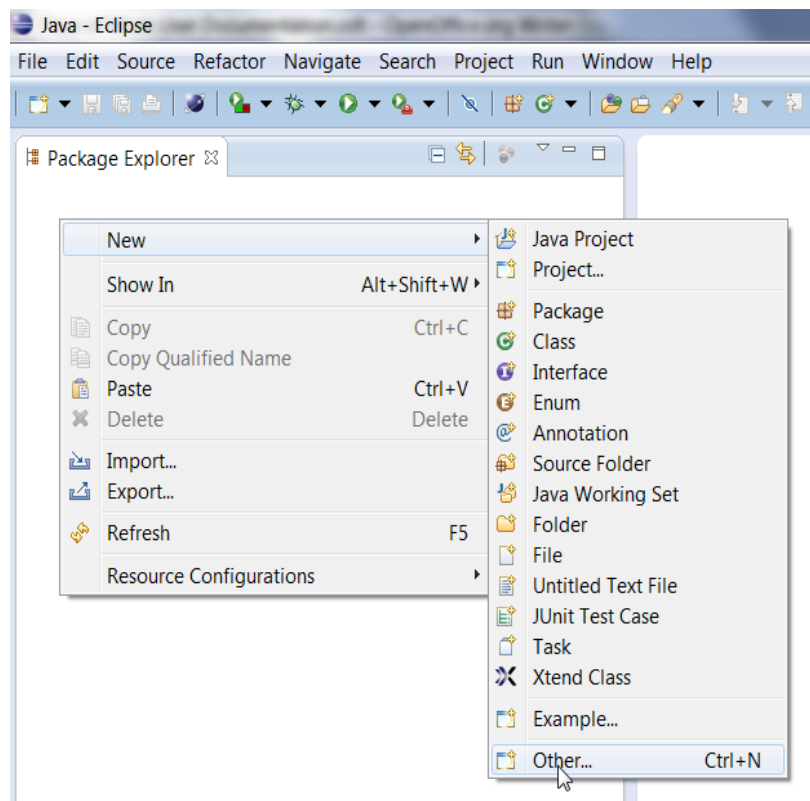
2.1.1 Create an empty project

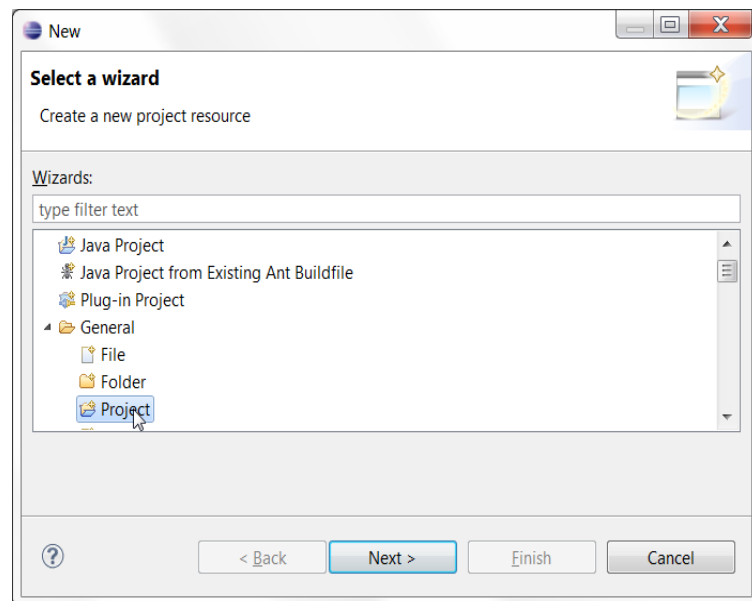
Start Eclipse and select an **empty** folder for your workspace, e.g. MyProcessWS.



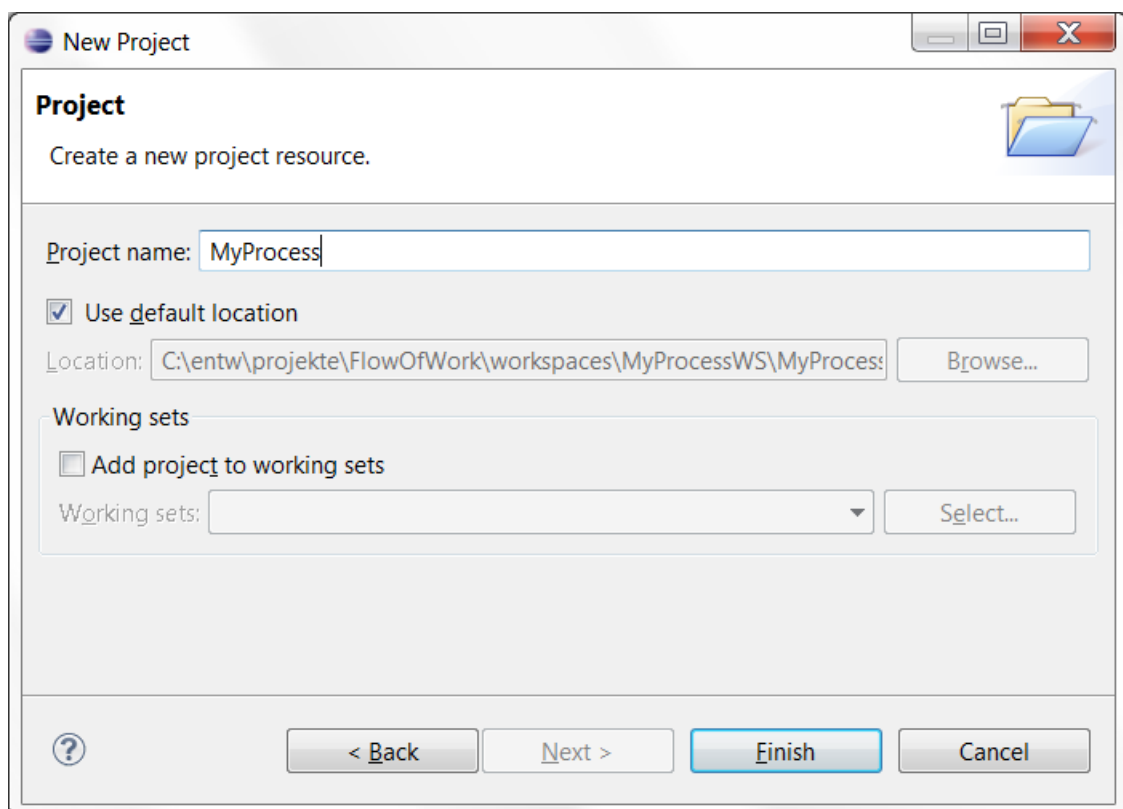
Create an empty project in the Package Explorer.

<right mouse click> -> New-> Other -> General -> Project





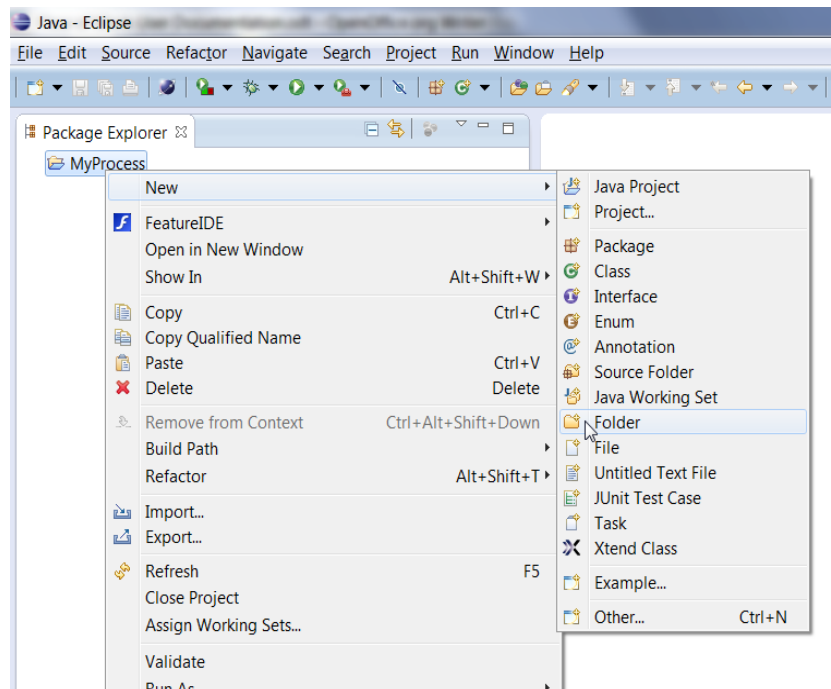
Define a Project Name, e.g. "MyProcess" and click Finish



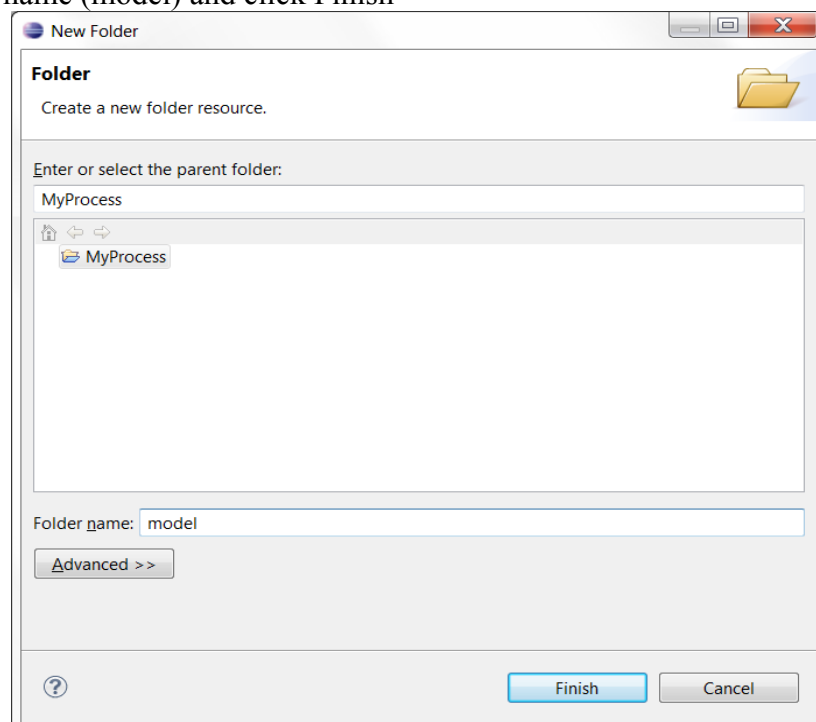
2.1.2 Create a folder and a file for the model

Create a new folder in the new project, named "model":

<right mouse click> -> New-> Folder

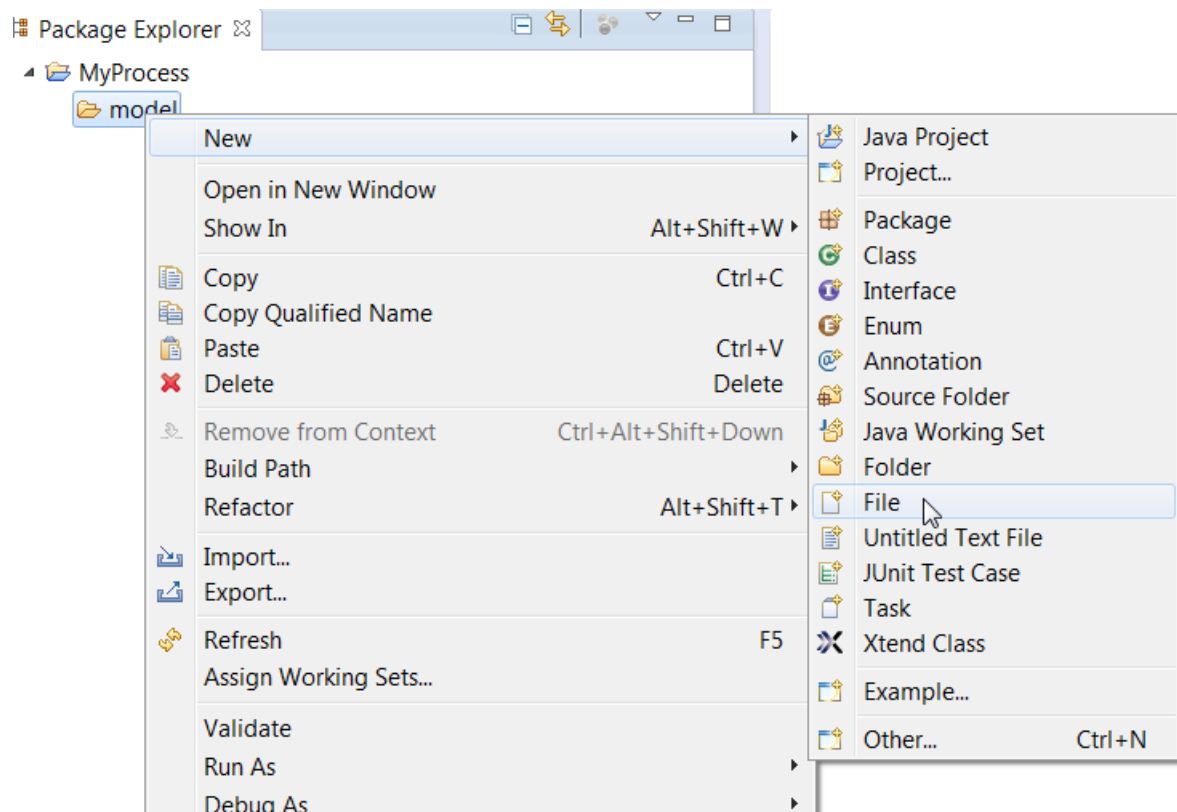


Define the folder name (model) and click Finish



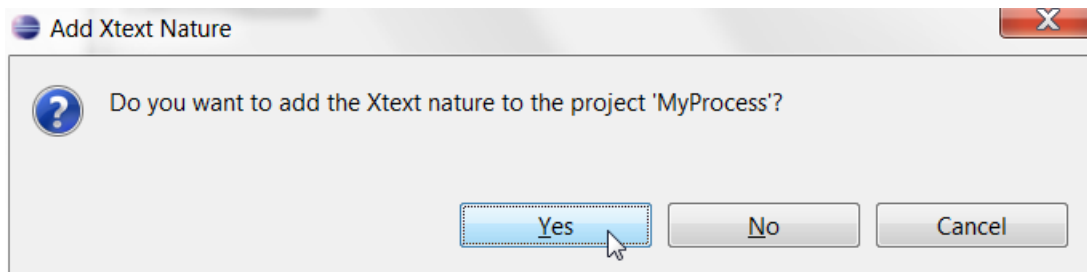
Create a new file in the model folder, named "MyProcess.fow":

<right mouse click> -> New-> File



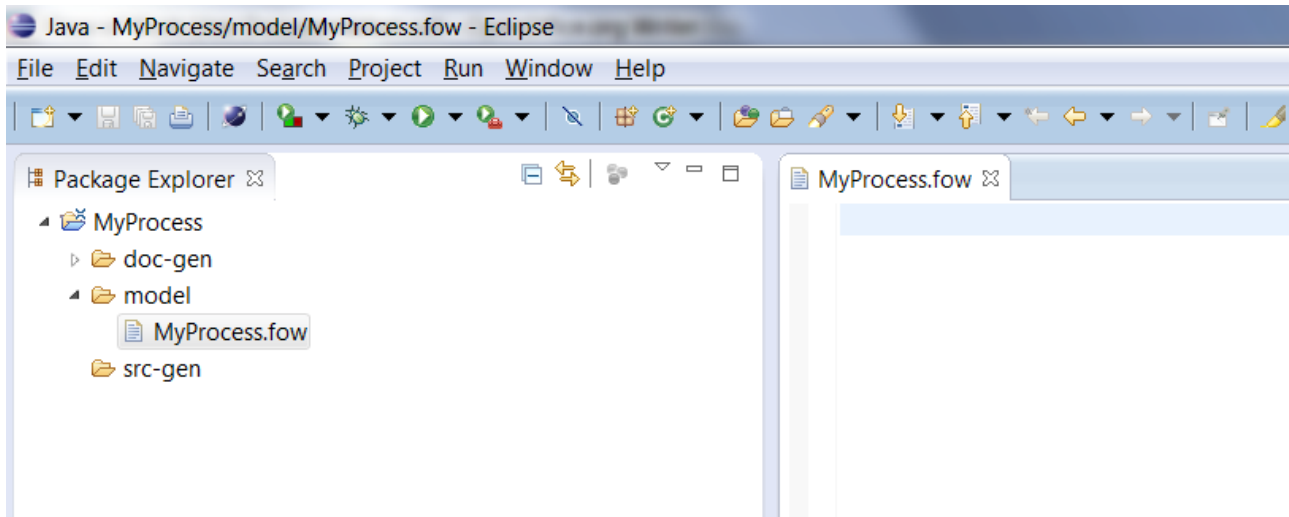
...

Make sure you add the Xtext nature with "Yes"



The empty file should open automatically in an editor, if not, doubleclick the file in the Package Explorer.

Result:

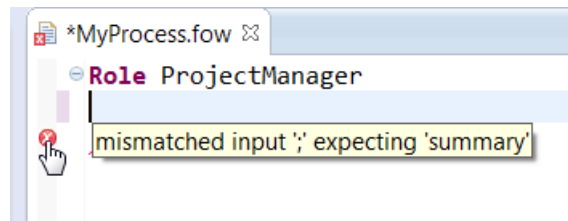


2.1.3 Creating our first Model

Now we create our first minimal process model. We start with the definition of a **Role**.

In the editor we enter the definition for a role. The syntax highlighting helps you doing it right. In the following example, you can see the error text by hovering with the mouse over the error marker.

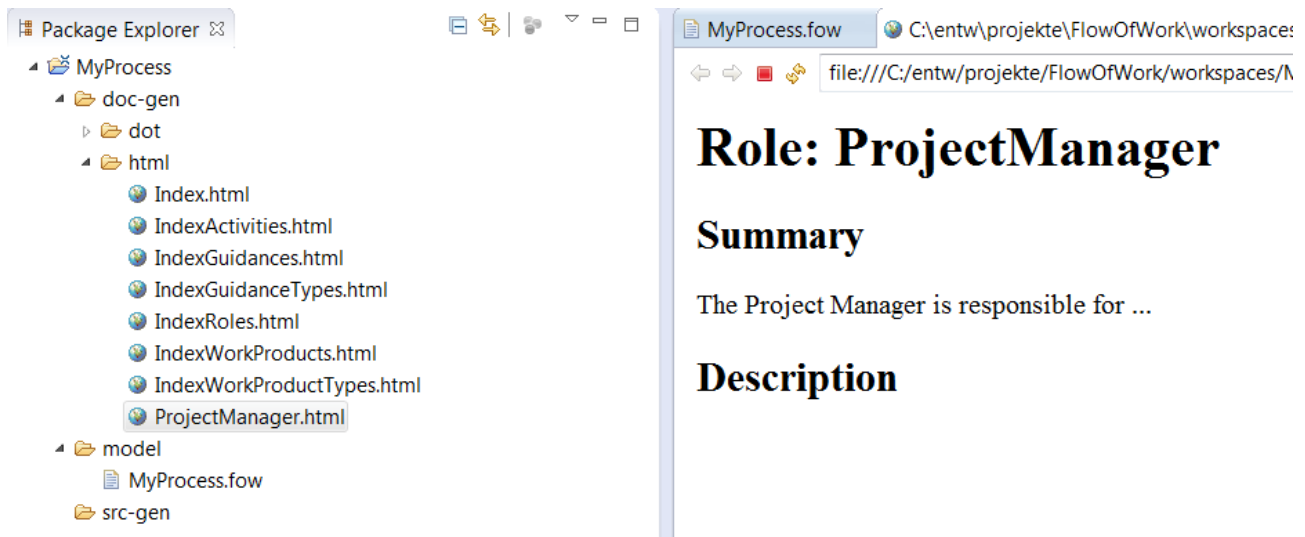
The Role needs at least a summary text.



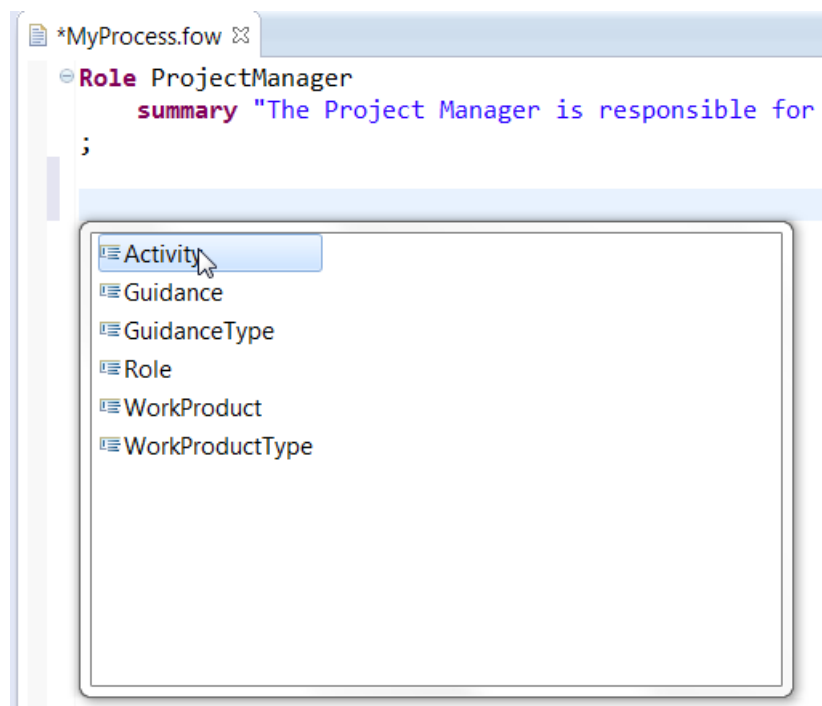
Here the correct model:

```
Role ProjectManager
    summary "The Project Manager is responsible for ..."
;
```

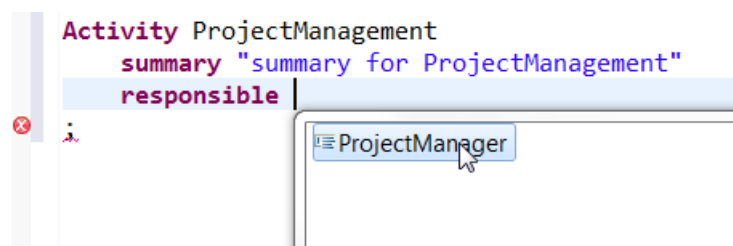
Save the model (shortcut: <ctrl>+s). When ever a model is saved the codegenerator updates the generated documentation, including all folders and index files. Look at the generated html documentation by double clicking doc-gen/html/ProjectManager.html. The file Index.html is the entry point to browse the complete documentation. The other Index... files contain a list of all Elements of a certain Type (e.g. IndexRoles.html contains a list for all Roles with the links to the generated html pages).



Now we want to create an Activity. You can use <ctrl>+<space> to see which model elements can be used at this position of the model. Select the Activity.



Complete the minimal definition of the Activity, using the code completion. After the keyword "responsible" press <ctrl>+<space> and select a role (only one available in our small model).



And here the complete model:

```
Role ProjectManager
    summary "The Project Manager is responsible for ..."
;

Activity ProjectManagement
    summary "summary for ProjectManagement"
    responsible ProjectManager
;
```

Browse the doc-gen/html folder and have a look at the generated Activity file (ProjectManagement.html):

Activity: ProjectManagement

Purpose:	summary for ProjectManagement		
Scope:	Where does the scope come from?		
Inputs:	no inputs	Outputs:	no outputs
Entry Criteria:	really needed???	Exit Criteria:	really needed???
Responsible:	ProjectManager		
Description:			

Activities and Responsibilities

Guidelines

SPEM Diagram

✖ SPEM Diagram: ProjectManagement

As you can see the SPEM Diagram is not yet created. In order to do that, open a **command prompt** (german: **Eingabeaufforderung**) and change the working directory to the dot folder of your project (e.g. C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot). The type make and the dot command creates the jpg files. In further versions, this call will be integrated into the generation process. Result in the command prompt:

```
C:\entw\projekte\FlowOfWork\workspaces>cd C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot
C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot>make
dot -Tjpg ProjectManagement.dot -o ProjectManagement.jpg
C:\entw\projekte\FlowOfWork\workspaces\MyProcessWS\MyProcess\doc-gen\dot>
```

Mark the project name in the Project Explorer and press <F5> to update your project with the new file and click into your html viewer for ProjectManagement.html and also update with <F5>.

Result in the eclipse project:

Activity: ProjectManagement

Purpose:	summary for ProjectManagement		
Scope:	Where does the scope come from?		
Inputs:	no inputs	Outputs:	no outputs
Entry Criteria:	really needed???	Exit Criteria:	really needed???
Responsible:	ProjectManager		
Description:			

Activities and Responsibilities

Guidelines

SPEM Diagram

ProjectManagement

3 Keyboard Shortcuts

3.1 *Shortcuts for textual FoW editor*

- <Shift> + <Ctrl> + F: Format File
- <Ctrl> + <SPACE>: Code Completion
- <F3>: Open Declaration (jump to Declaration)
- <Shift> + <Alt> + R: Rename Refactoring
- <Strg> + D: Remove Line