



# OCOPOMO

## Open Collaboration in Policy Modelling

### D4.2 SYSTEM AND USER DOCUMENTATION

#### D: USER MANUAL ON TOOLS FOR SIMULATION ANALYSIS AND OUTPUT SCENARIO GENERATION

---

Document Full Name	<b>OCOPOMO_D4.2-D_SimulationAT-UserManual.doc</b>
Date	<b>12/04/2013</b>
Work Package	<b>WP4: Integration of components</b>
Lead Partner	<b>Intersoft</b>
Authors	<b>Peter Smatana, Karol Furdik</b>
Document status	<b>v1.00 FINAL</b>
Dissemination level	<b>PUBLIC (PU)</b>

---



**Document Log**

<b>Version</b>	<b>Date</b>	<b>Comment</b>	<b>Author</b>
001	14/01/2013	Documentation describing the installation and use of the first prototype version of the Simulation Analysis Tool	Peter Smatana
002	04/02/2013	Updating for recent changes of prototype	Peter Smatana, Karol Furdik
003	21/02/2013	Corrections and updates	Peter Smatana
004	05/03/2013	Instructions on setting up the Simulation Analysis environment in Eclipse	Peter Smatana, Karol Furdik
005	27/03/2013	Included sections describing the annotations and upload of model-based scenario to Alfresco	Peter Smatana, Karol Furdik
100	12/04/2013	Integration of inputs, updates of all chapters. Consolidation, corrections and finalisation	Karol Furdik



## **TABLE OF CONTENTS**

<b>1. INTRODUCTORY NOTES .....</b>	<b>4</b>
<b>2. OVERALL DESCRIPTION OF THE TOOL .....</b>	<b>6</b>
2.1. USER ROLES .....	6
2.2. USER INTERFACE AND STRUCTURE OF THE SIMULATION ANALYSIS TOOLKIT .....	7
2.3. SETTING UP THE ECLIPSE WORKSPACE FOR SIMULATION ANALYSIS .....	8
<b>3. USE OF THE SIMULATION ANALYSIS TOOL .....</b>	<b>13</b>
3.1. INITIAL NARRATIVE SCENARIO SET UP .....	13
3.2. ANNOTATIONS .....	15
3.3. UPLOAD OF THE RESULTING SCENARIO TO ALFRESCO .....	20
<b>4. CONCLUSION.....</b>	<b>22</b>
<b>REFERENCES.....</b>	<b>23</b>



## 1. INTRODUCTORY NOTES

This user manual describes the usage of *Simulation Analysis* tools and related components of the integrated OCOPOMO toolkit, which enable a construction of output model-based scenarios on the basis of results of simulations executed upon the underlying agent-based policy model. The suite of Simulation Analysis tools is implemented as a set of Eclipse plug-ins and is proposed for end users, policy analysis and modelling experts, to operate in a local mode of stand-alone Eclipse environment. However, the resulting model-based scenario, after it is properly formulated and annotated in the Eclipse environment, needs to be uploaded to the shared web space of Alfresco CMS. It implies that the working environment of Simulation Analysis tools, which belongs to the Eclipse part of the OCOPOMO toolkit, should be installed and maintained in accordance with comprehensive usage guidelines and supporting materials for the underlying Eclipse platform, namely:

- Eclipse Indigo platform, (Eclipse Indigo, 2011), <http://www.eclipse.org/indigo/>;
- Documentation of Eclipse Indigo, version 3.7 (Eclipse Indigo Documentation, 2013), <http://help.eclipse.org/indigo/index.jsp>;
- General Eclipse site, <http://www.eclipse.org>.

Since the suite of Simulation Analysis tools works on outputs of running simulations (i.e., simulation logs), it should be installed in the same Eclipse environment as DRAMS-Repast suite of policy modelling and simulation tools, described in D4.2-C: *User Manual on Policy Modelling and Simulation Tools*, as well as in the main text of the D4.2 deliverable, section 4.4.

To enable an upload of resulting model-based scenarios, produced by here-presented Simulation Analysis tools, to the Alfresco web space requires that an instance of Alfresco CMS is installed and a data transfer connection between the local Eclipse and remote Alfresco environments is established. The Alfresco environment of OCOPOMO toolkit is built on the Alfresco Share platform (Alfresco Share, 2011), which provides a comprehensive usage guidelines and supporting materials, namely

- Alfresco Share wiki, [http://wiki.alfresco.com/wiki/Alfresco\\_Share](http://wiki.alfresco.com/wiki/Alfresco_Share);
- Documentation of Alfresco Share, version 3.5 (Alfresco Share Documentation, 2012), <http://docs.alfresco.com/3.5/topic/com.alfresco.team.doc/topics/uh-welcome.html>;
- General Alfresco site, <http://www.alfresco.com>.

To obtain instructions on the set up of connection between the Alfresco and Eclipse environments, provided by the Content Repository Client component, please refer to D4.2-B *User Manual on CCD Tools*, section 2.5.

Furthermore, this document complements the main text of the D4.2 deliverable by providing how-to instructions for use of tools and user interfaces during the fifth phase of the OCOPOMO process, i.e., the analysis of outputs obtained from experimental simulations of agent-based policy models, construction of respective model-based scenarios and publishing of these scenarios to the Alfresco web space for further evaluation and collaborative processing.

Usage guidelines are complemented by video presentations of key tools and components of the suite of Simulation Analysis tools as follows:

- *SimAnalysis\_Video.zip* - usage instructions for the *Simulation Analysis Tool*, which include the annotation of simulation logs and production of a model-based scenario;
- *MBSenarioPublish\_Video.zip* - usage instructions for the Simulation Analysis Tool (2nd part), focused on publishing of produced scenarios to Alfresco, together with a presentation of Simulation Output Visualisation components (see also in D4.2-E: *User Manual on Simulation Output Visualisation and Traceability Tools*).



Video presentations are also provided online at <http://www.ocopomo.eu/workspace/wp-04-integration-of-components-1/d4.2-system-and-user-documentation/how-to-videos> (only for users registered at the OCOPOMO web site, <http://www.ocopomo.eu>).

Other information related to the installation, maintenance, connection to the whole OCOPOMO platform, and technical details of the here-presented tools can be found in the main text of the D4.2 deliverable.

## 2. OVERALL DESCRIPTION OF THE TOOL

This chapter presents a general information on the *Simulation Analysis* tools and related components of the OCOPOMO toolkit, namely an overview of involved user roles, structure of provided user interface and location of particular tools, as well as a set of instructions for the maintenance and customisation of the tools.

### 2.1. USER ROLES

User roles for the underlying OCOPOMO process, which is supported by the here-presented set of tools, were proposed in the D2.1 deliverable (Mach et al, 2010) as it is presented in Table 1.

**Table 1:** User roles applied in the OCOPOMO system.

Icon	User Role	Icon	User Role
	Politician		<b>Analyst</b>
	Civil servant		<b>Modeller</b>
	Stakeholder		<i>Administrator</i>
	<i>Facilitator</i>		

The suite of *Simulation Analysis* tools supports the fifth phase of the OCOPOMO process, where results obtained from experimental simulations of agent-based policy models are processed and analysed by policy modelling and analysis experts. Respective model-based scenarios are developed as narratives summarising the facts and data dependencies proven by the underlying policy models. Finally, the produced narrative scenarios, as the main outputs of policy analysis and modelling process, are forwarded to the shared web space for evaluation and further discussions among policy analysis and modelling consumers represented by *Politician*, *Civil servant*, and *Stakeholder* roles.

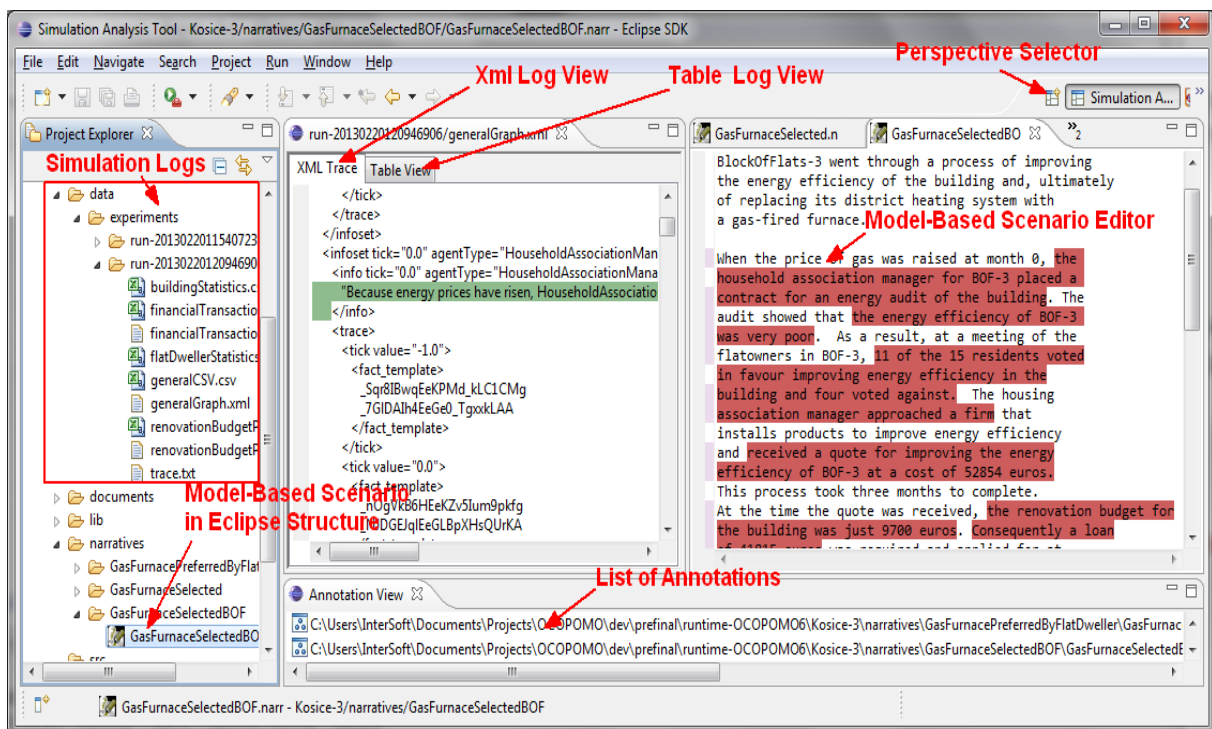
The respective actions of the fifth process phase, which is supported by the *Simulation Analysis* tools, are performed by experts, i.e., by *Analyst*, and *Modeller* user roles. Publishing of output narrative scenarios is accomplished by *Analyst*, with an assistance of *Facilitator* - user role responsible for overall process control and maintenance of the collaborative web space. Technical support for the here-presented tools, which includes a set up and maintenance of the Eclipse environment and its proper connection to the remote Alfresco web space, is handled by *Administrators*. End users - direct participants of the policy development (i.e., *Politician*, *Civil servant*, and *Stakeholder* roles) are not involved in the analysis of simulation results analysis and development of output narrative scenarios<sup>1</sup>; the produced model-based scenarios are provided for them in sixth process phase only.

The icons of user roles, presented in Table 1, are referenced and used as labels in "how to" instructions presented below. This labelling indicates the proposed actors/consumers of described operations.

<sup>1</sup> The distribution of user roles into the respective phases of the OCOPOMO process and organisation of the whole community of involved users could be, however, a more complex task that belongs to methodological issues. As such, it is detailed separately in the D8.1 deliverable (Scherer et al, 2013).

## 2.2. USER INTERFACE AND STRUCTURE OF THE SIMULATION ANALYSIS TOOLKIT

The default user interface of the *Simulation Analysis* tools is presented in Figure 1. Particular tools and data sources are accessible in the Eclipse environment in a structure of several views integrated in the Simulation Analysis perspective, which are included in the overall OCOPOMO Java project. However, since the *Simulation Analysis* tools are built upon the Eclipse Indigo framework (Eclipse Indigo, 2011), the layout of the user interface can be customised on each local installation by adjusting the plug ins and views in accordance with needs of particular users. For guidelines of maintaining the Eclipse development environment and customising the layout please refer to (Eclipse Indigo Documentation, 2013).



**Figure 1:** General user interface of Simulation Analysis Tool in the Eclipse environment.

The user interface of the *Simulation Analysis* toolkit and related data structures, depicted in Figure 1, consists of the following items:

- *Simulation logs*, generated by experimental simulations of executable agent-based policy model(s), are included into the structure of OCOPOMO Java project and are available as run-*<date-number>* sub-folders of the *../data/experiments/* folder;
- *Xml Log View* and *Table Log View* are visualisations of simulation log records provided by the *Simulation XML Log Transformation* component in XML and CSV Table formats;
- *Model-based Scenario files*, included into the OCOPOMO Java project structure in the *../narratives* sub-folder, contain inner XML representation of output model-based scenarios, which includes a text of the scenario narrative as well as all annotations of text fragments to particular records of simulation logs;
- *Model-Based Scenario Editor*, a text editor specifically designed for the development of output model-based scenarios in plain text format, which enables to annotate text portions by simulation log records and preserves the annotations during the text editing procedure.

The editor is built on *MyNote Annotator* plug-in, which is described in more details in D4.2-B *User Manual on CCD Tools*, section 3.5;

- *List of Annotations* view, which displays the list of created annotations - links between particular records of simulation logs and respective text fragments in the narrative scenario. Annotations are enabled by the inner *Annotation mechanism*, which is built upon the GATE annotation file mechanism (GATE, 2011).

From the perspective of system architecture, the *Simulation Analysis* toolkit consists of three main modules covering both inner (hidden) system components and end-user tools, namely:

- ***Simulation XML Log Transformation***, an inner system component that makes the produced simulation logs available for further simulation analysis, transforms the logs to proper formats (i.e., plain text, XML, and CSV Table - however, only the structured formats of XML and CSV table are employed for output scenario annotations) and provides the data for the *Simulation log views*.
- ***Simulation Analysis Tool***, an Eclipse plug-in that includes all the visual end-user tools as follows:
  - *Simulation log views* - visualisations of simulation log records, provided in *Xml Log View* and *Table Log View*, i.e., in the XML Trace and Table View tabs of the simulation logs window;
  - *Model-Based Scenario Editor*, described above in the outline of user interface components;
  - *Annotation mechanism*, a set of inner system components enabling creation and inner representation of the annotations - links between simulation log records and text portions of the output scenario. The annotations are visualised in the *List of Annotations* view, as well as in both *Simulation log views* and *Model-Based Scenario Editor* as highlighted log records and text fragments, respectively.
- ***Publishers***, inner components enabling an upload of the resulting model-based narrative scenario and all related data structures (i.e., the CCD model, simulation logs, annotations, and local files/documents used by experts during the policy modelling process). Publishers are provided as inner components of the *Content Repository Client* middleware, which is detailed in D4.2-B *User Manual on CCD Tools*, section 2.5.

In the following sections, usage instructions for the *Simulation Analysis* toolkit will be provided as follows. Global settings required for inner system components will be presented next in section 2.3. The focus of Chapter 3 will be given to usage instructions for visual end user interfaces, which are integrated in *Simulation Analysis Tool* plug-in.

### **2.3. SETTING UP THE ECLIPSE WORKSPACE FOR SIMULATION ANALYSIS**

This section contains some instructions on the set up of global Eclipse environment and its connection with the remote Alfresco web space. It is assumed that the Eclipse Indigo platform (Eclipse Indigo, 2011) is already installed together with all the OCOPOMO components and extensions according to the instructions provided in the main text of D4.2, section 3.2. In addition, the simulation analysis requires that previous steps of the OCOPOMO process, namely the conceptual modelling, development of agent-based policy model, and running simulation experiments on this model were accomplished successfully, in accordance with usage instructions provided in particular user manuals D4.2-B and D4.2-C. After all these procedures, the CCD model and simulation logs generated as outputs of executed simulations should be available in the OCOPOMO Java project in Eclipse. Based on these prerequisites, the simulation analysis may start.

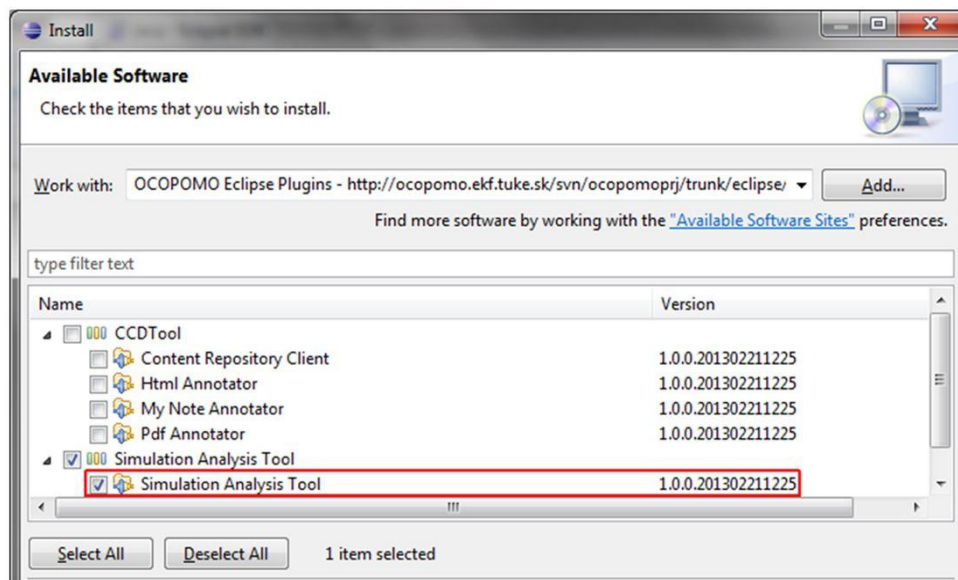


## How to check whether the Eclipse environment is ready for the simulation analysis procedure?



The simulation analysis can be performed within the OCOPOMO Java project, upon available simulation results - logs produced by running agent-based models. To check an availability of all the necessary data resources for the simulation analysis, please follow these instructions:

1. Ensure that the *Simulation Analysis Tool* package is properly installed in your Eclipse. During the installation procedure invoked by the *Help* → *Install New Software*, set *Location* to <http://ocopomo.ekf.tuke.sk/svn/ocopomoprj/trunk/eclipse/updatesite/site.xml> and in the displayed dialog check the *Simulation Analysis Tool* (see in Figure 2).



**Figure 2:** An installation dialog for the Simulation Analysis Tool.



***Install OCOPOMO\_Video.zip*** - see the video demonstrating the set up of the Eclipse for OCOPOMO. Instructions on the OCOPOMO plug-in installation to the Eclipse Indigo environment.

2. Check if the OCOPOMO Java project in your Eclipse installation contains all required data resources, as it is depicted in Figure 3. The resources namely include (1) the OCOPOMO Java project itself, properly configured for all the Eclipse OCOPOMO tools, (2) source files of developed CCD model, (3) simulation logs available in `../data` folder. Optionally, local documents/files used as background documents during the CCD model creation could be available in the `../documents` folder.
3. If some of required resources is missing, please return to previous steps of the OCOPOMO process, in accordance with usage instructions provided in D4.2-B: *User Manual on CCD Tools* and/or D4.2-C: *User Manual on Policy Modelling and Simulation Tools*.
4. If all required resources are available in your OCOPOMO Java project, then your Eclipse environment is ready for the procedure of simulation analysis and development of output model-based scenario.

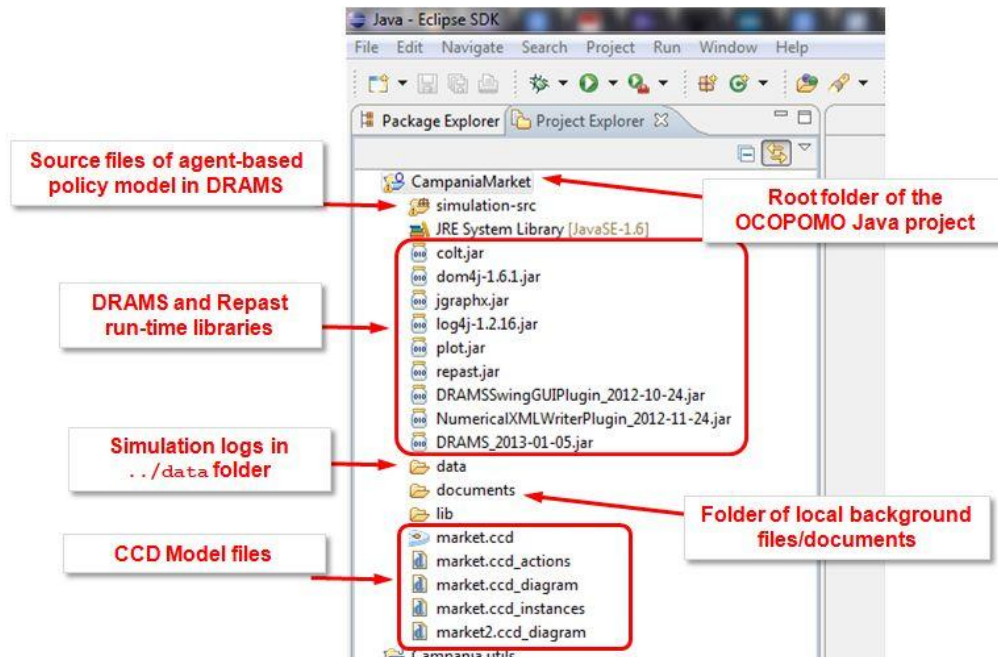


Figure 3: Structure of folders in the OCOPOMO Java project before the simulation analysis.

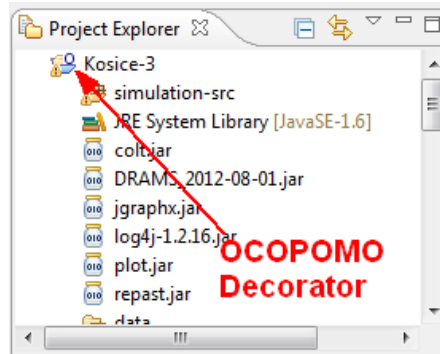
## How to connect the OCOPOMO project in Eclipse with remote Alfresco?



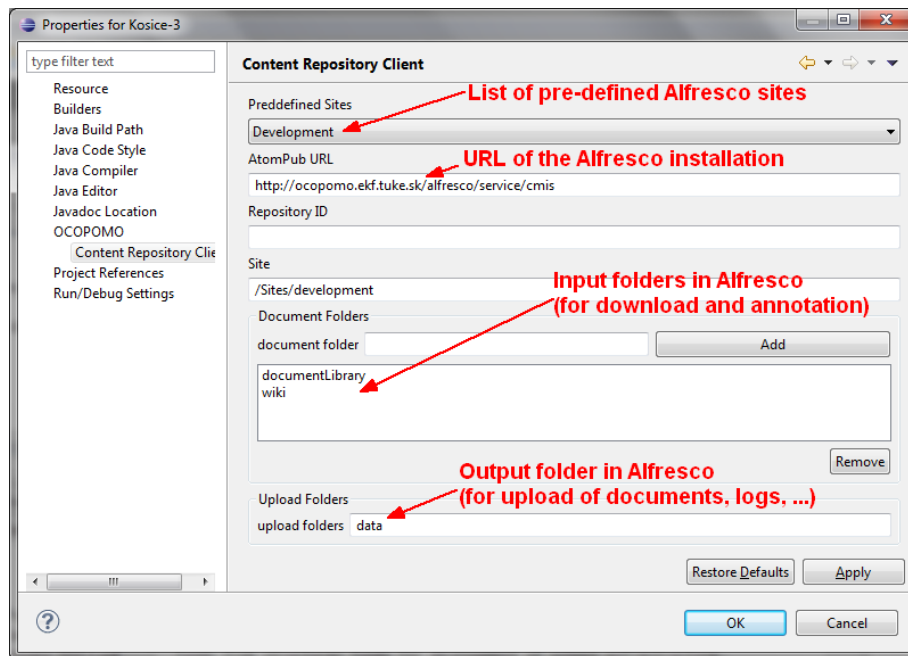
To enable an upload of resulting narrative scenario to the shared web space (and thus to provide stakeholders and policy makers with the produced scenario), a connection between the local Eclipse and remote Alfresco environments needs to be established and properly configured. It can be achieved by means of the *Content Repository Client* middleware, which supports the data transfer between Alfresco and Eclipse in both directions; however, for the publishing of output scenarios the data flow direction from Eclipse to Alfresco is applied.

Instructions for set up and configuration of the *Content Repository Client* are available in D4.2-B: *User Manual on CCD Tools*, since the Alfresco-Eclipse data transfer was required already for the conceptual policy modelling in the third phase of the OCOPOMO process. However, it is recommended to check (and update, if needed) the remote connection settings also in the phase of simulation analysis. To check the settings of the remote connection, please follow these steps:

1. Check if your OCOPOMO Java project in Eclipse is labelled by a decorator mark (see in Figure 4), which indicates that the project is connected to a remote Alfresco. If there is no decorator mark on the label of your Java project, then you should follow the instructions of setting up the *Content Repository Client*, presented in D4.2-B, section 2.5.
2. Even if there is a decorator mark on the label of your Java project, you should check the remote connection settings. To do so, please navigate to the properties of your Java project (by clicking with right mouse button on the project root in the *Project Explorer* view, or use the main menu command *File* → *Properties* when your project is selected in *Project Explorer* view).
3. Click on *OCOPOMO* → *Content Repository Client* to invoke the dialog presented in Figure 5. Check the connection settings and update them, if needed. Click *Apply* and *OK* buttons.



**Figure 4:** The OCOPOMO Java project connected with a remote Alfresco repository.

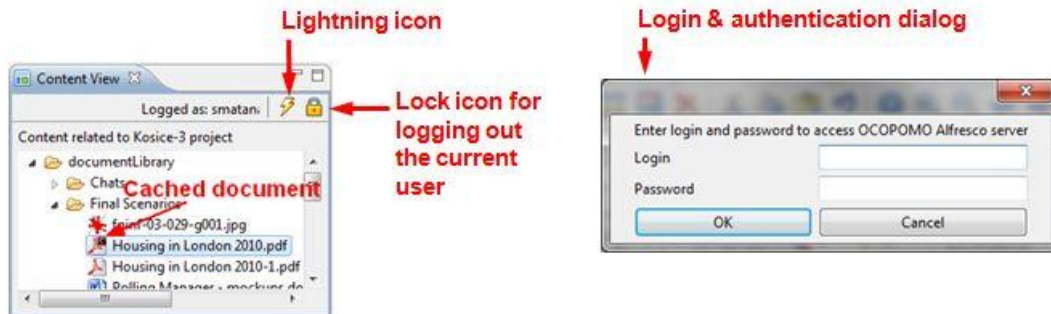


**Figure 5:** Content Repository Client properties for enabling a connection to Alfresco.


After the connection settings are checked and/or updated, it would be good to check if the data transfer works properly. To do so, please continue with the following these steps:

4. The Alfresco content is displayed in Eclipse in the *Content View* perspective (see in Figure 6). To open the *Content View*, please use the main menu *Window* → *Show View ...* → *Other ...* and select *Content View* under the OCOPOMO category.
5. If you have previously configured the connection, a structure of folders and files should be already available in the *Content View*.
6. To reload the locally stored content, click on the lightning icon ⚡ in the *Content View* perspective (see in Figure 6). The access to the Alfresco repository can be restricted, so you may be asked to authenticate yourself by entering the login and password data to the dialog depicted on the right hand side of Figure 6.

**Note:** Please be patient - the connection to the remote Alfresco site and the data download may take a while (typically it lasts 10-20 seconds, depending on the connection speed and amount of data to be transferred).



**Figure 6:** The OCOPOMO Java project connected with a remote Alfresco repository.

7. To log out from the Alfresco repository, please click on the lock icon .

After a successful reload of documents from the remote Alfresco repository, the check of data transfer connection is over and your Eclipse environment is ready for the simulation analysis and development of output model-based scenarios.

### 3. USE OF THE SIMULATION ANALYSIS TOOL

This chapter contains descriptions and instructions on how to perform the most common operations in the *Simulation Analysis Tool* towards the analysis of simulation results and development of output model-based scenario.

#### 3.1. INITIAL NARRATIVE SCENARIO SET UP

##### How to start with the model-based scenario development?



The model-based scenario is stored in a local `*.narr` file, which is included to the existing OCOPOMO Java project. The development of output model-based scenarios thus starts with creation of such a `*.narr` file, together with an initialisation of the Simulation Analysis perspective in Eclipse. To perform these operations, please follow the steps:

1. In the *Project Explorer* view, highlight the root of your OCOPOMO Java project folder e.g. by clicking on it with left mouse button.

**Note:** Your OCOPOMO Java project should be already connected to remote Alfresco web space and should be labelled by the decorator mark (see above in Figure 4). Moreover, all required data resources such as CCD model files and results of executed simulations should be already available in your OCOPOMO Java project - see the instructions for setting up the Eclipse workspace, which are presented in section 2.3 above.

2. From the main Eclipse menu, use the *File* → *New* → *Other ...* command. In the appearing wizard dialog, select the *Narrative Model-Based Scenario* option, which is located under the OCOPOMO group (see in Figure 7). Click on the *Next* button.
3. A simple dialog for configuration of output scenario will appear as it is depicted in the right hand side of Figure 7. Type the name of your scenario into the *Narrative name* field. Optionally, you can change the OCOPOMO Java project where the scenario will belong to, using the *Select Related Project* combo box. Click on the *Finish* button.

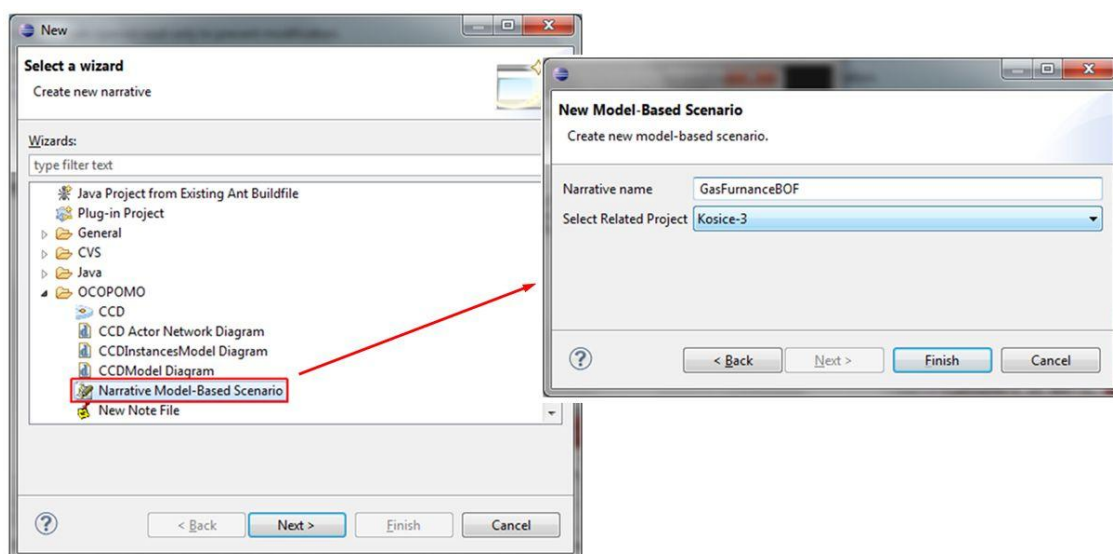
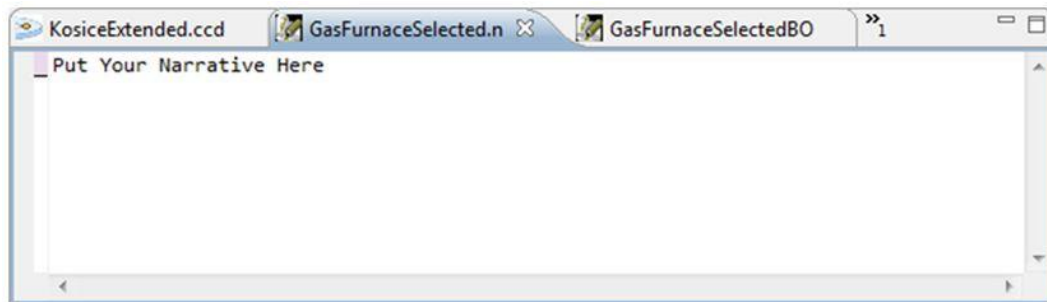


Figure 7: Wizards creating a new narrative model-based scenario.

- An empty model-based scenario will appear as a new window in Eclipse, as it is presented in Figure 8, and the respective \*.narr file is created on your local hard drive, under the path of the Java project. You can start typing the scenario text into the window, using standard editing functions of plain text editor. Changes can be saved by *Ctrl-S* or *File* → *Save* commands.



**Figure 8:** An empty model-based scenario.

**Hint:** When starting with a new (empty) model-based scenario, few blank lines or space characters should be added at the beginning of the text - e.g. by replacing the initial informative text "Put your Narrative Here" (see in Figure 8). This is useful for further annotations, since it allows to annotate the first line of the text and then, optionally, to add a non-annotated text in front of this annotation.

**Note:** The scenario editor is based on the *MyNote Annotator* plug in. To obtain more information on usage of this component, please refer to *D4.2-B: User Manual on CCD Tools*, section 3.5.




**SimAnalysis\_Video.zip** - see the video containing the first part of usage instructions for the *Simulation Analysis Tool*, namely initial procedures for creation of an empty scenario, editing of the scenario text, annotation of scenario text portions to simulation logs, and production of an output model-based scenario.

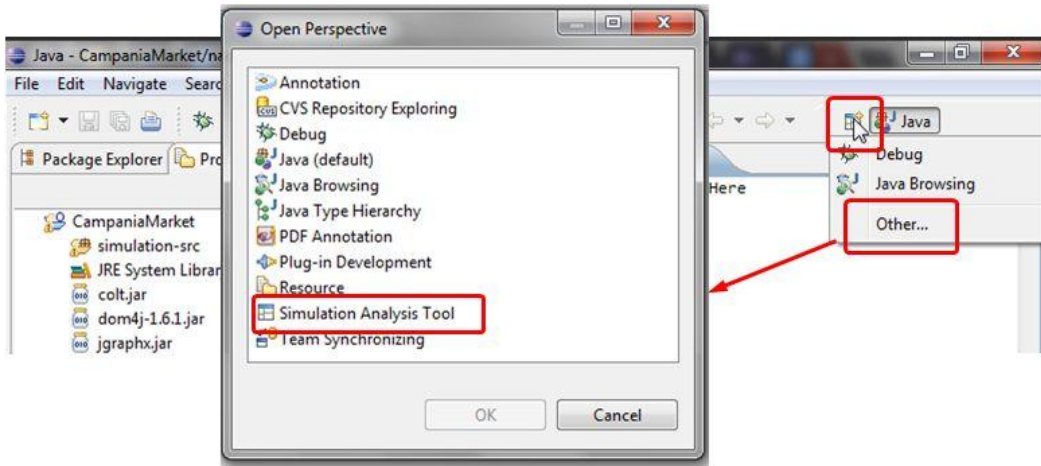
## How to include the simulation logs to the environment for model-based scenario development?



After the Eclipse environment was properly initialised and the \*.narr file was created, you should activate the simulation logs to enable the annotation of scenario text. The simulation logs, produced as outcomes of experimental simulations on agent-based policy model in previous step of the OCOPOMO process, are available in the `../data/experiments` sub-folder of your OCOPOMO Java project. To make these simulation logs available for further simulation analysis and output scenario development, please follow these instructions:

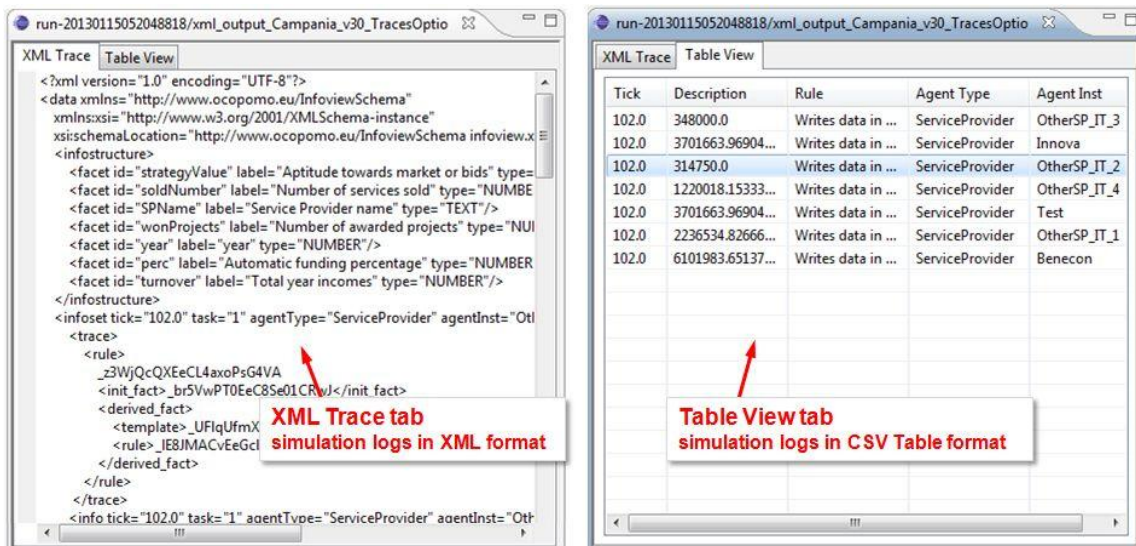
- At first, you need to switch the Eclipse perspective to the *Simulation Analysis Tool*. On the right upper corner of general Eclipse window, click with the right mouse button on the *Perspective Selector* button  (see in Figure 9), select the *Other...* option, and in the appearing *Open Perspective* dialog, select the *Simulation Analysis Tool* perspective.
- The perspective invokes a display of two new windows - *Simulation log views* window, where the simulation logs will be displayed in XML and CSV Table formats, and *Annotation View* that will contain a list of created annotations.

- To open a simulation log in the *Simulation log views* window and to make it available for annotations, expand the `../data/experiments` sub-folder in your OCOPOMO Java project, select a desired simulation run in the `/run-<date-number>` sub-folder, navigate to particular log file (i.e., a `xml_output_*.xml` file), and double-click on it. The respective simulation log will be filled in automatically into the *Simulation log views* window in both XML and CSV Table formats, in the *XML Trace* and *Table View* tabs of the view (see in Figure 10).



**Figure 9:** An empty model-based scenario.

- Use the same procedure for opening other simulation log files - they will be displayed as new tabs in the *Simulation log views* window, each having a separate *XML Trace* and *Table View* tabs for respective XML and CSV Table log formats. The number of simultaneously opened simulation log files is not restricted.
- The simulation logs are then available for annotations of the model-based scenario narrative.



**Figure 10:** Simulation logs available in XML and CSV Table formats.

**Hint:** In case of accidental closing of the "XML/Table Log View" tab, the fastest way to restore it would be to right-click on the *Simulation Analysis Tool* perspective and press "reset".

### 3.2. ANNOTATIONS

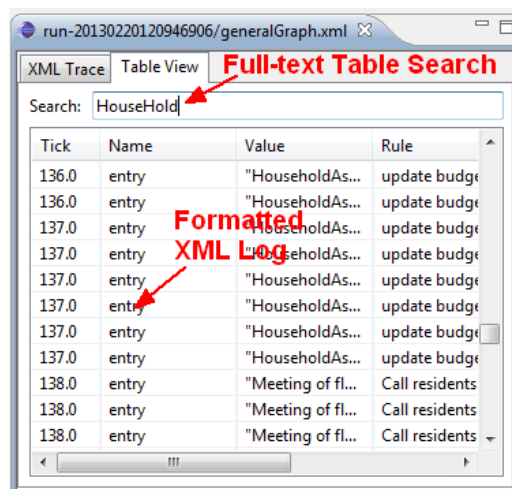
Annotations of the model-based scenario narrative by records of simulation logs can be seen as a fundamental mechanism enabling traceability of statements presented in the output scenario back to initial evidence-based scenarios, which were provided in first two phases of the OCOPOMO process by involved stakeholders, policy makers, and general public as their opinions on analysed and modelled policy case. From this perspective, the annotations, represented as links between particular simulation log records (or their parts) and respective text fragments in the output scenario, support a provenance of arguments and positions expressed in initial scenarios. It implies that the process of formulating and annotating the output scenario narrative is especially important and responsible task, which should be accomplished by skilled experts on policy analysis and modelling<sup>2</sup>.

#### How to annotate the scenario text?



Annotations are links between a text portion of scenario narrative and a particular simulation log record (or its part). In the *Simulation Analysis Tool*, the simulation logs are available in XML and CSV Table formats, as it is presented in Figure 10 above. Both of these formats can be used for annotations, as it is detailed in the following instructions:

1. To start inserting a new annotation, please select a text in the *Xml Log View* or a specific row in the *Table Log View* as follows:
  - a. Use the *Table Log View* in case you wish to link a particular record as whole. To select a particular record, just click on a corresponding row in the table. To navigate to a specific record, look for the values in columns (recommended: *Agent Type*, *Agent Inst*, or *Rule* columns). In the latest version of the *Simulation Analysis Tool*, you can also use the search and filtering functionality, as it is described in the following note:  
**Note:** To ease the navigation to a particular simulation log record, the *Simulation Analysis Tool* in its latest version provides a full-text search capability, which is useful if the list of records is extensive. To search for records, enter the text to the *Search* field located on the top of *Table Log View* (see in Figure 11). Entered text invokes the filtering feature that shows only rows containing the input text in their parameters - column values.



**Figure 11:** Searching simulation log records in the Table View.

<sup>2</sup> More details on methodology of developing and annotating the output model-based scenario narratives are provided in the D8.1 deliverable (Scherer et al, 2013).



- b. Otherwise, if you want to link a specific part of the simulation log to a scenario text fragment, use the *Xml Log View*. Highlight a fragment of XML (preferably a whole XML tag, for example, "<info [attributes]>value</info>" ) by left mouse button to mark a simulation log portion that will be linked to a scenario text.

**Note:** In the *Xml Log View*, simulation log records are specified by the <trace> XML tag. A selection of "<trace>...</trace>" XML text portion is equal to a selection of the log record in the *Table Log View*.

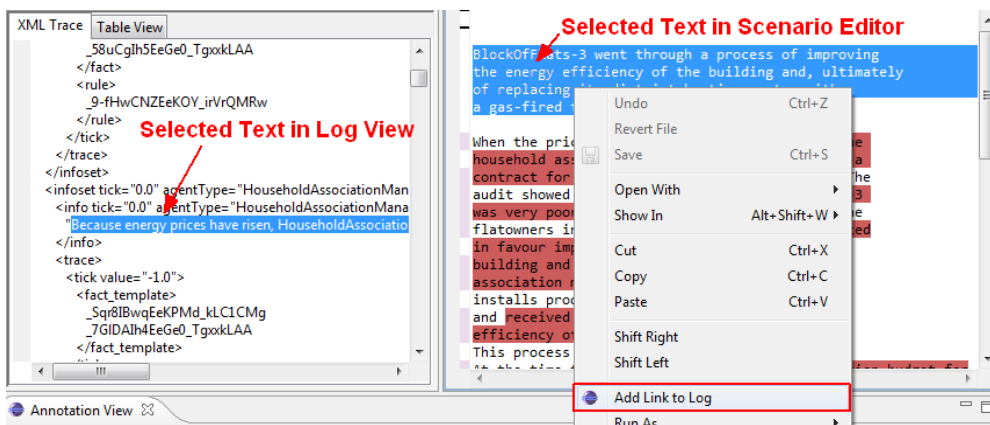
**Note:** To ensure backward traceability, the annotated logs have to contain all the traceability-related information, namely UUIDs of related CCD model concepts. The annotated <info> tags should include the <trace> tag which contains the UUID parameters, as it is presented in Figure 12.

```
<info tick="0.0" agentType="FlatDweller" agentInst="FlatDweller-59" rule="Find similar agents" >
  <info tick="0.0" agentType="FlatDweller" agentInst="FlatDweller-59" rule="Find similar agents" >
    "I have recognised FlatDweller-53 as being similar to me"
  </info>
  <trace>
    <tick value="0.0" >
      <fact_template>
        _9QXpMDVPEeK9aOzBKKFjEw
      </fact_template>
    </tick>
  </trace>
</info>
```

UUID of a CCD model concept

**Figure 12:** Annotating simulation logs in XML format.

2. Switch to the scenario text in the *Model-Based Scenario Editor*. Using the left mouse button, highlight a text portion that should be annotated - i.e., linked with the simulation log record (or its part) which is actually selected in the *Table Log View* or *Xml Log View*.
3. Inside the area of highlighted text in the *Scenario Editor*, use the right mouse button to invoke a contextual menu (see in Figure 13). Click on the *Add Link to Log* command.



**Figure 13:** Annotation of the model-based scenario.

4. The annotation is created - the annotated text fragment in the *Scenario Editor* is marked by a background colour. The linked simulation log record (or its part) is highlighted in the *Xml Log View*. Finally, the newly created annotation is included into the list of all existing scenario annotations, which is displayed in the *Annotation View* below the editor.

---

### How to update already annotated scenario text?



Since the provided *Model-Based Scenario Editor* is built on the *MyNote Annotator* plug-in, which preserves existing annotations during the text editing, the update of already annotated scenario text is straightforward. To do so, please follow these steps:

1. In the *Model-Based Scenario Editor*, click with left mouse button inside an already annotated text, which is marked by a background colour.
2. Do the desired text modifications by typing or removing text characters. The annotation will be preserved and automatically adapted to the updated text.
3. Do not forget to save the changes by *Ctrl-S* or *File* → *Save* commands.

---

### How to annotate the scenario text by several simulation logs?



In the provided version of the *Simulation Analysis Tool*, one scenario can be linked with multiple log files. It means that the output model-based narrative can be formulated as a summary of several simulation runs, possibly produced by different agent-based models. To perform annotations from several simulation logs, please follow these instructions:

1. To open a simulation log in the *Simulation log views* window to make it available for annotations, expand the `../data/experiments` sub-folder in your OCOPOMO Java project. Select a desired simulation run in the `/run-<date-number>` sub-folder, navigate to particular log file (i.e., a `xml_output_*.xml` file), and double-click on it. The respective simulation log will be filled in automatically into the *Simulation log views* window in both XML and CSV Table formats.
2. Do the scenario text editing and annotation operations in a standard way, in accordance with the above-presented instructions.
3. To open another simulation log, simply repeat the procedure described in step 1. Expand the `../data/experiments` sub-folder, select the simulation run, navigate to the log file you want to use for annotations and double-click on it. The *Simulation log views* window will load the content from this new simulation log into both *XML Trace* and *Table View* tabs.
4. Proceed with the scenario text editing and annotation operations as needed. Finally, save the changes by *Ctrl-S* or *File* → *Save* commands.

---

### How to navigate to a simulation log record from the annotated scenario text?



After several iterations of scenario formulation, editing, and annotation, the structure of annotations and underlying simulation logs can be quite complex. To enable an easier editing and manipulation with the annotated scenario narrative, a navigation from already annotated text portion to respective simulation log records (or their parts) is supported as follows:

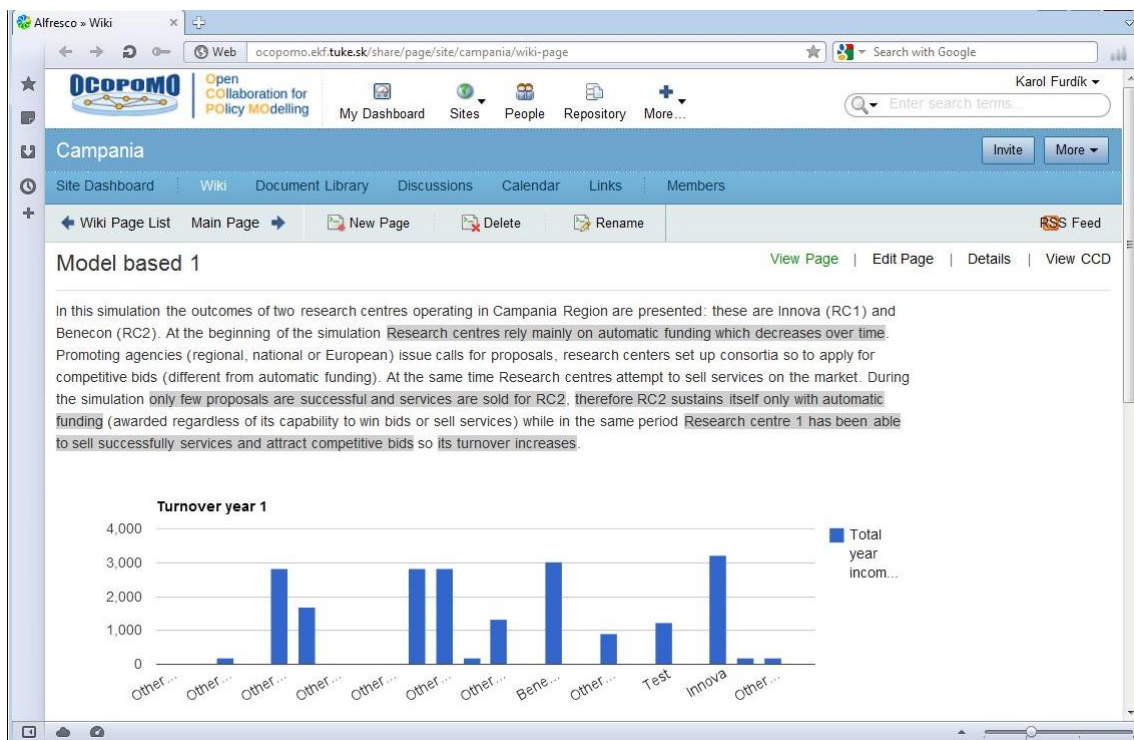
1. In the *Model-Based Scenario Editor*, double-click with left mouse button inside an already annotated text, which is marked by a background colour.
2. The simulation log file, which contains a record corresponding to the annotated text, will be automatically loaded into the *XML Trace* and *Table View* tabs.

- The simulation log record, which actually corresponds to the annotated text fragment in scenario narrative, will be highlighted in its XML format presented in the *XML Trace* tab of *Xml Log View*; moreover, the *Xml Log View* will scroll automatically to the highlighted simulation log record (or its part).

## How to include numerical data from simulations into the resulting model-based scenario?



Simulation experiments may provide some of outputs in a form of numerical data, which are typically presented in the output scenario as *charts* of various types, as it is depicted in Figure 14.



**Figure 14:** Upload of produced model-based scenario narrative to Alfresco.

The numerical outputs produced by simulation runs are provided by the *Output Writer* module of DRAMS in CSV format, as it is detailed in D4.2-C: *User Manual on Policy Modelling and Simulation Tools*, sections 3.2.3 and 6.2.1. The output data files are stored in respective `/run-<date-number>` sub-folders and therefore they are accessible from Eclipse environment, e.g. by opening and reading the numerical data logs in the *Simulation Analysis Tool*.

However, the Eclipse environment of the OCOPOMO toolkit, at least in its current version, does not support a direct insertion of charts into the output scenario narrative. This functionality is provided only by Alfresco-based tools, namely the *Wiki* and *Charts Management* components<sup>3</sup>. It implies that the output scenario, created in Eclipse as a plain text with annotations, needs to be uploaded to the remote Alfresco web space, and only after that the charts can be inserted into the scenario text. Usage instructions on inserting charts into the text of published scenarios are provided in D4.2-E: *User Manual on Simulation Output Visualisation and Traceability Tools*, section 3.2.

<sup>3</sup> For more information and usage instructions on inserting charts into the scenario wiki page text, please refer to D4.2-A: *User Manual on Collaboration and Scenario Generation Tools*, section 3.1, and D4.2-E: *User Manual on Simulation Output Visualisation and Traceability Tools*, section 3.2.

### 3.3. UPLOAD OF THE RESULTING SCENARIO TO ALFRESCO

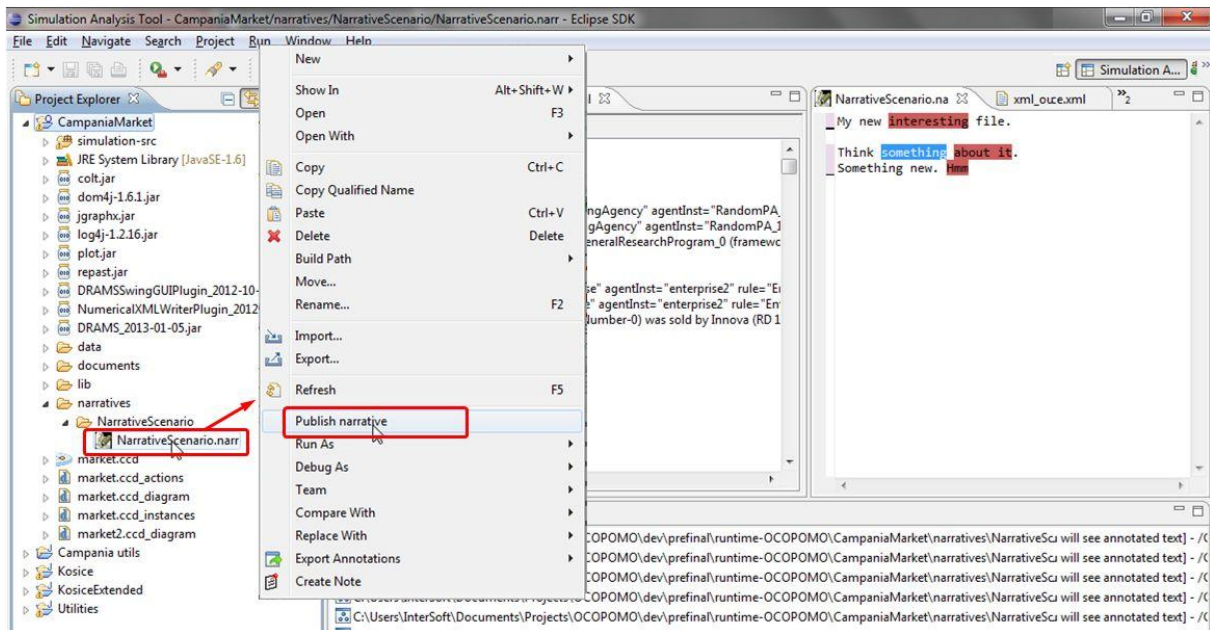
Created and properly annotated narrative scenarios can be finally published to the shared web space of Alfresco. This functionality is supported by the *Content Repository Client* middleware, installed as a plug-in in the Eclipse environment (see in section 2.3, as well as the user manual for the *Content Repository Client* included in D4.2-B *User Manual on CCD Tools*).

#### How to publish the resulting model-based scenario narrative to the Alfresco web space?



After several iterations of scenario formulation, editing, and annotation, the structure of annotations and underlying simulation logs can be quite complex. To enable an easier editing and manipulation with the annotated scenario narrative, a navigation from already annotated text portion to respective simulation log records (or their parts) is supported as follows:

1. In the *Project Explorer* view, navigate to the \*.narr file that contains an inner representation of the produced narrative scenario.



**Figure 15:** Upload of produced model-based scenario narrative to Alfresco.

1. Right click on the \*.narr file of the scenario.
2. In the invoked dialog, click on the *Publish Narrative* command (see in Figure 15). This will initiate the procedure of uploading the scenario and all the accompanying data resource files to the remote Alfresco web space.

**Note:** The procedure of upload and publishing of the whole structure of model-based scenarios and accompanying files to Alfresco is performed automatically, as a sequence of the following steps:

- a) Conversion of the produced model-based scenario, i.e., its plain text content and annotations, stored locally in the \*.narr file, to the Alfresco wiki format;
- b) Uploading of the converted *scenario wiki page* into the list of Alfresco wiki pages;
- c) Uploading of *DRAMS code* to the <OCOPOMO\_Alfresco\_site>/data/drams/ folder;



- d) Uploading of *CCD model* to the <OCOPOMO\_Alfresco\_site>/data/ folder;
- e) Uploading of simulation logs to the <OCOPOMO\_Alfresco\_site>/data/runs/ folder;
- f) Uploading of all locally created background documents, which were used for annotating input evidence-based scenarios and CCD model development, as well as for annotations of output model-based scenario narrative, to the <OCOPOMO\_Alfresco\_site>/data/documents/ folder.



***MBSscenarioPublish\_Video.zip*** - see the video containing the second part of usage instructions for *Simulation Analysis Tool*, namely the publishing of produced narrative scenario to the remote Alfresco web space.

Published scenario narrative and related documents can be explored on the Alfresco site by means of the *Collaboration* and *Scenario Generation Tools* or by the *CCD Model Explorer applet*. Usage instructions for these tools are provided in separate user manuals, namely in *D4.2-A: User Manual on Collaboration and Scenario Generation Tools* and *D4.2-E: User Manual on Simulation Output Visualisation and Traceability Tools*.



## **4. CONCLUSION**

This manual provides the documentation and usage instructions for the *Simulation Analysis* tools as a part of the integrated OCOPOMO toolkit. The manual describes common tasks for end users (policy analysts and modellers, assisted by facilitators and system administrators) during the analysis of simulation results generated from agent-based policy models and the development of output model-based narrative scenarios. An emphasis is given on tasks related to the annotation of scenario texts by simulation log records (see in section 3.2), since they are crucial for enabling the traceability from output scenarios back to agent-based policy models, conceptual CCD models, and finally to input evidence scenarios provided by involved stakeholders (Lotzmann, Wimmer, 2012).

Since the implementation of the *Simulation Analysis* is based on the Eclipse Indigo platform (Eclipse Indigo, 2011), where all the tools and components are implemented as Eclipse plug-ins, more information on the use of this platform can be found in (Eclipse Indigo Documentation, 2013) as well as on the Eclipse community, <http://www.eclipse.org>.

Further information resources:

- Methodology guidelines, OCOPOMO deliverable D8.1 (Scherer et al, 2013);
- Architecture and functionality description, OCOPOMO deliverable D2.1 (Mach et al, 2010);
- other public deliverables of the OCOPOMO project, available at <http://www.ocopomo.eu>.

Technical support on the presented tools can be provided upon request. To obtain such a support, please contact the coordinator of the OCOPOMO project (see contacts at <http://www.ocopomo.eu>).



## **REFERENCES**

- Alfresco Share* web client, [http://wiki.alfresco.com/wiki/Alfresco\\_Share](http://wiki.alfresco.com/wiki/Alfresco_Share). Alfresco Software, Inc., 23 March 2011.
- Eclipse Indigo Documentation*, HTML Help Center, Eclipse Indigo (3.7) Documentation, <http://help.eclipse.org/indigo/index.jsp>. The Eclipse Foundation, 2013.
- Eclipse Indigo*, the annual release of Eclipse projects, <http://www.eclipse.org/indigo/>, The Eclipse Foundation, June 22, 2011.
- GATE*, General Architecture for Text Engineering, The University of Sheffield, <http://gate.ac.uk>, 2011.
- Lotzmann, U., Wimmer, M. A.: *Provenance and Traceability in Agent-based Policy Simulation*. In: Proceedings of 26th European Simulation and Modelling Conference - ESM'2012, October 22-24, 2012, FOM, Essen, Germany, 2012.
- Mach, M., et al: *D2.1 Platform Architecture and Functional Description of Components*. Deliverable 2.1, OCOPOMO consortium, 2010.
- Scherer, S., et al: *D8.1 Manual of the Methodology for Process Development and Guide to Policy Modelling Toolbox*. Deliverable 8.1, OCOPOMO consortium, 2013.