OCOPOMO
Open Collaboration in Policy Modelling

D4.2 System and User Documentation

Document Full Name	**OCOPOMO_D4.2_v1.0.doc**
Date	**30/04/2013**
Work Package	**WP4: Integration of components**
Lead Partner	**Intersoft**
Authors	**Karol Furdik, Peter Smatana, Peter Bednar, Peter Butka, Ulf Lotzmann, Sabrina Scherer, Viera Furdikova, Jan Smatana, Carsten Hartenfels**
Document status	**v1.0 FINAL**
Dissemination level	**PUBLIC (PU)**

Document Log

Version	Date	Comment	Author
001	08/01/2013	Initial structure of the deliverable.	Karol Furdik
002	23/01/2013	Inputs to section 2.2 and Appendix A on functional requirements.	Peter Butka
003	29/01/2013	Updated structure of the deliverable, inputs to introduction and updates of section 2.2 on requirements.	Karol Furdik
004	11/02/2013	Update of requirements in Appendix A and small addition to section 2.2.	Peter Butka
005	13/02/2013	Inputs to user manual on CCD Tools.	Sabrina Scherer
006	21/02/2013	Update of requirements in Appendix A.	Peter Butka
007	25/02/2013	Integration of inputs, update of Chapter 3 Administration of OCOPOMO toolkit.	Karol Furdik, Viera Furdikova
008	04/03/2013	Inputs to user manual on CCD Tools - annotators, Content Repository Client.	Peter Smatana, Karol Furdik
009	05/03/2013	Inputs to user manual on Simulation Analysis Tool.	Peter Smatana, Jan Smatana
010	12/03/2013	Inputs to user manual on Collaboration and Scenario Generation Tools.	Peter Bednar
011	18/03/2013	Integration of inputs, update of section 3.3 system set up and maintenance.	Karol Furdik
012	22/03/2013	Inputs to user manual on Collaboration and Scenario Generation Tools.	Peter Bednar
013	27/03/2013	Integration of inputs, update of Chapter 2 on system architecture.	Karol Furdik, Viera Furdikova
014	03/04/2013	Inputs to user manual on DRAMS.	Ulf Lotzmann
015	03/04/2013	Integration of inputs, update of Chapter 4 on user manuals.	Karol Furdik
016	05/04/2013	Update of section 4.4 on the Simulation Environment module.	Ulf Lotzmann, Karol Furdik
017	10/04/2013	Update of section 4.5 on the Simulation Analysis module.	Peter Smatana, Karol Furdik
018	12/04/2013	Update of section 4.6 on the Traceability and evaluation of output scenarios.	Karol Furdik, Viera Furdikova, Peter Smatana
019	17/04/2013	Inputs to user manual on CCD Model Explorer applet.	Karol Furdik, Ulf Lotzmann, Carsten Hartenfels
020	19/04/2013	Integration of inputs, updates of Introduction and Conclusion sections. Preparation for internal review.	Karol Furdik
021	26/04/2013	Revisions based on internal review	Karol Furdik
100	30/04/2013	Final corrections and formatting	Karol Furdik, Maria Wimmer

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
CCD	Consistent Conceptual Description
CMIS	Content Management Interoperability Services
CMS	Content Management Server / Content Management System
CSET	Collaboration and Scenario Editing Tools
CSV	Comma Separated Values
DDG	Data Dependency Graph
DIV	A tag in the HTML code
DRAMS	Declarative Rule-based Agent Modelling System
EMF	Eclipse Modelling Framework
GATE	General Architecture for Text Engineering, http://gate.ac.uk
GMF	Graphical Modelling Framework (from Eclipse platform)
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HW	Hardware
ICT	Information and Communication Technologies
IDE	Integrated development environment
ID	Identifier
IDE	Integrated Development Environment
JDK	Java Development Kit
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
KSR	Kosice Self-governing region
OPS	Official Production System
OS	Operating system
PDF	Portable Document Format
RDG	Rule Dependency Graph
REST	Representational State Transfer
RESTful	A web service implemented using HTTP and the principles of REST
SD	System Documentation
SDK	Software Development Kit
SE	Simulation Environment
SPAN	A tag in the HTML code
SVN	Apache Subversion (source versioning system)
SW	Software

TRAC	Tracking system for software development projects (http://trac.edgewall.org)
TXT	Plain Text format
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
XML	eXtensible Markup Language
WP	Work Package

EXECUTIVE SUMMARY

This deliverable presents the user and system documentation of the integrated OCOPOMO platform, which was produced as outcome of the task T4.4. The documentation describes the OCOPOMO ICT toolkit in its final version, based on the integrated platform implemented within previous tasks of WP3 and WP4 and updated in accordance with the feedback obtained from the second trial of pilot applications. The included overview of user requirements documents the software development process and provides a detailed functionality description of particular end-user tools integrated in the platform.

The user documentation is organised in accordance with the specified OCOPOMO process. The usage instructions are provided for (1) a formulation of input scenarios, collaborative discussion on policy alternatives and production of evidence-based scenario narratives, for (2) conceptual modelling and annotation of scenarios, (3) programming of executable agent-based policy models and running simulations, (4) analysis of obtained simulation results and production of output model-based scenarios, and finally for (5) evaluation of output scenario narratives in a community of involved stakeholders, using the traceability information embedded in the provided scenarios.

The system documentation follows the division on the shared Alfresco-based collaborative discussion and content management web space and the local installations of Eclipse-based tools enabling experts to perform sophisticated policy analysis and modelling operations. The deliverable includes a detailed description of hardware and software requirements for running the OCOPOMO toolkit on both the environments, installation and maintenance instructions, set up guidelines, examples, and commented fragments of source code enabling a customisable configuration of provided end-user tools.

1. INTRODUCTION

The integrated OCOPOMO ICT toolkit, which is documented in the following chapters from the perspective of end users, system administrators, and software developers, was iteratively developed within WP 3 and WP 4 as a modular platform enabling collaborative formulation, modelling, and evaluation of policy scenarios. The ICT toolkit, provided as one of the main outcomes of the OCOPOMO project and described in D4.1 (Furdik et al, 2013), supports the underlying OCOPOMO process[1] consisting of six steps of collaborative policy development, as it is presented in Figure 1. Each of the process steps is performed by dedicated user roles (see below in section 4.1 and in D2.1 (Mach et al, 2010)) that use a specific package of tools provided in web-based Alfresco environment for stakeholders, policy decision makers and general public (steps 1, 2, 6) as well as in a local installation of Eclipse IDE for experts on policy analysis and modelling (steps 3, 4, 5). The collaborative and iterative process of policy development is supervised by roles of facilitators and system administrators who maintain tools in both Alfresco and Eclipse parts of the platform[2].

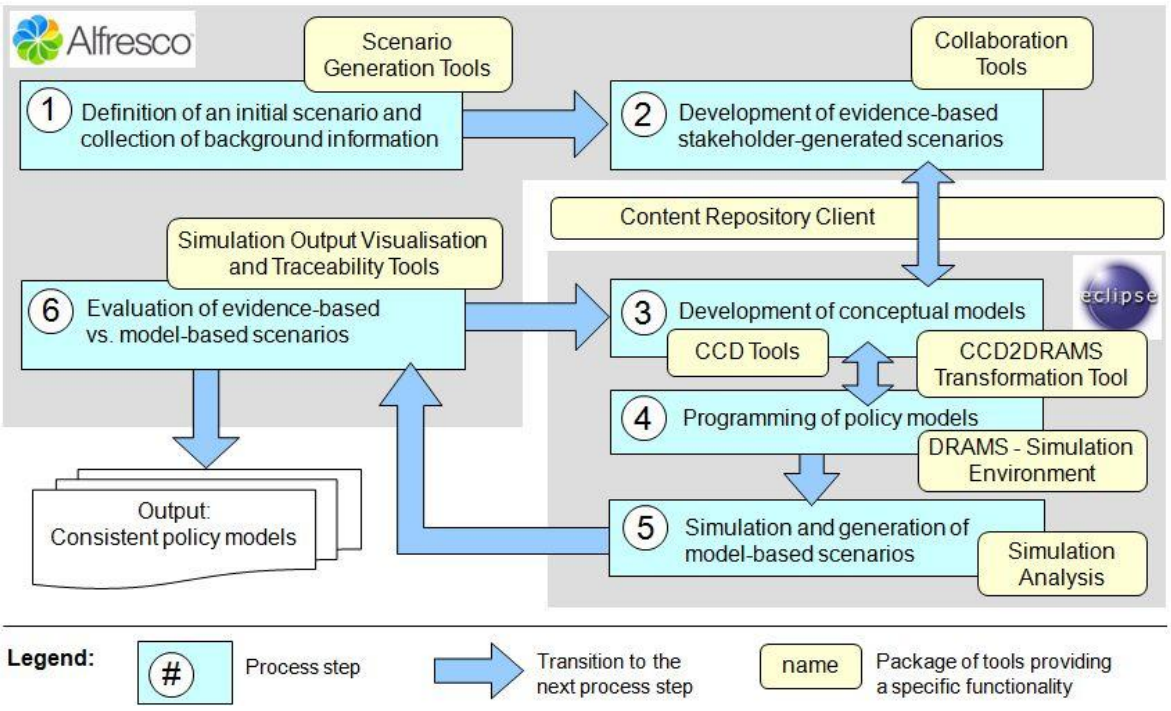

Figure 1: Tools provided for phases of the OCOPOMO policy development process.

Particular tools and system components of the OCOPOMO platform were designed, implemented, and integrated iteratively, in several development and integration phases, as it is described in D4.1 (Furdik et al, 2013). In parallel, the produced software modules were regularly documented in deliverables D3.1 (Bednar et al, 2011) and D3.2 (Butka et al, 2012), as well as in a set of user manuals that were drafted in the beginning of year 2012 for the first trial of pilot applications (cf. next in section 1.1). The intermediate user and system documentation has been developed with the aim to enable user partners to master the developed software, namely the end-user tools, during the first trial as well as to collect feedback and comments on both the documentation and implemented software in context of

[1] The OCOPOMO process was initially designed in D1.1 (Bicking et al, 2010), then it was updated and refined in D3.1 (Bednar et al, 2011) and D5.1 (Moss et al, 2011).

[2] The methodology for assigning user roles, competencies, and responsibilities on particular end-user tools of the platform will be detailed in the forthcoming deliverable D8.1.

usability and accessibility. The feedback provided by users was collected and processed using the internal TRAC system installed on http://ocopomo.ekf.tuke.sk/trac/ocopomoprj, which was utilized as a tool for the requirements maintenance. An overview of all the provided requirements, categorised and accompanied with attributes such as priority marks, implementation and usage notes, is provided in section 2.2 and in Appendix A. These requirements were taken as drivers for refining the system components (see also in D3.2 (Butka et al, 2012) and D4.1 (Furdik et al, 2013)) and consequently for producing the here-presented documentation of the OCOPOMO platform in its final version.

The documentation of the integrated OCOPOMO ICT toolkit, provided in the following chapters, is structured as follows. Overall description of the OCOPOMO platform, architecture of system modules and end-user tools, and overview of functional requirements is provided in Chapter 2. The description includes general implementation notes for the platform, limitations and known issues, as well as usage recommendations and lessons learnt that were identified during the system development and testing.

Administration, set up, and maintenance guidelines are presented in Chapter 3. The description includes HW/SW requirements for installing and operating the platform, package download and installation instructions. Chapter 4 is a usage manual for all end-user tools and interfaces provided within the OCOPOMO ICT toolkit, while the system documentation of the tools and inner system components is provided in Chapter 5.

In addition, the user documentation is enhanced by a set of video presentations containing life usage instructions of particular end-user tools, which are outlined in the introductory section of Chapter 4. A summary and conclusive remarks of the provided documentation is presented in Chapter 6.

1.1. PURPOSE AND CONTEXT OF THE DOCUMENT

This deliverable documents the usability and implementation details of the integrated ICT toolkit, which was developed within WP 3 and WP 4 as a final version of the OCOPOMO system for collaborative policy creation. The documented software solution was produced as an outcome of the second phase of the software integration and provides the toolkit consisting of a set of integrated platform modules that support all phases of the OCOPOMO process (cf. Figure 1 above).

The provided software documentation complements specifications and solution descriptions presented in other project deliverables, namely:

- Design of overall platform architecture, documented in D2.1 (Mach et al, 2010);
- Platform components, designed and implemented within the WP 3, which were documented in D3.1 ver. 1.0 (Bednar et al, 2011), the revised deliverable D3.1 (Bednar et al, 2012) and its update D3.2 (Butka et al, 2012);
- Integrated OCOPOMO ICT Toolkit, provided in its final version in D4.1 (Furdik et al, 2013).

In addition to that, the presented deliverable builds on and extends a set of manuals that were produced for particular tools for the first trial of pilot applications as follows:

- Manual to Collaboration and Editing Tools, version 0.4 (Bednar, Butka, 2012);
- CCD Tool Manual, version 0.7 (Scherer, Wimmer, 2012);
- CCD2DRAMS Tool Manual, version 0.1 (Scherer, Markisic, Lilge, 2012);
- DRAMS Manual, version 0.2 (Lotzmann, Meyer, 2012).

These manuals were extended and finalized in the here-presented report, namely in the Chapter 4 below. The development of user manuals and the accompanying system documentation was accomplished within the task T4.4, targeting all involved user roles such as administrators, facilitators,

end users (i.e., policy decision makers, stakeholders, and experts on policy analysis and modelling), as well as developers that can be interested in system customisation and functionality enrichment.

The provided documentation of the integrated software platform of OCOPOMO is envisioned to serve as the basic instruction material for set up, usage, and maintenance of end user tools in the second trial of pilot applications performed within WP 7. Together with a set of formal policy models, developed within WP 6 and published in D6.1 (Moss et al, 2013), and the overall methodology of the OCOPOMO process of collaborative policy modelling that is provided in D8.1 (Scherer et al, 2013), the here-presented user and system documentation complements the overall description of the OCOPOMO platform from the perspective of usage and possible adaptation of provided tools to other policy modelling use cases.

2. OVERALL DESCRIPTION OF THE OCOPOMO PLATFORM

The OCOPOMO platform is provided as a modular software system of end user tools and inner components integrated on both functional and data exchange levels. Particular system modules, schematically depicted in Figure 2, were designed with the aim to support all six phases of the OCOPOMO process of collaborative policy development, as it was presented in Chapter 1 (see in Figure 1).

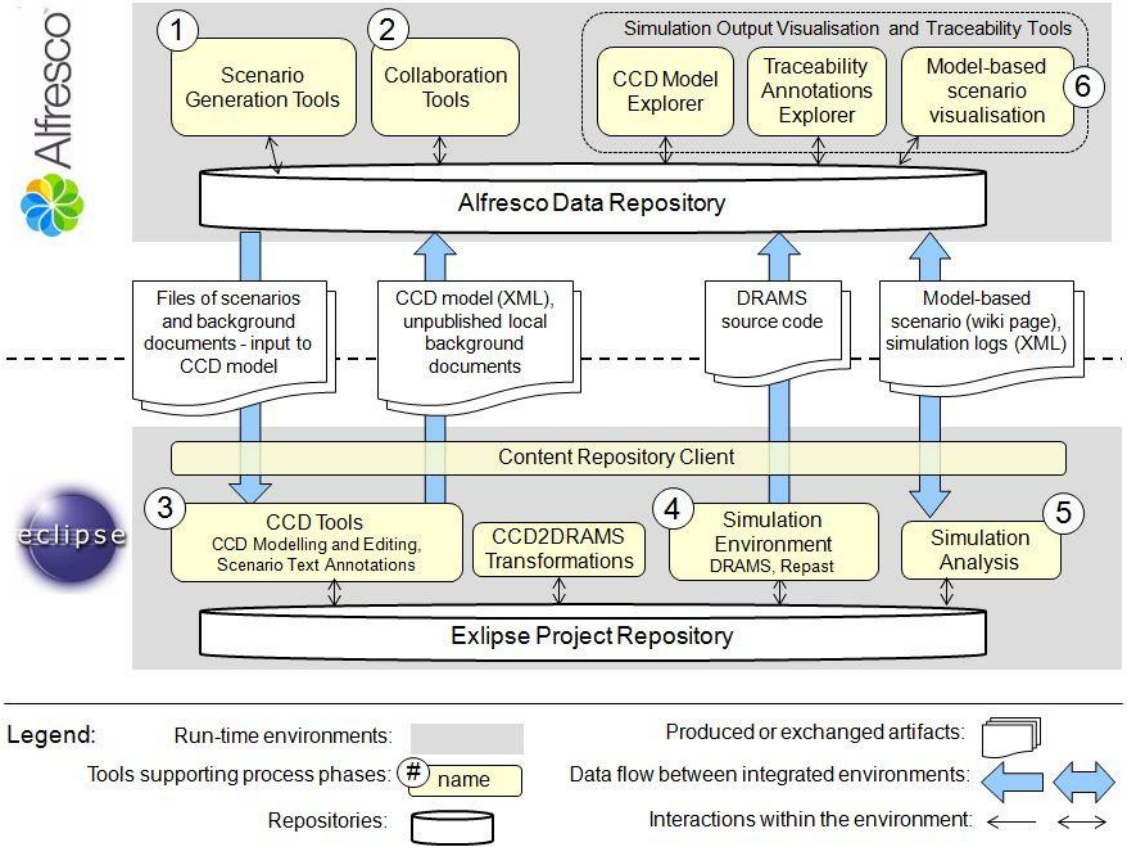

Figure 2: Modules of integrated OCOPOMO ICT toolkit in process phases.

The process phases 1, 2, and 6 are supposed to be accomplished by a broader audience of stakeholders, policy decision makers, various interest groups and general public as well. To enable a true collaboration of the involved participants, the tools and modules supporting these process phases are provided as a web application. Namely, the *Alfresco Share* platform (Alfresco Share, 2011) was employed for the modules of the OCOPOMO ICT toolkit, and respective process phases, as follows:

- ***Phase 1***: *Definition of an initial scenario and collection of background information* - is supported by:
    - *Scenario Generation Tools*, i.e., *Wiki* and *Documents Management* components supporting collaborative formulation of evidence-based scenarios of policy alternatives, together with the *Alfresco Data Repository* and *Content Management System* enabling to store, share, and collaboratively update the scenarios and various background materials.
- ***Phase 2***: *Development of evidence-based stakeholder-generated scenario* - is supported by:
    - *Collaboration Tools*, i.e., means for communication, discussion, planning, online opinion exchange, and shared presentation of ideas on policy alternatives in the

community of involved stakeholders. The policy alternatives are formulated as narrative scenarios, which are provided as inputs to phase 3 of the process.

- ***Phase 6***: *Evaluation of evidence-based vs. model-based scenarios* - is supported by:
    - *Simulation Output Visualisation and Traceability Tools*, i.e., means for presenting the results of policy modelling and simulation, performed by experts in phases 3, 4, and 5, to the stakeholders for further discussion. The toolkit includes presentation of output model-based scenarios as wiki pages in the Alfresco web space, traceability of annotations back to input evidence-based scenarios, visualisation and browsing facilities for CCD models, and presentation of dynamic charts embedded in Alfresco wiki pages. Further collaborative discussion on the provided output scenarios, their updates and modifications may re-invoke phases 1 and 2 as the next iteration of the OCOPOMO process.

Process phases 3, 4, and 5 cover an expert work of policy analysis, conceptual modelling, development of executable agent-based policy models, running simulations, and creation of output scenarios based on simulation results. To perform all these tasks, policy analysis and modelling experts are supported by a suite of tools provided as plug-ins into the *Eclipse Indigo* development environment (Eclipse Indigo, 2011). Plug-ins - tools and components that correspond to respective modules of the OCOPOMO ICT toolkit are provided for the process phases as follows:

- ***Phase 3***: *Development of conceptual models* - is supported by:
    - *CCD Tools*, i.e. a suite of tools and components for analysis and conceptual modelling of provided scenarios of policy alternatives, namely the *CCD Model Editor* for development, visualisation and maintenance of CCD models, as well as a set of *annotators* for plain text (both static and editable), PDF, and HTML formats.
    - *CCD2DRAMS*, a tool that automatically transforms the developed CCD models to stubs of Java and DRAMS code. This source code will be enhanced in the next process phase into a fully featured code of executable agent-based policy models.
- ***Phase 4***: *Programming of policy models* - is supported by:
    - *Simulation Environment* package, consisting of tools and components enabling development, debugging, and execution of declarative agent-based policy models. The package includes the DRAMS rule engine that executes declarative rules and clauses describing a behaviour of individual agents, as well as the Repast-based simulation environment for running Java code of executable policy models.
- ***Phase 5***: *Simulation and generation of model-based scenarios* - is supported by:
    - *Simulation XML Log Transformation*, a component that transforms the logs produced as outputs of executed simulations to proper XML- and text-based formats.
    - *Simulation Analysis Tool*, a specialised Eclipse plug-in that facilitates a production of output model-based narrative scenarios based on obtained simulation logs. By means of built-in *MyNote Annotator*, the tool enables experts, policy analysts and modellers, to create scenario narratives that are linked (i.e., annotated) to the simulation log records. This feature supports the backward traceability to DRAMS rules and clauses, CCD model elements, and input scenarios provided by stakeholders (see *Phase 6* above). Furthermore, the *Simulation Analysis Tool*, using the *Content Repository Client* interface, enables publishing of created output scenarios, together with supportive materials such as CCD model, DRAMS source code, and simulation logs, to the Alfresco shared space.

In addition to that, the *Content Repository Client* module serves as a communication and data exchange interface between Alfresco and Eclipse parts of the OCOPOMO platform. The *Content Repository Client* , implemented as Eclipse plug-in, enables a connection to the remote *Alfresco Data Repository* and provides a means (i.e., CMIS and REST web services) for data exchange between Alfresco and Eclipse environments. It acts as a data exchange enabler between process phases 2 and 3 (download of scenarios and background documents from Alfresco to Eclipse), as well as between phases 5 and 6 (upload of produced model-based scenarios, together with accompanying resources such as CCD model, DRAMS source code, and simulation logs, from Eclipse back to the Alfresco web space).

2.1. ARCHITECTURE OF THE INTEGRATED ICT TOOLKIT

The structure of tools and components of the OCOPOMO ICT toolkit, introduced in previous section, is presented in more details in Figure 3. For each of system modules - groups of tools, internal system components and tools are specified[3] together with data flow directions inside a platform (i.e., the internal data flow) as well as between the Alfresco and Eclipse platforms (the integration data flow).

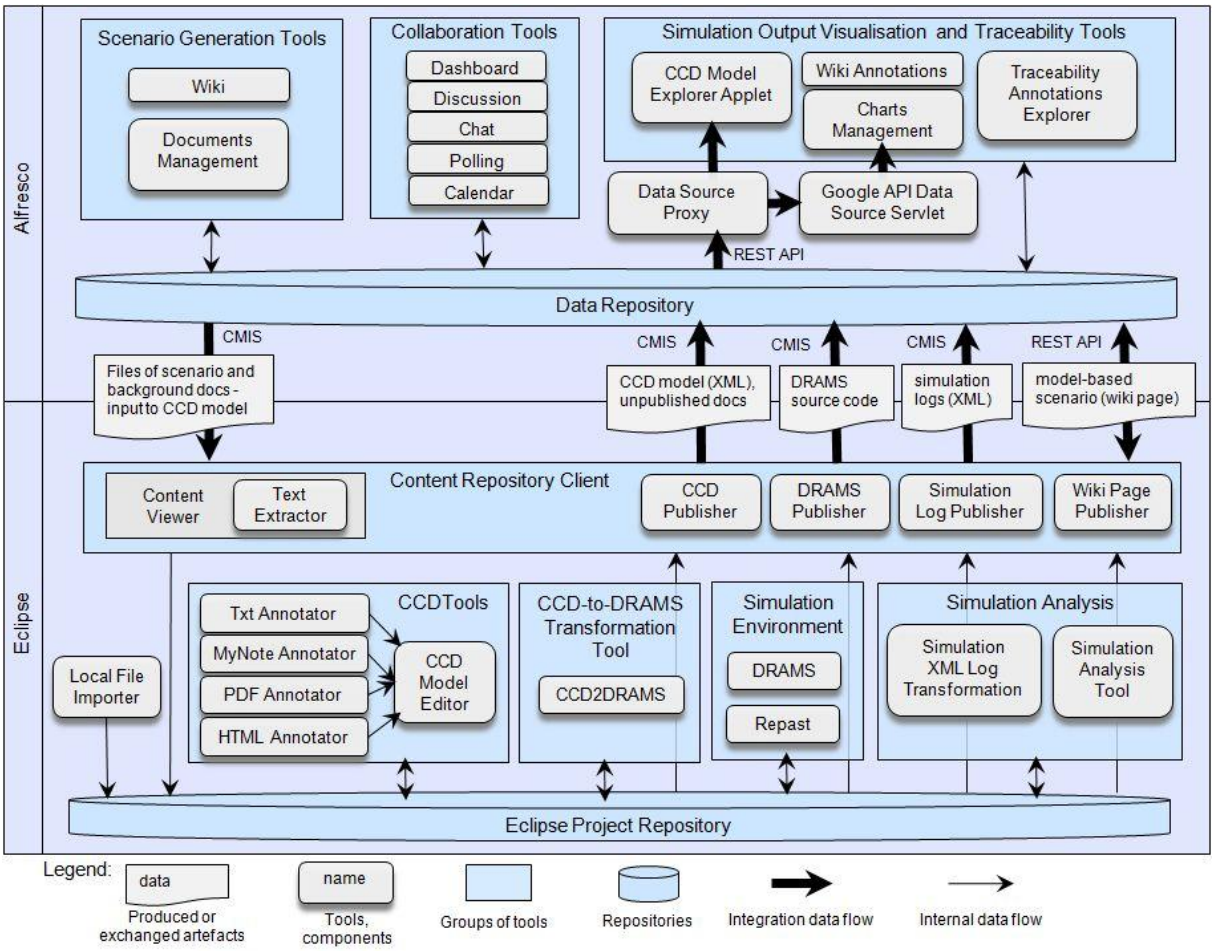

Figure 3: Architecture of system components integrated into the OCOPOMO ICT toolkit.

[3] Detailed description of all the system components, their characteristics and dependencies/relations between them, was presented in D4.1 (Furdik et al, 2013). These technical specifications basically correspond to the system documentation, which is provided in Chapter 5 and in accompanying documents.

According to the system architecture depicted in Figure 3, usage instructions for particular end-user tools are provided in Chapter 4, while the implementation details are described in the system documentation presented in Chapter 5. Installation and maintenance instructions for the OCOPOMO ICT toolkit as whole, even divided into Alfresco and Eclipse parts, is provided in Chapter 3.

The architecture of system components and end-user tools was specified in accordance with the requirements provided by user partners during the iterative design, development, and testing of the OCOPOMO system. The requirements, which are presented next in section 2.2 and detailed in Appendix A, reflect the functionality of implemented tools and components of the platform and this way they complement the user and system documentation of the toolkit.

2.2. PROCESSING THE REQUIREMENTS OF THE SYSTEM FUNCTIONALITY

In this section we provide the information on the processing of the requirements, which were taken as drivers for defining and implementing the needed system functionality of the OCOPOMO platform. The overview of the processing of the requirement can be divided into the two basic steps:

1. Definition and update of requirements *before* the development phase.
2. Definition and update of requirements *during* the development phase.

Within the step 1, the initial set of requirements was created before the start of WP3 (for development of the particular tools) and related tasks in WP4 (for integration of the whole OCOPOMO platform). The requirements here were created during several tasks within WP1 and WP2. After WP1 analysis, in D1.1 deliverable (Bicking et al, 2010), initial set of the user requirements was gathered from stakeholders and other user groups (i.e., the potential users of the final software). Specific methodology was used for investigation of requirements, where all steps of the identified scenario generation and policy modelling process were analysed, in order to find useful electronic support for their successful adoption. Then basic architecture of the platform was designed within the WP2. During this phase other requirements were added according to the state of the art analysis and more detailed descriptions of use cases for the platform, these requirements can be found in particular deliverable for WP2 – D2.1 (Mach et al, 2010). For this step, requirements were organized only as tables in deliverables and summarized on the project web page. All of them were used as an input for the creation of platform architecture. Some of requirements were also revised due to needs of WP3 and WP4 during their run, but were processed in a similar way as before. Any extension of some requirement during that time was processed as separate requirement in step 2.

In step 2 requirements were continuously processed during the run of WP3 and WP4. For this step we have used TRAC software (see http://trac.edgewall.org/ for the reference on software), which provides the enhanced wiki and issue tracking system for software development projects. If we compare the timeline of this step and integration phases provided in D4.1 deliverable in Figure 1 (Furdik et al, 2013), the processing of requirements using TRAC (Location of OCOPOMO TRAC instance – http://ocopomo.ekf.tuke.sk/trac/ocopomoprj) was used during all the phases – from the first release of components, first integration and trial, to revision of components, testing, final integration and second trial.

The issue tracking system was used for recording of new requirements or their update. Many of these requirements were also mentioned and commented (directly or as the platform functionality descriptions / needs) in WP3 and WP4 deliverables – D3.1 (Bednar et al, 2012) and D3.2 (Butka et al, 2012), and D4.1 (Furdik et al, 2013), i.e., it is also revision of requirements within WP3/WP4 related to implementation and integration issues, or update of previously defined requirements (but as new requirement in TRAC, where applicable).

The summary of requirements from step 1, containing the initial set of requirements from D1.1 and added requirements from D2.1, is presented with their final status (at the end of the project) in

Appendix A – Table A-1. An example from summary of these requirements is presented below in Table 1. The format used for their description in this document consists of:

- *ID*: Unique identifier of the requirement in D1.1 or D2.1.
- *Name*: Name of the requirement used in original document.
- *Priority*: Original priority identified for requirement – can be Must-have (it was expected that it is important for the platform), Should-have (optional for the platform, but can useful), or Nice-to-have (low priority for the platform). Some of the requirements were updated according to the priority during the project, for them additional information about revision of priority is added.
- *Component / Module*: This is information about the component(s) and module(s), which are related to current requirement. If requirement is not applicable to specific component or module, then it has 'General Requirement' in the column for Component/Module identification. Another option is requirement, which was not applied to the architecture at the start of architecture design (finished with D2.1 deliverable). For this option 'N.A.' is used in the Component/Module column. For the component identification, if it is not 'General Requirement' or 'N.A.', name(s) of the related managers from the architecture defined in D2.1 is used (e.g., Annotation Manager, Discussion Forums Manager, User Manager, etc.). For the module identification (if it is not 'General Requirement' or 'N.A.'), following abbreviations are used: CMS - Content Management System; CSET - Collaboration and Scenario Editing Tools; CCD Tool; SE - Simulation Environment.
- *Solution / Application / Usage*: This part contains final status of the requirement and related notes on this status. Final status options are: Implemented - if requirement was fully or partially implemented during the project; Rejected - if requirement was rejected due to some reasons; Postponed - requirement is still interesting (important), but during the project implementation was found out of scope for us (can be applied in next projects, or in business implementation for reuse after project, etc.). Notes is related to an information on decision, solution, application and usage of requirement within the project (e.g., how it was implemented, what is available by default, or why it was rejected or postponed, etc.).

Table 1: Description of the attributes of requirement from the initial and updated set within step 1.

ID	Name	Priority	Component / Module	Solution / Application / Usage
I-NF-2	Accessibility	Must-have	General Requirement	**Final status:** Implemented. **Notes:** Alfresco developers and community adapted WAI standards and provide necessary guidelines for achieving the accessibility (see http://wiki.alfresco.com/wiki/Accessibility for details). Due to fact that Alfresco provides heir styles with accessibility standards in mind and we have developed our modifications using their styles and guidelines, the accessibility standards (especially important for Alfresco Share part with stakeholders) are respected.

The summary of requirements from step 2 (i.e., requirements from development phases, processed by the TRAC software), is presented with their final status (at the end of the project) in Appendix A –

Table A-2. Only enhancements and tasks are in this summary, bugs and defects are not presented. An example from the summary of these requirements is presented in Table 2. In TRAC, the requirements were also divided into Trial 1 group (which were related to first trial), Trial 2 group (enhancements for second trial and final version of software), Anonymous (requirements not applicable as for some trial, continuous requirements). This division is not important for their summary for the end of the project and their final status; therefore it is not used in our table in this document. The format for their description in this document consists of:

- *ID*: Unique identifier of the requirement in TRAC.
- *Name*: Name of the requirement used in TRAC.
- *Priority*: Original priority for requirement – according to the settings in TRAC system, two priorities were used – Major (can be seen as equivalent of Must-have or Should-have, i.e., it is important requirement/issue), Minor (equivalent to Nice-to-have, low priority).
- *Module / Part*: This is information about the module and its part, which are related to current requirement. For the module identification, following abbreviations are used: CMS - Content Management System; CSET - Collaboration and Scenario Editing Tools; CCD Tool; SE - Simulation Environment. Part is the identification of part of some module of software according to the TRAC ticket system. Possible values were related to particular tools, e.g., Site and Personal Dashboard, Wiki, Pollings, Repository, CCD Modelling, CCD Annotations, DRAMS, etc.
- *Solution / Application / Usage*: This part contains the same information as in previous step presented in Table 1. Namely, the *Final status* options are: Implemented / Rejected / Postponed. The *Notes* entry is related to an information on decision, solution, application and usage of requirement within the project (e.g., how it was implemented, what is available by default, or why it was rejected or postponed, etc.).

Table 2: Table describing the attributes of a requirement - added TRAC requirements.

ID	Name	Priority	Module / Part	Solution / Application / Usage
TRAC-26	Add list of participants in current chat view	Major	CSET / Site and Personal Dashboard	**Final status:** Implemented. **Notes:** In chat view it is now possible to see current users of opened chat, implementation was based on the update of chat dashlet code.

At the end of this section, we add some details on all the requirements. The left-hand part of Figure 4 shows the distribution of requirements according to their origin. The table for step 1 with requirements originating from D1.1 and D2.1 contains 153 requirements, most of them are from initial set (INIT abbreviation used in figure, 135 requirements), others are from the state-of-the-art analysis provided in D2.1 (SOTA abbreviation used in figure, 8 requirements) or use case description / analysis (UC abbreviation used in figure, 10 requirements).

The table for step 2 with requirements from development phases processed as tickets in TRAC contains 50 requirements (as it was mentioned before, bugs and defects are not in the summary), TRAC abbreviation is used in figure.

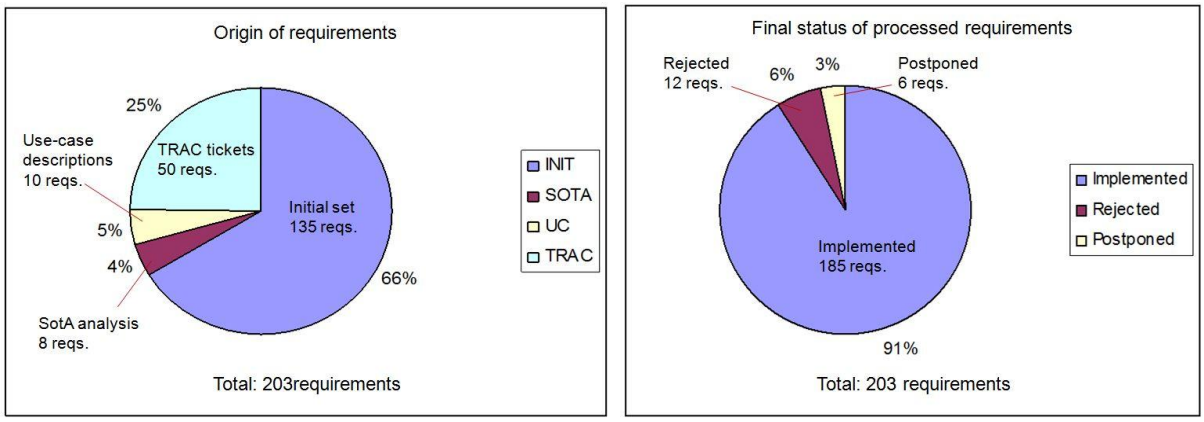

Figure 4: Distribution of requirements according to their origin (left part of the figure) and final status distribution of requirements (right part of the figure).

The right-hand part of Figure 4 shows the distribution of final status within the tables together (for all requirements). As we can see, most of 203 requirements are implemented at the end of the project, only small amount of them was rejected or postponed.

The rejection was mostly based on the fact that a requirement was revised during the project and was found not applicable for our project; postponed status usually means that described requirement seems to be interesting, but its realization is 'out of scope' for the implementation during the OCOPOMO project (due to several reasons, e.g., resources, currently suitable solution is not available, low priority makes it postponed for now, etc.) and could be possibly addressed in further revisions of the platform.

The list of all processed requirements is provided in Appendix A.

3. ADMINISTRATION OF THE OCOPOMO ICT TOOLKIT

The OCOPOMO ICT toolkit for collaborative policy modelling consists of several functional modules, components and tools, as it was presented in section 2.1 and depicted in Figure 3 above. The division of the toolkit to the web-based Alfresco and local Eclipse IDE parts implies specific requirements on the underlying hardware and software, installation and system maintenance, which will be presented in the next sections of this chapter.

The proposed installation of the OCOPOMO ICT toolkit is schematically depicted in Figure 5. Tools and user interfaces provided for collaborative formulation of policy scenarios are installed on the Alfresco web server and are accessible in a client-server mode. Alfresco provides a means for storage of shared data, namely scenarios and background documents, which can be accessed and remotely maintained by involved stakeholders using standard web browsers. Tools enabling discussions, online communication, opinion polling, or planning of collaborative scenario development process are provided as web applications as well.

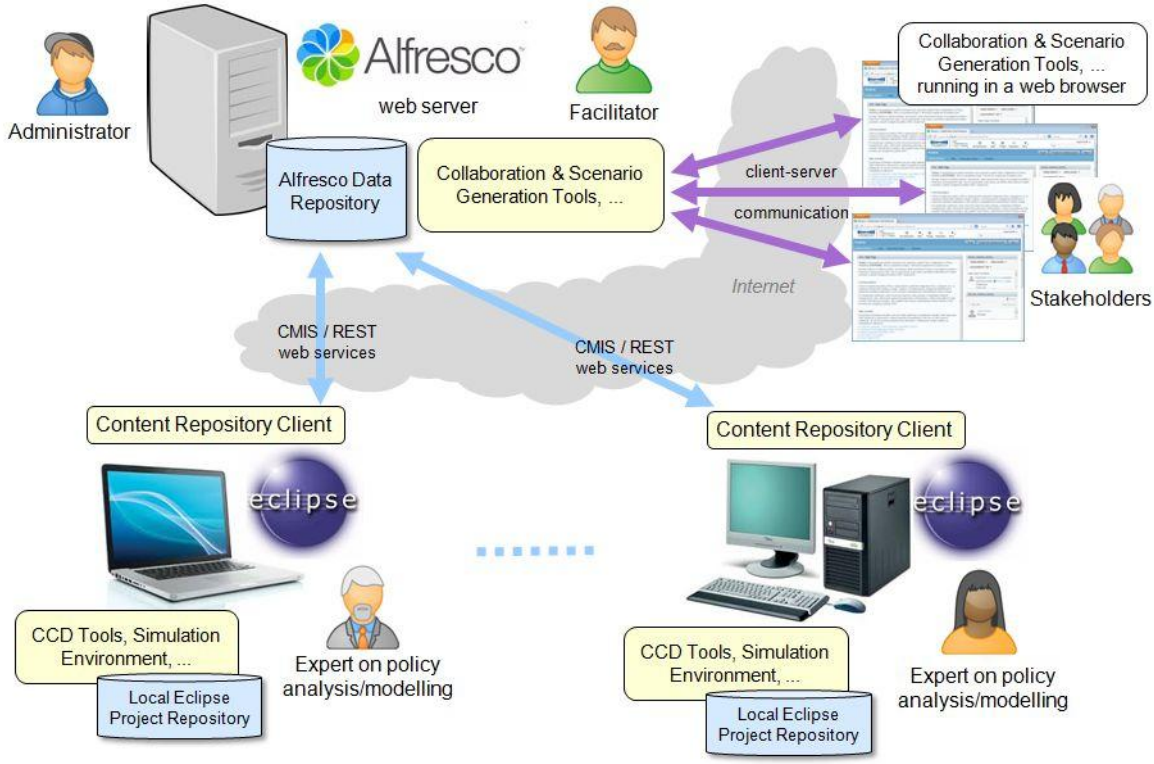

Figure 5: Schema of installed OCOPOMO ICT toolkit.

The toolkit dedicated for policy analysis and modelling experts is provided as a set of plug-ins running on a local Eclipse installation. Communication with Alfresco web space and data transfer to/from remote *Alfresco Data Repository* is supported by the *Content Repository Client* component, which is installed on each of Eclipse instances. It acts as an interface enabling file transfer between Eclipse and Alfresco environments via CMIS and REST web services. The communication is initiated and managed from the Eclipse side; the Alfresco repository enables remote upload and download of files upon proper authentication procedure.

The process of collaborative policy development is organised and controlled by facilitators who may use community moderation facilities (e.g., allowing or restricting access to some resources, initiating a discussion on some topic, controlling a flow of scenarios, etc.) provided by the Alfresco platform. Technical maintenance is under responsibility of administrators; however, since the OCOPOMO

system is composed of web-based Alfresco and locally installed Eclipse parts, administrators can directly manage only the Alfresco part. Installation, set up, and maintenance tasks related to the Eclipse part of OCOPOMO ICT toolkit should be performed mostly by particular end users - experts on policy analysis and modelling, with a remote assistance of facilitators and/or administrators of the OCOPOMO system. Methodology for managing the OCOPOMO process is provided separately, in the upcoming deliverable D8.1. Here we are focusing on general instructions for software installation and technical maintenance, which are presented in the next sections.

3.1. HARDWARE AND SOFTWARE REQUIREMENTS

The OCOPOMO system requires quite standard HW/SW equipment, which is determined by both underlying platforms of Alfresco and Eclipse. Please refer to the respective documentation of both vendors (Alfresco Share Documentation, 2012; Eclipse Indigo Documentation, 2013) and check the requirements against your infrastructure - e.g., operation system, configuration, etc.

Next we present the hardware and software requirements for both parts of the OCOPOMO system, assuming the installation and operation structure depicted in Figure 5 above - i.e., a single web server running Alfresco and at least one workstation running Eclipse IDE.

HW/SW requirements for Alfresco-based tools

The OCOPOMO components were developed and tested with Alfresco 4.0e Community edition software[4]. OCOPOMO components were partially tested and are compatible also with previous version Alfresco 3.6d Community edition. Older versions of Alfresco software are not supported. For newer versions of Alfresco software, please read release notes provided with the respective installation packages in order to check the configuration compatibility with the Alfresco 4.0e. Table 3 summarizes hardware and software requirements for the Alfresco-based OCOPOMO tools.

Table 3: Hardware and software requirements for the Alfresco part of OCOPOMO ICT toolkit.

Hardware requirements	
Server with at least 2x Core CPU, 2GB of main memory and 2GB of space on the hard disk (installation requires 1.5 GB for software, additional space is required depending on the size of the managed content, all content is stored as files on disk so for example for 1000 1MB documents, additional 1000+ MB of space will be required extended with the space for full-text index and temporary files).	
Windows, Linux or Mac OS operating system, recommended are 64-bit platforms	
Software requirements	
Oracle Java SDK version 1.6* or above (for Mac OS, Java 1.6 SDK from Apple is compatible); for CCD Model Explorer Applet the Java Runtime Environment (JRE) version 6 or higher is required	
Database server	Postgresql 9.0.4* MySQL 5.1, 5.5 Oracle 10g/11g MS SQL DB2 9.7.0.4

[4] https://wiki.alfresco.com/wiki/Community_file_list_4.0.e

Application server	Tomcat 6.0.4, 6.0.5* JBoss 5.1.1 Oracle WebLogic 11g
Client browser	Mozilla Firefox 4.0 MS Internet Explorer 7,9 Safari 5.1.1, 5.1.7 Chrome 15 Note that this is the list of browsers supported by the OCOPOMO Alfresco tools and the Alfresco software. Most of other versions can work with minor limitation and issues.

* default dependencies included in the installation package

HW/SW requirements for Eclipse-based tools

The Eclipse part of the OCOPOMO ICT toolkit was built on Eclipse Indigo IDE (Eclipse Indigo, 2011) and has been tested on Eclipse 4 Juno (Eclipse Juno, 2013) as well. Older versions are not supported. A summary of hardware and software requirements for the Eclipse-based OCOPOMO tools is provided in Table 4.

Table 4: Hardware and software requirements for the Eclipse part of OCOPOMO ICT toolkit.

Hardware requirements	
PC with at least 2GB of main memory (6 GB recommended); 1-1,5 GB of space on the hard disk. (Installation requires 0.5 GB for software, additional space is required depending on the size of the managed content - namely local files/documents, CCD models, DRAMS code, simulation logs. Additional 1+ GB of space is recommended.)	
Windows, Linux or Mac OS operating system, recommended are 64-bit platforms	
Software requirements	
Oracle Java SDK version 1.6* or above (for Mac OS, Java 1.6 SDK from Apple is compatible); for DRAMS simulation models Java JDK 1.7 is required	
Supported Eclipse packages	Eclipse *Indigo* or *Juno*, one of packages: • Eclipse Classic • Eclipse IDE for Java EE Developers • Eclipse IDE for Java Developers
Dependencies	Eclipse Rich Client platform* Eclipse modeling framework (EMF) * Eclipse graphical modeling framework (GMF) * Acceleo, ver. 3.3.0, http://www.eclipse.org/acceleo/ RepastJ 3, http://repast.sourceforge.net/repast_3/ Colt (distributed with RepastJ) Apache Log4j 1.2, http://logging.apache.org/log4j/1.2/ JGraphX, https://github.com/jgraph/jgraphx GATE, http://gate.ac.uk/download/ JPedal, http://www.idrsolutions.com/jpedal-downloads/

* default dependencies included in the installation package

3.2. DOWNLOAD AND INSTALLATION INSTRUCTIONS

Instructions for download of software components and their installation are provided for both Alfresco and Eclipse parts of the OCOPOMO system. The adopted modular system architecture, especially of the Eclipse part of the toolkit, allows a custom installation of provided plug-ins. However, to enable a smooth policy analysis and modelling process in its all required phases, it is recommended to install all the provided plug-ins and modules in the same Eclipse environment and make them available at the same time in a common Eclipse project. The same recommendation applies to the Alfresco part; however, a distributed installation of the Alfresco platform could be considered as well, as it is described later in the instructions.

Installation of Alfresco-based tools

The *Collaboration* and *Scenario Generation Tools*, together with the underlying *Content Management Server - Data Repository* and tools for *Traceability and Simulation Output Visualisation* (cf. Figure 3 in section 2.1), are based on the Alfresco software (Alfresco Share, 2011), an open source enterprise content management system. In order to install OCOPOMO Alfresco tools, please follow these steps:

1. Download the Alfresco installation package for your platform from https://wiki.alfresco.com/wiki/Community_file_list_4.0.e. Supported platforms include 32-bit and 64-bit Windows system, 64-bit Linux distributions and Mac OS platform. Before the installation, please check hardware and software requirements presented above in section 3.1.
2. Unpack the installation package if necessary and execute installation script/application. In the displayed dialog, select *Simple installation* and follow the instructions on the screen. Installation program will ask you to provide a path where the software will be installed on your hard drive, as well as a password for the system administrator of your computer. After the installation, please follow the post-installation procedure to check if the Alfresco installation is running properly.
3. After completing the installation of the Alfresco platform, please download and install the packages provided for OCOPOMO extensions, modules and configurations at http://ocopomo.ekf.tuke.sk/trac/ocopomoprj/wiki/Alfresco[5]. The suite of OCOPOMO Alfresco tools consists of the following installation packages which can be installed separately, depending on your preferences and requirements:
   - *commons* - contains Alfresco extensions for core components of the suite of OCOPOMO Alfresco tools, which are mostly related to *Document Library* and *Wiki* page components; localization files, traceability and annotation extensions are included as well.
   - *chat* - contains web application for chat server and *Chat* Alfresco Share dashlet.
   - *polling* - contains scripts for the *Pollings* page component.
   - *visualization* - contains web application and scripts for the *chart visualization* and *CCD model browsing*, integrated with the Alfresco collaboration tools.

   Download and extract selected packages on your hard drive to the path where the Alfresco software was installed. The *CCD Model Explorer* applet[6] should be placed into the `../tomcat/webapps/share/applets` folder.

[5] The mirror of installation packages and configuration bundles is available at http://www.ocopomo.eu/workspace/wp-04-integration-of-components-1/d4.1-integrated-platform/integrated-ict-toolkit/alfresco.

[6] The *CCD Model Explorer* applet can be downloaded and installed separately, as a stand-alone CCD model browser. Instructions and source files can be found at http://userpages.uni-koblenz.de/~ocopomo/ccdexplorer/.

4. Ensure that the Alfresco installation is currently not running (see instructions how to start or stop Alfresco platform in the subsequent section 3.3). Go to the expanded directory of the selected installation package and execute the deployment installation script for your platform (`deploy.bat` for Windows or `deploy.sh` for Linux or Mac OS). Deployment installation script will copy extensions files and configure the Alfresco installation.
5. Start your Alfresco installation and follow the post-installation procedure to check if Alfresco software and installed OCOPOMO components are working properly.

Note: The presented installation instructions are for the base installation where all components are installed on one server (see Figure 5 above). Depending on your requirements and number of users, Alfresco installation can be distributed in various ways where the database server, Alfresco Repository server, and Alfresco Share server can be installed on dedicated machine. For distributed installation, you need to install separately all required dependencies (i.e. Java SDK runtime, database server and application servers) on the particular machine and deploy separate packages for each component.

For more information about the Alfresco software installation please refer to the installation instructions for your particular OS/platform, which are available online at http://docs.alfresco.com/4.0/index.jsp?topic=%2Fcom.alfresco.enterprise.doc%2Fconcepts%2Fsimpleinstalls-community-intro.html.

Installation of Eclipse-based tools

Please follow these steps to install the Eclipse IDE:

1. Download Eclipse Indigo at http://www.eclipse.org/downloads/ and follow the instructions. Please ensure to use Indigo version. Older versions of Eclipse (e.g. Galileo or Helios) are not supported.
2. Start Eclipse.

After the general Eclipse environment is properly installed, you can proceed with installing particular plug-ins of the OCOPOMO ICT toolkit. Please follow the following steps to install the plug-ins for *CCD Tools*:

- Open the "*Install New Software*" Dialog by clicking on *Help -> Install New Software* in the Eclipse Menu.
- Click on the *Add* button to add a new installation repository. A dialog appears in which you need to give the repository a

  Name as e.g. "OCOPOMO" and a

  Location: http://userpages.uni-koblenz.de/~ocopomo/release/

  Click *OK* afterwards.
- If you are asked for user name and password, please use the following login data:

  User: ocopomo

  Password: Q$st*!56
- [optional] If the installation repository is not shown in the drop down box called "*Work with*", then please click on the drop down list and select the new installation repository which you just added in step 4.
- In the displayed *Install* dialog (see in Figure 6), select the check box *OCOPOMO* to install the *CCD Prototype*, the *CCD2DRAMS Prototype* and the *DRAMSEditorFeature*.

  Note: If nothing appears in the dialog, first deselect the *Group items by category* checkbox.

- Click *Next* and follow the instructions.

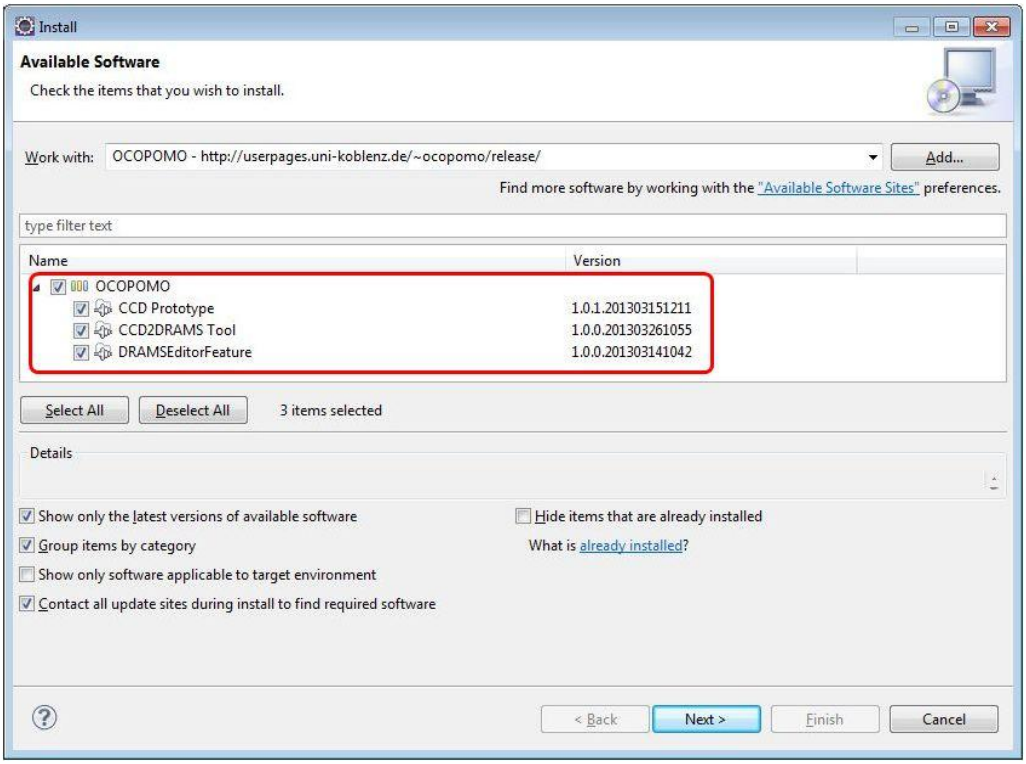

Figure 6: Installation of CCD Tools into the Eclipse.

After the installation of *CCD Tools* plug-ins, a set of accompanying plug-ins, so-called OCOPOMO Extensions, should be installed. The extensions include the *Content Repository Client* component, a set of *annotators* for PDF, HTML, and editable plain text formats, as well as the *Simulation Analysis Tool*. To install these extensions, please follow these instructions:

1. Open the "*Install New Software*" dialog by clicking on *Help -> Install New Software* in the Eclipse menu.
2. Click on the *Add* Button to add a new installation repository. A dialog appears in which you need to give the repository a

   Name as e.g. "OCOPOMO Extensions", and a
   Location: http://ocopomo.ekf.tuke.sk/svn/ocopomoprj/trunk/eclipse/updatesite/site.xml

   Click *OK* afterwards. No login/password is required for this installation repository.
3. [optional] If the installation repository is not shown in the drop down box called "*Work with*", then please click on the drop down list and select the new installation repository which you just added in step 2.
4. In the displayed *Install* dialog (see in Figure 7), select the plug-ins to be installed, namely:
   - *Content Repository Client*, *Html Annotator*, *MyNote Annotator*, and *Pdf Annotator* in the *CCDTool* package, and
   - *Simulation Analysis Tool* in the *Simulation Analysis Tool* package.

   Note: If nothing appears in the dialog, first deselect the *Group items by category* checkbox.
5. Click *Next* and follow the instructions.

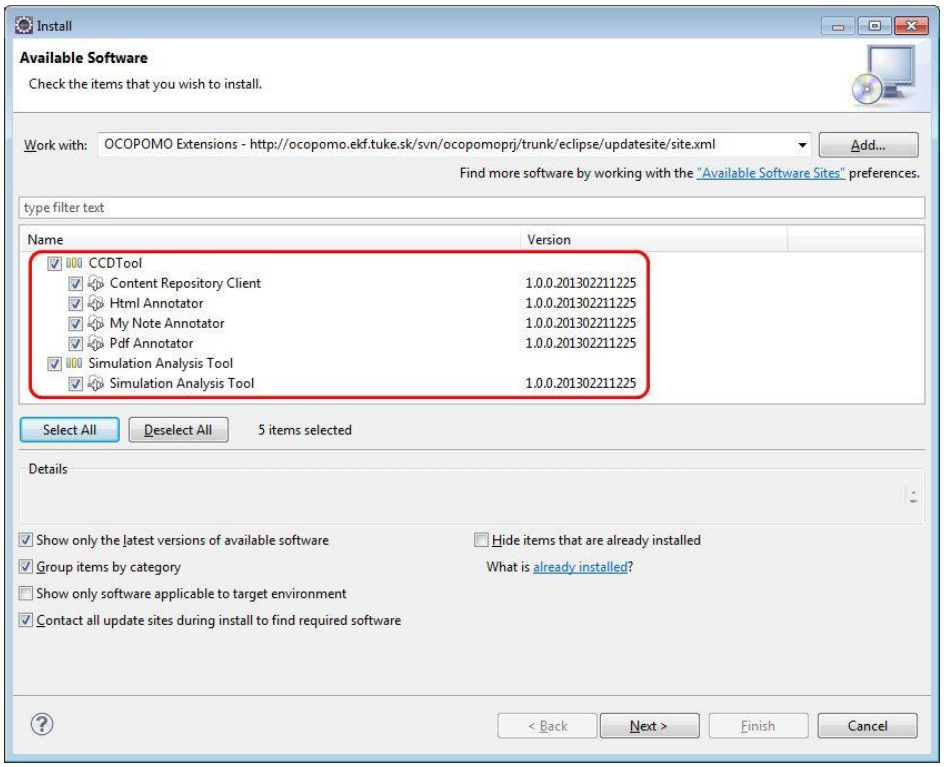

Figure 7: Installation of Annotation plug-ins into the Eclipse.

The installation of Eclipse tools continues with tools for agent-based policy modelling and simulation environment. This toolkit, provided as a number of Java class libraries (JAR files) in combination with an additional Eclipse editor plug-in, consists of DRAMS rule engine (Lotzmann, Meyer, 2011), supported by the run time simulation tool built on RepastJ 3.1 (Repast 3, 2008). To install the DRAMS policy modelling and simulation environment to Eclipse, please follow these instructions:

1. Install the DRAMS Editor plug-in for Eclipse (optional - supports editing of declarative rules):
   - Eclipse feature can be installed from the UKL OCOPOMO Eclipse Update Site at http://userpages.uni-koblenz.de/~ocopomo/release (access data - user name: ocopomo, password: Q$st*!56)
2. Install DRAMS:
   - by checking out the entire folder from SVN at https://svn.uni-koblenz.de/ocopomo/wp5/implementation/trunk/DRAMS (to obtain the access data, please send request to ocopomo@uni-koblenz.de), or
   - alternatively, by unpacking the DRAMS.zip file into an arbitrary folder on the local hard drive. To obtain the DRAMS.zip package containing complete installation and source files of DRAMS software, please contact the coordinator of the OCOPOMO project upon contacts available at http://www.ocopomo.eu.
3. Adapt project properties (if needed):
   - If there appear problem or error markers at the imported projects, this is usually due to missing libraries. All needed libraries (for the models supported by DRAMS) can be found in the /libraries subfolder of the DRAMS installation folder, so redirecting the respective Java build paths will solve the problems.

- Other missing libraries are usually a part of RepastJ 3.1, which can be downloaded from http://repast.sourceforge.net/repast_3/index.html.

Note: To update the installed plug-ins, please go to *Help -> Check for Updates*. Updates of some packages, e.g. CCD Tool, CCD2DRAMS, or DRAMS suite may require a prior un-installation of older versions from the Eclipse environment.

3.3. SYSTEM SET UP AND MAINTENANCE

This section outlines operational instructions and maintenance recommendations that are applicable after a successful installation of the OCOPOMO system. Again, the instructions are provided separately for Alfresco and Eclipse parts.

Set up and maintenance of Alfresco-based tools

Note: In the following text we will use %ALFRESCO to denote path to the directory where the Alfresco software is installed (default location is C:\Alfresco for Windows platform, \opt\alfresco for Linux and \Applications\alfresco-4.0.e for Mac OS X platform) and %DOMAIN to denote URL of the server where the Alfresco and OCOPOMO web applications are installed (if you are working at the same server where software is installed, you can use http://locahost:8080/ address).

How to start and stop OCOPOMO Alfresco tools?

Depending on your OS platform, you can use the following procedure to start or stop the OCOPOMO Alfresco Tools:

4. *Windows platform* - as default, Alfresco application server and Database server are installed as the system services. Please use *Services manager* tool in Control Panel/Administrative Tools to start or stop Alfresco and Alfresco Postgresql service. Optionally, you can use *Launcher* application installed by default in the Alfresco program group.
5. *Linux platform* - use `alfresco.sh` script in %ALFRESCO directory. To start tools, type `alfresco.sh start` and to stop running tools type `alfresco.sh stop`. Script will launch and stop both database server and application server. Additionally, for Ubuntu distribution, you can use Alfresco system start-up script in `/etc/init.d` repository.
6. Mac OS platform - you can use alfresco.sh script similarly as for Linux distribution or you can use Application Manager tool installed in Applications/alfresco-4.0.e folder.

Note that starting and stopping of the Alfresco server will take some time (especially the first start requires a more time to parse and cache initial scripts and databases). Check running processes and %ALFRESCO\alfresco.log file to track the start-up or stop process.

Post-installation procedure

In order to check if the OCOPOMO Alfresco tools were installed properly and are working correctly, please follow these steps:

1. Start OCOPOMO Alfresco tools (see previous set of instructions).
2. In your browser, navigate to %DOMAIN/alfresco page. Login dialog of the *Alfresco Explorer* should be displayed. Log in using the admin username and password you have selected during the installation. If you can log in successfully, then Alfresco repository is working properly.

3. Navigate to %DOMAIN/chat and %DOMAIN/visualization pages to check if chat server and visualization data source components were successfully installed. (Note that these components are optional).
4. Navigate to %DOMAIN/share page and login using the admin user and password you have selected during the installation. If you can log in successfully, Alfresco software is working properly.
5. After the log in into the Alfresco Share, navigate to the sample collaboration site created during the installation. Click on *Customize Dashboard* button and check if the *Chat* dashlet was successfully installed. Then click on the *More* button and select the *Customize Site* menu command. In the displayed page, check if the *Polling* page component was successfully installed. (Note that *Chat* and *Polling* components are optional.)
6. Now you can create your own collaboration site and invite collaborators. For more information, please read user manual for OCOPOMO Alfresco tools, provided separately in D4.2.-A *User Manual on Collaboration and Scenario Generation Tools*.

In case of any problems, please check %ALFRESCO\alfresco.log file and %ALFRESCO\tomcat\logs log files. Depending on the tracked problems, update your installation or change configuration options if required.

How to customize configuration options?

The main configuration file is located in %ALFRESCO\tomcat\shared\classes\alfresco-global.properties. In this file you can configure the following main aspects:

- *Common Alfresco properties* such as the location of the data directory where all user generated content is stored;
- *Database connection properties*, i.e., username, password, and database name for the connection to the database server;
- *External executables locations*, i.e., path to the external command line tools and applications used internally for example for the converting of the content into the different formats etc.;
- *Configuration of the full-text indexing*, i.e., to configure which tool will be used for full-text indexing and searching of the user content. Note that OCOPOMO Alfresco tools are compatible only with the internal Lucene indexing and external Solr indexing server is not supported. It means that the index.subsystem.name property has to be configured to Lucene.

Detailed information about the configuration options can be found on http://docs.alfresco.com/4.0/index.jsp?topic=%2Fcom.alfresco.Enterprise_3_4_0.doc%2Ftasks%2Fconfig-config.html.

How to backup and restore user data?

The following steps will describe how to (re)-store backup of the user data managed by the OCOPOMO Alfresco tools. Any modifications of the configuration options or any files installed for the extensions and components can be stored separately. To backup user data, please follow these steps:

1. Stop OCOPOMO Alfresco Server.
2. Backup the content from the following subdirectories in %ALFRESCO\alf_data directory:
   - contentstore
   - contentstore.deleted
   - audit.contentstore
   - backup-lucene-indexes

3. Backup database files using any tool required for you database server. For default configuration with Postgresql, use for example `pg_dump` command line utility to backup data in the dump SQL file. To use `pg_dump`, please ensure that database server is running (but not the Alfresco application server) and type:

   ```
   pg_dump -U alfresco alfresco -f {dumpfilename.sql}
   ```

   where `{dumpfilename.sql}` is the name of the SQL file where data will be stored. To login into the database, use default admin password.

To restore user data, please follow the same steps in the backward order. To restore Postgresql database files for the default configuration, type the following command

```
psql -U alfresco -d alfresco -f {dumpfilename.sql}
```

More information how you can backup and restore a content generated by users can be found on http://wiki.alfresco.com/wiki/Backup_and_Restore.

Other maintenance tasks

For troubleshooting and tracing of the problems, you can check logging files. Logs are primary recorded in the %ALFRESCO\alfresco.log file and in the %ALFRESCO\tomcat\logs directory. Logging can be configured in the %ALFRESCO\tomcat\webapps\alfresco\WEB_INF\classes \log4j.properties and logging.properties files.

The following issues are common:

- Alfresco repository did not start because index directory was locked. This may happen during the incorrect shutdown of the server. The solution is to delete the %ALFRESCO\alf_data\lucene.indexes and %ALFRESCO\alf_data\backup-lucene-indexes directory and re-index the content using the following procedure:
  1. Open global configuration properties file (see above the ***How to customize configuration options?*** set of instructions);
  2. Edit or add the *index.recovery.mode* property and set its value to *FULL*.
  3. Start the Alfresco server (see above the ***How to start or stop OCOPOMO Alfresco tools?*** set of instructions) and check progress of re-indexing in the log files.
  4. Stop Alfresco server and modify *index.recovery.mode* to default *AUTO* value.
  5. Start Alfresco server.
- Various exceptions are reported to the user during the login, such as content not found etc. This may happen during the incorrect shutdown of the server when the full-text index, database files and content files are not synchronized properly. In order to synchronize full-text index, follow the previous procedure for full re-indexing. If re-indexing will not solve the issues, restore all data from the backup copy (see above the ***How to backup and restore user data?*** set of instructions).

Set up and maintenance of Eclipse-based tools

How to prepare the Eclipse for OCOPOMO operation?

After the OCOPOMO tools are properly installed in Eclipse, a common working environment should be established for each of analysed policy cases. Such a working environment is represented by Eclipse Java project using *File* → *New*; then *General* → *Project*. Further instructions of setting up a working environment for CCD models are provided in section 2.4 of D4.2-B *User Manual on CCD Tools*.

Next, a data transfer connection between local Eclipse installation (namely, the selected Java project) and the remote Alfresco web space needs to be established. It can be done by setting up the properties of the *Content Repository Client* component. For a detailed description of this procedure, please refer to section 2.5 of D4.2-B *User Manual on CCD Tools*.

How to adapt Eclipse for DRAMS?

Please ensure that your Eclipse installation uses an JDK (Java Development Kit) instead of a JRE for all models using mathematical expressions in DRAMS. If a respective error message appears when running the model, the following procedure can be applied to make the JDK available for Eclipse (there might be other ways as well):

- go to "*Run Configurations*..." in Eclipse,
- select your model in the list viewer on the left,
- choose the tab "*JRE*",
- select "*Alternate JRE*",
- press "*Installed JREs*...", and
- "*Add*..." another entry by completing the wizard dialog.

During this procedure it is necessary to specify the location of the JDK on the hard disk by selecting the "*Directory*..." for "*JRE home*".

How to update new versions of plug-ins installed in Eclipse?

You can update the plug-ins installed in your Eclipse environment at any time, using a standard Eclipse functionality. To update the installed plug-ins, please go to *Help -> Check for Updates*.

4. USAGE OF TOOLS IN THE OCOPOMO PLATFORM

This chapter describes the usage of end-user tools in particular phases of the OCOPOMO process. It is assumed that the whole toolkit is properly installed (see instructions in section 3.2) and is ready for regular operational use.

For each set of end user tools, divided into five groups and presented in a sequence of OCOPOMO process phases, general descriptive information is presented next in sections 4.2-4.6. Furthermore, instructions for the most common operations that can be performed by end users are described in a "how to" form of questions and answers and are provided in five separate documents - specific user manuals, as follows:

1. *D4.2-A: User Manual on Collaboration and Scenario Generation Tools*;
2. *D4.2-B: User Manual on CCD Tools*;
3. *D4.2-C: User Manual on Policy Modelling and Simulation Tools*;
4. *D4.2-D: User Manual on Tools for Simulation Analysis and Output Scenario Generation*.
5. *D4.2-E: User Manual on Simulation Output Visualisation and Traceability Tools*

The user manuals are accompanied by video presentations demonstrating the most common operational tasks, namely:

Install OCOPOMO_Video.zip - instructions on the installation of the OCOPOMO plug-in to the Eclipse environment.
Applied to the OCOPOMO process phase **#3**, referenced in **D4.2.-B**.

CCD_Video.zip - instructions on how to create a CCD and how to create actors, objects, instances and relationships between them.
Applied to the OCOPOMO process phase **#3**, referenced in **D4.2.-B**.

Annotators_part1_Video.zip and ***Annotators_part2_Video.zip*** - usage instructions for *PDF Annotator*, *HTML Annotator*, and *Content Repository Client* components.
Applied to the OCOPOMO process phase **#3**, referenced in **D4.2.-B**.

MyNoteAnnotator_Video.zip - usage instructions for *MyNote Annotator*, i.e., the annotation of editable texts.
Applied to the OCOPOMO process phase **#3**, referenced in **D4.2.-B**.

SimAnalysis_Video.zip - usage instructions for *Simulation Analysis Tool*, i.e., the annotation of simulation logs and production of a model-based scenario.
Applied to the OCOPOMO process phase **#5**, referenced in **D4.2.-D**.

MBScenarioPublish_Video.zip - usage instructions for *Simulation Analysis Tool*, publishers, and *Simulation Output Visualisation* components.
Applied to the OCOPOMO process phases **#5 and #6**, referenced in **D4.2.-D** and **D4.2.-E**.

The user manuals containing instructions for all phases of the OCOPOMO process are introduced in the following sections and detailed in separate documents - specific user manuals, which are available online at http://www.ocopomo.eu/workspace/wp-04-integration-of-components-1/d4.2-system-and-user-documentation/user-manuals (again, only for users registered at the OCOPOMO web site, http://www.ocopomo.eu).

4.1. USER ROLES

For the underlying OCOPOMO process, which is supported by the set of tools presented in the following sections and the respective user manual documents, the user roles were proposed in the D2.1 deliverable (Mach et al, 2010) as it is presented in Table 5.

Table 5: User roles applied in the OCOPOMO system.

Icon	User Role	Description
	**Politician**	A decision maker; initiates collaborative policy development and participates in scenario formulation (namely the initial scenario).
	**Civil servant**	An assistant of decision makers; provides supportive background materials and participates in (initial) scenario formulation.
	**Stakeholder**	End user (citizen, NGO, SME, ...); actively participates in the construction of evidence-based scenarios.
	**Facilitator**	A mediator who supervises and methodologically controls the collaboration working space.
	**Analyst**	An expert who investigates scenarios and creates the corresponding conceptual model.
	**Modeller**	An expert who constructs formal policy model and deploys it to the respective simulation environment.
	**Administrator**	A person responsible for the installation and technical maintenance of the whole OCOPOMO system.

The "how to" descriptions of particular operations, presented in the five user manuals D4.2 A-E, are labelled by respective icons of user roles that are assumed to perform the actions. This labelling consequently allows the transformation of the guidelines into an on-line form that can be targeted to specific user groups.

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).

4.2. PRODUCTION OF EVIDENCE-BASED SCENARIOS

The first two steps of the OCOPOMO process (cf. Figure 1 in section 1), i.e. the definition of an initial policy scenario and the collaborative development of evidence-based scenarios, are supported by the OCOPOMO *Collaboration* and *Scenario Generation Tools*. This suite of tools is provided as an extension of the Alfresco Share platform (Alfresco Share, 2011) enabling highly collaborative web-based content management around projects and activities. An OCOPOMO policy modelling project, which typically corresponds to a single initial policy scenario, is represented in the dedicated Alfresco

Share installation by a shared collaboration space – web site consisting of several content management and collaborative discussion components, as it is schematically depicted in Figure 8.

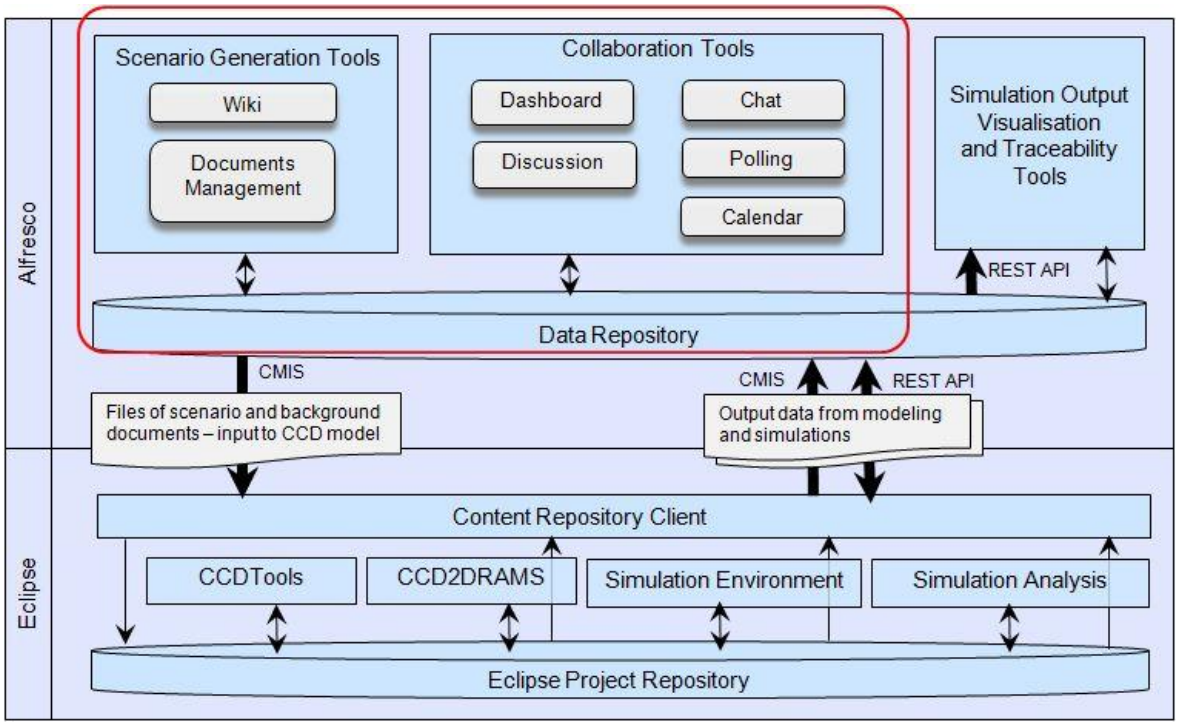

Figure 8: OCOPOMO ICT toolkit - highlighted tools and system components for a collaborative production of input evidence-based scenarios.

The Alfresco Share installation is highly customisable and could be adjusted globally by site administrators as well as locally by each of users (Alfresco Share Documentation, 2012). However, the OCOPOMO ICT toolkit provides a default configuration of the Alfresco Share-based policy modelling web site (see in Figure 9) that consists of the following components:

- ***Scenario Generation Tools***:
    - *Wiki* - a space of editable texts in the wiki HTML format, which serves as the main collaborative space for formulating user-generated policy scenarios. The *Wiki* link in the navigation bar displays the introductory wiki page, which may contain the initial scenario for a given policy case. Particular evidence-based scenario alternatives can be created after clicking the *New Page* link in the navigation sub-bar. The list of all available wiki pages is displayed after clicking the *Wiki Page List* link.
    - *Documents Management*, invoked by *Document Library* link in the navigation bar, provides the shared space for all involved users to store and access files in any format in a hierarchy of folders.

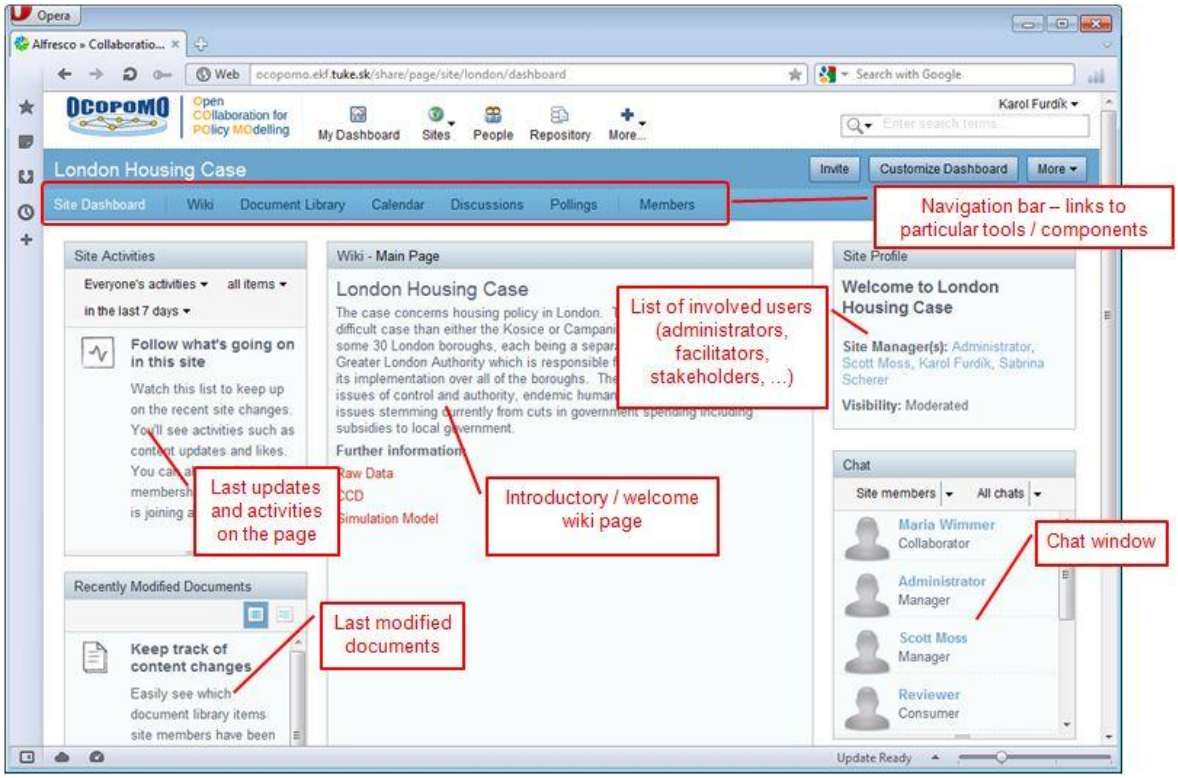

Figure 9: Web interface of the Alfresco Share-based policy modelling site.

- ***Collaboration Tools***:
  - *Dashboard* - global site and user personal dashboards, accessible after clicking on the *Site Dashboard* link in the navigation bar, provide an overview information about the project(s) together with the links to access the most recent content and activities. Users may configure the local layout of the personal dashboard page and displayed components (so-called dashlets) according to their preferences by means of the *Customize Dashboard* button above the navigation bar. Each site has its own default dashboard structure, which can be configured and adjusted by site administrators.
  - *Discussion* - a discussion forum space that allows creating discussion topics and posting messages.
  - *Chat* - enables a real-time online communication between the site members.
  - *Polling* - allows publishing of various questionnaires and opinion pollings.
  - *Calendar* - manages shared events assigned to the particular dates.

In addition to these OCOPOMO-specific components, several dashlets provided inherently by the Alfresco Share platform are available on the personal or site dashboard - for example, *Recently Modified Documents*, *RSS Feed*, *Image Preview*, etc. A complete list of the supported dashlets, as well as a detailed description of end-user operations that can be performed on all the implemented OCOPOMO-specific components, is provided in a separate document *D4.2-A: User Manual on Collaboration and Scenario Generation Tools*.

Finally, since the provided dashlets and components are often based on standard Alfresco software, useful guidelines can also be found on the Alfresco Share documentation page accessible at http://docs.alfresco.com/3.5/index.jsp.

4.3. ANALYSIS AND CONCEPTUAL MODELLING OF INPUT SCENARIOS

Third phase of the OCOPOMO process (cf. Figure 1 in section 1), i.e. the development of conceptual models of policy alternatives, is supported by a set of Eclipse plug-ins integrated into the *CCD Tools* package. This suite of tools is accompanied by components providing an interface and input data transfer from Alfresco to the Eclipse environment, which is enabled by the *Content Repository Client*, as well as the output data transformation from conceptual models to the simulation environment, provided by the *CCD2DRAMS Transformation Tool*. The scope of the conceptual modelling tools in the overall OCOPOMO toolkit architecture is presented in Figure 10.

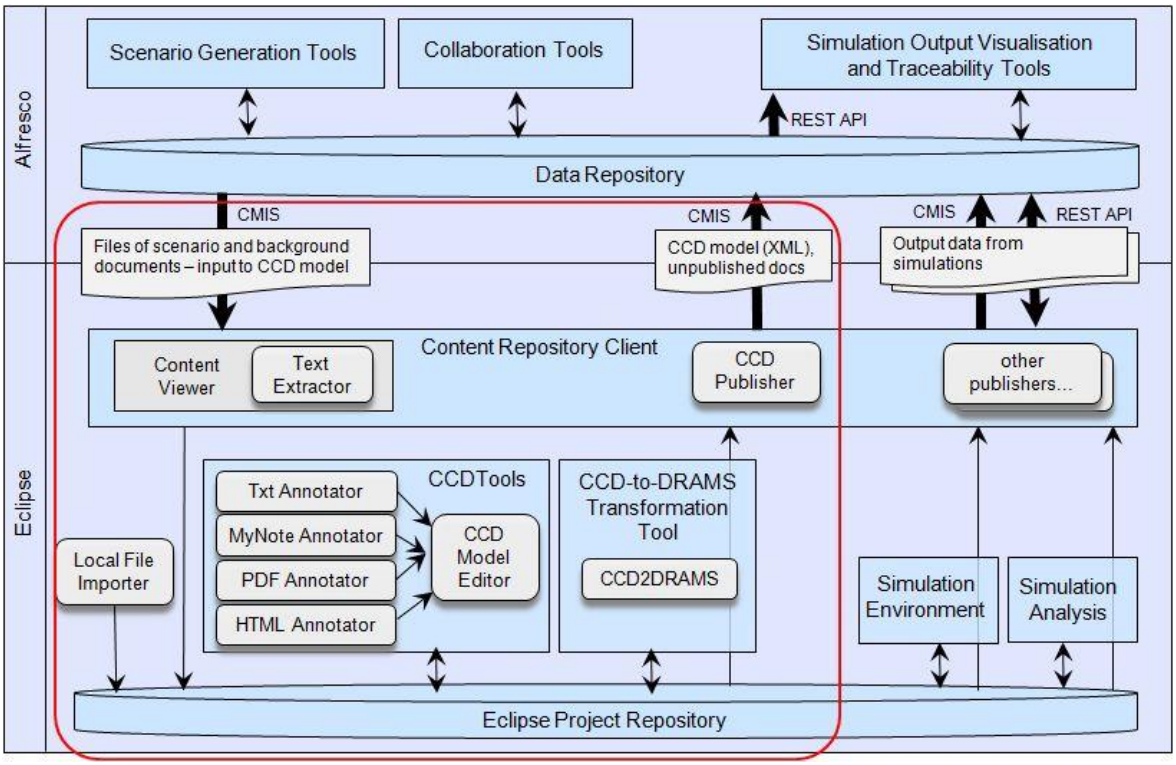

Figure 10: OCOPOMO ICT toolkit - highlighted tools and system components for conceptual modelling of policy alternatives.

The Eclipse Indigo environment (Eclipse Indigo, 2011), on which the *CCD Tools* are installed (see section 3.2), is highly customisable and could be adjusted according to the needs of particular end users. Typical layout of the user interface of *CCD Tools* and maintained information resources, provided as default configuration of the OCOPOMO ICT toolkit, is depicted in Figure 11. The suite of *CCD Tools* consists of the following components:

- ***Content Repository Client*** is the core integration component that enables a communication and data exchange between the Alfresco and Eclipse parts of the integrated OCOPOMO platform. Both parts can produce new data artefacts that have to be accessible simultaneously by the tools on the Alfresco as well as on the Eclipse site.
    - *Content Viewer* - an interface that visualises the *Alfresco Data Repository* in the Eclipse environment. It includes the *Text Extractor* component, which allows an extraction of plain texts from the wiki HTML format of Alfresco.
    - a set of *Publishers* (i.e., *CCD Publisher*, *DRAMS Publisher*, *Simulation Log Publisher*, and *Wiki Page Publisher*), provided as inner components of the *Content Repository Client*, enable upload and download of specific data types from Alfresco to

Eclipse and vice versa, by means of CMIS or REST protocols. Publishers do not provide a visible user interface; they are employed by other tools for transferring the data between Alfresco and Eclipse environments.

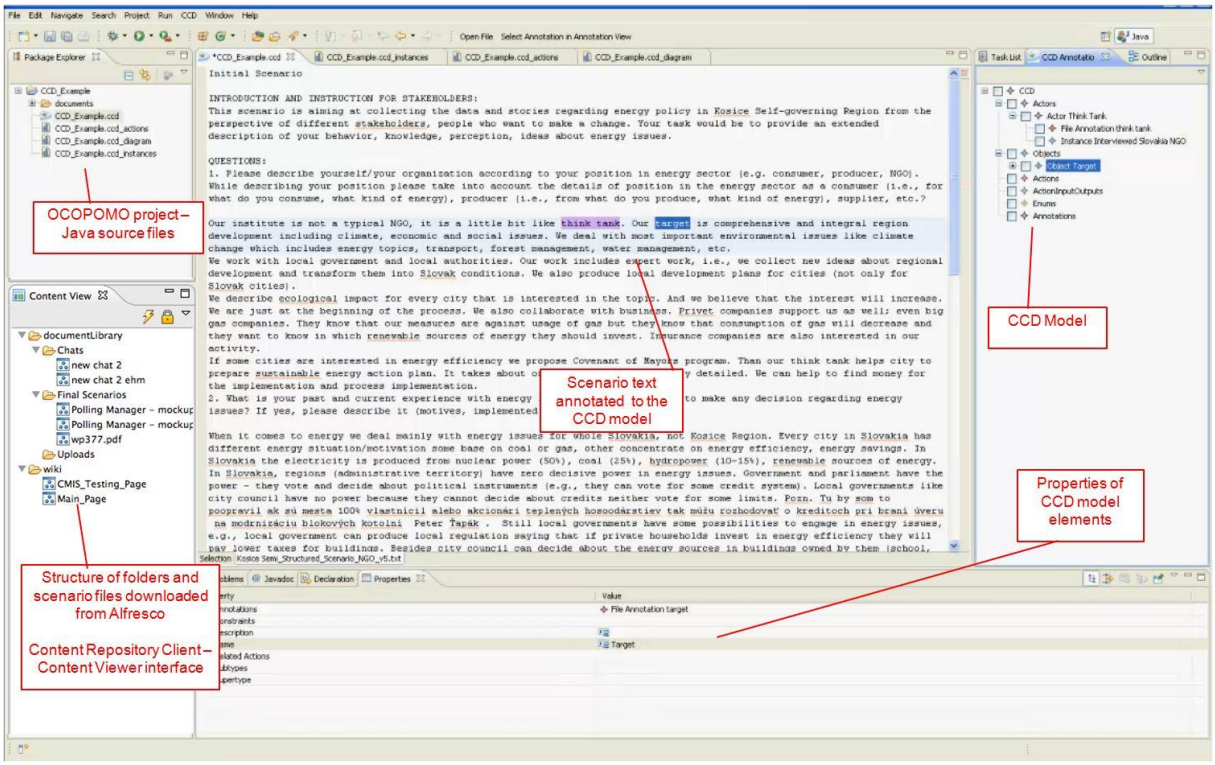

Figure 11: User interface of CCD Tools in Eclipse.

- ***CCD Tools***:
  - *CCD Model Editor* - a central tool for conceptual policy analysis and CCD model development. It enables creation and maintenance of CCD models consisting of various entities such as *Actor*, *Object*, *Relation*, *Attribute*, *Instance*, etc. The tool provides several views - diagrams for manipulation with the CCD elements. The CCD model is stored locally in a `*.ccd` file. In addition to the "native" CCD entities, the model includes all annotations represented as links/references between a CCD element and a text fragment in a document, e.g. an input scenario. This conceptual structure serves as a bridge between input evidence-based scenarios and executable agent-based policy models.
  - *Txt Annotator* - a plug-in that allows to annotate plain text documents, i.e., to create 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.
  - *PDF Annotator* - a plug-in enabling the annotation of documents in PDF format (both local files and external documents downloaded from Alfresco).
  - *HTML Annotator* - a plug-in enabling the annotation of documents in HTML format, namely Alfresco wiki pages containing the input evidence-based scenarios, as well as local HTML documents.

- *MyNote Annotator* - a plug-in that enables flexible editing and annotation of texts by preserving existing annotations during the text editing. This feature can be used for scenario analysis; however, it is especially employed during the 5th phase of OCOPOMO process to formulate an output model-based scenario iteratively, according to results obtained from several simulation cycles (see also in section 4.5).

- ***CCD-to-DRAMS Transformation Tool***:
  - *CCD2DRAMS* - inner plug-in component allowing an automatic transformation of produced CCD models to the respective stubs of Java and DRAMS source codes of corresponding agent-based policy models.

More detailed description and usage guidelines for all these tools and components belonging to the *CCD Tools* suite are provided in a separate document *D4.2-B: User Manual on CCD Tools*.

Finally, since the provided plug-ins and components are based on the Eclipse platform, useful guidelines can also be found on the Eclipse documentation page (Eclipse Indigo Documentation, 2013) as well as on the general Eclipse web site, http://www.eclipse.org.

4.4. FORMAL POLICY MODEL DEVELOPMENT AND SIMULATIONS

Next steps of the OCOPOMO process (cf. Figure 1 in section 1), namely its fourth phase that includes the programming of executable agent-based policy models, and partly also the fifth phase - execution of model-based simulations, are supported by the tools integrated into the *Simulation Environment* package. These tools, implemented as Eclipse plug-ins, provide an extensive IDE for developing, debugging, and running simulations of agent-based policy models. The scope of the *Simulation Environment* tools in the overall OCOPOMO toolkit architecture is presented in Figure12.

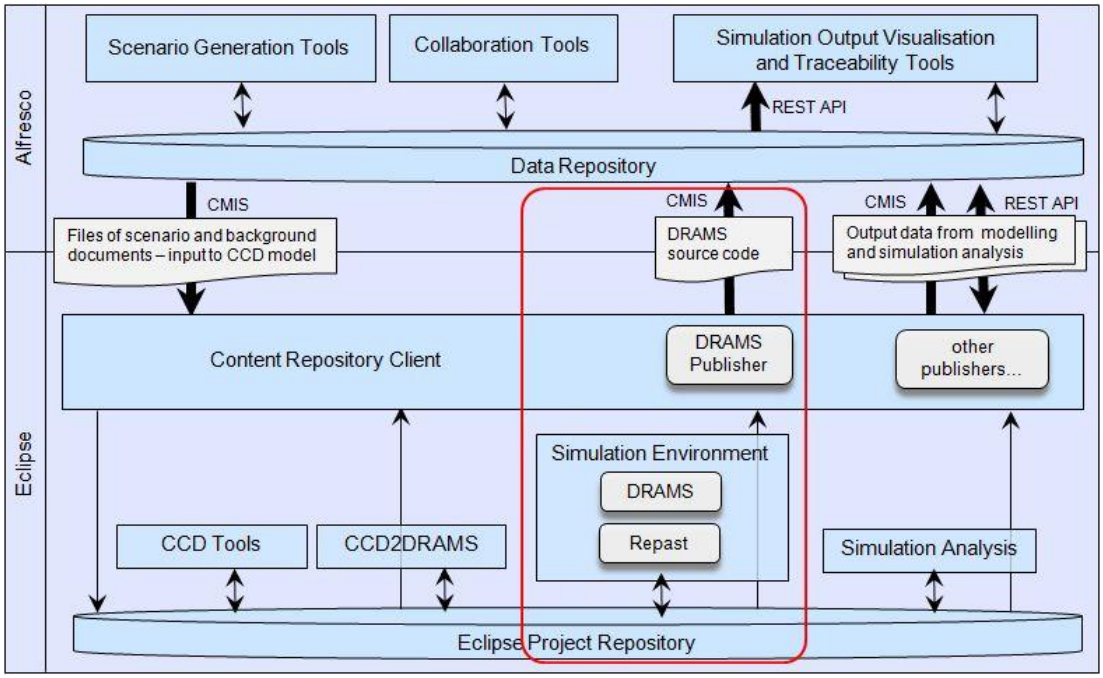

Figure 12: OCOPOMO ICT toolkit - highlighted tools and system components for agent-based policy modelling and simulations.

The agent-based policy model, together with related simulations, is developed within the OCOPOMO Java project created during the previous phase of the process. The Java project already contains the

CCD model as a conceptual representation of the analysed policy, as well as a stub of Java and DRAMS code of the agent-based model, created by the *CCD-to-DRAMS Transformation Tool* (see in previous section 4.3). The code stubs are taken as a basis for further encoding of executable models. The *Simulation Environment* package provides a rich set of tools - Eclipse plug-ins for editing and debugging both Java code of the executable models and DRAMS code of declarative rules describing a behaviour of agents included in the model. In addition, the package includes the environment for running simulations, which consists of the *DRAMS rule engine* (Lotzmann, Meyer, 2011) and the *Agent simulation engine* based on RepastJ 3.1 (Repast 3, 2008). Inner structure of the *Simulation Environment* module is rather complex; however, it could be logically divided into several groups of end user tools and system components as follows:

- ***Code editing tools***, i.e., editors of declarative rules and Java code editors, provided as Eclipse plug-ins, namely:
    - *DRAMS rule editors* - a set of views enabling the development and editing of declarative rules for the DRAMS model of agents, expressed in DRAMS-specific OPS5-like formal language. The respective user interface of DRAMS editors is depicted in Figure 13. The DRAMS model is constructed as a structure of facts and rules, which is presented in the *Data Dependency Graph* (DDG). Relationships between particular rules are visualised in the *Rule Dependency Graph* (RDG). The built-in scheduling mechanism may be used to control the order and priority of invoking rules during the model execution in a simulation run (see in Figure 14).
    - *Java objects editor* - an Eclipse built-in environment for editing the Java code of Repast models, based on the Repast J 3.1framework. To obtain guidelines and tutorials for encoding Repast-based simulation models, please refer to the Repast documentation at http://repast.sourceforge.net/repast_3/tutorials.html.

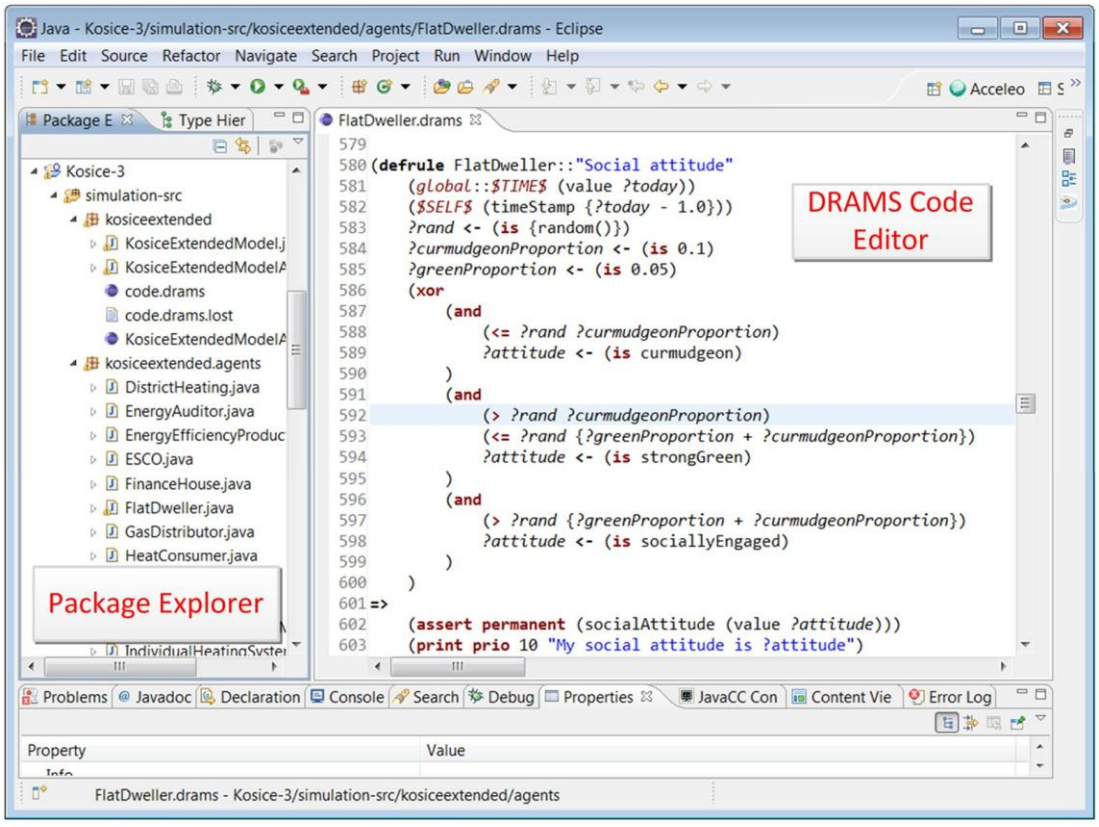

Figure 13: Programming of declarative rules in DRAMS.

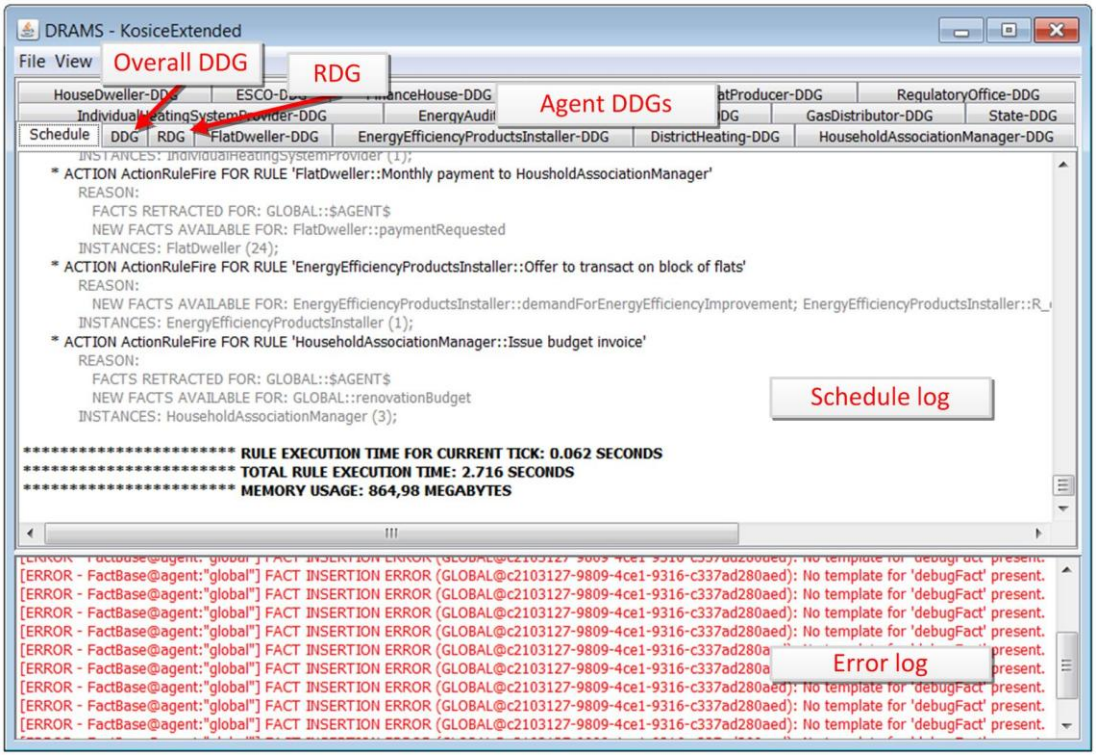

Figure 14: Monitoring model structure and execution with DRAMS.

- ***Code debugging tools*** for both Java code in Repast and for declarative rules DRAMS, namely:
  - *Java simulation code debugger* - a Repast-based mechanism providing the debugging and code inspection functionalities for the created Java code of simulation models.
  - *DRAMS code debugger* - the tracing and testing mechanism that enables an integrated debugging of Repast/DRAMS models. It allows, for example, running the simulation model step by step, setting breakpoints at any point within any rule and to display the internal state of rule evaluation after reaching a breakpoint.

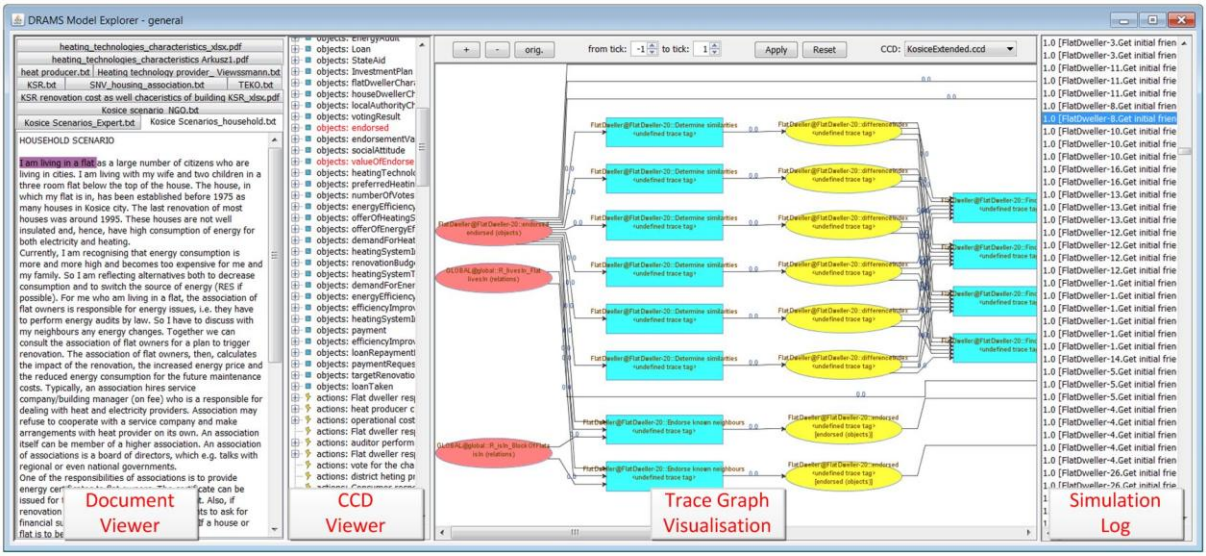

Figure 15: Views of the DRAMS Model Explorer toolkit in Eclipse.

- ***DRAMS-Repast simulation environment***, a set of tools and components that provide a means for executing the simulations of developed agent-based policy models, as well as to generate and store outputs obtained from the executed simulations, namely:
    - *DRAMS Model Explorer* - the Eclipse plug-in enabling execution and debugging of the simulation model together with the related conceptual information and input narrative scenarios. The tool interface, presented in Figure 15, includes views of a scenario text with highlighted text portions (*Document Viewer*), which were linked to particular CCD model elements (*CCD Viewer*), which were transformed to the rules and clauses of the declarative policy model (*Trace Graph Visualisation*). The executed simulation of the policy model results in a list of log records (*Simulation Log*), where a log corresponding to particular rule or clause can be identified and traced back to respective CCD model elements and consequently to related text fragments of input scenario.
    - *Repast simulation environment* - a set of Eclipse plug-ins, provided by the RepastJ 3.1 framework (Repast 3, 2008), that enable running the simulations of the policy model expressed in Java code. The user interface (see Figure 16) includes the *Control Panel* for the simulation run maintenance, as well as output views of simulation results in both graphical and textual formats.
    - *Output writer* - a component that collects the output data (i.e., simulation logs) produced during the simulation run and stores it persistently to the space of underlying Eclipse Java project.
    - *DRAMS Publisher* - inner component of the *Content Repository Client* (see above in section 4.3), which uploads the produced simulation logs to the Alfresco web space. The publisher is invoked by the *Simulation Analysis Tool* during the fifth phase of the OCOPOMO process, as it is described next in section 4.5.

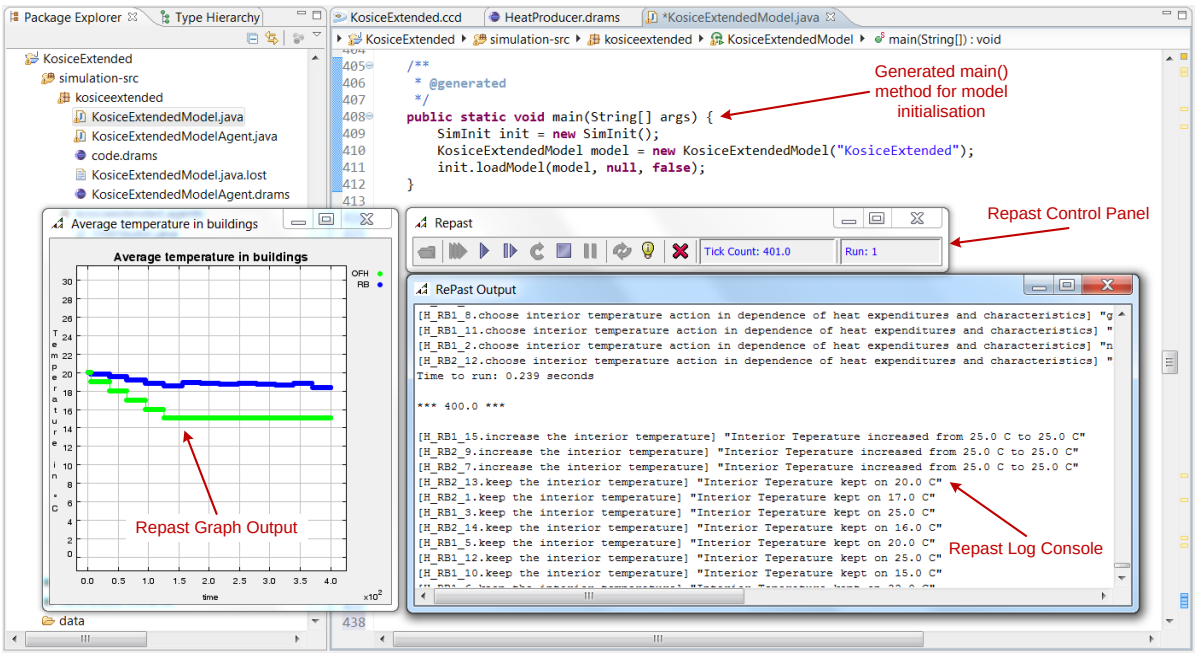

Figure 16: Running model-based simulations in Eclipse.

The development of declarative agent-based policy models is rather complex and long-lasting task, which should be performed by skilled expert users that belong to the *Modeller* user role (cf. section 4.1). Technical maintenance of local Eclipse installations should be performed by *Modellers* as well, with an assistance of *Administrators*.

More detailed description and usage guidelines for the tools included in the *Simulation Environment* can be found in a separate document *D4.2-C: User Manual on Policy Modelling and Simulation Tools*. Namely, the guidelines include a detailed presentation of the DRAMS language, its syntax and usage description. The DRAMS-Repast toolkit is also presented in several scientific publications such as, for example, (Lotzmann, Meyer, 2011; Lotzmann, Wimmer, 2012).

To obtain methodological instructions and examples on how to construct the declarative agent-based policy models, please refer to OCOPOMO deliverables D5.1 (Moss et al, 2011), D6.1 (Moss et al, 2013) and D8.1 (Scherer et al, 2013).

4.5. ANALYSIS OF SIMULATION RESULTS AND PRODUCTION OF OUTPUT MODEL-BASED SCENARIOS

Fifth phase of the OCOPOMO process (cf. Figure 1 in section 1) covers the analysis of outputs obtained from experimental simulations of agent-based policy models constructed in previous step (see in section 4.4). Simulation logs are analysed and transformed to narrative model-based scenarios, which are then uploaded back to the Alfresco web space. These actions of simulation output analysis, model-based scenario generation, and publishing of scenarios, together with all related data such as CCD models, simulation logs, etc., are supported by a set of Eclipse plug-ins integrated into the *Simulation Analysis* package. Upload of resulting scenarios and related data structures to Alfresco is enabled by the *Content Repository Client*, which acts as a middleware interface between Eclipse and Alfresco environments (see also in section 4.3). The scope of the *Simulation Analysis* tools in the overall OCOPOMO toolkit architecture is presented in Figure 17.

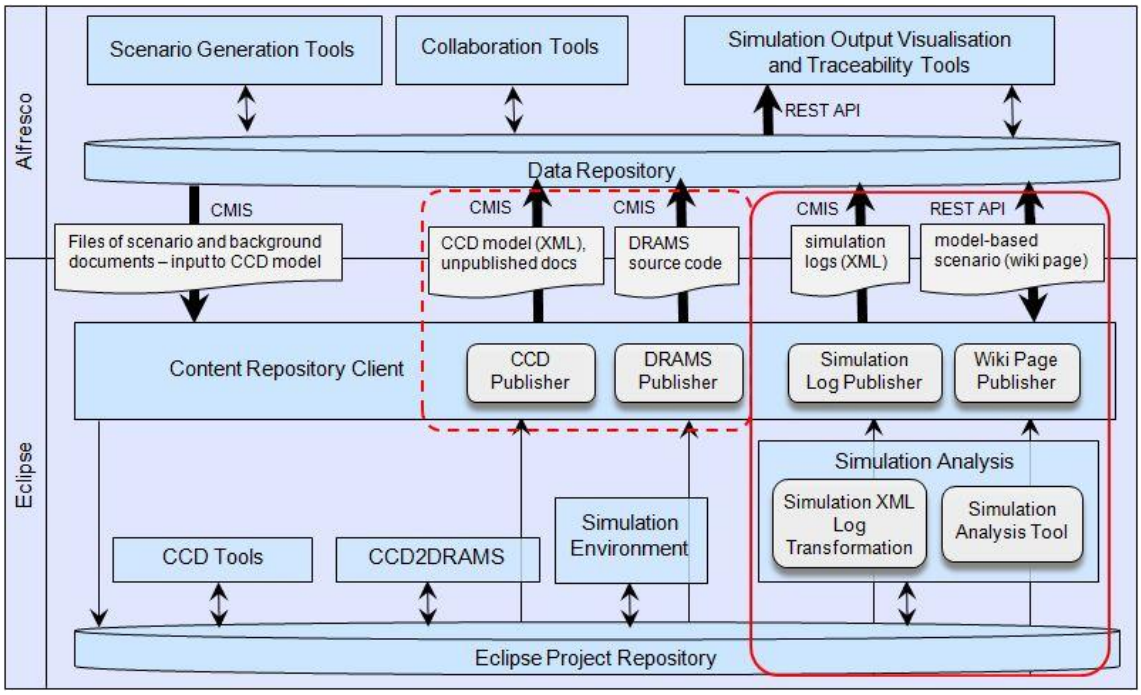

Figure 17: OCOPOMO ICT toolkit - highlighted tools and system components for analysis of simulation results and development of output model-based narrative scenarios.

Functionality of the simulation results analysis and model-based scenario development is provided by the *Simulation Analysis Tool*, an Eclipse plug-in that serves as the main user interface and managing console for all the tools included in the *Simulation Analysis* package. As it is depicted in Figure 18, the *Simulation Analysis Tool* integrates the underlying components enabling partial tasks such as the

format transformation and visualisation of simulation logs, the editing of scenario narratives, the annotation of scenario text fragments to respective simulation log records, and publishing of resulting scenarios to the Alfresco web space. Namely, the *Simulation Analysis* package consists of the following tools and system components:

- ***Simulation XML Log Transformation***, i.e., a set of inner components providing the simulation logs in supported formats for further annotation of the textual content of model-based scenarios. Simulations of developed agent-based policy models, executed in DRAMS-Repast environment, provide outcomes in a form of simulation logs, which are available in several data formats such as XML, plain text records, and CSV tables. The *Simulation XML Log Transformation* module makes the simulation logs available in the Java project space of Eclipse, namely in the `../data/experiments/run-<date-number>` folders of the project space (see on the left hand side of Figure 18). The logs may then serve as the main background information for further processing and development of output model-based scenarios, which is performed by means of the *Simulation Analysis Tool*.

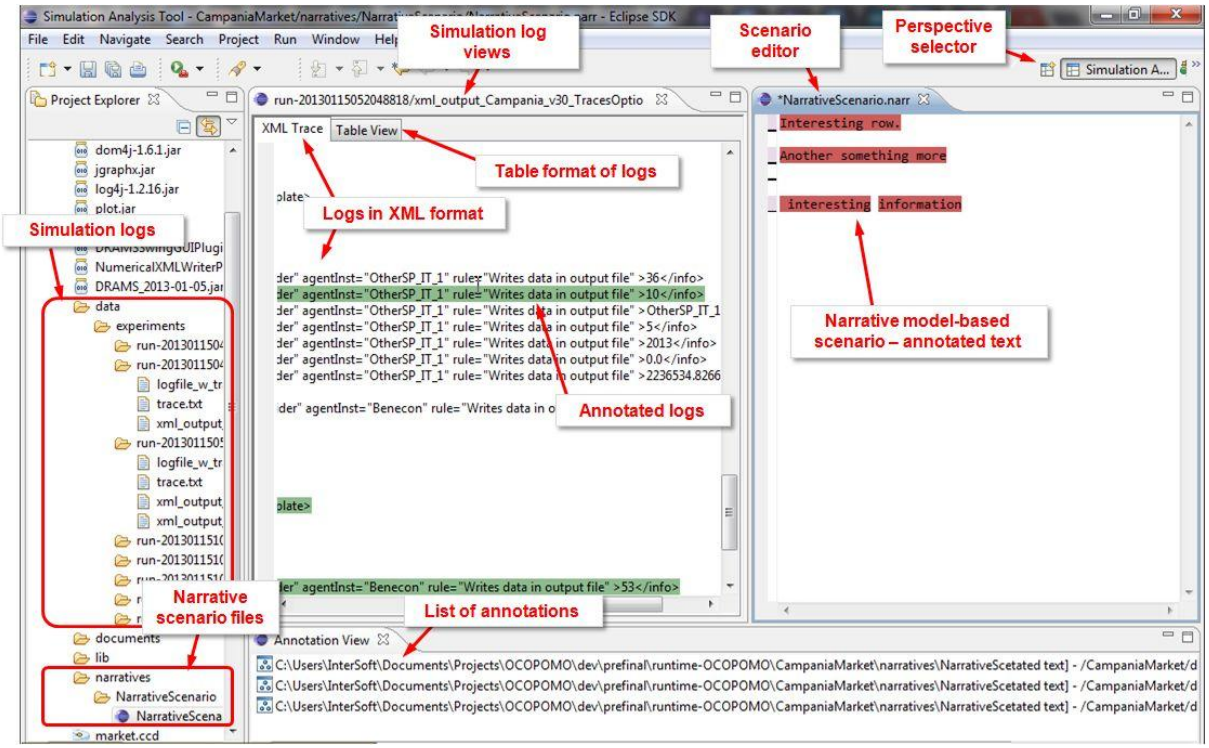

Figure 18: User interface of the Simulation Analysis Tool.

- ***Simulation Analysis Tool***, provided as Eclipse plug-in, is a toolkit for analysing simulation results and constructing related model-based scenarios. It consists of several components - views, presented in Figure 18, which support particular tasks of model-based scenario development process, namely:
    - *Simulation log views* - an interface that visualises the simulation logs provided by the *Simulation XML Log Transformation* module. The logs are available in the tool in structured XML and Table formats (see in Figure 19), which are specifically supported for a creation of annotation links between particular rules of the agent-based model and text portions of the scenario.
    - *Scenario editor* - a text editor specifically designed for development of output model-based scenarios. The editor enables to highlight text portions in the scenario content

and link them to particular records of simulation logs in both XML and CSV Table formats. Since the editor is based on the *MyNote Annotator* plug-in (see above in section 4.3), the annotations remain preserved in the text even during the further text editing and scenario creation.

- *Annotation mechanism* - a set of inner system components, based on the GATE annotation file mechanism (GATE, 2011), allowing creation and persistent storage of links between simulation log records and text fragments of output model-based scenario. Depending on the Table or XML visualisation of log records, the annotation mechanism enables to select the whole log record or its part, e.g. a particular DRAMS rule, fact, or even CCD concept included with the XML log by its UUID. These data elements can be linked to a highlighted text portion of the scenario - a sentence, paragraph, word or a sequence of words. Resulting annotations are stored in a separate XML-based file with the `.narr` extension, located in the Java project space in the `/narratives` folder, together with the output scenario.

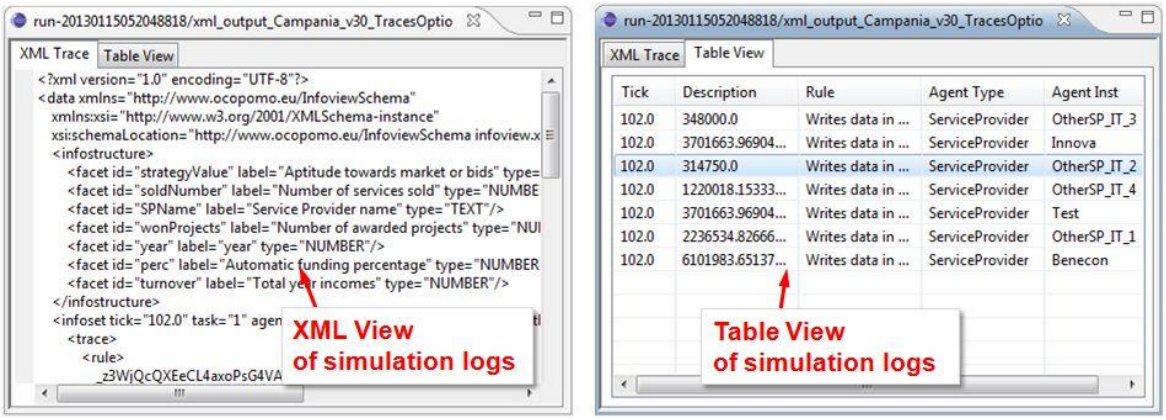

Figure 19: XML and CSV Table formats of simulation logs.

- ***Publishers***, i.e., inner components of the *Content Repository Client* middleware, which are invoked from the Simulation Analysis Tool for uploading the resulting model-based scenario and all related data resources to the Alfresco CMS. The publishers are provided as follows:
  - *Wiki Page Publisher* - a component that transforms the text of produced model-based scenario to the wiki HTML format required by the Alfresco CMS. Annotations inside the scenario text are represented as parameterised hypertext links, which are able to browse annotations on the Alfresco web site. The *Wiki Page Publisher* then provides the REST API interface for accessing the Alfresco web space and uploads the transformed scenario to the *Alfresco Data Repository*.
  - *Simulation Log Publisher* - a component that identifies all the simulation logs that were used for annotation of the produced model-based scenario, and consequently uploads the simulation logs to the Alfresco web space by means of CMIS interface.
  - *DRAMS Publisher* - a component that identifies all the DRAMS source files (i.e., files representing agent-based simulation models) that are related to the produced model-based scenario - it means that the logs produced by running simulations of these models were annotated to the scenario text. Finally, the publisher uploads the DRAMS source files to the Alfresco web space by means of CMIS interface.
  - *CCD Publisher* - a component that identifies the CCD model and all local files that were used by experts during the whole procedure of policy modelling and scenario

development, and thus are referenced by scenario annotations. The publisher then, using the CMIS interface, uploads the CCD model and the local files to the Alfresco web space.

More detailed description and usage guidelines for the outlined tools enabling simulation analysis and production of output model-based scenarios can be found in a separate document *D4.2-D: User Manual on Tools for Simulation Analysis and Output Scenario Generation*. For further methodological instructions on how to formulate and develop output model-based scenarios from simulation logs and conceptual descriptions of policy cases, please refer to OCOPOMO deliverables D6.1 (Moss et al, 2013) and D8.1 (Scherer et al, 2013)

4.6. PROVENANCE OF ARGUMENTS AND EVALUATION OF OUTPUT SCENARIOS

Output scenarios, created by policy analysis and modelling experts in accordance with developed agent-based policy models, are forwarded for evaluation and further discussion to the Alfresco web site. Availability of output model-based scenarios in the shared space of Alfresco CMS initiates the final sixth phase of the OCOPOMO process (cf. Figure 1 in section 1), which is supported by a suite of *Simulation Output Visualisation and Traceability Tools*. These tools are provided as extensions of the Alfresco Share platform (Alfresco Share, 2011), specifically its functionality of collaborative text sharing and editing in a wiki style (see also *Scenario Generation Tools* → *Wiki* in section 4.2 above). The feature of traceability, implemented as an extension of Alfresco wiki editor, enables to navigate from highlighted (i.e., annotated) text portions of output scenario back to statements in initial evidence-based scenarios, which were provided by stakeholders at the beginning of the OCOPOMO process. In addition to that, the *CCD Modeller Explorer* applet is provided for stakeholders and policy decision makers to browse the underlying CCD model structures and investigate a conceptual background of produced policy scenarios. The scope of model-based scenario browsing tools in the overall OCOPOMO toolkit architecture is presented in Figure 20.

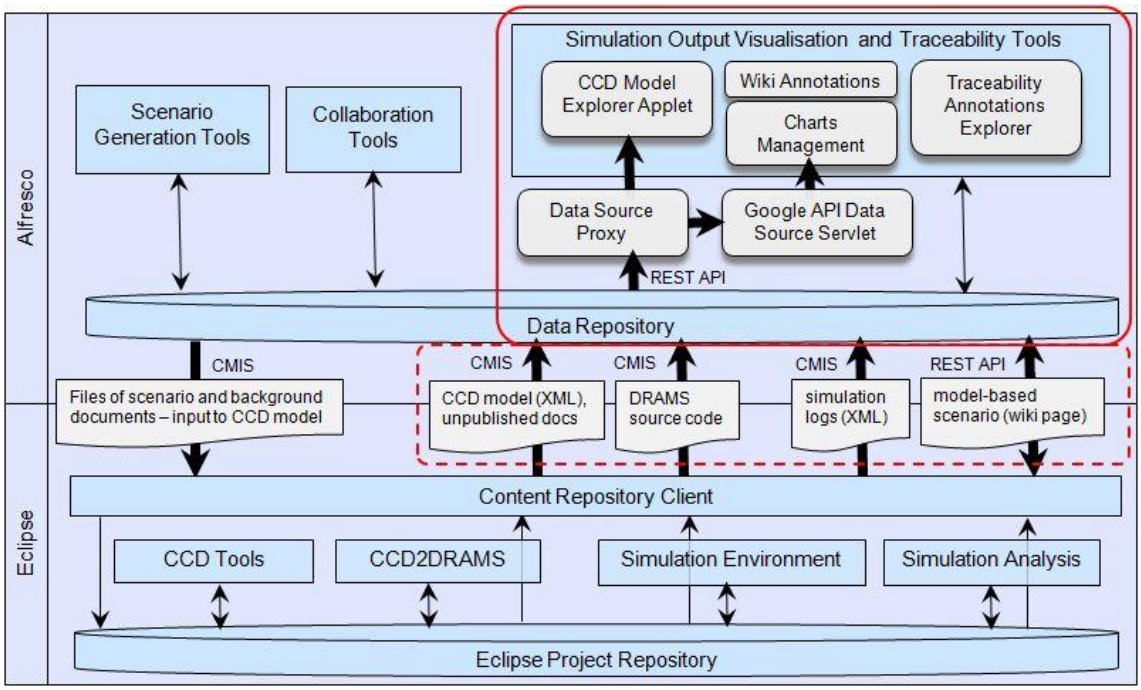

Figure 20: OCOPOMO ICT toolkit - highlighted tools and components for visualisation and investigation of model-based scenarios.

Since the output scenarios are presented in Alfresco as wiki pages, the *Wiki* component of *Scenario Generation Tools* (see in section 4.2) is employed for standard text editing and content maintenance. Furthermore, the *traceability* feature is provided as a mechanism supporting the provenance of arguments, comments, and opinions provided by stakeholders (Lotzmann, Wimmer, 2012). The traceability tools are based on *Wiki Annotations*, which were included into the scenario content during its development and are presented in Alfresco wiki as hypertext links with highlighted text sequences (see in Figure 21). After clicking such a link, the *Traceability Annotations Explorer* popup window is displayed and links to related parts of initial evidence-based scenarios are presented. In addition, the *View CCD* link enables browsing of underlying conceptual model that was employed for construction of the output model-based scenario presented on the wiki page.

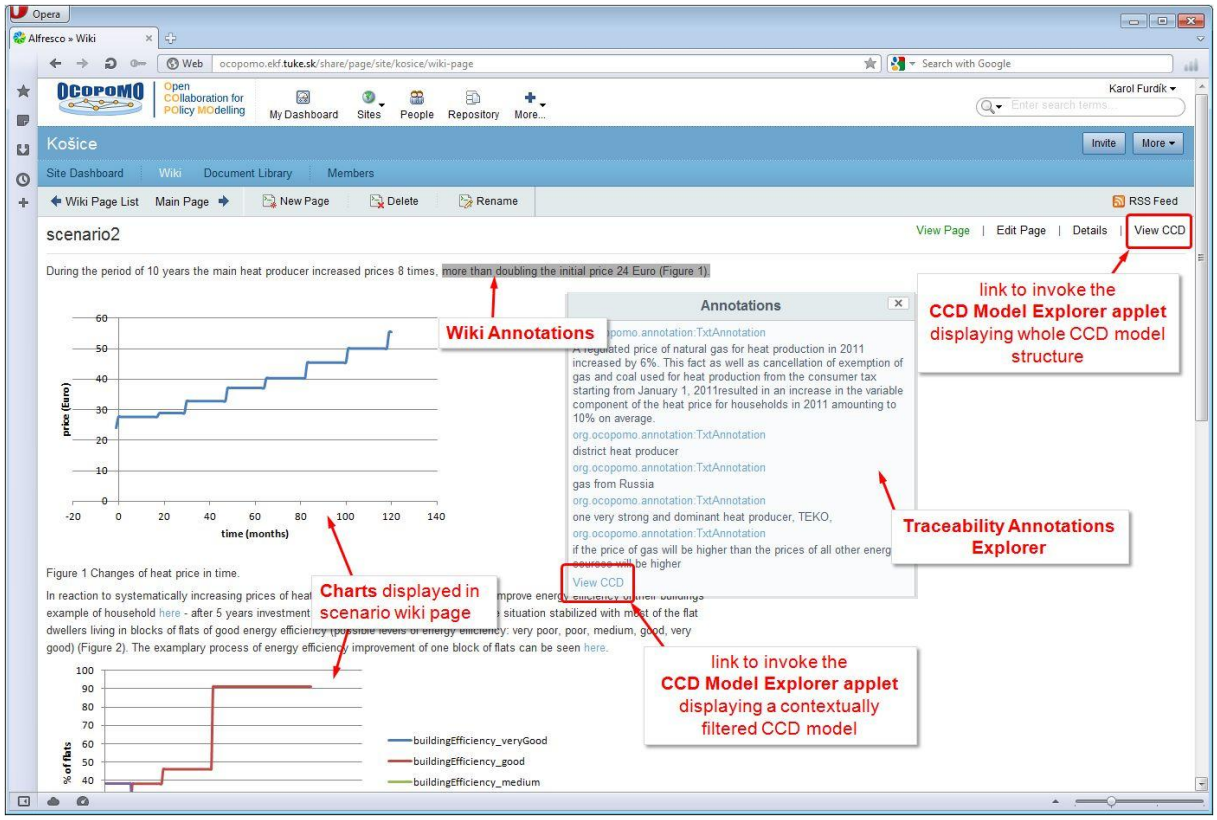

Figure 21: User interface of Simulation Output Visualisation and Traceability Tools.

The Alfresco Share web environment (Alfresco Share, 2011), where the output scenarios and related traceability tools are presented, is highly customisable and could be adjusted according to the needs of particular end users. However, a typical layout of the user interface of *Simulation Output Visualisation and Traceability Tools* is depicted in Figure 21. The toolkit consists of the following components:

- ***Wiki Content, Annotations and Traceability Tools***:
    - *Wiki Annotations* - a mechanism that visualises the annotations of model-based scenarios. The annotations, presented on the wiki page as highlighted hypertext links, are internally represented by JSON literals containing traceability metadata such as descriptive text extracted from simulation traces, links to the relevant evidence-based scenarios and unique identifiers of relevant CCD concepts. The JSON literals are embedded into the HTML code of the wiki page as SPAN elements with custom data attributes, which are evaluated after the annotation link is invoked.

- *Traceability Annotations Explorer* - an interface that visualises all the traceability-related information for a given wiki annotation. To see this information, just click on the annotation link to invoke it - as a result, the *Annotations* popup window is displayed on the screen. The window contains reference links to related statements extracted from initial evidence-based scenarios, which were taken during the process of policy modelling and simulation as resources for the formulation of the output scenario statement highlighted as the annotation link.
- *Charts Management* - a mechanism that enables an embedding of dynamic charts to the HTML content of the Alfresco wiki page. Charts are embedded into the HTML content as hidden DIV tags with the JSON literal containing the chart settings specified in the *Google Visualization API*. Several chart types are supported, for example, the Pie, Line, Area, Column chart, etc. All the available chart types are described in more details in the attached document *D4.2-E: User Manual on Simulation Output Visualisation and Traceability Tools*.

- ***Data Source Components***:
  - *Data Source Proxy* - an interface that enables the CMIS and REST API services for uploading the model-based scenario and all related data files from a local Eclipse environment to the *Alfresco Data Repository*. The interface is installed on the Alfresco site as an implementation of the Alfresco Public REST API (Alfresco 3.0 REST API, 2011) and CMIS API (Alfresco CMIS, 2012).
  - *Google API Data Source Servlet* - an interface that implements the Google Visualization API (Google Visualization API Reference, 2012) for querying of chart data to visualise all the supported types of charts on the Alfresco web client site.

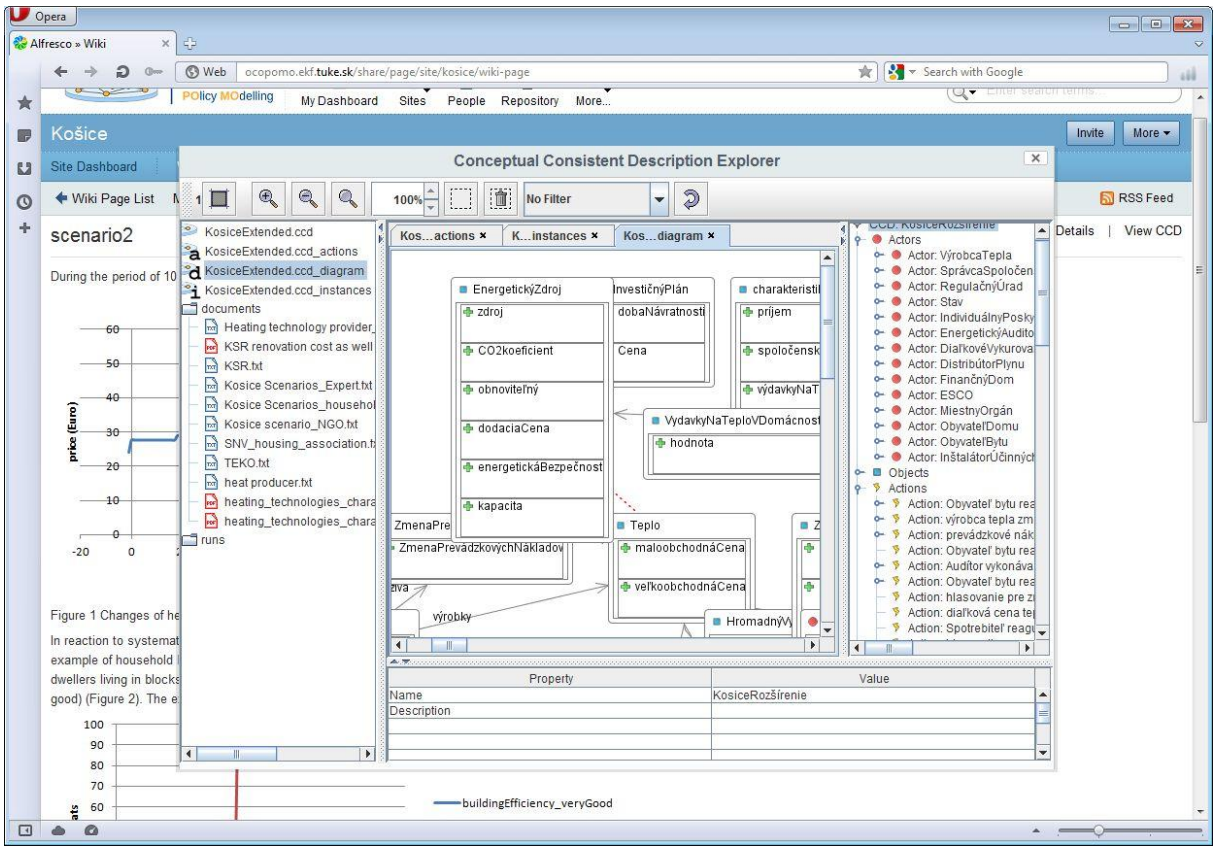

Figure 22: CCD Model Explorer applet invoked from a wiki page.

- ***CCD Model Explorer Applet*** - a component that visualises the conceptual model, which was used for a construction of the presented model-based scenario. This way, the presented backward traceability information is enhanced on the CCD model, which was created as an intermediate step between input evidence based scenarios and agent-based policy models. The *CCD Model Explorer applet*, depicted in Figure 22, can be invoked from the main menu of the wiki page, as well as from the *Annotations* popup window (see also in Figure 21). In the latter case, the displayed CCD model is filtered by the context - i.e., by the list of CCD concepts taken from the source annotation from which the *Annotations* popup window was invoked.

More detailed description and usage guidelines for *Simulation Output Visualisation and Traceability Tools* are provided in a separate document *D4.2-E: User Manual on Simulation Output Visualisation and Traceability Tools*.

The output model-based scenario can be further discussed and modified in the collaborative web space of Alfresco, using the *Collaboration* and *Scenario Generation Tools*, as it is described in section 4.2 and *D4.2-A: User Manual on Collaboration and Scenario Generation Tools*. This way, the provided model-based policy scenario may be transformed to several evidence-based scenarios formulated by involved stakeholders and policy makers in a collaborative discussion. Availability of new evidence-based scenarios will initiate the next iteration of the OCOPOMO process.

5. SYSTEM DOCUMENTATION

This chapter presents the system documentation that was developed as a description of technical aspects and implementation details of particular end-user tools and inner system components of the integrated OCOPOMO ICT platform.

An inner structure of the provided system documentation follows the specifics of underlying implementation technology that was applied upon the Alfresco and Eclipse frameworks. In particular, it covers an architecture of internal modules, classes and libraries, applied configuration settings and customisation guidelines, as well as a description of exposed interfaces of the documented tools and system components. For Java-based tools and components, a documented source code is provided in the JavaDoc format as well.

The overall system documentation is organised into a set of separate documents - specific system documentation materials for respective OCOPOMO tools and components, which are provided in the following structure:

1. *D4.2-SD-1: System Documentation of OCOPOMO Alfresco tools*, which includes:
   - applied technologies - Spring Web Scripts and Spring Surf;
   - customisation of general Alfresco platform for the OCOPOMO toolkit - architecture of inner components, file structure, etc.;
   - Alfresco extensions on Polling, Chat, Wiki Page Formatting, Traceability and Wiki Annotations, embedding and visualisation of charts, and localisation issues;
   - CCD Model Explorer applet.
2. *D4.2-SD-2: System Documentation of CCD Tool and Annotation Extensions*, namely:
   - CCD Tool - its inner structure, classes and interfaces;
   - Annotation extensions - PDF Annotator, HTML Annotator, MyNote Annotator;
   - Content Repository Client;
   - Simulation Analysis Tool.
3. *D4.2-SD-3: System Documentation of DRAMS tools*, which includes:
   - DRAMS-Repast simulation environment - its inner structure, classes and interfaces.

The system documentation of particular tools is accompanied by various files containing JavaDoc description of key classes and methods, source code examples, etc., namely:

CCDExplorerApplet_JavaDoc.zip - documented source code in JavaDoc and implementation code examples for the CCD Model Explorer applet.
Referenced in **D4.2-SD-1**.

CCDTool_JavaDoc.zip - documented source code in JavaDoc for the CCD Tool.
Referenced in **D4.2-SD-2**.

CCDToolsExtensions_JavaDoc.zip - documented source code in JavaDoc for the Content Repository Client component and for the Annotation Extensions of the CCD Tools suite.
Referenced in **D4.2-SD-2**.

SimulationAnalysisTool_JavaDoc.zip - documented source code in JavaDoc for the Simulation Analysis Tool and related utilities.

Referenced in **D4.2-SD-2**.

DRAMS_JavaDoc.zip - documented source code in JavaDoc for the DRAMS-Repast simulation environment.

Referenced in **D4.2-SD-3**.

6. CONCLUSION

This deliverable documents the developed OCOPOMO ICT platform from both tools usage and technical implementation perspectives. The user documentation, provided in Chapter 4 and accompanying documents D4.2 A-E, follows the concept of the underlying OCOPOMO process for collaborative policy development, which is presented in Chapter 1 as the basis for further explanation of use of particular tools.

The architecture of the integrated OCOPOMO ICT toolkit is presented in Chapter 2. The role of particular system tools and components is described in more details, with regard to the six phases of the OCOPOMO process. Functionality and proposed use of particular tools within the process phases is demonstrated on the requirements that were taken as key drivers during the platform design, development, and implementation.

Instructions for the installation, set up, administering, and maintenance of the overall platform are provided in Chapter 3. Based on the division of the platform on collaborative Alfresco web space and local Eclipse-based toolkit for policy analysis and modelling experts, the hardware and software requirements, download and installations instructions, and system maintenance guidelines are described for both the applied policy development environments.

The user documentation is provided in sections of Chapter 4 and related documents D4.2 A-E. The documentation is organised in accordance with respective process phases. Usage instructions are formulated in the format of answers on "how-to" questions that describe the most common maintenance and use operations for end users of target user groups. A set of videos presenting key actions of handling the tools from an end-user perspective is included as well.

Finally, the system documentation, introduced in Chapter 5, is presented again as a set of accompanying documents D4.2 SD-1-3, where D4.2 SD-1 covers all the Alfresco-based tools and components, while the implementation details of Eclipse-based tools are provided in the rest of enclosed SD documents.

In addition to the provided user manuals and system documentation materials, the OCOPOMO ICT platform is deployed on the Alfresco web environment, where particular pilot applications are located on the following URLs:

- *KSR* pilot: http://ocopomo.ekf.tuke.sk/share/page/site/kosice/dashboard;
- *Campania* pilot: http://ocopomo.ekf.tuke.sk/share/page/site/campania/dashboard;
- *London Housing* pilot: http://ocopomo.ekf.tuke.sk/share/page/site/london/dashboard;

These installations can be seen as reference applications of the OCOPOMO ICT toolkit and therefore they complement the user and system documentation by presenting the platform in a run-time operational environment. Other information related to the overall methodology of the OCOPOMO process, development of narrative scenarios of policy alternatives and creation of policy models is published in respective OCOPOMO deliverables D8.1 (Scherer et al, 2013), D5.1 (Moss et al, 2011), and D6.1 (Moss et al, 2013). All these materials should support the exploitation of the developed ICT platform, methodology, and information structures of scenarios and policy models towards further adaptations and applications in other policy modelling use cases.

The integrated OCOPOMO ICT platform, presented in this deliverable, was produced in its final release as outcome of the OCOPOMO project. It is, however, expected that further enhancements and extensions will be applied in future and new versions of the platform will be released. The user and system documentation for the next releases will be then provided as an update of this document and will be published at http://www.ocopomo.eu.

REFERENCES

Alfresco CMIS, Content Management Interoperability Services, http://wiki.alfresco.com/wiki/CMIS. Alfresco Software, Inc., 28 November 2012.

Alfresco Share Documentation, Using Alfresco Share. Alfresco Team 3.5 Documentation, http://docs.alfresco.com/3.5/topic/com.alfresco.team.doc/topics/uh-welcome.html. Alfresco Software Inc., 2012.

Alfresco Share web client, http://wiki.alfresco.com/wiki/Alfresco_Share. Alfresco Software, Inc., 23 March 2011.

Alfresco 3.0 REST API, http://wiki.alfresco.com/wiki/3.0_REST_API, Alfresco 3.0 Repository Public RESTful API, Alfresco Software, Inc., 17 February 2011.

Bicking, M. et al: *D1.1 Stakeholder Identification and Requirements for Toolbox, Scenario Process, and Policy Modelling*. Deliverable 1.1, OCOPOMO consortium, 2010.

Bednar, P., et al: *D3.1 Platform Components* (Supportive Report to Software Components). Deliverable 3.1, v.1.00, OCOPOMO consortium, 2011.

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

Bednar, P., Butka, P.: *Manual to Collaboration and Editing Tools*. WP3 Deliverable, Draft, Version 0.4, OCOPOMO consortium, 2012.

Butka, P., et al: *D3.2 Revisions Report to Platform Components*. Deliverable 3.2, v. 1.00, OCOPOMO consortium, 2012.

Eclipse Indigo, the annual release of Eclipse projects, http://www.eclipse.org/indigo/, The Eclipse Foundation, 2011.

Eclipse Indigo Documentation, HTML Help Center, Eclipse Indigo (3.7) Documentation, http://help.eclipse.org/indigo/index.jsp. The Eclipse Foundation, 2013.

Eclipse Juno, version 4.2. The annual release of Eclipse projects, http://www.eclipse.org/juno/, The Eclipse Foundation, 2013.

Furdik, K. et al: *D4.1 Integrated Platform*. Deliverable 4.1, OCOPOMO consortium, 2013.

GATE, General Architecture for Text Engineering, The University of Sheffield, http://gate.ac.uk, 2011.

Google Visualization API Reference, Google Developers, Google Inc., https://developers.google.com/chart/interactive/docs/reference, 9 July 2012.

Lotzmann, U., Meyer, R.: *A Declarative Rule-Based Environment for Agent Modelling Systems*. In: The Seventh Conference of the European Social Simulation Association, ESSA 2011, Montpellier, France, 2011.

Lotzmann, U., Meyer, R.: *DRAMS Manual*. WP6 Deliverable, Draft, Version 0.2, OCOPOMO consortium, 02 February 2012.

Lotzmann, U., Wimmer, M. A.: *Provenance and Traceability in Agent-based Policy Simulation*. In: Proceedings of 26th European Simulation and Modelling Conference - ESM'2012, October 22-24, 2012, FOM, Essen, Germany, 2012.

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

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

Moss, M. et al: *D6.1 Narrative Scenarios and Policy Models*. Deliverable 6.1, OCOPOMO consortium, 2013.

Repast 3, Recursive Porus Agent Simulation Toolkit. The Repast Suite, Argonne National Lab, http://repast.sourceforge.net/repast_3/, 2008.

Scherer, S., Markisic, S., Lilge, B.: *CCD2DRAMS Tool Manual*. WP3 Deliverable, Draft, Version 0.1, OCOPOMO consortium, 30 January 2012.

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.

Scherer, S., Wimmer, M. A.: *CCD Tool Manual*. WP3 Deliverable, Draft, Version 0.7, OCOPOMO consortium, 29 January 2012.

APPENDIX A: REQUIREMENTS ON SYSTEM FUNCTIONALITY

Table A-1: Initial, added and revised requirements before the 1st pilot.

ID	Name	Priority	Component / Module	Solution / Application / Usage
I-NF-2	Accessibility	Must-have	General Requirement	**Final status**: Implemented **Notes**: Alfresco developers and community adapted WAI standards and provide necessary guidelines for achieving the accessibility (see http://wiki.alfresco.com/wiki/Accessibility for details). Due to fact that Alfresco provides heir styles with accessibility standards in mind and we have developed our modifications using their styles and guidelines, the accessibility standards (especially important for Alfresco Share part with stakeholders) are respected.
I-28	Action–based and rule-based role playing of stakeholders in simulation	Must-have	Simulation Manager / SE	**Final status**: Postponed **Notes**: This requirement was skipped due to resource limitations in favour of more significant developments in the focus of the project (CCD, traceability).
T-16	Agent-based simulation tool	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: For running of simulations and preparing of the simulation-specific code, Repast is used as a basic background simulation execution engine, with the agents and all defined and controlled using rule-based engine developed within the project - DRAMS (Declarative Rule-based Agent Modelling System).
I-36	All personal preferences in one place	Nice-to-have	User Manager / CMS	**Final status**: Implemented **Notes**: Due to the fact that preferences are only those for Alfresco Share accounts (and these are then used for other components, if necessary), all of them are covered by particular Alfresco user management services.
I-NF-4	Authentication	Must-have	User Manager / CMS	**Final status**: Implemented **Notes**: Applied and used according to the Alfresco user management services.
I-NF-5	Authorization	Must-have	User Manager / CMS	**Final status**: Implemented **Notes**: Applied and used according to the Alfresco user management services.
T-37	Authorization/authentication issues are taken into account	Must-have	User Manager / CMS	**Final status**: Implemented **Notes**: All places where authorization and/or authentication is needed uses Alfresco

	in individual tools			user management services.
T-4	Chat	Must-have	Chat Manager / CSET	**Final status**: Implemented **Notes**: Chat manager was implemented from the scratch and consists of two parts: 1. Chat servlet - implements server part of the manager, it is used as a simple server for exchange XMPP protocol messages, which supports creation and management of multiuser chat rooms and its occupants. 2. Chat client - Alfresco Share dashlet for site dashboards, implemented using JavaScript, Freemarker templates, etc. It uses XMPP-based protocol in message-based server-client communication with the Chat Servlet in order to achieve its required chat functionality. Most of the functionality is covered. For details of some other issues, see requirements related to Chat Manager before 2nd trial (for final implementation).
I-11	Closing the scenario generation process / versioning	Must-have	Process Manager, Version Manager / CSET, CMS	**Final status**: Implemented **Notes**: Current implementation is based on the Alfresco (Share) implementation for wiki pages and document library, with their versioning support and possibility to close (disallowing the edit functionality) scenario versions.
T-25	Commenting functionality	Must-have	Polling and Rating Manager / CSET	**Final status**: Implemented **Notes**: This issue is mainly covered by reuse of Alfresco Share functionality - aspect-based component for commenting the content and positive feedback ("like" feedback) for content. Additionally to the standard Share, commenting of wiki pages was added also directly to their view (of content) and in the preview of the list of versions. However, it is possible only to comment whole content due to limitations of API, but user can use direct editing of the wiki page content to add comments for the selected text using the different formatting.
I-18	Comparison of simulations	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: When applying the OCOPOMO process on pilot model development, it was found that the added value of comparing simulation runs and raw results on the SE level is low. Thus, this requirement has been rejected for the purposes of the OCOPOMO project. Instead, support for traceability and simulation analysis allow comparison of results on a more aggregated level (initial vs. model generated scenarios).
T-39	Computer-assisted	Must-have	Annotation Manager /	**Final status**: Implemented

	Qualitative Data Analysis Software Tool – Coding of text passages and clustering of codes		CCD Tool	**Notes**: The CCD tool allows coding of narrative text passages into a conceptual model based on the CCD meta model. The CCD meta model as vocabulary defines the clusters.
T-40	Computer-assisted Qualitative Data Analysis Software Tool – flexible querying of codes and issues	Must-have	Annotation Manager / CCD Tool	**Final status**: Implemented **Notes**: It is possible to query the elements (codes) for text passages (issues) and text passages for elements.
T-41	Computer-assisted Qualitative Data Analysis Software Tool – statistics	Nice-to-have	Annotation Manager / CCD Tool	**Final status**: Postponed **Notes**: Statistics of available CCD elements and annotations was postponed (within the scope of this project) as nice-to-have but not must-have feature.
T-5	Content Management System (CMS) functionality	Must-have	Content Manager, Document Manager, Version Manager / CMS	**Final status**: Implemented **Notes**: OCOPOMO Content Management Server is based on the Alfresco Content Repository. The managed content includes all information artefacts produced by the collaboration tools, i.e. documents, wiki pages, discussion forum's messages, calendar events, chat history, polling posts, etc. Content and Collaboration Services embedded in the Alfresco application server are available for other applications (e.g., tagging, content transformation, workflow, user profile management, feeds, application logic for CSET tools, etc. The functionalities provided by the services are accessible for clients through REST APIs or CMIS services (which defines two way - REST-like protocol based on the extensions of the AtomPub format and SOAP binding).
SOTA-2	Content/WYSIWYG	Should-have	Document Manager / CSET	**Final status**: Implemented **Notes**: Alfresco Share uses as standard TinyMCE editor for inline editing of html files, which is used for wiki pages content, as well as for other content (e.g. messages in discussions).
I-13	Control of scenario generation process phases	Must-have	Process Manager, Version Manager / CMS, CSET	**Final status**: Implemented **Notes**: To control scenario generation process and its documents (versions), we are able to use Alfresco workflow engine (Activiti or JBPM implementation) on documents processes, with the support for versioning of them and management of permissions to them (all as Alfresco functionality). Therefore, control over the scenario generation process can be also achieved by the combination of versioning and access rights setup with the organizational work of facilitators (as was applied

				in our trials).
I-4	Creation of stakeholder groups for the scenario generation process	Must-have	Collaboration Space Manager, Notification Manager, Process Manager / CSET	**Final status**: Implemented **Notes**: For the stakeholder groups specific group from Alfresco Share site have been used (as implemented by the Alfresco with the support of user management with groups), with the particular access rights (different for standard stakeholder, different for decision makers and for facilitators). Invitation (invite people to site, etc.) and notification services (activities and documents/content feeds) are the part of dashboard and/or site services in Alfresco.
I-23	Creation of stakeholders groups for policy modelling process	Must-have	Collaboration Space Manager, User Manager / CSET, CMS	**Final status**: Implemented **Notes**: Role and user management from Alfresco CMS was reused. Stakeholder and facilitator roles were mapped to the security roles of Alfresco Share. Besides of security roles which define access rights for the content, user can be arbitrary grouped according to any criteria.
I-22	Defining scenario for policy modelling	Must-have	Document Manager, Annotation Manager / CCD Tool, CSET	**Final status**: Implemented **Notes**: Evidence-based scenarios and background documents, provided by stakeholders, can be annotated using the CCD Annotation Tool during the modelling process. The facilitator can introduce any additional documentation, which can be annotated similarly as background documents.
UC-10	Development of social network	Should-have	Link Manager, Concept Manager / CCD Tool	**Final status**: Implemented **Notes**: The CCD tool allows the definition of available actors and their relationships as a kind of social network in a CCD. The Actor Network Diagram visualises such networks.
I-17	Discussion about simulation results and decisions of human agents in simulation	Must-have	Discussion Forums Manager / CSET	**Final status**: Implemented **Notes**: Discussions module from Alfresco Share implementation was used for Discussion forums. Specific topic for discussion on simulation results or decisions is organizational responsibility of modeller, facilitator or analysts (as it was not finally expected in trials).
T-1	Discussion forums	Must-have	Discussion Forums Manager / CSET	**Final status**: Implemented **Notes**: Discussions module from Alfresco Share implementation was used for Discussion forums.
T-1-4	Discussion forums - Authorisation on level of the	Must-have	Discussion Forums Manager, User Manager	**Final status**: Implemented **Notes**: Discussions module from Alfresco Share implementation was used for

	discussion forum		/ CSET, CMS	Discussion forums. Specific permissions (access rights) on topics for discussion and their posts are based on default Alfresco implementation, where the main difference is on the ownership of post or topic, and on the default roles like Collaborator, Contributor, etc. More details on roles can be found in Alfresco documentation. Decision to stay with these settings is based on the revision of requirement, where particular pilot users do not expect more enhanced access rights setup. However, in Alfresco it is possible to setup customized roles and manage the forum and post specific nodes in repository in order to manage permissions of specific topics using specific roles.
T-1-5	Discussion forums - condition of use	Must-have	Discussion Forums Manager / CSET	**Final status**: Implemented **Notes**: There is possibility to apply Condition of use in the form of organizational information page for stakeholders (e.g., starting wiki page for current case) or using some default text, which will be added at the start of the discussion. In our pilot runs it was not used in this way, but only due to fact that we have found not needed for our particular case (generally it is simply applicable).
T-1-2	Discussion forums - entries should be organised in threads	Must-have	Discussion Forums Manager / CSET	**Final status**: Implemented **Notes**: Discussions module from Alfresco Share implementation (which was used for Discussion forums implementation) organizes entries in expected way using threads.
T-1-3	Discussion forums - possibility to order entries in chronological order and for topics	Must-have	Discussion Forums Manager / CSET	**Final status**: Implemented **Notes**: Discussions module from Alfresco Share implementation (which was used for Discussion forums implementation) organizes entries in expected way using its threads.
T-12	Discussion forums – moderated and non-moderated discussions	Must-have	Discussion Forums Manager / CSET	**Final status**: Implemented **Notes**: The usage of Discussions module from Alfresco Share implementation (which was used for Discussion forums implementation) is generally non-moderated, but can be used with the specific access rights settings (e.g., facilitator will be able to remove / edit problematic inputs / topics), as well as organization rules of collaborative work could support moderation of forums.
T-14	Discussion forums – rating of contributions and contributors (analysis of discussions based on a	Must-have	Polling and Rating Manager / CSET	**Final status**: Implemented **Notes**: This requirement was updated during the project implementation and due to the implementation of Discussions module in Alfresco Share and not expected significant added value for more complicated extension (too much resources for

	relevance feedback)			that), actually implemented simple 'like'-based rating of contributions was used for fulfilling this requirement.
T-34	E-mail notification system	Must-have	Notification Manager / CMS	**Final status**: Implemented **Notes**: Alfresco functionality for sending of emails is available within the CSET platform module (SMTP configuration is required), also with quite strong support for Google's Gmail. Some of the email (or notification) functionality can be also supported by RSS feeds.
UC-7	Expertise-based relations	Should-have	Annotation Manager / CCD Tool	**Final status**: Implemented **Notes**: Expertise-based relations and elements in the CCD can be annotated with expert annotations.
SOTA-3	File types supported	Should-have	Document Manager / CMS, CCD Tool	**Final status**: Implemented **Notes**: Thanks to Alfresco Share and its repository, the system is able to support many different formats. Also annotation tool (CCD) is able to support text files, pdf and html.
I-39	Full dependency graph including dependency of rules on lagged clauses	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: DRAMS supports lagged fact base retrieval and queries.
UC-6	Generation of relations	Should-have	Annotation Manager / CCD Tool	**Final status**: Implemented **Notes**: Annotation tool supports the functionality where links between text phrases and elements (issues) in the CCD can be expressed.
I-NF-11	Help and assistance	Must-have	Notification Manager / CMS, CSET	**Final status**: Implemented **Notes**: Especially for CSET part, Alfresco help and manuals are available, together with the help icons on dashlets. For particular pilot evaluation, there is also simpler help with the description of customised features with the mostly used components.
T-C1	Hints for interesting topics	Nice-to-have	Annotation Manager, Document Manager, Notification Manager / CSET	**Final status**: Implemented **Notes**: CSET part of the system, which is based on the Alfresco Share, provides the summary of links to last activities and content added (or changed) within the current site () on its dashboard (in dashlets). It is also possible to filter information about the content showed there, e.g. by time or ownership.
I-29	Human actions analysis	Should-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Simulation logs can be written e.g. in CSV format, so spread sheet tools

				(e.g. Excel) can be used for all kinds of filtering and analyses. However, the gaming requirement (I-28, FR24_PM) has been rejected.
I-1	ICT toolbox functionality provided through one portal-based interface	Must-have	Collaboration Space Manager, Process Manager, Search Manager / CSET	**Final status**: Implemented **Notes**: This requirement was partially revised after the first review. Due to the reviewer comments, which were reflected as useful, we have decided to divide the application to Alfresco and Eclipse parts.
T-18	Import of the previously exported simulation data	Should-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: Rejected in favour of more important features.
UC-4	Initiate project	Must-have	Collaboration Space Manager, Process Manager / CMS, CSET	**Final status**: Implemented **Notes**: Current implementation is in the way that facilitator (or site administrator) is able to create new project (site) and manage it (prepare organizational rules, upload relevant document, etc.). This way for implementation is based on the needs for our first trials. Alfresco Share also support the creation and management of workflow-based processes (using Activiti or JBMP implementation).
SOTA-6	Information structuring	Should-have	Annotation Manager, Rule Manager / CCD Tool	**Final status**: Implemented **Notes**: Identifying and structuring the information extracted from unstructured texts (e.g. scenarios and/or support documents) is done in a tree structure.
I-5	Integration of components within the e-participation tools for scenario generation – data exchange / annotation	Must-have	Collaboration Space Manager, Concept Manager, Discussion Forums Manager, Link Manager, Search Manager / CSET, CMS, CCD Tool	**Final status**: Implemented **Notes**: Alfresco Share application covers the needs for integration between different e-participation tools within CSET module (site, documents, forums, chat, pollings, etc.). For data exchange and annotations scenarios are created using wiki with all necessary functionality, also Eclipse modules (for annotation, model creation, simulation analysis) are connected to Alfresco repository (e.g., using CMIS protocol) and are able to exchange all necessary data (in both ways) in order to achieve traceability of information and sharing of knowledge.
I-6	Integration of components within the e-participation tools for scenario generation – search	Must-have	Search Manager / CMS, CSET	**Final status**: Implemented **Notes**: Alfresco Share application covers the needs for integration between different e-participation tools within CSET module (site, documents, forums, chat, pollings, etc.). The search is based on the Alfresco repository Search component, which is available for advanced search through all Alfresco-based content. Due to fact that also Eclipse content is uploaded to (and also from other side, Alfresco to Eclipse, for annotation of scenarios or background documents) Alfresco , it is

				possible to search also in documents created by Eclipse plugin.
I-7	Integration of components within the e-participation tools for scenario generation – workspace	Must-have	Collaboration Space Manager, Notification Manager, Process Manager, Search Manager / CSET	**Final status**: Implemented **Notes**: Alfresco Share application covers the needs for integration between different e-participation tools within CSET module (site, documents, forums, chat, pollings, etc.), which are together shared through site component for particular case. Actually, all cases (sites) are available and integrated through Alfresco interface using their own sites (projects), which can users join.
I-25	Integration of policy modelling tool and simulation / analysis tools – data exchange / annotation	Should-have	Concept Manager, Content Manager, Link Manager, Rule Manager / CCD Tool, SE, CSET	**Final status**: Implemented **Notes**: Several tools integrated within the platform are able to support policy modelling, creation of conceptual models, preparing the simulation models, analysis of simulation outputs and provision of information back to users (and its traceability). The main division is in separation of Eclipse application tools and Alfresco tools (distinction follows different types of users and fulfils the usability requirements). Most of the Alfresco tools are related to the presentation of information from policy modelling and simulations, while Eclipse tools are mostly related to modelling, analysis and annotation. The connection of components which achieve traceability of information fulfils the requirement issue.
I-NF-10	Integrity	Must-have	General Requirement	**Final status**: Implemented **Notes**: Support of integrity for content stored within the platform is based on the integrity tools and functionality of Alfresco Repository (which is also used for large and commercial implementations) based on the standard industry-based database (anyone who reuse repository is able to choose database backend, e.g. MySQL, Postgres, etc.).
UC-2	Invitation – send and receive	Should-have	Notification Manager, Process Manager, User Manager / CMS, CSET	**Final status**: Implemented **Notes**: Alfresco Share support the authorised users (site administrators) to send invitations to the site or accept/reject requests to join particular site.
I-20	Log of activities within policy modelling / simulation	Must-have	Process Manager, Search Manager, Simulation Manager / CCD, SE	**Final status**: Implemented **Notes**: Activities in CCD Tool can be logged using a standard versioning software as e.g. SVN. Implemented for SE by DRAMS output writers.
I-19	Log of activities within scenario generation	Must-have	Annotation Manager, Process Manager, Search Manager / CSET	**Final status**: Implemented **Notes**: Activities of users, especially creation and modification of content, as well as changes in site structure or membership, are logged within the Alfresco Share

				and are shown to the users in specific dashlets on their site dashboard (activities, documents, etc.).
I-F-I3	Login	Must-have	User Manager / CMS	**Final status**: Implemented **Notes**: Alfresco user management facilities are used for registered users for their login into the system. Same login is used also for other integrated tools within the OCOPOMO platform (e.g. for connection of Eclipse annotation tool to Alfresco repository of directories with the files of the current case), so user needs to be registered only once and can use one password for all needs.
I-NFT-8	Look and feel	Must-have	General Requirement	**Final status**: Implemented **Notes**: According to the original description of requirement, necessary content for stakeholders and decision makers is displayed (mostly within the Alfresco Share site for particular case) on the site and available for users. Facilitators helped to customise the site (in order to simplify the provision of information) and to provide information in the shape, which is understandable to users and they are able to follow scenarios and outputs from the simulation in narrative form with graphs and other visualisation elements. All is accomplished by the tools and methods supporting the traceability of information between scenarios and simulation outputs.
I-14	Maintaining of scenarios and rules within the ICT toolbox	Must-have	Concept Manager, Link Manager, Rule Manager, Version Manager / CMS, CCD	**Final status**: Implemented **Notes**: Default usage of Alfresco Repository functionalities supports the versioning of the wiki pages, files in document library, etc. As Eclipse tools (used for annotation, modelling, etc.) are usually using standard connections to the repository in order to retrieve files and work with them, the maintaining of necessary information is mostly available within the whole platform.
SOTA-7	Memos	Should-have	Document Manager / CSET, CMS	**Final status**: Implemented **Notes**: Users can manage memos as the Wiki pages or as the plain text files stores in the Document library.
I-35	Multilingual interface	Must-have	General Requirement	**Final status**: Implemented **Notes**: All tools within the platform support the multilingual interface using standard internalization methods. From the practical reasons, where specific language was not needed within the run of our pilots, we have not added the translation for every interface. For example, due to fact that models were created in English by the experts from non-pilot partners, these interfaces are mostly in

				English (but can be changed using language resources files). On the other side, for Alfresco Share interface, which is important for stakeholders, decision makers, facilitators in their scenario generation phase of the process, we have prepared multilingual interface for all parts, which are available to the pilot's native language users. The language for web browser interfaces in platform is automatically setup by the web browser settings for language.
UC-9	Network visualisation	Should-have	Rule Manager / CCD Tool	**Final status**: Implemented **Notes**: Visualisation of conceptual model in CCD Tool with Actor-Network, Instances and Activity diagrams.
T-24	News functionality	Must-have	Notification Manager / CMS, CSET	**Final status**: Implemented **Notes**: News are realized using the RSS channels, which are available for users in Alfresco Share environment, and also can be read by any other RSS reader (which user wants to use for news reading) using its address.
T-C2	News – rating/polling functionality	Should-have	Polling and Rating Manager / CSET	**Final status**: Implemented **Notes**: Polling functionality was implemented into the Alfresco Share and it is possible to setup any polling regarding the news, but currently it is related to the organizational work of facilitator or admin, who are able to create particular pollings.
T-29	Newsletter	Must-have	Notification Manager / CMS, CSET	**Final status**: Implemented **Notes**: Several tools can be used for the preparation of the newsletter. The main way is to provide document in the document library and use some of the information channels in order to get the attention of users – email, RSS, main wiki page, etc. For small messages RSS feeds can be used for the whole newsletter.
SOTA-8	Non-RETE rule engine	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: DRAMS was developed as a rule engine relying on a data-driven rule scheduling mechanism instead using the RETE algorithm.
I-NF-7	Operational	Must-have	General Requirement	**Final status**: Implemented **Notes**: As it was expected by the initial description of requirement, thanks to Alfresco-based solution (with Share collaborative platform), the solution runs as a web server application. Also Eclipse plugin for modellers communicates with the server and its repository. All interfaces are provided via industry-standard web browsers or standard Eclipse GUI (needed due to its essential position for

				modelling). Support of different file types is related to the usage of standard technologies within Alfresco or Eclipse solutions. For practical issues within the pilots, we have supported administration email, which can be used by the users in order to resolve technical problems. Of course, the system is capable of allowing simultaneous access by more than one user.
I-10	Opinion polling about the current version of scenario generation resources	Must-have	Polling and Rating Manager, Process Manager / CSET, CMS	**Final status**: Implemented **Notes**: Polling component was implemented (from the scratch) in order to achieve expected functionality. It consists of underlying data model for polling and polling posts (modelled as nodes in Alfresco Repository), form editor (for creation of questions / answers, their types, etc.), form engine (renders and validates forms, accessible as REST service) and results service (also available as REST service, aggregates the data in polling). User interface for polling is integrated into the Alfresco Share as a site page component. It provides list of polling which can be filtered according to the modification time and tagging, access to the form editor and links to the page with the rendered result statistics.
T-10	Opinion polling tool – different types of questions & answers	Must-have	Polling and Rating Manager / CSET	**Final status**: Implemented **Notes**: Polling component was implemented (from the scratch) as site page component in Akfresco Share. Polling form editor provides a user interface for editing of polling posts types. It registers new data model type and form definition for this type. Data model defines list of possible properties of the post node where each property corresponds to one question of the polling. Type and range of the values can be constrained, what allows to define various types of the questions (free text inputs, yes/no questions, single and multi-select from the predefined list of values). The form definition allows to specify layout how the form will be rendered on html page and customize help/validation messages and controls for each question.
T-7	Opinion polling tool – open forms	Must-have	Polling and Rating Manager / CSET	**Final status**: Implemented **Notes**: Due to fact that polling is represented using nodes in Alfresco repository, it is also possible to setup access rights on their creation, editing, etc. Therefore, authorised users (with granted access rights, e.g. facilitator in case of the scenario generation process) are able to conduct an opinion poll and define the users who are allowed to participate or to organize an open poll (using groups). Closing of polls is currently based on the organizational rules between users (usually setup by facilitators).

T-8	Opinion polling tool – participation of users in polls – one vote per person	Nice-to-have (refined from must-have)	Polling and Rating Manager, Version Manager / CSET	**Final status**: Postponed **Notes**: Polling component was implemented (from the scratch) in order to achieve expected functionality. This requirement was refined to nice-to-have priority due to simplification of polling implementation and low expected added value (in comparison to needed effort for the implementation). For possible solution in future, it is possible to check registered users, but it was difficult to identify unique guest users.
T-9	Opinion polling tool – participation of users in polls – possibility to modify the answers provided (versioning)	Must-have	Polling and Rating Manager, Version Manager / CSET, CMS	**Final status**: Implemented **Notes**: Polling component was implemented (from the scratch) in order to achieve expected functionality. This requirement is implemented in this way: user has possibility to post new answers, while previous answers are still available.
T-11	Opinion polling tool – presentation of the results	Must-have	Polling and Rating Manager / CSET, CMS	**Final status**: Implemented **Notes**: Polling component was implemented (from the scratch) in order to achieve expected functionality. The results service (available as REST service) aggregates the data in polling and provide them to the users. There is currently not available visualisation using graph or bars in polling page component, but the data from results can be imported into any available (and suitable) tool for visualisation, e.g., the graph from data can be shown on the wiki. As postponed condition, it can be also imported into polling page component using similar Google graphs generation.
I-F-I1	Password reminder	Nice-to-have (refined from must-have)	User Manager / CMS	**Final status**: Postponed **Notes**: This requirement was refined to nice-to-have priority due to trials runs (where it is/was not expected) and low expected added value (in comparison to needed effort for the implementation). Currently, any issues for user management are on system administrators, which are able to support the user by the application of organizational rules for run of pilot sites (user can contact administrator for help, etc.). It can be added to the system by implementing of such feature or as a part of Alfresco updates (if it will be available as functionality in some of the next versions).
I-F-I6	Personalise overview	Must-have	Collaboration Space Manager, Notification Manager, Process Manager, Search	**Final status**: Implemented **Notes**: Alfresco Share site page is customised and can be personalised according to the current case. There is also personal dashboard, where particular user can add his/her own dashlets for presentation of documents, content changes, activities

			Manager, User Manager / CSET, CMS	RSS, etc. It is possible for user aggregates the information for personal view throughout the all cases (sites), where he/she is involved.
T-17	PM (Analysis) - Export of simulation-related data	Should-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in form of DRAMS plugins for creating different kinds of structured output files (CSV, XML).
TP-3	PM (Analysis) - Narrative output	Must-have	Concept Manager, Link Manager, Simulation Manager / SE	**Final status**: Implemented **Notes**: DRAMS provides a print clause for writing textual logs; all pilot models make appropriate use of this functionality.
T-23	PM (Analysis) - Qualitative representation of the simulation results	Must-have	Concept Manager, Link Manager, Simulation Manager / SE	**Final status**: Implemented **Notes**: All models generate textual logs, that are used for development of narrative model-based scenarios.
TP-5	PM (Analysis) - Visualisations of non-numerical outcomes/events	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Provided through the implementation of traceability features.
TP-2	PM (Analysis) – Experiment and rule development browser	Should-have	Concept Manager, Link Manager, Simulation Manager / SE	**Final status**: Implemented **Notes**: This requirement is basically fulfilled with the DRAMS console (on-the-fly rule execution).
TP-1	PM (Analysis) – Within-timestep dependency graph visualisation	Must-have	Concept Manager, Link Manager, Simulation Manager / SE	**Final status**: Implemented **Notes**: This feature is provided by DRAMS Model Explorer plugin.
FR26_PM	PM (Experimentation) - Automated experimentation	Must-have	Simulation Manager / SE	**Final status**: Postponed **Notes**: Functionality rudimentarily provided by Repast simulation platform, but not used for pilot models.
FR25_PM	PM (Experimentation) - Change simulation parameters	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Part of Repast/Java model class.
FR22_PM	PM (Experimentation) - User engagement in simulation	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Vague requirement formulation, inherently fulfilled.
FR23_PM	PM (Experimentation) - User interaction	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Vague requirement formulation, inherently fulfilled.

FR24_PM	PM (Experimentation/Gaming) - Gaming	Must-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: This requirement was skipped due to resource limitations in favour of more significant developments in the focus of the project (CCD, traceability).
FR27_PM	PM (Gaming) - Feedback on simulation	Must-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: Rejected due to dependency on also rejected requirement FR24_PM.
T-32	PM (Gaming) – Role-playing games (single user)	Must-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: Rejected due to dependency on also rejected requirement FR24_PM.
T-33	PM (Gaming) – User interface for human player	Must-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: Rejected due to dependency on also rejected requirement FR24_PM.
FR07_PM	PM (Modelling process) - Agent type creation	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates the agent types defined in the CCD model.
FR08_PM	PM (Modelling process) - Agents at different aggregation levels	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool	**Final status**: Rejected **Notes**: Multi-level modelling was found not to be useful for the pilot applications.
NFR03_PM	PM (Modelling process) - End states	Must-have	Rule Manager / SE	**Final status**: Implemented **Notes**: Implemented within the simulation tool in SE.
FR10_PM	PM (Modelling process) - Environment definition - general	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates a simulation model for each CCD model.
FR09_PM	PM (Modelling process) - Exogenous factors	Must-have	Concept Manager, Link Manager, Rule Manager, Simulation Manager / SE	**Final status**: Implemented **Notes**: External events can trigger rule execution in simulation models.
NFR07_PM	PM (Modelling process) - General model description	Must-have	Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD tool allows to add descriptions to a CCD and each element of a CCD. CCD2DRAMS tool automatically creates a simulation model for each CCD model and adds descriptions as comments. Suitability as multi-purpose modelling approach has been proven by testing with different models (pilots and others).
NFR04_PM	PM (Modelling process) - Initial model definition	Should-have	Rule Manager / CCD Tool	**Final status**: Implemented **Notes**: CCD2DRAMS tool automatically creates a simulation model for each CCD

	(Beginner's mode)			model and adds descriptions as comments.
NFR05_PM	PM (Modelling process) - Iterations (Expert's mode)	Should-have	Rule Manager / CCD Tool	**Final status**: Rejected **Notes**: CCD2DRAMS tool does not revise existing code in iterations.
NFR06_PM	PM (Modelling process) - Model description	Must-have	Rule Manager / CCD	**Final status**: Implemented **Notes**: CCD tool allows to add descriptions to a CCD and each element of a CCD. CCD2DRAMS tool automatically creates a simulation model for each CCD model and adds descriptions as comments.
FR13_PM	PM (Simulation setup) - Initial state definition	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR12_PM	PM (Simulation setup) - Setup initial agent facts	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR11_PM	PM (Simulation setup) - Setup world facts	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR17_PM	PM (Simulation termination) - Adjustable parameters	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR14_PM	PM (Simulation termination) - End state	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR15_PM	PM (Simulation termination) - Irregular termination events	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR16_PM	PM (Simulation termination) - Regular termination events	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR21_PM	PM (Simulation termination) - Simulation abort	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR20_PM	PM (Simulation termination) - Simulation interrupt	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR19_PM	PM (Simulation termination) - Simulation start	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR18_PM	PM (Simulation termination)	Must-have	Simulation Manager / SE	**Final status**: Implemented

	- State validation			**Notes**: Implemented in Repast/DRAMS.
NFR08_PM	PM (Simulation) - Event handling	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
NFR09_PM	PM (Simulation) - Exception handling	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
NFR12_PM	PM (Simulation) - Parameter locking	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
NFR11_PM	PM (Simulation) - Parameter presentation	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
NFR14_PM	PM (Simulation) - Simulation execution	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
NFR10_PM	PM (Simulation) - Simulation visualization	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
NFR13_PM	PM (Simulation) - State handling for inspection	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Implemented in Repast/DRAMS.
FR06_PM	PM (Transformation process) - Assumption definition	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates a simulation model for each CCD model. Elements in the CCD are transferred too.
NFR01_PM	PM (Transformation process) - Data representation	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates a simulation model for each CCD model and adds descriptions as comments.
FR01_PM	PM (Transformation process) - Define initial policy modelling aspects	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates a simulation model for each CCD model. Elements in the CCD are transferred too. Actions are transferred into rule stubs.
FR03_PM	PM (Transformation process) - Environment generation	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates a simulation model for each CCD model. Elements in the CCD are transferred too.

FR04_PM	PM (Transformation process) - Goal definition	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates rule stubs for actions. Goals can be expressed in the CCD as comments of actions and are transferred in comments of rule stubs.
NFR02_PM	PM (Transformation process) - Language transition	Should-have	Annotation Manager / CCD Tool	**Final status**: Implemented **Notes**: This requirement is implemented as annotation of narrative texts.
FR05_PM	PM (Transformation process) - Rule generation	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates rule stubs for actions defined in the CCD.
FR02_PM	PM (Transformation process) - Stakeholder extraction	Must-have	Concept Manager, Link Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: CCD2DRAMS tool (as a part of CCD Tool) automatically creates agent classes for actors defined in the CCD.
T-22	Preview simulation mode – focusing on a part of the used model	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Partly implemented by the DRAMS console in SE (on-the-fly rule execution).
T-20	Preview simulation mode – level of details and/or time scale	Must-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: Requirement was found not compatible with the applied modelling approach during the run of the project.
T-21	Preview simulation mode – searching for a specified event	Should-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: This requirement is fulfilled by the implementation of traceability features within the platform.
T-19	Previewing of a simulation (means: state of running simulation can be observed)	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: This requirement is implemented in Repast/DRAMS.
I-NF-6	Privacy	Must-have	User Manager / CMS	**Final status**: Implemented **Notes**: Privacy of the information is correctly preserved thanks to Alfresco software (and its user management) and secured communication with Eclipse plugin.
I-24	Publishing of simulation results by the publishing tool (content management tool)	Must-have	Document Manager / CMS, CSET, SE	**Final status**: Implemented **Notes**: Simulation analysis tool allows to access the Eclipse Project Repository, where simulation logs, CCD model files, and other information artefacts necessary

				for creating a model-based scenario are persistently stored. In addition to that, the tool implements the interface provided by the Content Repository Client and communicates with the Alfresco Data Repository by means of both CMIS and REST protocols. The Simulation Analysis Tool provides the following functionality: creation of a narrative model-based scenario from logs of running simulations; publishing the created scenario to the Alfresco CMS as a wiki page; uploading the supportive materials such as CCD model, DRAMS source code, and simulation logs to the Alfresco shared space.
UC-8	Quantitative data analysis	Should-have	Annotation Manager / CCD Tool	**Final status**: Implemented **Notes**: The Quantitative Data Analysis is provided with annotation tool (within CCD Tool). The Analyst can go through the document and annotate relevant sections, code them into the CCD and links the section (quantitative information) to an existing issue. The Analyst can enter metadata of analysed section, revise it as well as delete.
SOTA-5	Real-time co-editing	Nice-to-have	Document Manager / CSET, CMS	**Final status**: Implemented **Notes**: This requirement is fulfilled in this way: there is possibility to publish some document on Google Docs document and there it can be co-edited (together with comments). This document can be then manually (e.g. by facilitator) uploaded back to the Alfresco Repository. Use case for this requirement is simple: Facilitator publishes document, asks the stakeholders to write, update and comment it, and he/she will close it and upload to Alfresco, when it is necessary.
I-F-I2	Removing profile	Must-have	User Manager / CMS	**Final status**: Implemented **Notes**: This requirement was slightly modified according to the functionality of Alfresco. The removing of the user (and its profile) is possible within the system, but it is available only for administration users, therefore currently, if some user wants to be removed, he has to contact administrator of the system. For membership within the current case it is possible to automatically leave the site using on action link 'Leave site'.
I-NF-3	Response Time	Must-have	General Requirement	**Final status**: Implemented **Notes**: We have not tested response times for particular settings mentioned in the initial version of requirement, but according to the benchmark tests of scalability and running of Alfresco platform (which are widely available for free) response time is not the issue even for large commercial applications based on Alfresco.

				Also, we have not encountered problems with the response of Alfresco Share during the project (of course, apart from the moments with bugged installation or crash of the server, which represent non-standard situation).
UC-1	Rights management	Must-have	Collaboration Space Manager, Process Manager, User Manager / CMS, CSET	**Final status**: Implemented **Notes**: User management functionality of Alfresco software is used for rights management.
T-30	RSS	Must-have	Notification Manager / CMS, CSET	**Final status**: Implemented **Notes**: RSS channels are supported within the Alfresco platform. It is possible to read them using particular dashlet in Alfresco Share interface, but it is also possible to use its address and read it in any RSS reader preferred by the user.
UC-3	Send request for invitation	Nice-to-have	Process Manager, User Manager / CMS, CSET	**Final status**: Implemented **Notes**: Due to low priority we have decided for simple solution. Request for invitation is based on 'request to join' function within the particular site (to current case) in Alfresco Share. For more information, the user is able to contact site administrator or facilitator using other channel (it can be detailed within the organizational rules for case).
SOTA-4	Several document editors	Should-have	Document Manager / CMS, CSET	**Final status**: Implemented **Notes**: The users are able to edit documents in Alfresco Repository, mostly the wiki page (which was found as fundamental for preparation of scenario), but also the other sources. For co-editing of one file by more users Google Docs document can be used (see Real-time co-editing requirement for more details).
T-28	Shared calendar with events related to the current processes	Should-have	Calendar Manager, Notification Manager / CSET, CMS	**Final status**: Implemented **Notes**: Shared calendar is based on its default implementation in Alfresco Share, where it is possible to share events within the current site (case).
I-34	Simulation back-end integrated with the ICT toolbox	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Supported by CCD Tool, CCD2DRAMS and Repast/DRAMS.
I-27	Simulation preview tool available from different physical locations – remote access	Must-have	Simulation Manager / SE	**Final status**: Rejected **Notes**: According to the OCOPOMO process running of simulation models is performed by policy modellers/simulation analysts. This is supported by the Eclipse part of the toolbox. Remote access is implemented for model artefacts and

				simulation results/model-based scenarios.
T-36	Simulation tool – performance in simulation cycles	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Simulation is done in discrete time mode (cycles). The number of necessary steps and computing time is dependent on model, number of agents and used hardware. Pilot models can be executed in matter of minutes.
T-35	Simulation tool – the number of agents	Must-have	Simulation Manager / SE	**Final status**: Implemented **Notes**: Pilot models are initialised with up to a few hundred of agent, which is a reasonable number.
I-3	Starting the scenario generation process - initial scenario	Must-have	Process Manager / CMS, CSET	**Final status**: Implemented **Notes**: Alfresco supports the workflow engine (based on the Activiti or JBPM engine), which can be used for automatic control of some predefined workflow. Therefore, it is possible to support such feature, but due to the refinement of the OCOPOMO process application in our pilots, where automatic control was not finally expected, we have followed organizational rules and control by the facilitators in scenario generation process (who was responsible to control - with the help and suggestions of other users - when it is time to write scenario, close it, etc.).
I-12	Support for direct export/import of information between scenario generation process and policy modelling	Should-have	Concept Manager, Link Manager, Process Manager, Rule Manager / CCD, SE, CMS	**Final status**: Implemented **Notes**: Several channels for sharing the information are available and implemented for traceability of information within the process. Most of the connections are realized using defined standard APIs like CMIS or REST services.
I-15	Support for the policy modelling tool to create a new scenario generation iteration	Must-have	Process Manager / CMS, SE, CSET	**Final status**: Implemented **Notes**: Alfresco supports the workflow engine (based on the Activiti or JBPM engine), which can be used for automatic control of some predefined workflow. Therefore, it is possible to support such feature, but due to the refinement of the OCOPOMO process application in our pilots, where automatic control was not finally expected, we have followed organizational rules and control by the facilitators, modellers or analysts in their processes. In this case, it is up to responsible person (e.g. facilitator) who decided (with the help of others) to start with the creation of new scenario.
T-42	Tags	Should-have	Calendar Manager, Document Manager,	**Final status**: Implemented **Notes**: Tags are fully supported within the Alfresco Share. It is possible to setup

			Discussion Forums Manager, Polling and Rating Manager / CMS, CSET	scheme with predefined tags, as well as add them incrementally by the users. Tags are realized as aspects, which can be added to (almost any) content shared in Alfresco.
T-43	Tags - automatic support	Nice-to-have	N.A.	**Final status**: Rejected **Notes**: This requirement was rejected before the design of the architecture (due to low priority).
T-6	Teleconferencing tool	Nice-to-have	N.A.	**Final status**: Rejected **Notes**: This requirement was rejected before the design of the architecture (due to low priority).
T-38	Transcription tool	Should-have	N.A.	**Final status**: Rejected **Notes**: This requirement was rejected before the design of the architecture (due to relatively low priority and high expected efforts).
I-2	Transformation table - connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox	Must-have	Annotation Manager, Concept Manager, Link Manager / CCD Tool, CMS	**Final status**: Implemented **Notes**: This requirement is fulfilled by the Simulation Analysis Tool which allows policy analysts link traceable logs from simulation run to created narrative model-based scenario.
I-40	Transition table browser	Must-have	Concept Manager, Link Manager, Rule Manager, Annotation Manager, Document Manager / CCD Tool, CSET	**Final status**: Implemented **Notes**: This requirement is fulfilled by the Simulation Analysis Tool, which publishes simulation scenarios as a wiki pages annotated with the CCD concepts and links to evidence-based scenarios. Wiki pages were extended with the visualisation of annotations and with the CCD explorer for the browsing of relevant CCD concepts.
I-30	Translation of agent rules from a tool neutral syntax into simulation back-end language	Must-have	Annotation Manager, Rule Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: This requirement is fulfilled by the annotation of texts with CCD Tool software and its transformation into DRAMS Code.
UC-5	Update description of the project	Must-have	Collaboration Space Manager, Document Manager / CSET	**Final status**: Implemented **Notes**: Site in Alfresco Share represents the project or case (for policy modelling). Its description can be updated, together with the other related documents (e.g.

				starting wiki page).
I-NF-1	Usability	Must-have	General Requirement	**Final status**: Implemented **Notes**: In order to achieve better usability we have decided (based on the experience of users and comments from reviewers) to divide platform into two basic interfaces: Alfresco Share part, which is mostly used for scenario generation, publishing of results and collaboration, and Eclipse plugin with tools more related to the expert process members (modellers, facilitator), where it is necessary to do annotations, modelling, coding, simulation running and analysis of results. Also, customisation of Alfresco site component (which dashlets are shown on site dashboard and which page components are used within the site for particular policy case) is the way for improvement of usability. Another important point is support for traceability of information from simulation results to CCD and scenarios, what is also supported for better usability (it helps to understand the results without problems with the long searching for information).
I-F-I5	User profile	Must-have	Collaboration Space Manager, Notification Manager, User Manager / CMS, CSET	**Final status**: Implemented **Notes**: User profile is used from the Alfresco Share user management.
I-F-I4	User registration	Must-have	User Manager / CMS, CSET	**Final status**: Implemented **Notes**: The registration of users is based on the user management implementation of Alfresco Share.
I-26	Version control of process models and/or agent models	Must-have	Version Manager, Rule Manager, Concept Manager, Link Manager / CCD Tool, SE	**Final status**: Implemented **Notes**: Versioning of the documents specific for models is done by the standard versioning software (SVN).
I-32	Workflow support	Must-have	Collaboration Space Manager, Notification Manager, Process Manager / CMS, CSET	**Final status**: Implemented **Notes**: Workflow functionality is available within the Alfresco system with Activiti and JBPM implementation, with its support for process management. Due to revisions of our needs for the trials, we have decided to use management based on the organizational rules defined by the responsible users (facilitator, modeller, etc.), who controls the process. For more sophisticated business process management (can be applied in next projects or reuse by others), Alfresco supports preparing of the BPM workflows for management of the whole process.

SOTA-1	Workflow engine	Should-have	Document Manager, Process Manager / CMS	**Final status**: Implemented **Notes**: : Workflow engine functionality is available within the Alfresco system with Activiti (default) and JBPM implementation, with its support for process management (compatible with BPMN standard).

Table A-2: Added requirements for the final implementation.

ID	Name	Priority	Module / Part	Solution / Application / Usage
TRAC-3	Facilitator should be able to delete some versions in version history	Major	CMS / Repository	**Final status**: Rejected **Notes**: Versions are organized in tree-like structure; delete of some version can cause deletion of whole subtree.
TRAC-4	Commenting on selected text in Wiki	Major	CSET / Wiki	**Final status**: Implemented **Notes**: Commenting functionality was added to the Wiki page. Comments are directly visible on the page with the content or on the page preview in the detailed view with the list of versions. However, it is possible only to comment whole content (commenting of the selected text will require extensible modifications of the comment API which was already integrated to many components). User can use direct editing of the wiki page content to add comments for the selected text using the different formatting.
TRAC-6	Only the facilitator and person who uploaded document can delete it	Major	CMS / Document Library	**Final status**: Implemented **Notes**: Each document has associated different access rights for different user groups or specific users, however this has to be configured in Alfresco Explorer client, Alfresco Share provides only limited configuration settings.
TRAC-8	Answer to particular comment about the content	Major	Unspecified	**Final status**: Implemented **Notes**: This is supported in Discussion forum manager. User can create manually new discussion topic for the particular content.
TRAC-9	Indicate presence of the user on the site after the login	Minor	CSET / Site and Personal Dashboard	**Final status**: Rejected **Notes**: Online indicator is still unsupported by the latest Alfresco version of services (in 4.0d, which will be used in Trial 2). Moreover, due to minor priority, this requirement was rejected (as planned also in previous reports, if it will be not

				supported in Alfresco version till Trial 2).
TRAC-10	Improve and complete Slovak localization	Major	Unspecified	**Final status**: Implemented **Notes**: Improvement of Slovak localization was realized for Alfresco components compatible with version 4.0d.
TRAC-11	Upgrade to current version of Alfresco	Major	Unspecified	**Final status**: Implemented **Notes**: Upgrade of Alfresco server from Trial 1 version (4.0a) to new version for Trial 2 (4.0d) was realized. Some components have been updated according to their specific need for new version.
TRAC-12	Add Italian resources for improvement of localization for new components	Major	Unspecified	**Final status**: Implemented **Notes**: Italian resources for new components (e.g., chat, pollings) in Alfresco were updated using resource files.
TRAC-13	Support more question types	Major	CSET / Pollings	**Final status**: Implemented **Notes**: Plain text answer type was added besides of the one-to-many and many-to-many enumerated answers.
TRAC-14	Allow to publish polling results as a document in Document library	Major	CSET / Pollings	**Final status**: Implemented **Notes**: It is now possible to publish the results of polling as the comma separated file stored in the selected folder in the Document library.
TRAC-16	Filtering Diagrams	Minor	CCD / CCD Modelling	**Final status**: Implemented **Notes**: 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*. • 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*. • Hide unconnected elements: It is possible in the diagrams to select one or more nodes and hide all unconnected nodes (see **Chyba! Nenalezen zdroj odkazů.**). With "arrange all" it is possible to arrange the remaining elements. To show again all elements please click "Show hidden elements".
TRAC-17	Validation	Major	CCD / CCD Modelling	**Final status**: Postponed **Notes**: This requirement was postponed as out of scope due to no validation

				functionality implemented.
TRAC-20	Design mock-up user interface for presentation of CCD concepts and traceability links on Alfresco site	Major	Unspecified	**Final status**: Implemented **Notes**: Mock-ups of user interfaces for traceability were developed accordingly to the needs of the modellers.
TRAC-21	Implement export of CCD model and links from Eclipse to Alfresco	Major	Unspecified	**Final status**: Implemented **Notes**: Simulation Analysis Tool is able to upload CCD model from Eclipse to Alfresco during publication of narrative model-based scenario.
TRAC-22	Implement Alfresco services for CCD concepts and traceability links	Major	Unspecified	**Final status**: Implemented **Notes**: All CCD artefacts (i.e. diagram files, traces, etc.) are accessible through data proxy. Proxy provides authenticated access web access to the CMS repository for web client and CCD Explorer applet.
TRAC-23	Implement user interface extensions for presentation of CCD concepts and traceability links	Major	Unspecified	**Final status**: Implemented **Notes**: Wiki page manager was extended with the preview of traceability annotations, annotation popup dialog and CCD Explorer applet dialog.
TRAC-24	Integrate data source for simulation results to Alfresco site	Major	Unspecified	**Final status**: Implemented **Notes**: Data source provides Google visualization API for the visualization of simulation data on the Alfresco site. It is connected to the data proxy which fetch simulation traces stored in the CMS repository.
TRAC-25	Implement extensions of Wiki editor for embedding of charts	Major	CSET / Wiki	**Final status**: Implemented **Notes**: Base settings for the charts are supported, such as configuration of the source data table and columns and base settings as the title, width and height of embedded charts. Chart is embedded into the HTML as JSOM mark-up which can be further customized in HTML preview mode.
TRAC-26	Add list of participants in current chat view	Major	CSET / Site and Personal Dashboard	**Final status**: Implemented **Notes**: In chat view it is now possible to see current users of opened chat, implementation was based on the update of chat dashlet code.
TRAC-27	Implement log library for DRAMS with the traceability	Major	SE / DRAMS	**Final status**: Implemented **Notes**: Plugins for writing log information and numerical data with trace

	support			information have been implemented.
TRAC-28	Prepare package with CMIS module for Eclipse	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Content Repository Client provides CMIS and REST communication between Alfresco and Eclipse repositories.
TRAC-29	Integrate CMIS to current version of CCD Tools with Text Annotation	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Content Repository Client provides CMIS and REST communication between Alfresco and Eclipse repositories.
TRAC-30	Integrate PDF Annotation	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: PDF Annotation is integrated into existing CCD Tool as new annotation extension.
TRAC-31	Integrate HTML Annotation	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: HTML Annotation is integrated into existing CCD Tool as a new annotation extension.
TRAC-37	Filtering literals after selecting an enum	Major	CCD / CCD Modelling	**Final status**: Implemented **Notes**: By choosing a specific enum as an attribute, only the literals which are defined for this enum should be viewable as the possible target value.
TRAC-42	In CCD, lists should be possible attribute values	Major	CCD / CCD Modelling	**Final status**: Rejected **Notes**: In CCD, lists should be possible attribute values. No workaround available for CCD Modelling, but lists can be used in simulation model code.
TRAC-44	Finding annotations from marked-up document passages	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Finding annotations from marked-up document passages. This feature is implemented in the CCD Viewer applet (CCD Web Interface) and in combination with the new Annotation plugins.
TRAC-45	Deleting documents should delete annotations	Major	CCD / CCD Annotations	**Final status**: Rejected **Notes**: The deletion of annotations with documents would kill important evidences. Another solution is necessary as e.g. to avoid deletion of documents as long annotations are available.
TRAC-48	In console, show only instantiations of selected agents	Minor	SE / DRAMS	**Final status**: Implemented **Notes**: A filter function for the fact base browser has been implemented.

TRAC-49	Edit message in chat	Minor	CSET / Site and Personal Dashboard	**Final status**: Rejected **Notes**: First, it was postponed for decision after Trial 1. Then, as it was not found important, and it has only nice-to-have priority in D2.1, and XMPP does not support such operation (e.g. in Skype it is a vendor-specific solution), this requirement was finally rejected.
TRAC-50	Chat - necessary updates for migration to Alfresco 4.0d	Major	CSET / Site and Personal Dashboard	**Final status**: Implemented **Notes**: Chat source code was updated in order to achieve successful migration to new version of Alfresco.
TRAC-51	Advanced roster management with non-Alfresco users	Minor	CSET / Site and Personal Dashboard	**Final status**: Rejected **Notes**: Advanced management is the possibility for the integration with external communication tools using Jabber protocol. It was decided that support for external tools is out of the project scope.
TRAC-52	Online indicator for users which can be used in chat	Minor	CSET / Site and Personal Dashboard	**Final status**: Rejected **Notes**: Postponed for decision after Trial 1: it was not found important, and due to fact that online indicator is still unsupported by the latest Alfresco version of services and rejection of advanced management of chat users (#51), this requirement was finally rejected.
TRAC-53	MyNote Annotion plug-in for CCD Tool	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: New annotation plugin was created to support requested functionality.
TRAC-54	Creating and killing agents	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Clauses for creating and killing agents have been added, as well as extensions to the model interface for related Java-based functionality.
TRAC-55	List processing in DRAMS	Major	SE / DRAMS	**Final status**: Implemented **Notes**: List processing clauses available in DRAMS.
TRAC-57	Remove new project wizard	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Project wizard configuration of OCOPOMO Eclipse project was re-implemented as new project property page.
TRAC-58	Simulation Analysis Tool	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Simulation Analysis Tool is used to support creation of traceable model-based scenario, which can be then shown to the users.

TRAC-59	Upload narrative model based scenario as Alfresco wiki page	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Narrative model-based scenario can be published as a new wiki page via Alfresco REST API.
TRAC-60	XML to wiki transformation	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: Model-based scenario created as XML file is transformed to HTML format. Ready for publishing as Alfresco Wiki page.
TRAC-61	Locking mechanism	Major	CMS / Repository	**Final status**: Implemented / Rejected / Postponed **Notes**: Implemented using the Alfresco CMS access rights management.
TRAC-62	Known-by relation (CCD Modelling)	Major	CCD / CCD Modelling	**Final status**: Implemented **Notes**: Extension to CCD meta model was implemented.
TRAC-63	Generation of link tags (CCD2DRAMS)	Major	CCD / CCD Modelling	**Final status**: Implemented **Notes**: Extensions of CCD2DRAMS for preserving links to CCD elements in simulation model code have been fully implemented.
TRAC-64	Environment for editing, debugging and inspecting Java code of Repast models	Major	SE / DRAMS	**Final status**: Implemented **Notes**: Environment for editing, debugging and inspecting of code in Repast is realized as built-in functionality of Eclipse IDE.
TRAC-65	User interface for editing declarative rules of the DRAMS model	Major	CCD / CCD Annotations	**Final status**: Implemented **Notes**: User interface for rules editing in DRAMS is implemented as Editor part of DRAMS Eclipse plugin.
TRAC-66	Environment for integrated debugging of Repast/DRAMS models	Major	SE / DRAMS	**Final status**: Implemented **Notes**: DRAMS Console with on-the-fly rule execution, fact base inspection and detailed log/error information has been implemented in order to fulfil this requirement.
TRAC-67	Plug-in that provides (interactive) views with several different graphical representations of the model structure (data/rule and rule dependency graphs; currently under development)	Major	SE / DRAMS	**Final status**: Implemented **Notes**: Filtering functionality has been added to existing dependency graph visualisations. A Model Explorer plugin for DRAMS has been developed; this tool allows for analysing and visualising the inference process of the rule engine, as well as inspecting traceability information.

TRAC-68	Plug-in which allows adding and maintaining special tags attached to code fragments, referring to CCD elements from which the code fragments originate (for both Java and DRAMS code)	Major	SE / DRAMS	**Final status**: Implemented **Notes**: The link preservation/traceability functionality has been implemented into DRAMS and the transformation tool (CCD2DRAMS).
TRAC-69	Plug-in for inspecting simulation outcomes, with the possibility to link from any log phrase back to the originating rule (and furthermore to the referenced CCD element)	Major	SE / DRAMS	**Final status**: Implemented **Notes**: Model Explorer plugin for DRAMS has been developed; this tool allows for inspecting traceability information. Integration into Eclipse has been rejected as not necessary.
TRAC-70	Locking mechanism	Major	CSET / Wiki	**Final status**: Implemented **Notes**: When modeller starts modelling scenarios and background documents on Alfresco site will be prevented of further modification.