OCOPOMO
Open Collaboration in Policy Modelling

D4.2 System and User Documentation

B: User Manual on CCD Tools

Document Full Name	**OCOPOMO_D4.2-B_CCD-UserManual.doc**
Date	**25/03/2013**
Work Package	**WP4: Integration of components**
Lead Partner	**Intersoft**
Authors	**Sabrina Scherer, Maria A. Wimmer, Scott Moss, Peter Smatana, Karol Furdík**
Document status	**v1.00 FINAL**
Dissemination level	**PUBLIC (PU)**

Document Log

Version	Date	Comment	Author
001	08/08/2011	Documentation describing the Installation and Use of the first prototype version of the CCD Tool	Sabrina Scherer
002	11/08/2011	Updating for recent changes of prototype	Sabrina Scherer
003	25/01/2012	Updating for recent changes of prototype	Sabrina Scherer
004	27/01/2012	Updating for recent changes of prototype	Sabrina Scherer
005	27/01/2012	Included section describing the installation and use of the PDF-HTML Annotation tool.	Peter Smatana, Karol Furdik
006	29/01/2012	Reviewing and editing	Maria Wimmer
007	30/01/2012	Editing of CCD Model Editor description	Sabrina Scherer
008	26/03/2012	Updating for newest changes in the CCD Tool	Sabrina Scherer
009	18/04/2012	Correcting text	Sabrina Scherer, Scott Moss
010	24/04/2012	Updating installation information for Eclipse Update Site information	Sabrina Scherer
011	10/05/2012	Updating the manual for the new filter functionality and the “known by” reference.	Sabrina Scherer
012	15/05/2012	Editing of CCD Diagrams section	Sabrina Scherer
013	13/02/2013	Updating with new features of CCD, adding new figures	Sabrina Scherer
014	19/02/2013	Updating of CCD2DRAMS section	Sabrina Scherer
015	21/02/2013	Updating of Content Repository Client description, connection to Alfresco	Peter Smatana
016	06/03/2013	Editing of description of annotators	Peter Smatana, Karol Furdik
100	25/03/2013	Updating of CCD section. Integration of inputs, updates of all chapters. Consolidation, corrections and finalisation.	Sabrina Scherer, Karol Furdik

TABLE OF CONTENTS

1. INTRODUCTORY NOTES

This user manual describes the usage of *Consistent Conceptual Description (CCD) Tools* and related components of the integrated OCOPOMO toolkit, which provide the means for developing conceptual models of policy alternatives. The suite of CCD 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. It implies that the working environment needs to 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.

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 third phase of the OCOPOMO process, i.e., the development of conceptual policy models by analysis and annotation of evidence-based scenarios produced in previous phases of the process.

Usage guidelines are complemented by video presentations of key tools and components of the *CCD Tools* suite as follows:

- *Install_OCOPOMO_Video.zip* - instructions on the installation of the OCOPOMO plug-in to the Eclipse environment;
- *CCD_Video.zip* - instructions on how to create a CCD and how to create actors, objects, instances and relationships between them;
- *Annotators_part1_Video.zip* and *Annotators_part2_Video.zip* - usage instructions for PDF Annotator, HTML Annotator, and Content Repository Client components;
- *MyNoteAnnotator_Video.zip* - usage instructions for MyNote Annotator, i.e., the annotation of editable texts (see in section 3.5).

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 TOOLS

This chapter presents a general information on the *CCD 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 in the OCOPOMO system.

Icon	User Role
👤	Politician
👤	Civil servant
👥	Stakeholder
👤	**Facilitator**

Icon	User Role
👤	**Analyst**
👤	**Modeller**
👤	**Administrator**

The suite of *CCD Tools* supports the third phase of the OCOPOMO process, i.e., a transformation of textual policy scenarios into their conceptual representation. The respective actions of this process phase are performed by *Facilitator*, *Analyst*, and *Modeller* user roles, while the set up and maintenance of the Eclipse environment and its proper connection to the Alfresco content repository 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 conceptual model development[1]; the produced CCD model is 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.

2.2. POSITION OF CCD TOOLS IN THE OCOPOMO PROCESS

A core objective of OCOPOMO is to ensure the consistency between scenarios of policy alternatives, which were collaboratively created by involved stakeholders, and the simulation models developed by policy modellers (cf. D2.1 (Mach et al, 2012) and D5.1 (Moss et al, 2011)). The suite of *CCD Tools* supports policy analysis and modelling experts in developing a stakeholder-accessible formalisation of a policy model in the form of a conceptual model (the CCD) and thereby ensuring and documenting the consistency. Figure 1 visualises different steps of the OCOPOMO process[2], in terms of exchanged

[1] 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] http://www.ocopomo.eu/results/glossary/ocopomo-policy-development-process; see also Figure 1 in the main D4.2 document.

artefacts and supporting ICT tools. The suite of *CCD Tools* has a central position in the process since it conceptually and functionally connects the scenarios produced by public with simulation models developed by experts.

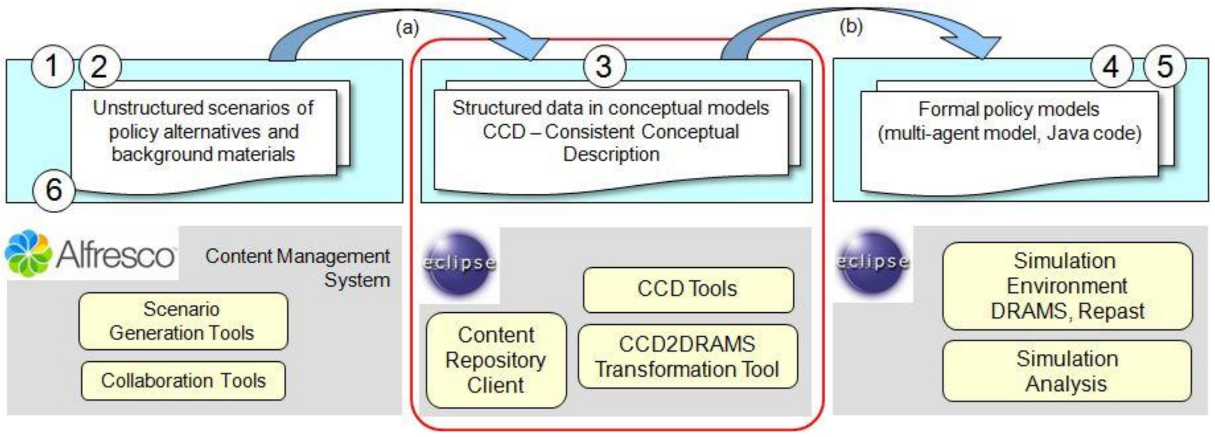

(a)
(b)
1
2
6
Unstructured scenarios of policy alternatives and background materials
3
Structured data in conceptual models CCD – Consistent Conceptual Description
4
5
Formal policy models (multi-agent model, Java code)
Alfresco
Content Management System
Scenario Generation Tools
Collaboration Tools
eclipse
CCD Tools
Content Repository Client
CCD2DRAMS Transformation Tool
eclipse
Simulation Environment DRAMS, Repast
Simulation Analysis

Figure 1: Position of the CCD Tools in the OCOPOMO policy development process.

Figure 1 highlights the transformation steps necessary to bridge the gap between provided scenarios and developed formal simulation models as follows:

a) Transformation of a scenario text into CCD, which includes:
   - Download of scenarios and background documents from the Alfresco web space and their availability in the Eclipse environment - using the *Content Repository Client* tool (see in section 2.5);
   - Construction of the CCD model by defining the CCD model entities (actors, objects, relations, etc.) and diagrams - using the *CCD Model Editor* (see in section 3.1);
   - Annotation of scenario texts by linking text fragments to CCD model elements - using a set of *annotation tools* presented in sections 3.2-3.5.

b) Transformation of conceptual models into programming code of agent-based policy models:
   - Automatic generation of Java/DRAMS source code stubs for policy models and simulations - using the *CCD2DRAMS* tool, described in section 3.6.

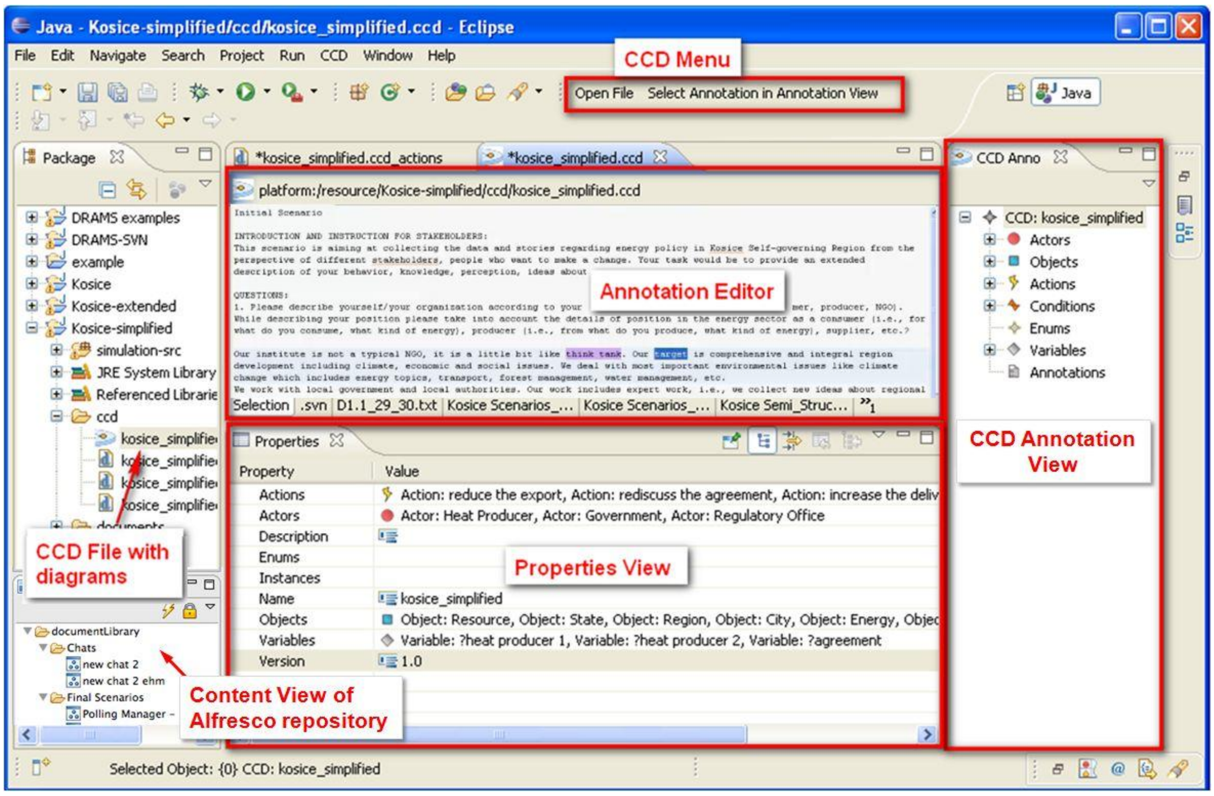
2.3. USER INTERFACE AND STRUCTURE OF THE TOOLS

The default user interface of the *CCD Tools* is presented in Figure 2. Particular tools and data sources are accessible in the Eclipse environment in a structure of several views/perspectives. However, since these 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).

The *CCD Tools* user interface depicted in Figure 2 consists of the following items:

- *CCD menu*, providing shortcuts to the most common operations;
- *CCD File with diagrams*, included into the OCOPOMO Java project;
- *Content View of Alfresco repository*, a structure of folders and files downloaded from the Alfresco web space by means of the *Content Repository Client*;
- *Annotation Editor*, a space for displaying the CCD model and the content of documents in various formats (i.e., TXT, PDF, HTML) that are annotated to the CCD model elements; all the data resources are displayed in a multi-tab view;

- *CCD Annotation View*, a tree structure of the CCD model, together with annotation nodes;
- *Properties View*, a list of properties of the CCD element node that is selected in the CCD Annotation View.



```
Initial Scenario

INTRODUCTION AND INSTRUCTION FOR STAKEHOLDERS:
This scenario is aiming at collecting the data and stories regarding energy policy in Kosice Self-governing Region from the
perspective of different stakeholders, people who want to make a change. Your task would be to provide an extended
description of your behavior, knowledge, perception, ideas about

QUESTIONS:
1. Please describe yourself/your organization according to your                            mer, producer, NGO).
While describing your position please take into account the details of position in the energy sector as a consumer (i.e., for
what do you consume, what kind of energy), producer (i.e., from what do you produce, what kind of energy), supplier, etc.?

Our institute is not a typical NGO, it is a little bit like think tank. Our target is comprehensive and integral region
development including climate, economic and social issues. We deal with most important environmental issues like climate
change which includes energy topics, transport, forest management, water management, etc.
We work with local government and local authorities. Our work includes expert work, i.e., we collect new ideas about regional
```

Property	Value
Actions	Action: reduce the export, Action: rediscuss the agreement, Action: increase the deliv
Actors	Actor: Heat Producer, Actor: Government, Actor: Regulatory Office
Description	
Enums	
Instances	
Name	kosice_simplified
Objects	Object: Resource, Object: State, Object: Region, Object: City, Object: Energy, Objec
Variables	Variable: ?heat producer 1, Variable: ?heat producer 2, Variable: ?agreement
Version	1.0

Figure 2: General user interface of CCD Tools in the Eclipse environment.

2.4. SETTING UP THE ECLIPSE WORKSPACE

This section contains a more detailed instructions on the maintenance and customisation of a local installation of the OCOPOMO ICT Toolkit in Eclipse Indigo (Eclipse Indigo, 2011). To support a getting started procedure, we are presenting here some of the most fundamental instructions on the usage and maintenance of the provided CCD Tools in the Eclipse environment, including a set up of data transfer connection with the Alfresco web space by means of the *Content Repository Client*.

How to set up a new OCOPOMO project in Eclipse?

All the work on conceptual policy analysis, modelling, and simulation in the Eclipse part of the OCOPOMO ICT toolkit is organised under a single Eclipse Java project. To start using the CCD Tools, the Java project has to be created at first, together with a CCD file that will contain the results of policy analysis - CCD model and annotations of input scenarios. To set up a new project in Eclipse, please follow these instructions:

1. Create a new Java Project: Click on *File* -> *New*; select *General* -> *Project* and follow the instructions, give it a name of your choice.
2. Select the new project (name you gave in step 1) and click on *File* -> *New* -> *Other*.
3. Select *CCD* in the appearing wizard and click *Next*.
   Hint: if you do not find *CCD*, type this into the text field under "*Wizards:*".

4. Give the CCD file a *Name* and click on *Next*.
5. Select *CCD* in the drop down box under *Model Object* and click on *Finish*.

A new file with ending `*.ccd` appears in the project and in the middle of Eclipse the *Annotation Editor* is opened (see in Figure 2). If the *Annotation Editor* is not shown, please click with the right mouse button on the CCD file and select “*Open with -> CCD Annotation Editor*”.

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.

How to set global properties for the CCD model?

To start the CCD model development, a CCD file has to be created at first. The CCD file, included into the underlying OCOPOMO Java project, will contain a representation of the CCD model. The CCD file can be created by performing the following steps in Eclipse:

1. Click with the right mouse button on the *CCD* node in the *CCD Annotation view* (right frame, see above in Figure 2), then select the *Show Properties View* command in the context menu.
2. A list of properties for the CCD model will appear in the *Properties view* (see the bottom frame in Figure 2).
3. A *Name*, a *Version*, and a *Description* can be defined here for the root CCD file, which represents the CCD model as whole.
4. Click into the *Value* cell in the *Name* row and type in the name of the CCD. The same you can do with the *Version*.

Are there any examples of OCOPOMO projects and/or CCD models available?

Yes, the examples are available and can be uploaded to the Eclipse environment. In the case you want to view an existing CCD project (*.zip), please perform the following steps:

1. Go to *File -> Import*; Select *General -> Existing Projects into Workspace* in the wizard and click on *Next*.
2. Next step is to *Select archive file* by *Browsing* your local space. Choose your example(s) (e.g. “CCD_Example.zip”[3] – you may add several examples). If not already selected, click the example(s) you want to import and hit *Finish*.

You may now view and navigate through the example you have imported.

CCD_Video.zip - see the video demonstrating the interaction and use of the CCD. Instructions on how to create a CCD and how to create actors, objects, instances and relationships between them.

[3] A simple example is available under http://www.ocopomo.eu/reviewer-space/review-y2/deliverables/ccd-prototype/CCD_Example.zip

2.5. CONTENT REPOSITORY CLIENT - CONNECTION TO ALFRESCO

After the OCOPOMO Java project and the CCD file was properly created, the workspace set up procedure continues with establishing a data transfer connection between Eclipse and Alfresco environments. The connection is enabled by the *Content Repository Client* plug-in, which is included in the installation package of CCD Tools.

How to connect the OCOPOMO project with Alfresco?

The OCOPOMO Java project can be connected with an instance of the Alfresco Content Repository, identified by site URL and repository ID, to enable a file transfer - upload and download of the content from/to the Alfresco web site. Namely, scenarios of policy alternatives, which were collaboratively produced by involved stakeholders and are available on the remote Alfresco repository, will be downloaded to the local Eclipse installation and will be visualised in the *Content view* of Eclipse user interface in a tree structure (see in Figure 2). To connect to a remote Alfresco web site, please follow these instructions:

1. Navigate to the properties of the OCOPOMO 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).
2. Click on *OCOPOMO → Content Repository Client* to invoke the dialog presented in Figure 3.
3. Specify the entries required for identifying the Alfresco repository instance. You can select one from pre-defined sites or change individual preferences according to your needs. When the connection is correctly configured, you should see an OCOPOMO decorator mark on the label of your Java project (see in Figure 4).

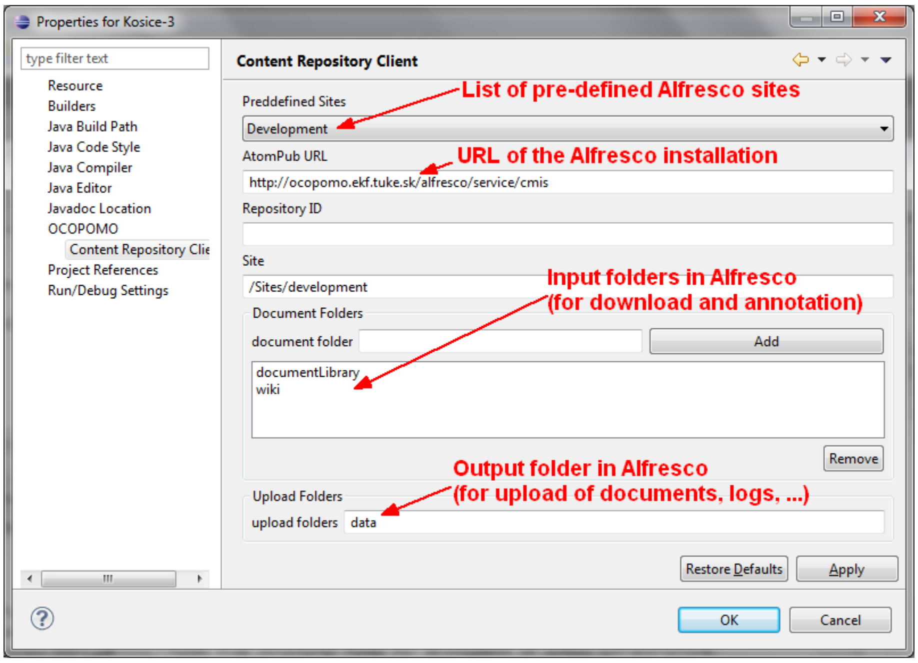

Properties for Kosice-3
Content Repository Client
Preddefined Sites: Development
AtomPub URL: http://ocopomo.ekf.tuke.sk/alfresco/service/cmis
Repository ID:
Site: /Sites/development
Document Folders
document folder:
documentLibrary
wiki
Upload Folders
upload folders: data

Figure 3: Content Repository Client properties for enabling a connection to Alfresco.

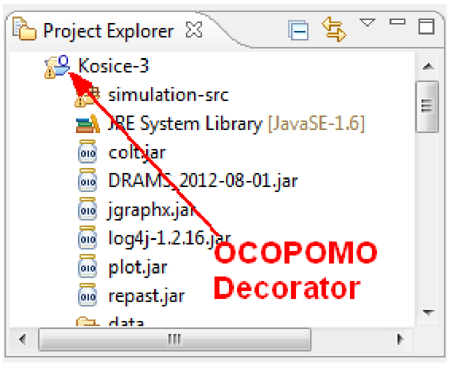


Figure 4: The OCOPOMO Java project connected with a remote Alfresco repository.

How to download the published connect from Alfresco to the OCOPOMO Java project?

When the connection between local Eclipse and remote Alfresco environments is successfully established, you can download the Alfresco content (namely, the scenarios of policy alternatives) into your OCOPOMO Java project in Eclipse by following these steps:

1. The Alfresco content will appear in Eclipse in the *Content View* perspective (see in Figure 2). To open the *Content View*, please use the main menu *Window → Show View … → Other …* and select *Content View* under the OCOPOMO category.
2. To initiate the download of remote content, click on the lightning icon ⚡ in the *Content View* perspective (see in Figure 5). The access to the Alfresco repository can by 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 5.

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).

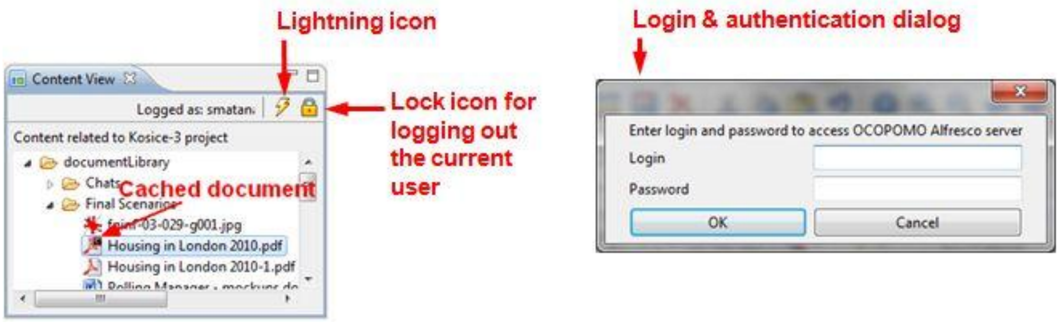

Lightning icon
Login & authentication dialog
Content View
Logged as: smatan
Lock icon for logging out the current user
Content related to Kosice-3 project
documentLibrary
Chats
Cached document
Final Scenarios
fgint-03-029-g001.jpg
Housing in London 2010.pdf
Housing in London 2010-1.pdf
Enter login and password to access OCOPOMO Alfresco server
Login
Password
OK
Cancel

Figure 5: The OCOPOMO Java project connected with a remote Alfresco repository.

3. After the download, the *Content View* perspective displays the tree structure of folders and documents downloaded from the Alfresco site. The documents are internally marked as related to the currently selected OCOPOMO Java project (please note that Eclipse enables to have several Java projects opened at the same time; however, only a single project can be selected as currently active).

Note: Selection of the Java project (for a *Content View* perspective) is related to the CCD file, which is opened in the *Annotation Editor*. It means that the *Content View* will be refreshed by a content related to configuration of the project in which the opened CCD file appears. When you will switch to another CCD file, then the *Content View* will be refreshed by its related content. The hierarchy of folders and documents is automatically downloaded but first time you are trying to connect to Alfresco you will be asked to authenticate yourself (see in Figure 5).

4. You can browse the structure of folders and documents in the *Content View* perspective. However, documents displayed in the hierarchy are not directly downloaded to your computer. To download a particular document to your local hard drive, please double click on it (again, please be patient, it can take some time).

Note: Downloaded document is decorated by *"c"* decorator (i.e., a *cached* document), as it is depicted in the left hand side of Figure 5. All cached documents are stored locally and can be found in the *"<related project>/documents"* folder.

5. You can reload the structure of Alfresco repository at any time by clicking the lightning icon ⚡. This functionality is used when the content on the Alfresco site has changed.
6. To log out from the Alfresco repository, please click on the lock icon 🔒.

After the download of documents from the Alfresco repository, the provided narrative scenarios and other background materials can be analysed and conceptually modelled by *CCD Tools*, as it is described in the following chapter.

3. USE OF THE CCD TOOLS

The following sections contain descriptions and instructions on how to perform the most common operations of conceptual modelling on provided scenarios of policy alternatives. Operations are grouped according to the components of the *CCD Tools*.

3.1. CCD MODEL EDITOR

The *CCD Model Editor* is the central tool for conceptual policy analysis and CCD model development. The user interface of the *CCD Model Editor* consists of three frames (i.e., Eclipse views or perspectives), as it is presented above in Figure 2. It namely includes (1) *Annotation Editor*, a multi-tab space for displaying the CCD model and the content of annotated documents, (2) *CCD Annotation View*, a tree structure of CCD model elements and annotation nodes, and (3) *Properties View*, a list of properties of the CCD element selected in the *CCD Annotation View*.

Functionality of the *CCD Model Editor* is rather complex, since it covers the maintenance of various types of CCD elements and related diagrams. It will be described in detail in the following sub-sections.

CCD_Video.zip - see the video demonstrating the interaction and use of the CCD. Instructions on how to create a CCD and how to create actors, objects, instances and relationships between them.

3.1.1. Entities of CCD model

The CCD model provides a number of entities which can be described in a CCD file. The entities are defined in the CCD Meta-model - the CCD vocabulary. Table 2 lists the entities. The first column shows the name of the entity and how it is visualised in the *Annotation Editor* when a text is highlighted for annotation (see in section 3.2). The second column gives a short description. The last column names the section, where the entity is described in more detail.

Table 2: Overview of entities defined in the CCD Meta-model.

CCD entity	Short Description	Section
● Actor	Classes of stakeholders in the policy model. They are transformed to agents later in the consistent transformation process	3.1.2
■ Object	Classes of things, which are relevant in the policy model such as groupings, institutions or artefacts (e.g. “House”, “Material”) as well as virtual helper objects (e.g. “Heat Delivery”)	3.1.3
➡ Relation	A conceptual relationship between two actors, two objects or an actor and an object.	3.1.4
✚ Attribute	*Characteristic or property of concepts Actor* or *Object*. Attribute have a primitive data value (Integer, Double, Enum, String)	3.1.5
Enum	An enumeration as a primitive value type	3.1.6
Instance	Concrete occurrences / instantiations of an *Actor* or an *Object*.	3.1.7
➡ Relation Instance	Relationship between two *Instances* based on a defined conceptual *Relation* between two concepts (of type *Actor* and/or *Object*).	3.1.7.1
✚ Attribute Instance	Concrete occurrence of an *Attribute* of an *Instance*.	3.1.7.2

Action	Representation of tasks or actions in the policy case. Actions are usually executed by actors and pursue an objective. In our concept, we also model natural events and consequences, e.g. "*it rains*" and as consequence "*it is wet*". With such actions, no actor is affiliated.	3.1.8
Condition	Representation of an pre-condition or post-condition (meant as consequence) of an action. A *Condition* as post-condition of one action can be used as pre-condition for another action. This way, behaviour of and interaction between actors are modelled. *Conditions* describe only one particular characteristic on an element.	3.1.8
Variable	Representation of objects or actors used as flexible declaration in one or more *Conditions*.	3.1.8
Literal	An enumeration value.	3.1.6
File Annotation (Txt Annotation and Pdf Annotation)	The CCD tool makes it possible to **annotate text** by selecting text in the "Annotation Editor" and dragging it on the appropriate node in the "Annotation View". By dropping it in a node, a *File Annotation* (either Txt or Pdf) is added into the selected node in the CCD.	3.2
Expert Annotation	Representation of an opinion of an expert which is not described in documents or an argument to describe a modelling decision.	3.2

For each of these entities, a *Name* and a *Description* can be defined. The CCD Meta-model for the concepts is described in detail in D 3.1 (Bednar et al, 2012), namely in section 3.3 of the deliverable. How the concepts are to be used by the facilitators, policy analysts and modellers is detailed subsequently. The extraction of such concepts from scenario texts and background documents is described in sections 3.2-3.5 presenting various annotation tools.

3.1.2. Actors

Actors and *Objects* allow the description of *Actor Network Diagrams*. It is possible to describe relations between different actors, between actors and objects and between different objects.

The concept *Actor* represents classes of stakeholders in the policy domain and can be transformed to agents later in the consistent transformation process. Actions executed by a class of actors can be linked with the actor class.

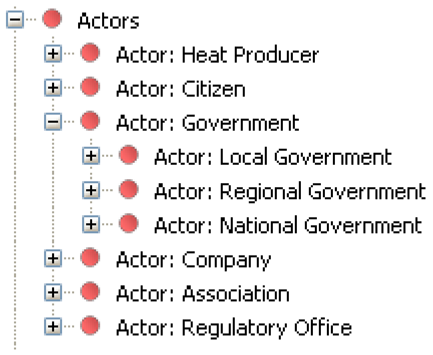

Actors
Actor: Heat Producer
Actor: Citizen
Actor: Government
Actor: Local Government
Actor: Regional Government
Actor: National Government
Actor: Company
Actor: Association
Actor: Regulatory Office

Figure 6: Examples for Actors: Heat Producer, Citizen, Government, Company, etc.

How to add an Actor entity into the CCD model?

To add an Actor, select the CCD node and select *New Child -> Actor*:

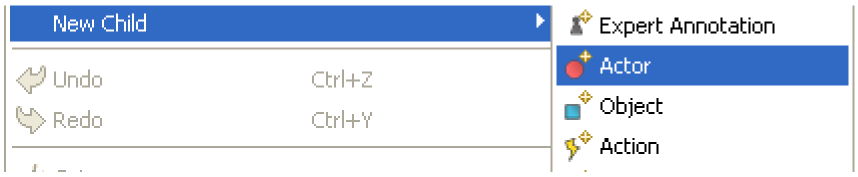


Figure 7: Add an Actor entity to the CCD model.

The following properties can be set for actors in the *Properties View* (select the appropriate *Actor* node, click on the right mouse button and select *Show Properties View*):

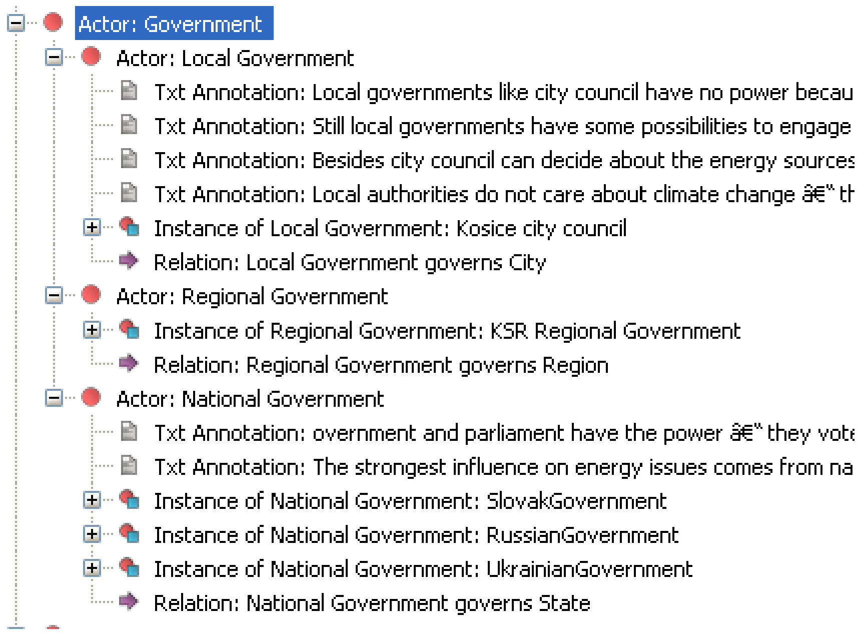
Table 3: Properties of Actors.

Property	**Description**
Actions	List of actions, which the *Actor* performs.
Description	A description of the *Actor*.
Name	Name of the *Actor*. Also name of the generated java agent classes.

The possible types of sub nodes of an *Actor* are listed in Table 4. To add them select the appropriate *Actor* node, click on the right mouse button and select *New Child -> <Type of Subnode>* (not possible with Txt Annotation or PDF Annotation, because those can be only added by using the annotation feature, see sections 3.2 and 3.3).

Table 4: Possible types of sub nodes of an Actor node.

Type of Subnode	**Description**
Txt Annotation	Textual annotation which references a text phrase in a document. Double click on this node opens the appropriate text phrase.
Expert Annotation	Expert annotation which gives evidence for this node based on an expert.
Instance	Instantiation of the *Actor*. See section 3.1.6.
Relation	Relationship to another Actor or Object. See section 3.1.4.
Attribute	Primitive value attribute. See section 3.1.5.
● Actor	Sub-actors representing a hierarchical relationship between *Actors*. Example: *Government* with sub actors *Local Government*, *Regional Government*, *State Government*.

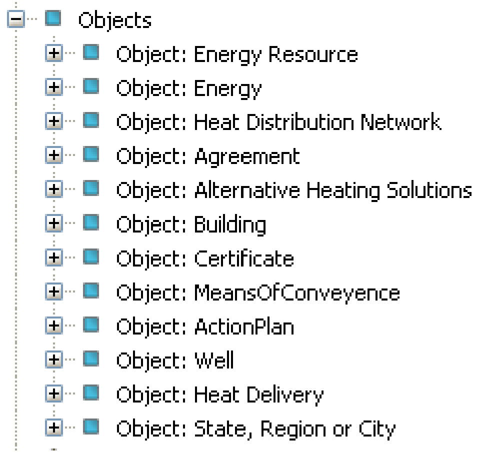


Actor: Government
Actor: Local Government
Txt Annotation: Local governments like city council have no power becau
Txt Annotation: Still local governments have some possibilities to engage
Txt Annotation: Besides city council can decide about the energy sources
Txt Annotation: Local authorities do not care about climate change â€" tł
Instance of Local Government: Kosice city council
Relation: Local Government governs City
Actor: Regional Government
Instance of Regional Government: KSR Regional Government
Relation: Regional Government governs Region
Actor: National Government
Txt Annotation: overnment and parliament have the power â€" they vote
Txt Annotation: The strongest influence on energy issues comes from na
Instance of National Government: SlovakGovernment
Instance of National Government: RussianGovernment
Instance of National Government: UkrainianGovernment
Relation: National Government governs State

Figure 8: Example of an Actor with sub-actors, annotations, relations and instances.

3.1.3. Objects

The concept *Object* represents classes of things, which are relevant in the policy model such as groupings, institutions or artefacts (e.g. "House", "Material") as well as virtual helper objects (e.g. "Heat Delivery"). The concept *Object* is a representation of a class of objects in the policy case, i.e. all entities that cannot take actions. Hence actions are not linked with an object class.



Objects
Object: Energy Resource
Object: Energy
Object: Heat Distribution Network
Object: Agreement
Object: Alternative Heating Solutions
Object: Building
Object: Certificate
Object: MeansOfConveyence
Object: ActionPlan
Object: Well
Object: Heat Delivery
Object: State, Region or City

Figure 9: Example for Objects: Energy Resource, Energy, Heat Delivery, etc.

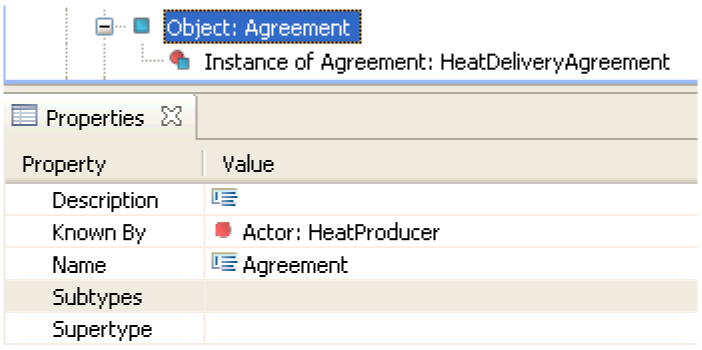
How to add an Object entity into the CCD model?

To add an object, select the CCD node and select *New Child -> Object*. The list of properties is shown in Table 5.

Table 5: Properties of Objects.

Property	**Description**
Known By	List of type of Actors who know the object. If the list is empty, the object is *globaly* known (i.e. by all actors).
Description	A description of the *Actor*.
Name	Name of the *Actor*. Also name of the generated java agent classes.

Figure 10 shows an example of an *Object* with name "Agreement". This Agreement is only known by the type of Actor "HeatProducer".


Object: Agreement
Instance of Agreement: HeatDeliveryAgreement

Properties

Property	Value
Description	
Known By	Actor: HeatProducer
Name	Agreement
Subtypes	
Supertype	

Figure 10: Example of an *Object*.

The list of possible sub nodes is the same as for *Actors* (see in Table 4) with one difference: it is only possible to add sub *Objects* and not sub *Actors*.

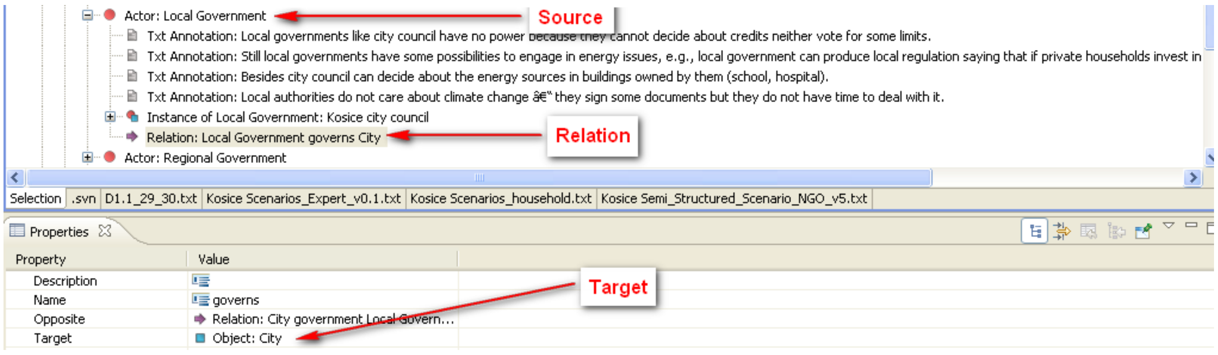
A diagram visualises and supports editing of actors, objects and their relations and attributes: see in section 3.1.9.1.

3.1.4. Relations

A *Relation* describes a conceptual relationship between two actors, two objects or an actor and an object. A relation is directed. Therefore a Relation has to have a *Source Concept* and a *Target Concept*.

How to add a Relation entity into the CCD model?

To create a *Relation*, select the appropriate source node in the CCD with the right mouse button, select *New Child -> Relation*. Each relation needs to have a source concept, which is automatically the concept under which the relation hangs, and a target concept. To define the target concept of the relation, select it under *Target* in the *Property View*, as it is presented in Figure 11.


Actor: Local Government
Txt Annotation: Local governments like city council have no power because they cannot decide about credits neither vote for some limits.
Txt Annotation: Still local governments have some possibilities to engage in energy issues, e.g., local government can produce local regulation saying that if private households invest in
Txt Annotation: Besides city council can decide about the energy sources in buildings owned by them (school, hospital).
Txt Annotation: Local authorities do not care about climate change â€" they sign some documents but they do not have time to deal with it.
Instance of Local Government: Kosice city council
Relation: Local Government governs City
Actor: Regional Government

Selection | .svn | D1.1_29_30.txt | Kosice Scenarios_Expert_v0.1.txt | Kosice Scenarios_household.txt | Kosice Semi_Structured_Scenario_NGO_v5.txt

Properties

Property	Value
Description	
Name	governs
Opposite	Relation: City government Local Govern…
Target	Object: City

Figure 11: Example: *Actor*: Local Government governs *Object*: City.

The following properties can be set for relations in the *Properties View* (select the appropriate *Relation* node, click on the right mouse button and select *Show Properties View*):

Table 6: Properties of Relations.

Property	**Description**
Description	A description of the *Actor*.
Known By	List of type of Actors who know the relation. If the list is empty, the relation is *global* known (i.e. by all actors).
Name	Name of the *Actor*. Also name of the generated java agent classes.
Opposite	A Relation can have an opposite relation, which means that this is the symmetric relation. To set an opposite relation, select it under *Opposite* in the *Properties View*.
Target	Target concept of the relation.

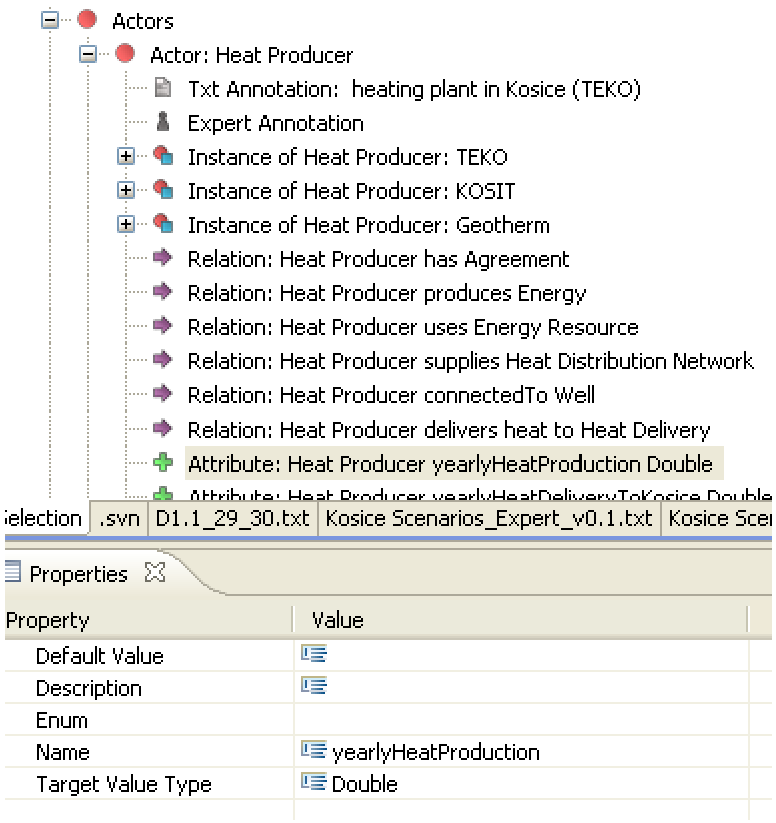
A special type of *Relations* are *Is-a Relations* that are describing subtype relations (see sub actors and sub objects in section 3.1.2 and section 3.1.3). They are possible between actors and between objects. In order to create them, click on an Actor and then right mouse button *New Child* -> *Actor* or click on an Object and then right mouse button *New Child* -> *Object*.

Example: Local Government is-a Government

A relation can have as sub nodes *Txt Annotations, Pdf Annotations*, and *Expert Annotations*.

3.1.5. Attributes

Attributes describe characteristics of concepts *Actor* or *Object*, which can be represented with a primitive value type (i.e. an enumeration, integer, double, string or Boolean). Example of an *Attribute*, i.e. the "yearly heat production" as an attribute of the "Heat Producer" *Actor*, is presented in Figure 12.



Actors
Actor: Heat Producer
Txt Annotation: heating plant in Kosice (TEKO)
Expert Annotation
Instance of Heat Producer: TEKO
Instance of Heat Producer: KOSIT
Instance of Heat Producer: Geotherm
Relation: Heat Producer has Agreement
Relation: Heat Producer produces Energy
Relation: Heat Producer uses Energy Resource
Relation: Heat Producer supplies Heat Distribution Network
Relation: Heat Producer connectedTo Well
Relation: Heat Producer delivers heat to Heat Delivery
Attribute: Heat Producer yearlyHeatProduction Double

Property	Value
Default Value	
Description	
Enum	
Name	yearlyHeatProduction
Target Value Type	Double

Figure 12: Example: Heat Producer with Attribute yearly heat production.

How to add an Attribute entity into the CCD model?

To create an *Attribute*, select the appropriate super node in the CCD with the right mouse button, select *New Child -> Attribute*.

The following properties can be set for attributes in the *Properties View* (select the appropriate *Attribute* node, click on the right mouse button and select *Show Properties View*):

Table 7: Properties of Attributes.

Property	**Description**
Default Value	Default value of the attribute.
Description	A description of the *Attribute*.
Enum	If the *Target Value Type* is *Enum*, select the appropriate *Enum* here. See in section 3.1.6.
Name	Name of the *Attribute*.
Target Value Type	An *Attribute* needs to have a target value type. Here you can choose between *Enum* for an enumeration value, *Integer*, *Double*, *Boolean* and *String*. If you have selected one of the last three values types, nothing else is to do. If you select *Enum*, it is necessary to select the appropriate *Enum* under the corresponding name in the property view.

An *Attribute* can have as sub nodes *Txt Annotations, Pdf Annotations*, and *Expert Annotations*.

3.1.6. Enums and Literals

An *Enum* represents an enumeration as a primitive value type.

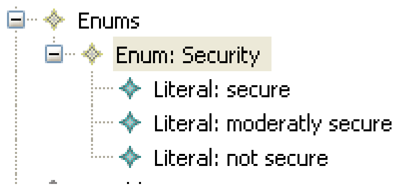

Enums
Enum: Security
Literal: secure
Literal: moderatly secure
Literal: not secure

Figure 13: Example: Enum Security.

How to add Enum and Literal entities into the CCD model?

To add a new *Enum* select the top level node and "*New Child* -> *New Enum*".

To fill the *Enum* with a list of values, add a *Literal* for each value. Select the appropriate *Enum* node and "*New Child* -> *New Literal"*.

The following properties can be set for Enums or Literals in the *Properties View* (select the appropriate node, click on the right mouse button and select *Show Properties View*):

Table 8: Properties of Enums and Literals.

Property	**Description**
Description	A description of the *Enum/Literal*.
Name	Name of the *Enum/Literal*.

An *Enum* or *Literal* can have as sub nodes *Txt Annotations*, *Pdf Annotations*, and *Expert Annotations*.

3.1.7. Instances with Relation Instances and Attribute Instances

Instances for concepts *Actor* or *Object* describe concrete occurrences / instantiations of a concept.

How to add an Actor entity into the CCD model?

In order to create an *Instance*, select the appropriate super node (a concrete *Actor* or *Object*) in the CCD with the right mouse button, select *New Child* -> *Instance*.

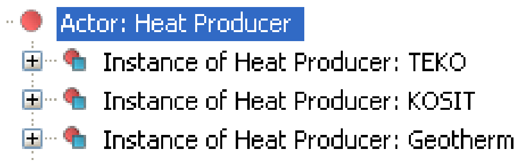

Actor: Heat Producer
Instance of Heat Producer: TEKO
Instance of Heat Producer: KOSIT
Instance of Heat Producer: Geotherm

Figure 14: Example: Instances of Heat Producer: TEKO, KOSIT, and Geotherm.

Properties outlined in Table 9 can be set for *Instances* in the *Properties View*. To do so, please select the appropriate node, click on the right mouse button and select *Show Properties View*.

Table 9: Properties of Instances.

Property	Description
Description	A description of the *Enum/Literal*.
Name	Name of the *Enum/Literal*.
Number of Instances	Default number of instances is 1. But here the number of instances can be varied as e.g. necessary for Actor Household. Code generation automatically creates the number of facts in the fact basis.

The possible types of sub nodes of an *Instance* are listed in Table 10. To add them select the appropriate *Instance* node, click on the right mouse button and select *New Child -> <Type of Subnode>* (not possible with Txt Annotation or Pdf Annotation, because those can be only added by using the annotation feature, see in section 3.2.

Table 10: Possible types of sub nodes of an Instance node.

Type of Subnode	Description
Txt Annotation or Pdf Annotation	Textual annotation which references a text phrase in a document. Double click on this node opens the appropriate text phrase.
Expert Annotation	Expert annotation which gives evidence for this node based on an expert.
Relation Instance	A *Relation Instance* presents a concrete relationship or with other words an instantiation of a relation between two instances. A *Relation Instance* has to have a *Relation*, a *source* instance and a *target* instance.
Attribute Instance:	An *Attribute Instance* presents a concrete characteristic of an instance. An *Attribute Instance* has to have an *Attribute*, a *source* instance and a *value*..

3.1.7.1. *Relation Instance*

The *Relation Instance* describes a relationship between two *Instances* based on a defined conceptual *Relation* between two concepts (of type *Actor* and/or *Object*). As a Relation, a *Relation Instance* is also directed. This means that a *Relation Instance* has to have a *Source Instance* and a *Target Instance*. Figure 15 shows a Relation Instance between Instance “Source Instance” and “Target Instance”.

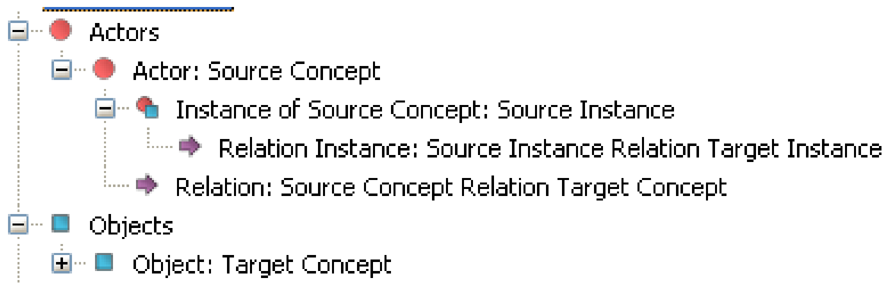

Actors
Actor: Source Concept
Instance of Source Concept: Source Instance
Relation Instance: Source Instance Relation Target Instance
Relation: Source Concept Relation Target Concept
Objects
Object: Target Concept

Figure 15: Example: Relation Instance between "Source Instance" and "Target Instance".

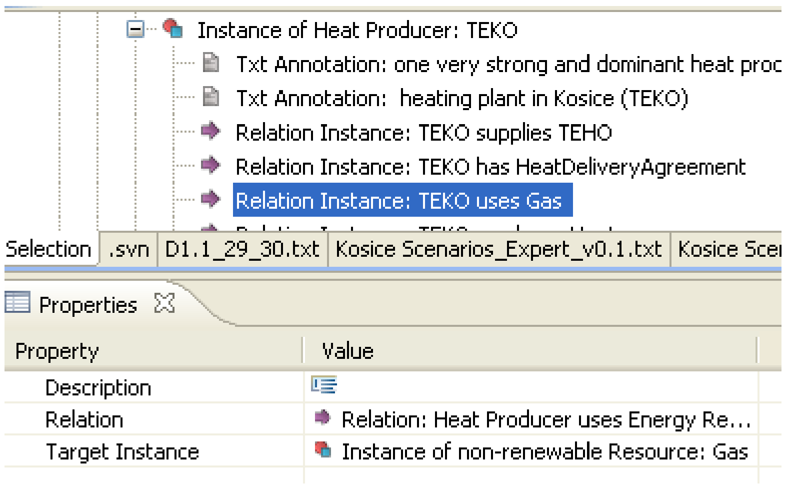
How to add a Relation Instance entity into the CCD model?

To create a *Relation Instance* select the appropriate Instance and then click right mouse button *New Child -> Relation Instance*. This defines the source of the *Relation Instance* entity.

Define then the *Relation* and the *Target Instance* in the *Properties View*:

Table 11: Properties of Relation Instances.

Property	**Description**
Description	A description of the *Enum/Literal*.
Relation	The conceptual base relation. The source instance of this relation must be of the same concept as the source concept of the underlying relation.
Target Instance	The target instance of the relation. The target instance of this relation must be of the same concept as the target concept of the underlying relation.



Instance of Heat Producer: TEKO
Txt Annotation: one very strong and dominant heat proc
Txt Annotation: heating plant in Kosice (TEKO)
Relation Instance: TEKO supplies TEHO
Relation Instance: TEKO has HeatDeliveryAgreement
Relation Instance: TEKO uses Gas

Selection | .svn | D1.1_29_30.txt | Kosice Scenarios_Expert_v0.1.txt | Kosice Scer

Properties

Property	Value
Description	
Relation	Relation: Heat Producer uses Energy Re...
Target Instance	Instance of non-renewable Resource: Gas

Figure 16: Example: TEKO uses GAS.

3.1.7.2. Attribute Instance

The *Attribute Instance* presents a concrete characteristic of an instance. An *Attribute Instance* has to have an *Attribute*, a *source* instance and a *value*.

How to add an Attribute Instance entity into the CCD model?

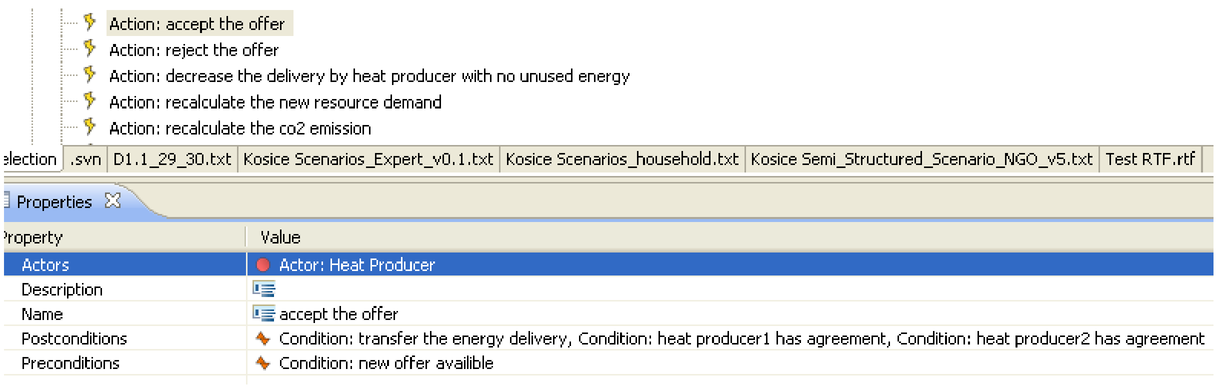
To create an Attribute Instance select the appropriate Instance and then click right mouse button *New Child -> Attribute Instance*. This defines the source of the *Attribute Instance* entity. Define then the *attribute* and the *value* in the *Properties View*.

3.1.8. Actions and Conditions

The *Action* is a representation of tasks or actions in the policy case. Actions are usually executed by actors and pursue an objective. In our concept, we also model natural events and consequences, e.g. “*it rains*” and as consequence “*it is wet*”. With such actions, no actor is affiliated.

How to add an Action entity into the CCD model?

To create an *Action* select the CCD node then click right mouse button *New Child ->Action* or use the *Actions Diagram* (see in section 3.1.9.2).



Action: accept the offer
Action: reject the offer
Action: decrease the delivery by heat producer with no unused energy
Action: recalculate the new resource demand
Action: recalculate the co2 emission

election | .svn | D1.1_29_30.txt | Kosice Scenarios_Expert_v0.1.txt | Kosice Scenarios_household.txt | Kosice Semi_Structured_Scenario_NGO_v5.txt | Test RTF.rtf

Properties

Property	Value
Actors	Actor: Heat Producer
Description	
Name	accept the offer
Postconditions	Condition: transfer the energy delivery, Condition: heat producer1 has agreement, Condition: heat producer2 has agreement
Preconditions	Condition: new offer availible

Figure 17: Example: Action "accept the offer".

The following properties can be set for *Actions* in the *Properties View* (select the appropriate node, click on the right mouse button and select *Show Properties View*):

Table 12: Properties of Actions.

Property	**Description**
Actors	The actors, if available, which perform the action.
Description	A description of the *Action*
Name	Name of the *Action*.
Postconditions	Each Action can have several post-conditions represented by *Condition* objects. A post-condition can be understood as the consequence of the action. The post-condition can be a pre-condition of another action. See Figure 18 for a visualisation of pre- and post conditions relations.
Preconditions	Each Action can have several preconditions represented by *Condition* objects. A precondition is meant as a condition that results in the performance of the action. The pre-condition can be a post-condition of another action. See Figure 18 for a visualisation of pre- and post conditions relations.

The *Condition* is a representation of an pre-condition (or post-condition (meant as consequence) of an action. A *Condition* as post-condition of one action can be used as pre-condition for another action. This way, behaviour of and interaction between actors are modelled. *Conditions* describe only one particular characteristic on an element.

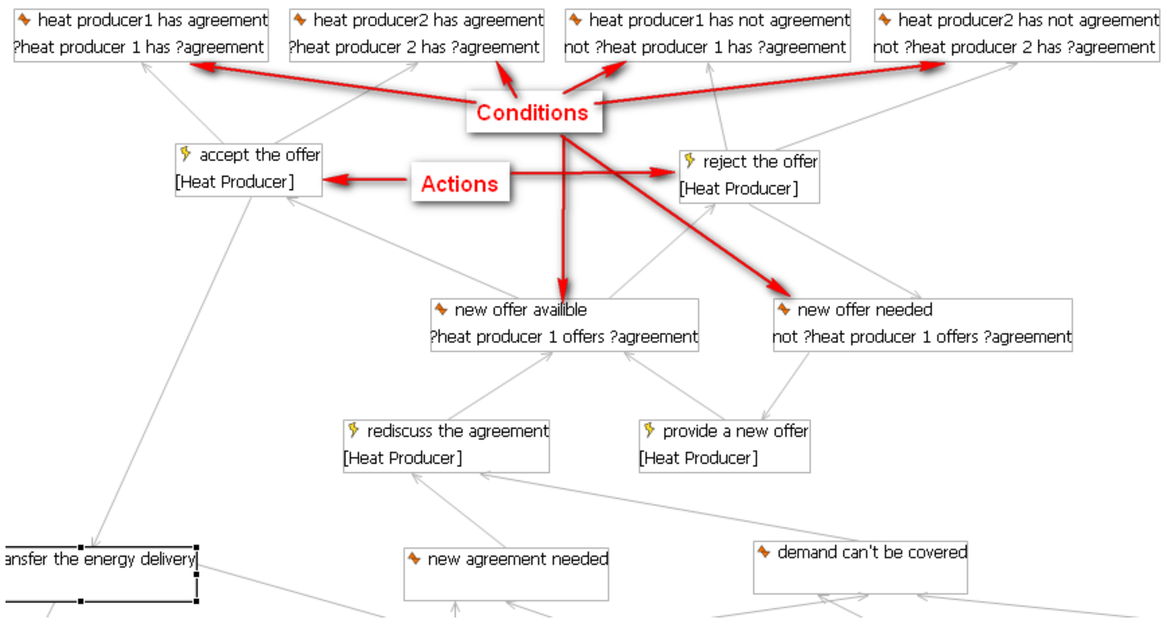
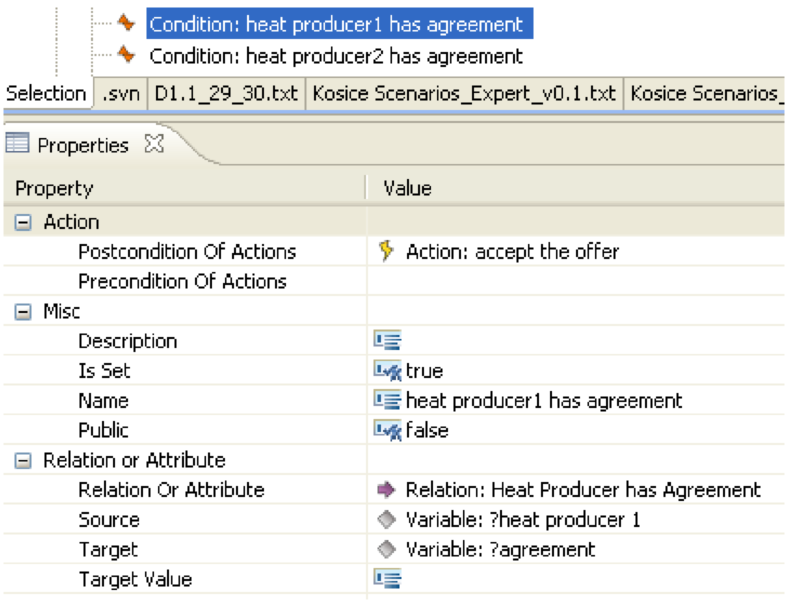

heat producer1 has agreement
?heat producer 1 has ?agreement
heat producer2 has agreement
?heat producer 2 has ?agreement
heat producer1 has not agreement
not ?heat producer 1 has ?agreement
heat producer2 has not agreement
not ?heat producer 2 has ?agreement
Conditions
accept the offer
[Heat Producer]
Actions
reject the offer
[Heat Producer]
new offer availible
?heat producer 1 offers ?agreement
new offer needed
not ?heat producer 1 offers ?agreement
rediscuss the agreement
[Heat Producer]
provide a new offer
[Heat Producer]
ansfer the energy delivery
new agreement needed
demand can't be covered

Figure 18: Action with pre-conditions and post-conditions.

Figure 18 shows an actions diagram that visualises the different elements described above. The figure shows a number of actions and how different *Conditions* build a flow between these elements. For example, Condition "new offer available" is a pre-condition of action "accept the offer" while "heat provider 1/2 has agreement" are post-conditions of the same action. Another pre-condition is still needed to differentiate when action "accept the offer" or action "reject the offer" is performed.

How to add a Condition entity into the CCD model?

To create a *Condition* select the CCD node then click right mouse button *New Child ->Condition* or use the *Actions Diagram* (see in section 3.1.9.2).


Condition: heat producer1 has agreement
Condition: heat producer2 has agreement

Selection | .svn | D1.1_29_30.txt | Kosice Scenarios_Expert_v0.1.txt | Kosice Scenarios_

Properties

Property	Value
Action	
Postcondition Of Actions	Action: accept the offer
Precondition Of Actions	
Misc	
Description	
Is Set	true
Name	heat producer1 has agreement
Public	false
Relation or Attribute	
Relation Or Attribute	Relation: Heat Producer has Agreement
Source	Variable: ?heat producer 1
Target	Variable: ?agreement
Target Value	

Figure 19: Example for a Condition: Heat producer 1 has agreement.

The following properties can be set for *Conditions* in the *Properties View* (select the appropriate node, click on the right mouse button and select *Show Properties View*):

Table 13: Properties of Conditions.

Property	**Description**
Postcondition of Actions	Each Action can have several post-conditions represented by *Condition* entities. A post-condition can be understood as the consequence of the action. The post-condition can be a pre-condition of another action. See Figure 18 for a visualisation of pre- and post conditions relations.
Precondition of Actions	Each Action can have several preconditions represented by *Condition* entities. A precondition is meant as a condition that results in the performance of the action. The pre-condition can be a post-condition of another action. See Figure 18 for a visualisation of pre- and post conditions relations.
Description	A description of the *Condition*
Is Set	If the *Condition* is set or not.
Name	Name of the *Condition*.
Public	*Not clear if this is necessary*.
Relation or Attribute	The underlying relation or attribute of the condition. With the relation a relationship between instances and/or variables can be set as condition that an action is performed or the consequence of an action.. Example: “?heat producer 1” “has” “?agreement”
Source	Source *Instance* or *Variable* of the condition.
Target	Target *Instance* or *Variable or Literal* of the condition.
Target Value	*Target Value* (String, Double, Integer) of the condition (if an *Attribute* is selected under *Relation or Attribute*)

The ◆ *Variable* means a representation of objects or actors used as flexible declaration in one or more *conditions*. To create a *Variable* select the CCD node then click right mouse button *New Child ->Variable.*

A diagram supports the user in editing the actions and their conditions: see section 3.1.9.2.

3.1.9. CCD Diagrams

This section presents available diagrams in the CCD Tool.

3.1.9.1. Actor Network Diagram

An Actor Network Diagram visualizes the following elements and their relationships of a CCD: actors, objects, attributes and relations.

How to work with the Actor Network Diagram in the CCD model?

In order to open an *actor network diagram* of the CCD, a `*.ccd_diagram` file needs to be created. In order to do so, please select the appropriate ccd file with the right mouse button. Figure 20 shows the appearing context menu.

Select *Initialize ccd_diagram diagram file* to open the wizard that supports you in creating an *"ccd_diagram actor network diagram file"*.

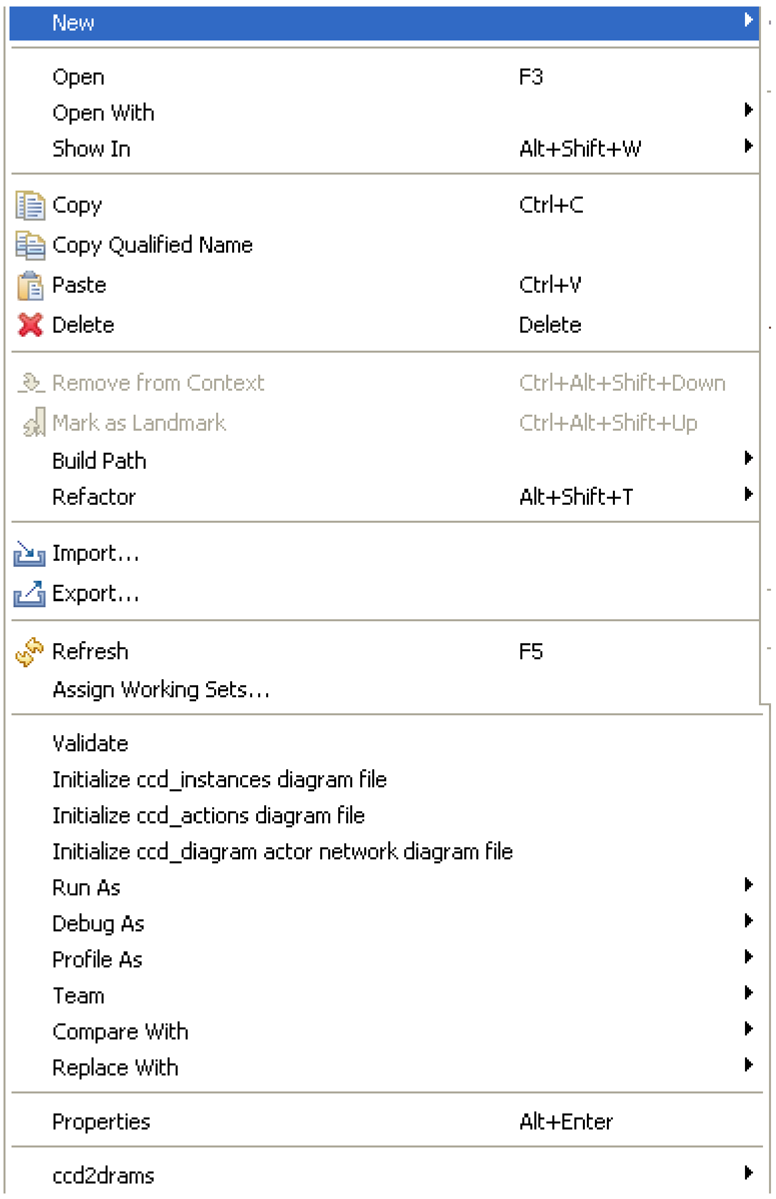


Figure 20: Context menu to create diagram files.

If the wizard appears, only click on *Finish*. A `ccd_diagram` file is created and the CCD "Diagram Editor "appears (if not double-click on the `ccd_diagram` file).

An example of an actor network diagram is depicted in Figure 21.

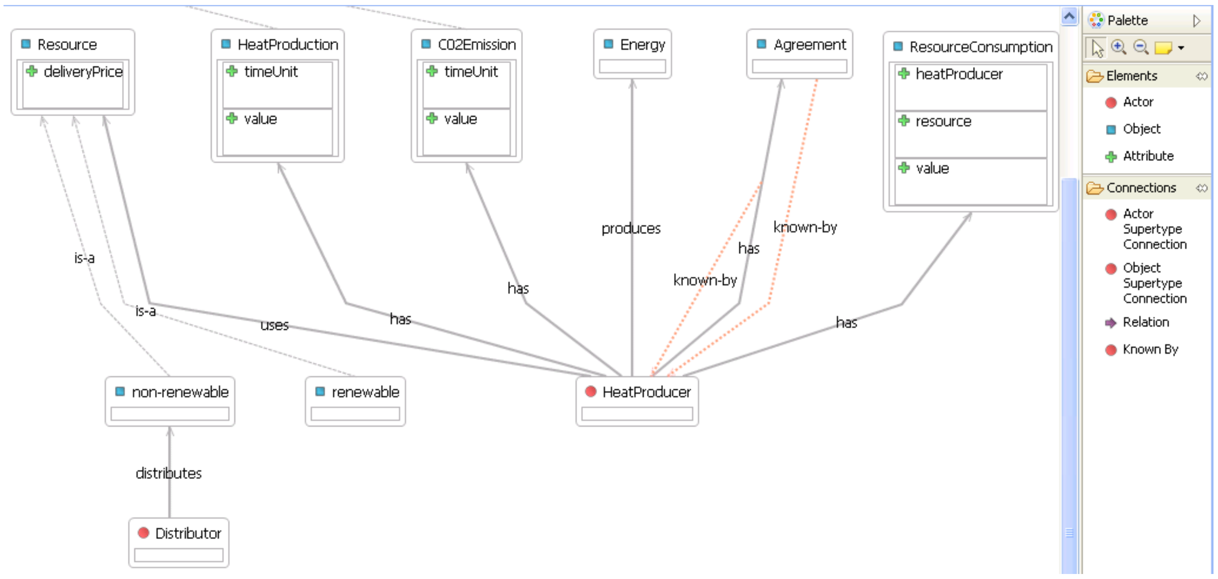


Figure 21: Add Example for Actor Network Diagram.

The “Diagram Editor” shows a diagram of actors, objects and relations created in the CCD file. You can add new actors and objects by clicking on the appropriate element in the palette and then clicking on the drawing area. *Objects* are visualized using blue rectangles and *Actors* using red circles as icons.

To create a connection, select the appropriate connection (*Relation*, *Actor Supertype*, *Object Supertype*, *Known-By*), click on the source node in the area and draw it to the target node. *Supertype* connections are visualised with a dashed line, *Relations* with a solid line, *Known By* connections with a dotted line. In difference to other connections, the *Known By* connections can also be drawn from *Relations* to *Actors* (cf. the property *Known-By* in section 3.1.4 “Relations”). In Figure 21 a *Known By* connection is visible between the *Relation* “has” and the *Actor* “Heat Producer”. To create such a connection, select a *Known By* connection in the palette, click on the *Relation* in the drawing area and draw it to the target *Actor* node.

An *Attribute* can be created, by clicking on and *Attribute* in the palette and clicking afterwards in the inner box of an *Actor* or an *Object*.

It is possible to minimise the box showing the attributes of an actor or an object. If you select an actor or an object, a minus is visible in the left top corner of the box showing the attributes. Click on the minus and the box is minimising. Click on the plus and the box and the box is maximising again. Figure 22 shows a maximised and a minimised box.

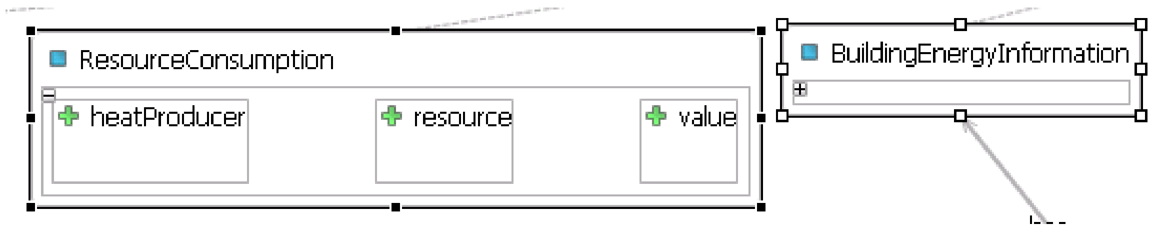


Figure 22: Minimising and maximising the box showing the attributes.

3.1.9.2. Actions Diagram

An Actions Diagram visualizes actions and conditions and their connections.

How to work with the Actions Diagram in the CCD model?

Select *Initialize ccd_actions diagram file* to open the wizard that supports you in creating an actions diagram file. If the wizard appears, only click on *Finish*. A `ccd_actions` file is created and the CCD Actions Diagram Editor appears (if not double-click on the `ccd_actions` file).

The CCD “Actions Editor” shows a diagram of *Actions* and *Conditions* created in the CCD file. You can add new *Action* or *Condition* by clicking on the appropriate entity in the palette and then clicking on the drawing area. To create a connection, select the appropriate connection, click on the source node in the area and draw it to the target node. An example for an “Actions Diagram” is presented in Figure 23.

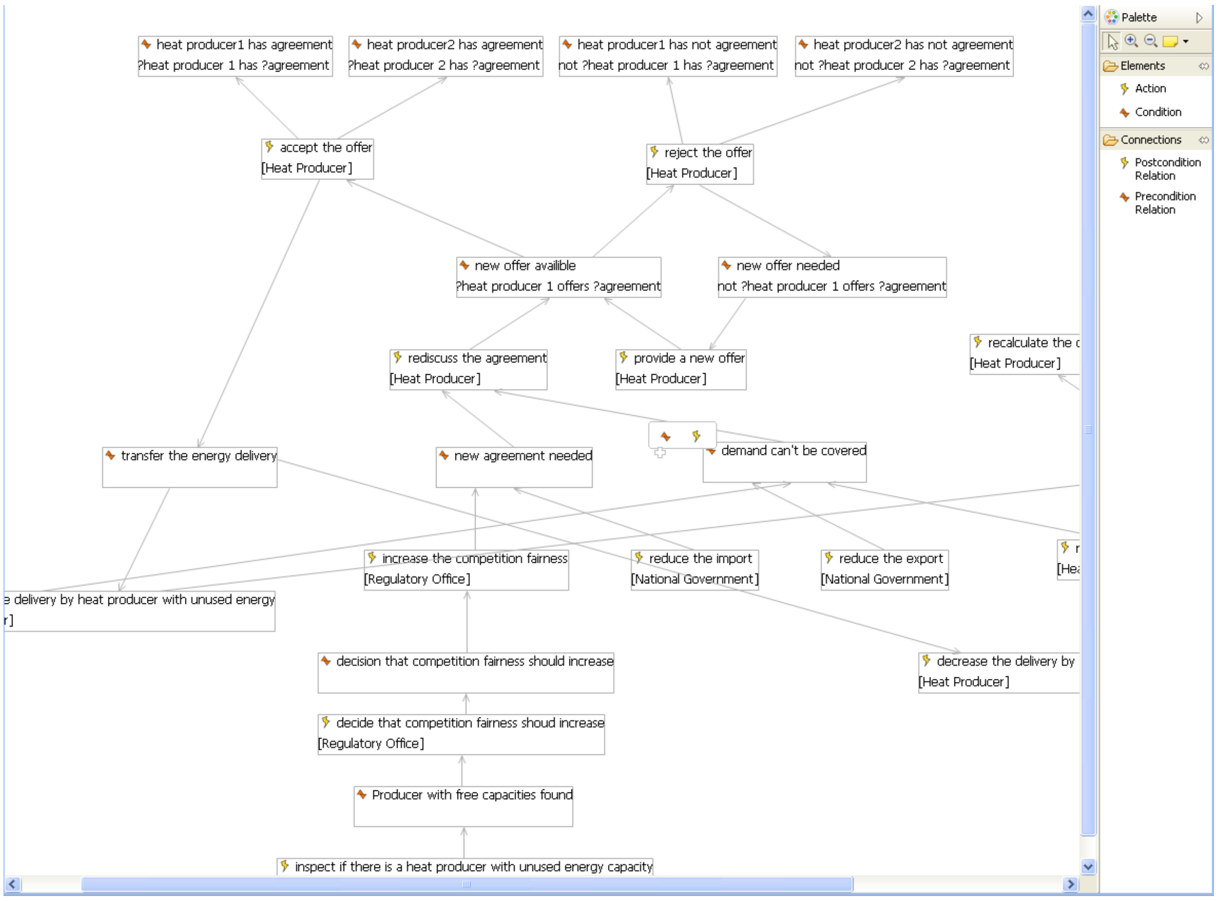

heat producer1 has agreement
?heat producer 1 has ?agreement
heat producer2 has agreement
?heat producer 2 has ?agreement
heat producer1 has not agreement
not ?heat producer 1 has ?agreement
heat producer2 has not agreement
not ?heat producer 2 has ?agreement
accept the offer
[Heat Producer]
reject the offer
[Heat Producer]
new offer availible
?heat producer 1 offers ?agreement
new offer needed
not ?heat producer 1 offers ?agreement
rediscuss the agreement
[Heat Producer]
provide a new offer
[Heat Producer]
recalculate the c
[Heat Producer]
transfer the energy delivery
new agreement needed
demand can't be covered
increase the competition fairness
[Regulatory Office]
reduce the import
[National Government]
reduce the export
[National Government]
e delivery by heat producer with unused energy
r]
decision that competition fairness should increase
decrease the delivery by
[Heat Producer]
decide that competition fairness shoud increase
[Regulatory Office]
Producer with free capacities found
inspect if there is a heat producer with unused energy capacity
Palette
Elements
Action
Condition
Connections
Postcondition Relation
Precondition Relation

Figure 23: Example for an Action Diagram.

3.1.9.3. Instances Diagram

How to work with the Instances Diagram in the CCD model?

Select *Initialize ccd_instances diagram file* to open the wizard that supports you in creating an actions diagram file. If the wizard appears, only click on *Finish*. A `ccd_instances` file is created and the CCD Actions Diagram Editor appears (it not double-click on the `ccd_instances` file).

The CCD "Instances Editor" shows a diagram of *Instances* and *Instance Relations* (i.e. relations between these instances) created in the CCD file. Each node visualising an Instance shows the concept of an instance in italic font as e.g. *KOSIT* is of type *Heat Producer*. You can add a new *Relation Instance (i.e. a relation between existent instances*) by selecting the appropriate connection, click on the source node in the area and draw it to the target node. Instances cannot be added in the diagram. This is only possible in the CCD Editor (Tree).

An example for an Instances Diagram is presented in Figure 24.

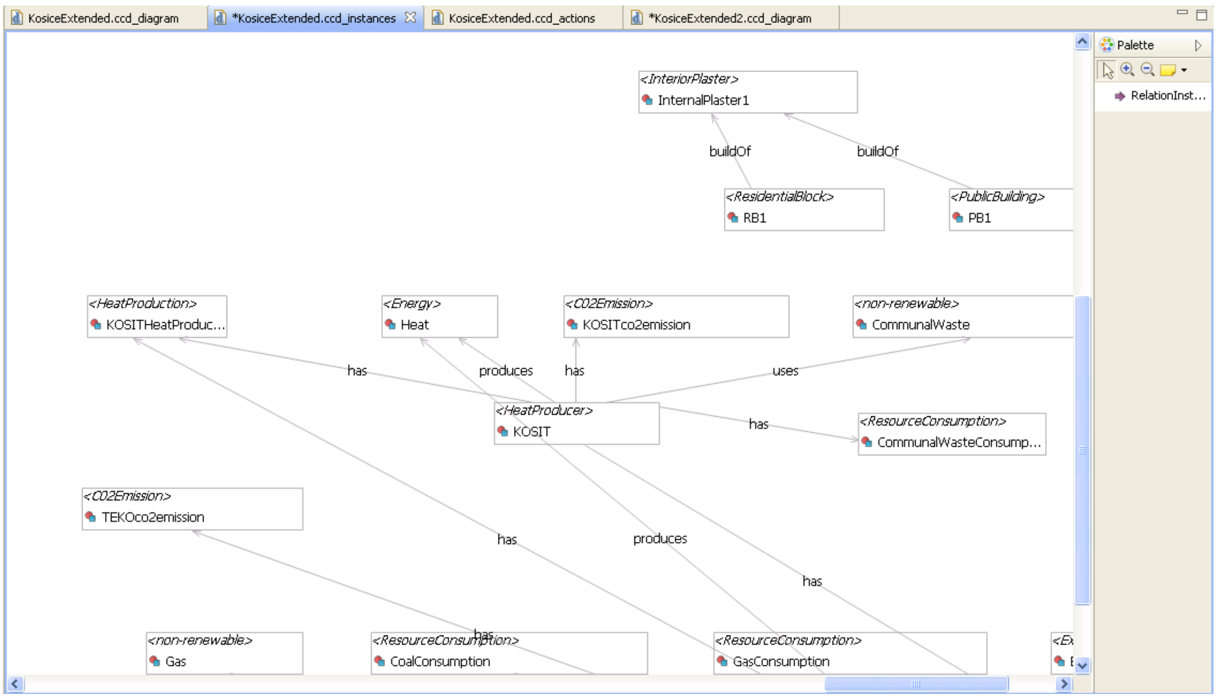


Figure 24: Example for an Instances Diagram.

3.1.9.4. Filtering Diagram

How to work with the Filtering Diagram in the CCD model?

It is possible to filter the diagrams for selected nodes or edges as well as for nodes and edges of a certain type (as e.g. Actor or Relation):

- Filter selected nodes or edges: Select the nodes or edges which you want to hide. Click the right mouse button and select Filters -> Hide Selection (see Figure 25).
- Filter nodes or edges of a certain type (as e.g. Actor or Relation): Select a node or edge of the type you want to hide, e.g. select an actor. Click the right mouse button and select *Filters -> Hide VisualType* (see Figure 25).
- Hide unconnected elements: It is possible in the diagrams to select one or more nodes and hide all unconnected nodes (see Figure 26). With "arrange all" it is possible to arrange the remaining elements. To show again all elements please click "Show hidden elements".

Note: The *Hide SemanticModel* only hides the selected node.

To bring hidden elements back, click the right mouse button somewhere in the diagram and select *Filters -> Show All Hidden Parts*.

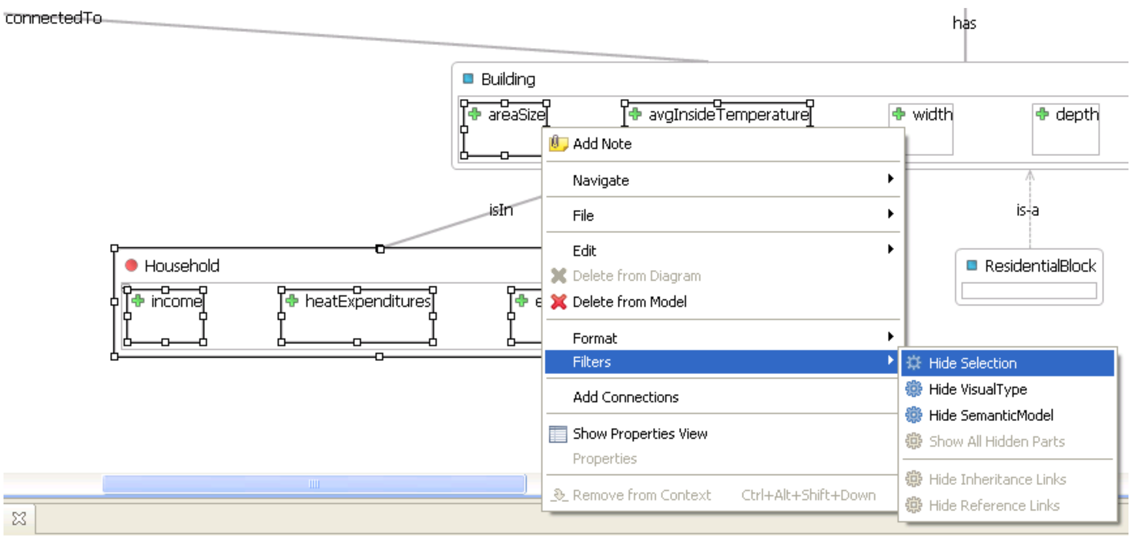


Figure 25: Filtering context menu.

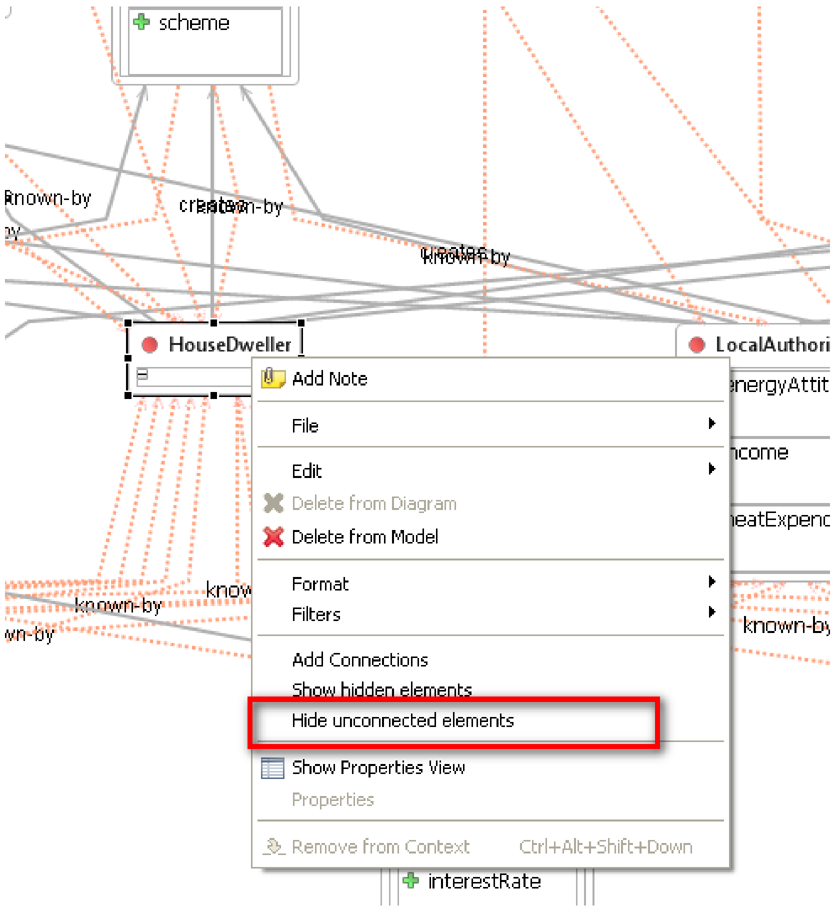


Figure 26: Hide unconnected elements.

3.2. TXT ANNOTATOR

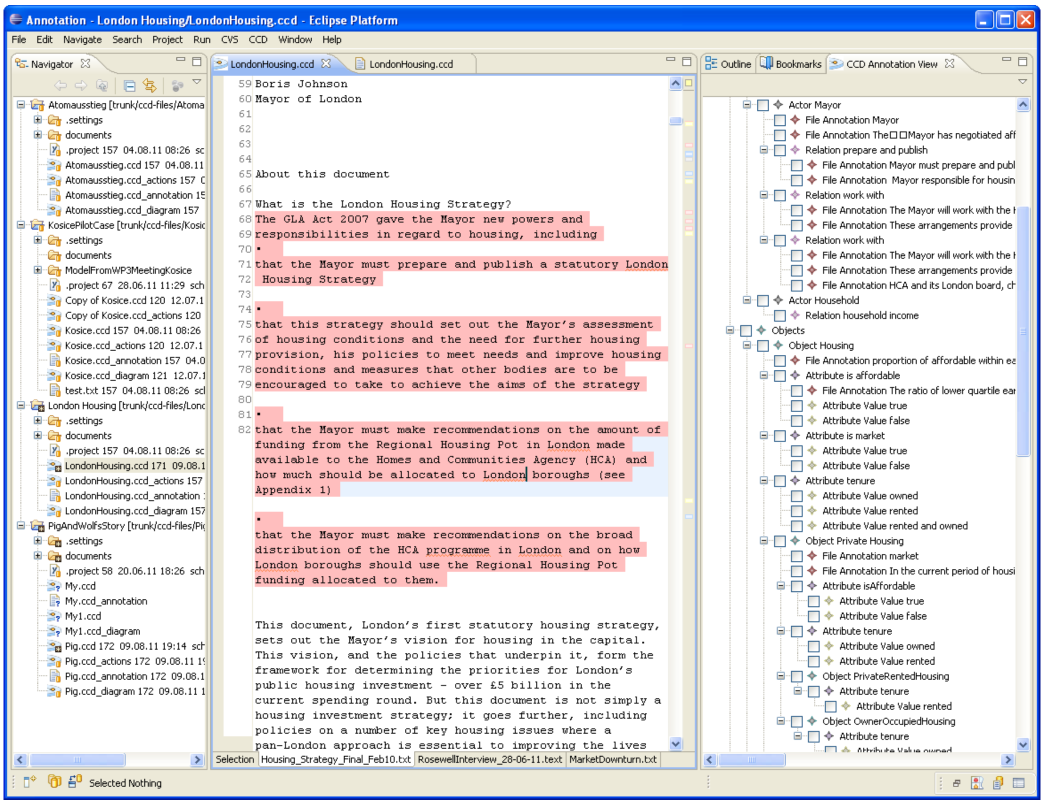
How to open a plain text document for annotations?

The *Txt Annotator* plug-in allows the creation of links between CCD model elements and text fragments of the documents in plain text format - both local files and external TXT documents (background materials) downloaded from the Alfresco content repository. To open a plain text document for annotations, please follow these steps:

1. Make sure that your project is correctly set and the CCD file appears in the root of your project. Activate the *Annotation Editor* and *CCD Annotation view* by clicking with the right mouse button on the CCD file that will be used for annotations. In the invoked dialog, select the *Open With → CCD Annotation Editor* command. The CCD file will be opened in the *Annotation Editor* view and in the *CCD Annotation view* as well.
2. Navigate to the plain text document you want to annotate - in *Content View* for documents downloaded from Alfresco, or in the project folder of *Package Explorer* for local TXT files. Double-click on the plain text document to open it in the respective tab of the *CCD Annotation Editor* view - annotation perspective, as it is depicted in Figure 27.

Note: To **open a new local text file** in the *Annotation Editor*, click on *CCD -> Open File* in the Eclipse menu. In the appearing file dialog, choose a text file from your local hard drive. The text file is loaded into the documents folder in the active project. To ensure a consistency of annotations, please never change this folder of local files. The same procedure of uploading new local files for annotations can be used for any file format - however, only TXT, PDF, and HTML formats are supported for CCD model annotations (see also sections 3.3 and 3.4 - description of PDF and HTML annotators).

The CCD Annotation Editor opens those files that are already annotated in the tabs.



```
Atomausstieg [trunk/ccd-files/Atoma
    .settings
    documents
    .project 157  04.08.11 08:26  sc
    Atomausstieg.ccd 157  04.08.11
    Atomausstieg.ccd_actions 157  C
    Atomausstieg.ccd_annotation 15
    Atomausstieg.ccd_diagram 157
KosicePilotCase [trunk/ccd-files/Kosic
    .settings
    documents
    ModelFromWP3MeetingKosice
    .project 67  28.06.11 11:29  sch
    Copy of Kosice.ccd 120  12.07.1
    Copy of Kosice.ccd_actions 120
    Kosice.ccd 157  04.08.11 08:26
    Kosice.ccd_actions 120  12.07.1
    Kosice.ccd_annotation 157  04.0
    Kosice.ccd_diagram 121  12.07.1
    test.txt 157  04.08.11 08:26  scl
London Housing [trunk/ccd-files/Lonc
    .settings
    documents
    .project 157  04.08.11 08:26  sc
    LondonHousing.ccd 171  09.08.1
    LondonHousing.ccd_actions 157
    LondonHousing.ccd_annotation :
    LondonHousing.ccd_diagram 157
PigAndWolfsStory [trunk/ccd-files/Pi
    .settings
    documents
    .project 58  20.06.11 18:26  sch
    My.ccd
    My.ccd_annotation
    My1.ccd
    My1.ccd_diagram
    Pig.ccd 172  09.08.11 19:14  sch
    Pig.ccd_actions 172  09.08.11 1
    Pig.ccd_annotation 172  09.08.1
    Pig.ccd_diagram 172  09.08.11 1
```

```
Boris Johnson
Mayor of London




About this document

What is the London Housing Strategy?
The GLA Act 2007 gave the Mayor new powers and
responsibilities in regard to housing, including
•
that the Mayor must prepare and publish a statutory London
 Housing Strategy

•
that this strategy should set out the Mayor's assessment
of housing conditions and the need for further housing
provision, his policies to meet needs and improve housing
conditions and measures that other bodies are to be
encouraged to take to achieve the aims of the strategy

•
that the Mayor must make recommendations on the amount of
funding from the Regional Housing Pot in London made
available to the Homes and Communities Agency (HCA) and
how much should be allocated to London boroughs (see
Appendix 1)

•
that the Mayor must make recommendations on the broad
distribution of the HCA programme in London and on how
London boroughs should use the Regional Housing Pot
funding allocated to them.


This document, London's first statutory housing strategy,
sets out the Mayor's vision for housing in the capital.
This vision, and the policies that underpin it, form the
framework for determining the priorities for London's
public housing investment - over £5 billion in the
current spending round. But this document is not simply a
housing investment strategy; it goes further, including
policies on a number of key housing issues where a
pan-London approach is essential to improving the lives
```

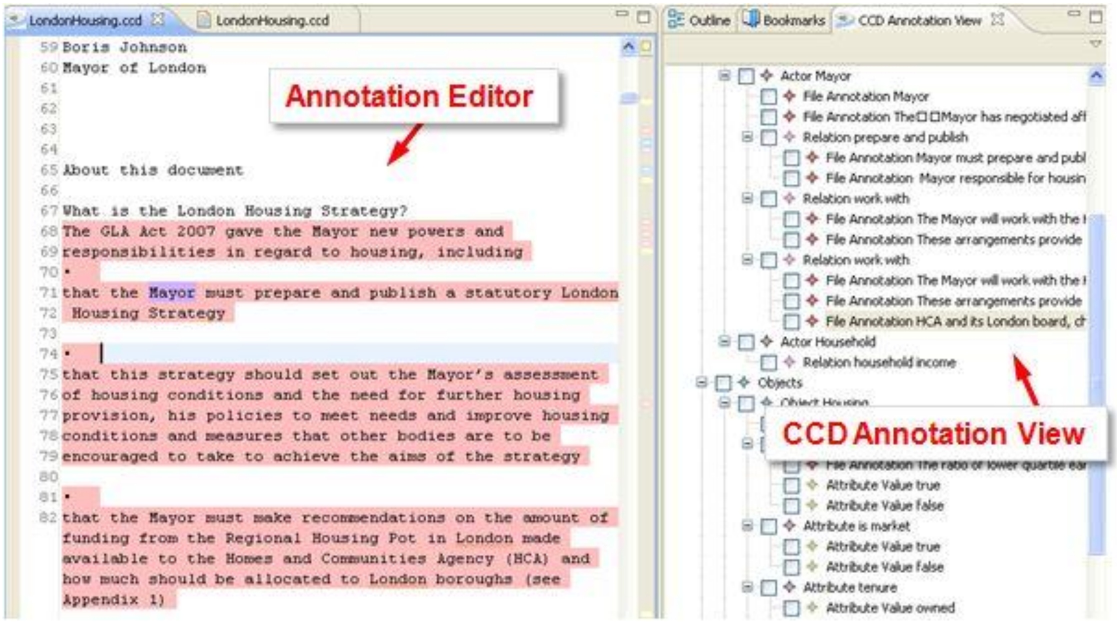
```
    Actor Mayor
        File Annotation Mayor
        File Annotation The□□Mayor has negotiated aff
        Relation prepare and publish
            File Annotation Mayor must prepare and publ
            File Annotation  Mayor responsible for housin
        Relation work with
            File Annotation The Mayor will work with the I
            File Annotation These arrangements provide
        Relation work with
            File Annotation The Mayor will work with the I
            File Annotation These arrangements provide
            File Annotation HCA and its London board, ch
    Actor Household
        Relation household income
Objects
    Object Housing
        File Annotation proportion of affordable within ea
        Attribute is affordable
            File Annotation The ratio of lower quartile ear
            Attribute Value true
            Attribute Value false
        Attribute is market
            Attribute Value true
            Attribute Value false
        Attribute tenure
            Attribute Value owned
            Attribute Value rented
            Attribute Value rented and owned
        Object Private Housing
            File Annotation market
            File Annotation In the current period of housi
            Attribute isAffordable
                Attribute Value true
                Attribute Value false
            Attribute tenure
                Attribute Value owned
                Attribute Value rented
            Object PrivateRentedHousing
                Attribute tenure
                    Attribute Value rented
            Object OwnerOccupiedHousing
                Attribute tenure
                    Attribute Value owned
```

Figure 27: Annotation perspective of a plain text document.

How to annotate a plain text document?

The plain text document is ready for annotations when it is opened in the respective tab of the *Annotation Editor* view and the CCD model structure is available in the *CCD Annotation view*. To create an annotation, please follow these steps:

1. Inside the content of the plain text document, highlight a text portion and drag it on the appropriate CCD node in the *Annotation* view.
2. By dropping it in a node, a *File Annotation* is added into the selected node in the *CCD Annotation view*. If you drop a selected text phrase on one of the nodes *Actors*, *Objects* or *Actions*, a new node of the same type (i.e., *Actor* or *Object* or *Action*) is added. The appropriate text phrase is marked in the text editor. Figure 28 shows an example of the *Annotation Editor* in action.



```
Boris Johnson
Mayor of London



About this document

What is the London Housing Strategy?
The GLA Act 2007 gave the Mayor new powers and
responsibilities in regard to housing, including
•
that the Mayor must prepare and publish a statutory London
 Housing Strategy

•
that this strategy should set out the Mayor's assessment
of housing conditions and the need for further housing
provision, his policies to meet needs and improve housing
conditions and measures that other bodies are to be
encouraged to take to achieve the aims of the strategy

•
that the Mayor must make recommendations on the amount of
funding from the Regional Housing Pot in London made
available to the Homes and Communities Agency (HCA) and
how much should be allocated to London boroughs (see
Appendix 1)
```

```
Actor Mayor
  File Annotation Mayor
  File Annotation The□□Mayor has negotiated aff
  Relation prepare and publish
    File Annotation Mayor must prepare and publ
    File Annotation  Mayor responsible for housin
  Relation work with
    File Annotation The Mayor will work with the I
    File Annotation These arrangements provide
  Relation work with
    File Annotation The Mayor will work with the I
    File Annotation These arrangements provide
    File Annotation HCA and its London board, ch
Actor Household
  Relation household income
Objects
  Object Housing
      File Annotation The ratio of lower quartile ear
      Attribute Value true
      Attribute Value false
    Attribute is market
      Attribute Value true
      Attribute Value false
    Attribute tenure
      Attribute Value owned
```

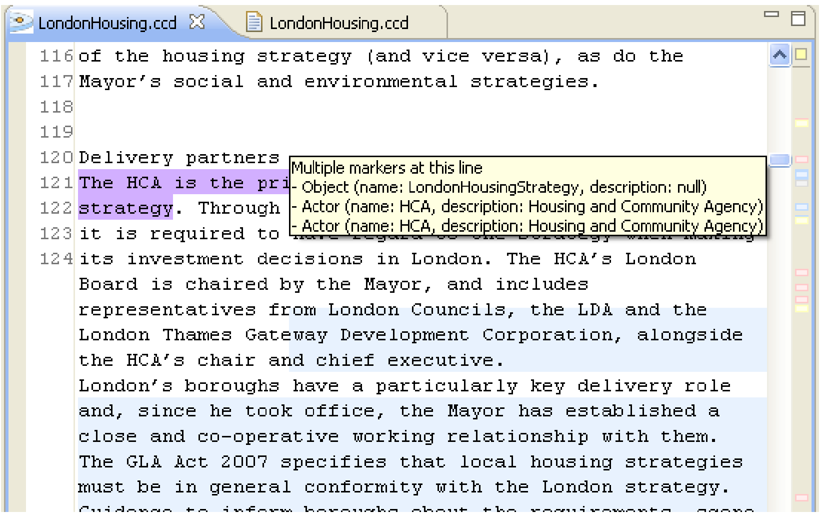
Figure 28: Annotation of a plain text to CCD nodes.

How to navigate through the annotations?

There are two options for navigating through file annotations, namely:

1. From CCD to text:
   - In the *CCD Annotation view*, double click on a *File Annotation* entity. The related annotation in the plain text content will be displayed in the *Annotation Editor*.
2. From text to CCD:
   - In the *Annotation Editor*, click on the small boxes at the right side bar. It will select the appropriate annotation inside the text displayed in the *Annotation Editor*. Mouse over the text shows you a short description of the corresponding annotated object, as it is depicted in Figure 29.

- Click with the mouse into the annotated text portion, then select in the menu the *CCD -> Select Annotation in Annotation View* command. Then the *CCD Annotation view* selects all corresponding annotations in the view (please note: The *CCD Annotation view* must be already visible).



```
of the housing strategy (and vice versa), as do the
Mayor's social and environmental strategies.


Delivery partners
The HCA is the pri
strategy. Through
it is required to
its investment decisions in London. The HCA's London
Board is chaired by the Mayor, and includes
representatives from London Councils, the LDA and the
London Thames Gateway Development Corporation, alongside
the HCA's chair and chief executive.
London's boroughs have a particularly key delivery role
and, since he took office, the Mayor has established a
close and co-operative working relationship with them.
The GLA Act 2007 specifies that local housing strategies
must be in general conformity with the London strategy.
```

```
Multiple markers at this line
- Object (name: LondonHousingStrategy, description: null)
- Actor (name: HCA, description: Housing and Community Agency)
- Actor (name: HCA, description: Housing and Community Agency)
```

Figure 29: Show information about an annotation.

Note: Several texts can be opened in the *Annotation Editor*. For each text a new tab page is added.

Is it possible to add annotations that are not anchored in any text?

Yes, so-called ***Expert Annotations*** can be added as other CCD entities into the CCD model hierarchy displayed in the CCD Annotation view. To add an expert annotation, please follow these instructions:

1. In the tree of *CCD Annotation view*, use the right mouse button to invoke the context menu. Select the *Add Expert Annotation* command.
2. Type the text of the annotation and confirm by OK. The expert annotation will appear in the CCD hierarchy of *CCD Annotation view*.

3.3. PDF ANNOTATOR

How to open a PDF document for annotations?

The *PDF Annotator* plug-in allows to create links between CCD model elements and text fragments of the documents in PDF format - both local files and external PDF documents (background materials) downloaded from the Alfresco content repository. To open a PDF document for annotations, please follow these steps:

1. Make sure that your project is correctly set and the CCD file appears in the root of your project. Activate the *Annotation Editor* and *CCD Annotation view* by clicking with the right mouse button on the CCD file that will be used for annotations. In the invoked dialog, select the *Open With* → *CCD Annotation Editor* command. The CCD file will be opened in the *Annotation Editor* view and in the *CCD Annotation view* as well.

2. Navigate to the PDF document you want to annotate - in *Content View* for documents downloaded from Alfresco, or in the project folder of *Package Explorer* for local PDF files. Double-click on the PDF document to open it in the respective tab of the *CCD Annotation Editor* view.

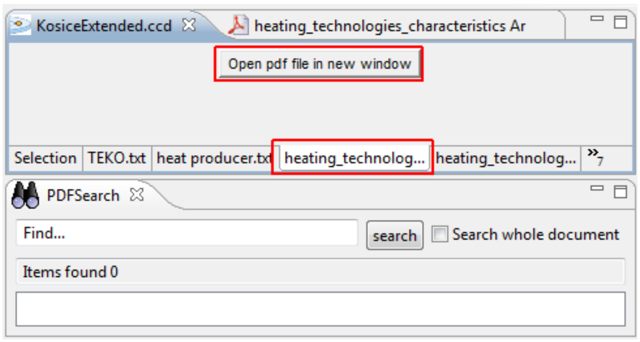


Figure 30: The tab displayed for PDF documents.

Note: The *PDF Annotator* component could consume a big amount of computer's “working memory”. Therefore displaying of PDF file is on demand. The tab of the *CCD Annotation Editor* view, which is related to PDF documents, shows only a button (see in Figure 30) which will open PDF document in new window and Eclipse perspective will be changed as well (see in Figure 31).

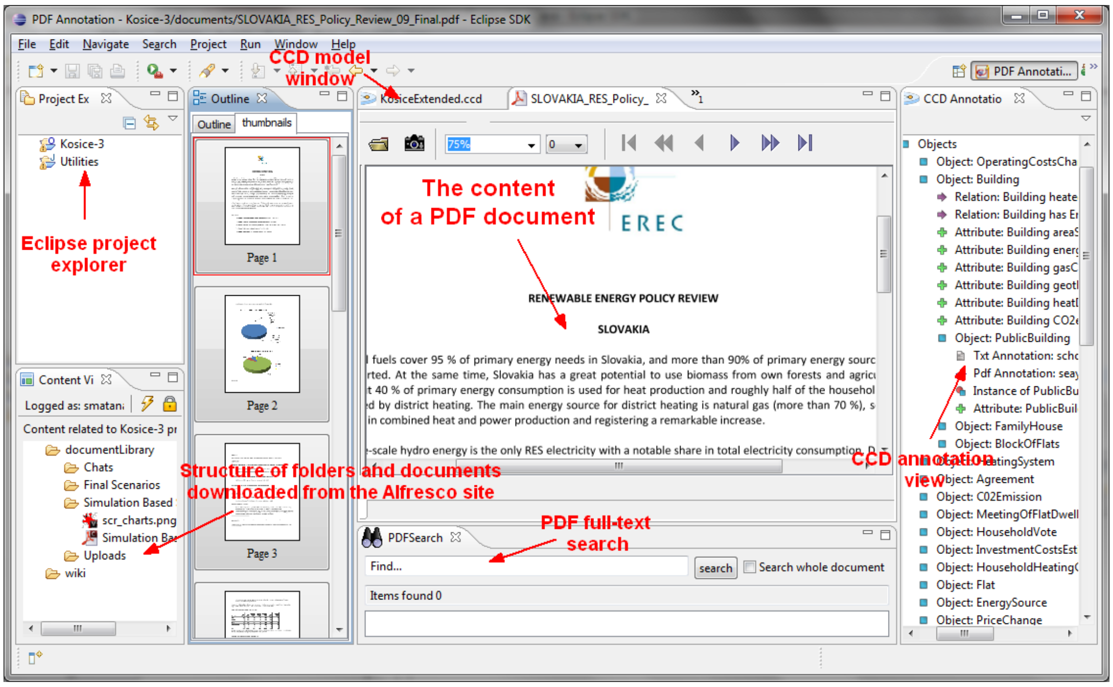


Figure 31: The PDF annotation perspective.

Annotators_part1_Video.zip and ***Annotators_part2_Video.zip*** - see the video demonstrating the use of PDF Annotator and HTML Annotator tools. Instructions on how to annotate PDF/HTML documents downloaded from the Alfresco space.

How to annotate a PDF document?

The PDF document is ready for annotations when it is opened in the respective tab of the *Annotation Editor* view and the CCD model structure is available in the *CCD Annotation view*. To create an annotation, please follow these steps:

1. In the CCD model structure displayed in the *CCD Annotation view*, select a proper CCD concept on which you wish to annotate a text.
2. Inside the content of the PDF document, highlight a text portion that will then be associated with the selected CCD concept.

Note: For PDF annotation, it is necessary to select a CCD concept in advance, prior to the highlighting of a text fragment. Drag & drop functionality is not supported.

3. Right click on the highlighted text portion to invoke the context menu (see in Figure 32).
4. Click on the *Annotate* command in the displayed context menu. The annotation is created immediately - the highlighted text is associated with the selected CCD concept; i.e., it is displayed as a child node of the concept in the CCD hierarchy in *CCD Annotation view*.

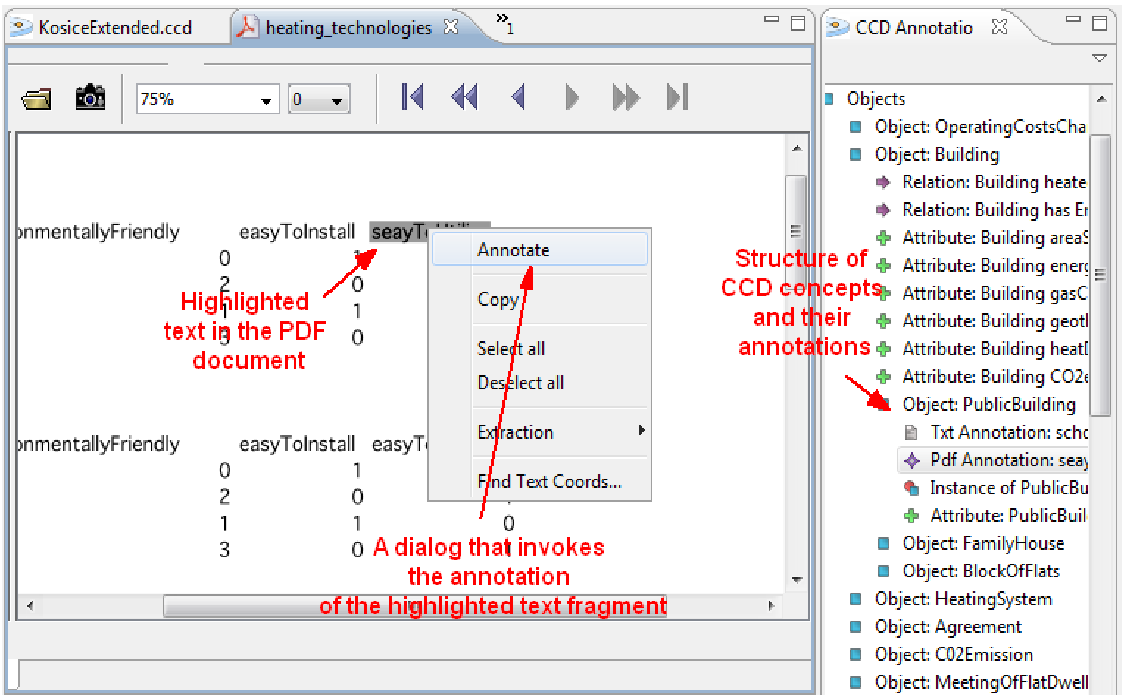


Figure 32: Annotation of a PDF document.

In the structure of CCD concepts, annotations are labelled by first line of the text that was annotated. To distinguish particular annotations in the PDF text, they are marked with different colours. Moreover, by clicking on an annotation in the structure of CCD concepts, the PDF content scrolls to the respective annotated text. This way, the annotations can serve as a navigation mechanism through the (possibly long and complex) content of a document.

Note: The PDF Annotation tool allows opening and simultaneous annotation of several documents.

How to navigate through the PDF annotations?

There are two options for navigating through the PDF annotations, namely:

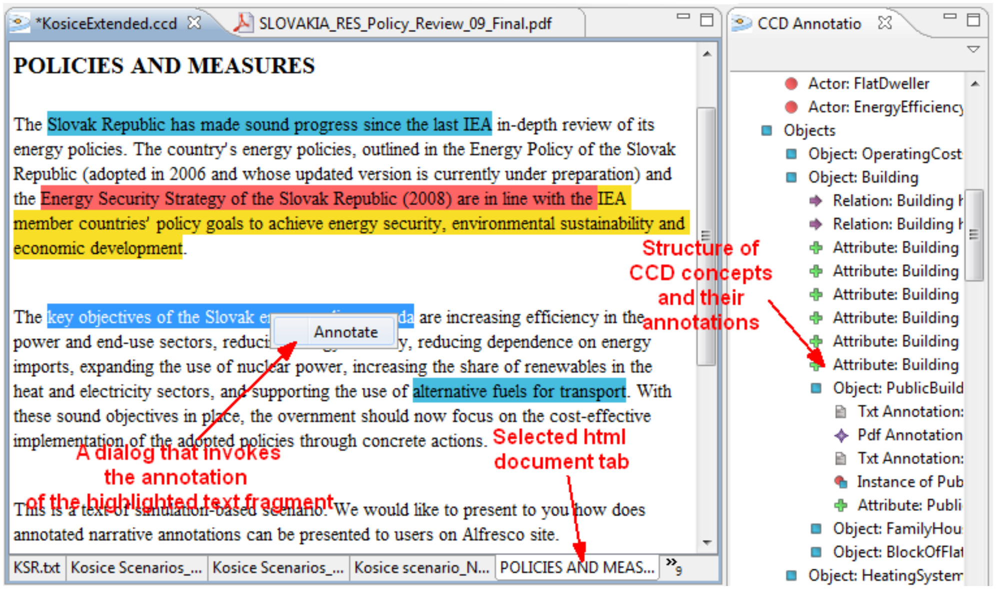
1. From CCD to text:
   - In the *CCD Annotation view*, which displays the structure of CCD concepts and their annotations, double click on an annotation - a child node of the CCD concept. The related annotation in the PDF content will be displayed in the *Annotation Editor*.
2. From text to CCD:
   - In the *Annotation Editor*, right click on the text portion which is highlighted as an annotation. In the context menu, select the *Find Text Coordination* command. The *CCD Annotation view* then selects all corresponding annotations in the view (please note: The *CCD Annotation view* must be already visible).

3.4. HTML ANNOTATOR

How to open a HTML document for annotations?

The *HTML Annotator* plug-in allows to create links between CCD model elements and text fragments of the documents in HTML format, namely the wiki pages downloaded from the Alfresco content repository. To open a HTML document for annotations, please follow these steps:

1. Make sure that your project is correctly set and the CCD file appears in the root of your project. Activate the *Annotation Editor* and *CCD Annotation view* by clicking with the right mouse button on the CCD file that will be used for annotations. In the invoked dialog, select the *Open With → CCD Annotation Editor* command. The CCD file will be opened in the *Annotation Editor* view and in the *CCD Annotation view* as well
2. Navigate to the HTML document you want to annotate - in *Content View* for wiki pages downloaded from Alfresco, or in the project folder of *Package Explorer* for local HTML files. Double-click on the HTML document to open it in the *HTML* tab of the *CCD Annotation Editor* view, as it is depicted in Figure 33.


*KosiceExtended.ccd | SLOVAKIA_RES_Policy_Review_09_Final.pdf

POLICIES AND MEASURES

The Slovak Republic has made sound progress since the last IEA in-depth review of its energy policies. The country's energy policies, outlined in the Energy Policy of the Slovak Republic (adopted in 2006 and whose updated version is currently under preparation) and the Energy Security Strategy of the Slovak Republic (2008) are in line with the IEA member countries' policy goals to achieve energy security, environmental sustainability and economic development.

The key objectives of the Slovak e[...]da are increasing efficiency in the power and end-use sectors, reduci[...]y, reducing dependence on energy imports, expanding the use of nuclear power, increasing the share of renewables in the heat and electricity sectors, and supporting the use of alternative fuels for transport. With these sound objectives in place, the overnment should now focus on the cost-effective implementation of the adopted policies through concrete actions.

This is a text of simulation-based scenario. We would like to present to you how does annotated narrative annotations can be presented to users on Alfresco site.

KSR.txt | Kosice Scenarios_... | Kosice Scenarios_... | Kosice scenario_N... | POLICIES AND MEAS... | »9

CCD Annotatio

- Actor: FlatDweller
- Actor: EnergyEfficiency
- Objects
  - Object: OperatingCost
  - Object: Building
    - Relation: Building
    - Relation: Building
    - Attribute: Building
    - Attribute: Building
    - Attribute: Building
    - Attribute: Building
    - Attribute: Building
    - Attribute: Building
  - Object: PublicBuild
    - Txt Annotation:
    - Pdf Annotation
    - Txt Annotation:
    - Instance of Pub
    - Attribute: Publi
  - Object: FamilyHou
  - Object: BlockOfFlat
- Object: HeatingSystem
Annotate
Structure of CCD concepts and their annotations
A dialog that invokes the annotation of the highlighted text fragment
Selected html document tab

Figure 33: Annotation of a HTML content.

How to annotate a HTML document?

The HTML document is ready for annotations when it is opened in the *HTML* tab of the *Annotation Editor* view and the CCD model structure is available in the *CCD Annotation view*. To create an annotation, please follow these steps:

1. In the CCD model structure displayed in the *CCD Annotation view*, select a proper CCD concept on which you wish to annotate a text.
2. Inside the content of the HTML document, highlight a text portion that will then be associated with the selected CCD concept.

Note: For HTML annotation, it is necessary to select a CCD concept in advance, prior to the highlighting of a text fragment. Drag & drop functionality is not supported.

3. Right click on the highlighted text portion to invoke the context menu.
4. Click on the *Annotate* command in the displayed context menu. The annotation is created immediately - the highlighted text is associated with the selected CCD concept; i.e., it is displayed as a child node of the concept in the CCD hierarchy in *CCD Annotation view*.

How to navigate through the HTML annotations?

There are two options for navigating through the HTML annotations, namely:

1. From CCD to text:
   - In the *CCD Annotation view*, which displays the structure of CCD concepts and their annotations, double click on an annotation - a child node of the CCD concept. The related annotation in the HTML content will be displayed in the *Annotation Editor*.
2. From text to CCD:
   - In the *Annotation Editor*, right click on the text portion which is highlighted as an annotation. In the context menu, select the *Find Text Coordination* command. The *CCD Annotation view* then selects all corresponding annotations in the view (please note: The *CCD Annotation view* must be already visible).

3.5. MYNOTE ANNOTATOR

How to open an editable plain text document for annotations?

The aim of the *MyNote Annotator* plug-in is to provide an editor enabling flexible editing and annotation of texts (in plain text format, which is only supported in current version of the plug-in). The distinction between the *MyNote Annotator* and other provided annotators (i.e., TXT, PDF, and HTML - see in previous sections) is that the *MyNote Annotator* preserves existing annotations during the text editing. This feature helps users, namely modellers, to change their notes and do not lose already created annotations during the modelling phase. To create a new note file, please follow these steps:

1. Use the "*File → New → Other …*" command from the main Eclipse menu. In the appearing wizard, select the "*New Note File*" wizard type, which is located under the OCOPOMO group (see Figure 34).

2. In the displayed configuration wizard, specify the name of the new note file and the related OCOPOMO project. Click *Finish* to confirm the operation. The newly created note file is displayed in the *Annotation Editor* window.

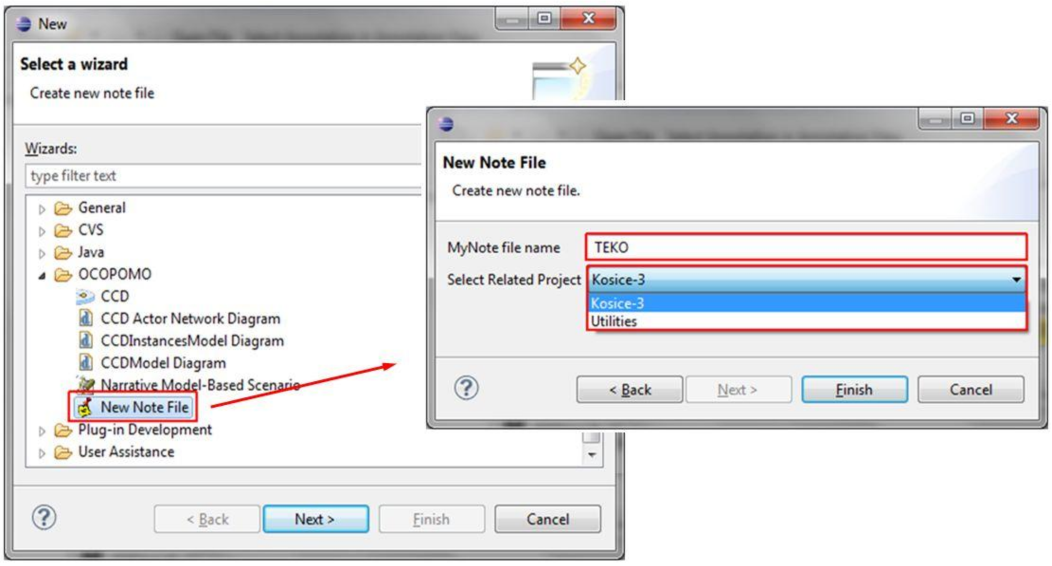


Figure 34: Invoking the *New Note File* wizard.

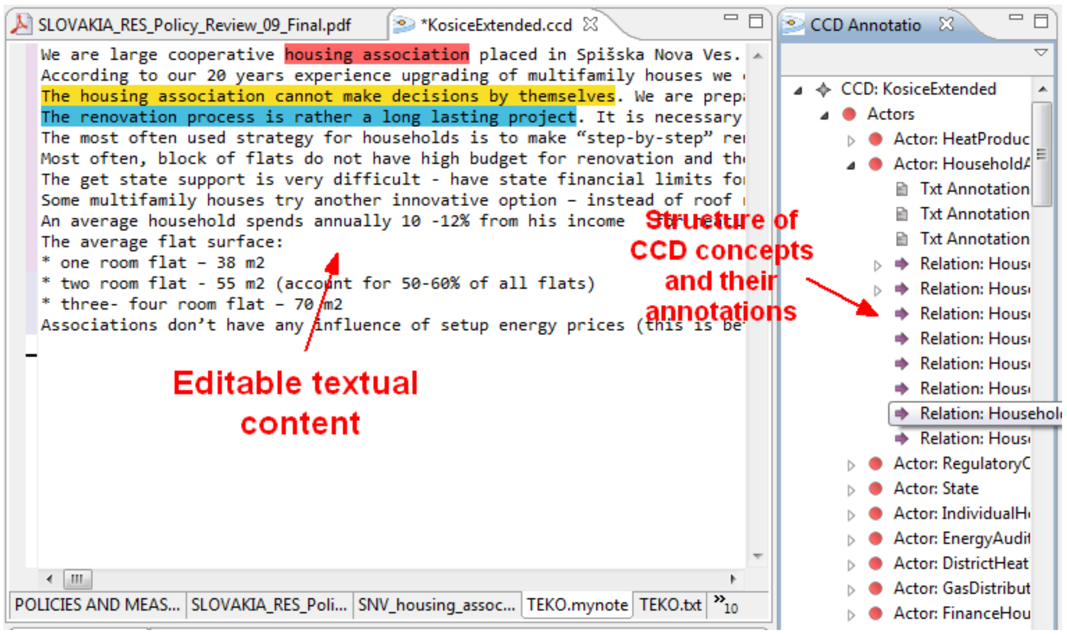
Note: The *MyNote Annotator* is specifically proposed for use during the 5th phase of the OCOPOMO policy development process (see in section 2.2 above), namely for creation of model-based scenarios. Please refer to D4.2-D *User Manual on Simulation Output Analysis and Traceability Browsing Tools* for more information.

MyNoteAnnotator_Video.zip - see the video demonstrating the use of the tool. Instructions on how to edit the text, annotate it, and preserve the annotations during the editing.

How to annotate and edit the text simultaneously?

After the new note file is created, it is available for standard manual editing in the *Annotation Editor* window. Annotations can be performed by drag & drop functionality, in a similar way as it is in the Txt Annotator tool (cf. section 3.2). To annotate and edit the text simultaneously in the *MyNote Annotator*, please follow these steps:

1. Highlight a text portion in the *Annotation Editor* window and drag it to the related CCD concept in the *CCD Annotation view*.
2. Drop the highlighted text to the note of the CCD concept. New annotation will appear in CCD model, as it is depicted in Figure 35).
3. Now you can edit the text inside the created annotation. The annotation will remain valid and will be adjusted to the text changes.



```
SLOVAKIA_RES_Policy_Review_09_Final.pdf    *KosiceExtended.ccd

We are large cooperative housing association placed in Spišska Nova Ves.
According to our 20 years experience upgrading of multifamily houses we
The housing association cannot make decisions by themselves. We are prep
The renovation process is rather a long lasting project. It is necessary
The most often used strategy for households is to make "step-by-step" re
Most often, block of flats do not have high budget for renovation and th
The get state support is very difficult - have state financial limits fo
Some multifamily houses try another innovative option - instead of roof
An average household spends annually 10 -12% from his income
The average flat surface:
* one room flat - 38 m2
* two room flat - 55 m2 (account for 50-60% of all flats)
* three- four room flat - 70 m2
Associations don't have any influence of setup energy prices (this is be

POLICIES AND MEAS... | SLOVAKIA_RES_Poli... | SNV_housing_assoc... | TEKO.mynote | TEKO.txt | »10
```

```
CCD Annotatio
◢ CCD: KosiceExtended
  ◢ Actors
    ▷ ● Actor: HeatProduc
    ◢ ● Actor: HouseholdA
        Txt Annotation
        Txt Annotation
        Txt Annotation
      ▷ Relation: Hous
      ▷ Relation: Hous
        Relation: Hous
        Relation: Hous
        Relation: Hous
        Relation: Hous
        Relation: Househol
        Relation: Hous
    ▷ ● Actor: RegulatoryC
    ▷ ● Actor: State
    ▷ ● Actor: IndividualH
    ▷ ● Actor: EnergyAudit
    ▷ ● Actor: DistrictHeat
    ▷ ● Actor: GasDistribut
    ▷ ● Actor: FinanceHou
```

Figure 35: Annotation of an editable plain text using the MyNote Annotator.

Hint: When starting with a new (empty) note file, few blank lines or space characters should be added at the beginning of the text. It allows to annotate the first line of the text and then, optionally, add a non-annotated text in front of this annotation.

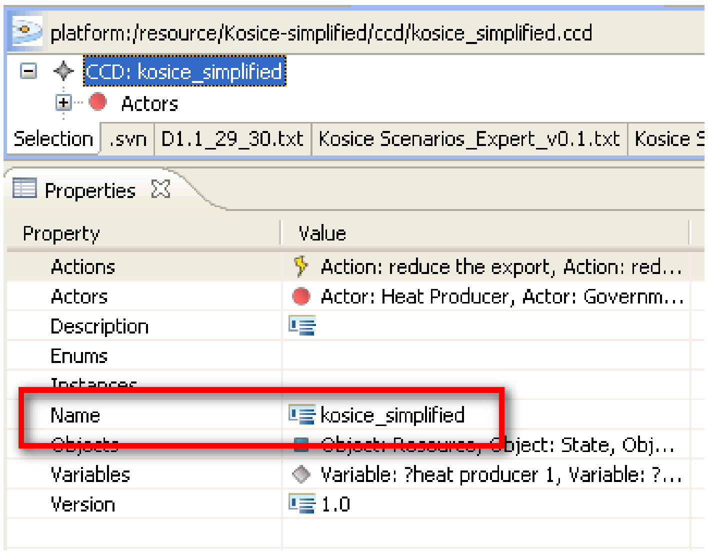
Note on further extensions: The *MyNote Annotator* currently does not support text formatting - only the plain text format can be used. It could be, however, helpful for modellers/analysts to use some WYSIWYG functionality, which will be considered in the next versions of the OCOPOMO toolkit.

3.6. CCD2DRAMS

How to generate the DRAMS code from created CCD models?

The CCD2DRAMS plug-in simplifies the generation of simulation models by an automated transformation of developed CCD models to the stubs of Java and DRAMS source codes. To perform the CCD model transformation, please follow these steps:

1. To start code generation, make sure that the CCD model, and the respective source file, has a proper name (see in Figure 36).


platform:/resource/Kosice-simplified/ccd/kosice_simplified.ccd

CCD: kosice_simplified
Actors

Selection | .svn | D1.1_29_30.txt | Kosice Scenarios_Expert_v0.1.txt | Kosice S

Properties

Property	Value
Actions	Action: reduce the export, Action: red...
Actors	Actor: Heat Producer, Actor: Governm...
Description	
Enums	
Instances	
Name	kosice_simplified
Objects	Object: Resource, Object: State, Obj...
Variables	Variable: ?heat producer 1, Variable: ?...
Version	1.0

Figure 36: The CCD with given name.

2. If the CCD has a name, please click with the right mouse button on the CCD file and select *"ccd2drams -> Start Generate"* (see in Figure 37).

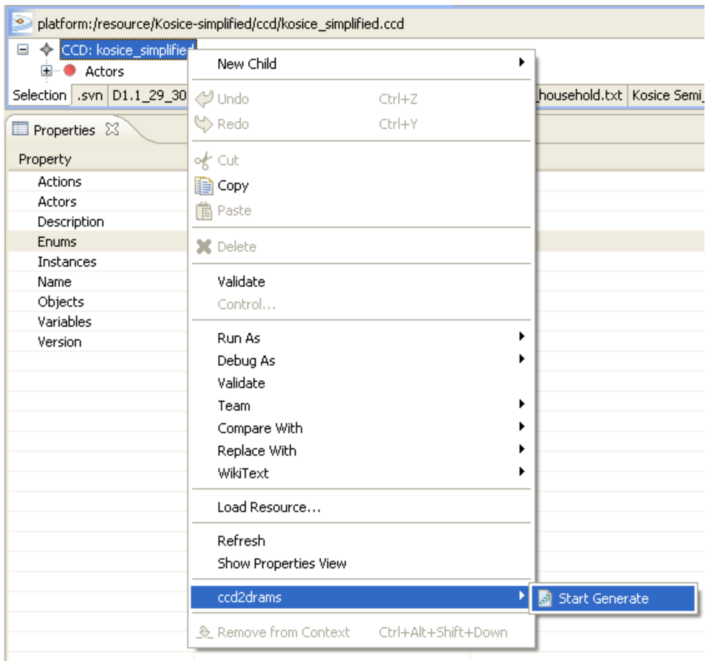


Figure 37: Initiate the CCD2DRAMS code generation.

3. After the "*Start Generate*" command has been clicked, a new source folder is created in the same project with the name "*simulation-src*", as it is depicted in Figure 38. The folder contains stubs of both Java and DRAMS code of the agent-based policy model, which were automatically generated for each of CCD files in the project.

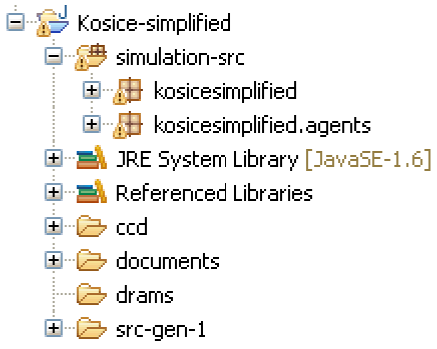


Figure 38: The "*simulation-src*" folder.

How to change the generated DRAMS code?

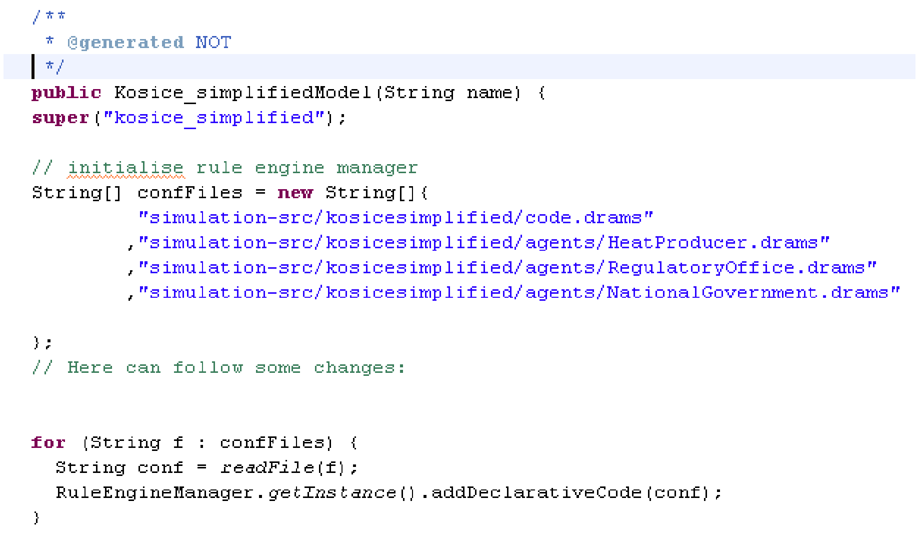
The code of Java classes and DRAMS rules was generated from the CCD model in its basic stub-like form of both types of source code files. To extend the code into an operational agent-based policy model that could be deployed for running simulations, a manual programming is necessary. If you wish to update the Java or DRAMS source code, which was generated by CCD2DRAMS, please follow these instructions:

1. In the hierarchy of project files in Package Explorer view, navigate to the "*simulation-src*" folder containing generated Java and DRAMS source code files for all available CCD models.
2. Expand a respective package - <*CCD_name*> for Java classes and <*CCD_name.agents*> for DRAMS rules.
3. Double-click on a source file, which will then appear in a separate editing view. Do the code updates - for the Java source code, all features provided natively by Eclipse IDE can be used. DRAMS code editing and debugging is supported by the installed DRAMS module (see in D4.2-C).
4. After the code updates are done, save changes by Ctrl-S.

How to re-generate the DRAMS code and preserve manual changes made in the code?

After the Java or DRAMS source code was manually updated, the underlying CCD model might be changed again, in an iterative process of scenario analysis and conceptual policy modelling. The CCD model change implies a necessity to re-generate the Java/DRAMS source code in the next interation; however, the changes that were already done in the source code should be preserved. It is possible to mark the automatically generated code so that the next time the code is generated, it will not be overwritten. To protect the source code there are two options:

1. "@generated" tag: If the code is in a Java method or class, the "@generated" tag needs to be changed to "@generated NOT". The whole method or class labelled with this tag will not be overwritten (see in Figure 39).
2. the "user code" section: The code that is manually placed inside the "user code" section (see in Figure 40) will not be overwritten.



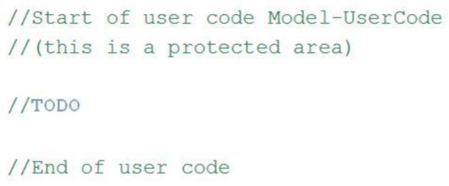
```
/**
 * @generated NOT
 */
public Kosice_simplifiedModel(String name) {
super("kosice_simplified");

// initialise rule engine manager
String[] confFiles = new String[]{
          "simulation-src/kosicesimplified/code.drams"
         ,"simulation-src/kosicesimplified/agents/HeatProducer.drams"
         ,"simulation-src/kosicesimplified/agents/RegulatoryOffice.drams"
         ,"simulation-src/kosicesimplified/agents/NationalGovernment.drams"

};
// Here can follow some changes:


for (String f : confFiles) {
  String conf = readFile(f);
  RuleEngineManager.getInstance().addDeclarativeCode(conf);
}
```

Figure 39: The "@generated NOT" tag used for preserving the source code.



```
//Start of user code Model-UserCode
//(this is a protected area)

//TODO

//End of user code
```

Figure 40: The "user code" section used for preserving the source code.

Further information on the classes, methods, and DRAMS rules, together with a description of the development environment for agent-based policy models, can be found in the *DRAMS User Manual* provided in the D4.2-C document.

4. CONCLUSION

This manual provides the documentation and usage instructions for the OCOPOMO *CCD Tools* and accompanying components such as *Content Repository Client* and *CCD2DRAMS Transformation tool*. The manual describes common tasks for end users (policy analysts, modellers, and facilitators) during the analysis of provided policy scenario alternatives and development of respective conceptual policy models - CCD, which are then, by means of *CCD2DRAMS*, transformed to source code stubs of executable agent-based policy models.

Since the implementation of the *CCD Tools* 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

Bednar, P., et al: D3.1 *Platform Components* (Supportive Report to Software Components). Deliverable 3.1, v. 1.04, OCOPOMO consortium, 2012.

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.

Mach, M., et al: *D2.1 Platform Architecture and Functional Description of Components*. Deliverable 2.1, OCOPOMO consortium, 2010.

Moss, S., et al: *D5.1 Scenario, Policy Model and Rule-based Agent Design*. Deliverable 5.1, OCOPOMO consortium, 2011.

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.