D8.1: Manual of the Methodology for Process Development and Guide to Policy Modelling Toolbox

Document Full Name	**OCOPOMO_D8.1_v1**
Date	**19/05/2013**
Work Package	**WP8: Development of methodology, guidelines and user support**
Lead Partner	**University of Koblenz-Landau**
Authors	**Sabrina Scherer, Ulf Lotzmann, Maria A. Wimmer, Karol Furdik, Marian Mach, Tomas Sabol, Peter Butka, Peter Bednar, Scott Moss, Daniele Pinotti, Marta Kacprzyk, Peter Smatana, Magda Roszczynska- Kurasinska**
Document status	**V1.0**
Dissemination level	**PUBLIC (PU)**

Table of Contents

List of Figures and Tables

List of Abbreviations

CCD	Consistent Conceptual Description
CCD2DRAMS	CCD to DRAMS Transformation
CMS	Content Management Server
CSET	Collaboration and Scenario Editing Tools
DRAMS	Declarative Rule-based Agent Modelling System
MOF	Meta Object Facility
OMG	Object Management Group
QVT	Query/Transformation/View
XML	eXtensible Markup Language

Executive Summary

OCOPOMO created an online consultation and open collaboration approach to involve stakeholders in the policy formation process thereby being participatory, consensus oriented, transparent, and best possible inclusive (i.e. implementing good governance principles). Stakeholders are collaboratively involved in the development of scenario texts relevant in the context of a policy under discussion in the political agenda. These scenarios and further background information are used as base for the development of respective policy models by expert policy modellers. In the context of the OCOPOMO project, a policy model means a conceptual model and/or a simulation model designed and implemented to explore the individual actions and combinations of actions that are believed to be available to governments or key policy decision makers for the purpose of achieving specific and well-formulated objectives. The agent-based simulation of these policy models shall give insights into the impact of a new policy options to decision makers and stakeholders.

In this deliverable, guidelines are provided in terms of how to use the OCOPOMO outcomes. The guidelines and recommendations comprise indications for how to use and how to apply the overall OCOPOMO policy development process as well as its respective ICT support tools developed in OCOPOMO. Such guidelines are given in two directions: first, methodical grounds; second, guidelines and how-tos for applying the solution (process and ICT tools). The deliverable also outlines the actors targeted with the solution and it provides insights on whether the solutions can be transferred to different policy domains and what potential limitations may exist in application or customisation of the toolbox or the overall process. The deliverable finally outlines the added value of the OCOPOMO solutions for policy makers, stakeholders and policy modellers.

1. Introduction

Work package 8 brings together the different strands of work done in WP1, WP5, WP6 and WP7 on the particular approach to policy modelling proposed in OCOPOMO. The objectives of this WP are to document the methodology for processes used in policy modelling and scenario development and to develop guidelines easy to understand and easy to use for transferring the solution to other application domains. Based on the experiences and results of the requirements analysis (WP1), the policy modelling and scenario process design (WP5, WP6), a detailed description of the methodology developed and applied in the course of the project is given. This includes instructions on how to use the integrated IT platform (WP2 - WP4) for the purposes of open collaboration in policy making. The resulting manual (this deliverable) is aimed at expert stakeholders (policy operators, policy analysts, domain experts, etc.) but will be understandable to a wider audience (interest groups, general public), too.

OCOPOMO created an online consultation and open collaboration approach to involve stakeholders in the policy formation process thereby being participatory, consensus oriented, transparent, and best possible inclusive (i.e. implementing good governance principles). Stakeholders are collaboratively involved in the development of scenario texts relevant in the context of a policy under discussion in the political agenda. These scenarios and further background information are used as base for the development of respective policy models by expert policy modellers. In the context of the OCOPOMO project, a policy model means a simulation model designed and implemented to explore the individual actions and combinations of actions that are believed to be available to governments for the purpose of achieving specific and well-formulated objectives. The agent-based simulation of these policy models shall give insights into the impact of the new policy to decision makers and stakeholders.

This deliverable provides guidelines along the process in regards to which actors are to be involved, which methods are to be applied and which tools of the OCOPOMO ICT toolbox are to be used. Abstracting from the specifics of the three case studies, guidelines for the development and processing of policy models in other application domains are identified and elaborated.

The report is structured as follows: Section 2 introduces the solution proposed in OCOPOMO. This includes the overall OCOPOMO policy development process, actors as well as an outline of the supporting ICT toolbox. The methodologies applied in the OCOPOMO policy development process are described in section 3. Section 4 focuses on stakeholder engagement methodologies as they are supposed to be applied during the whole OCOPOMO policy development process in order to support the different methods. Section 5 presents a guideline for transferring the OCOPOMO solution to other application domains. Thereby general, methodological and technical considerations are given and experiences from the OCOPOMO pilot cases are provided. In section 5, the added value of the OCOPOMO approach for the stakeholder groups is discussed. Finally, conclusions are given in section 6.

2. OCOPOMO Policy Development Process

2.1 Process phases

The main stages of the OCOPOMO policy development process are visualised in Figure 1: *Initiation* of policy development process, *collaborative discussion* of the policy case based on stakeholder scenarios, *knowledge representation* in the form of a conceptual model of the policy case, *modelling and simulation* of the policy model, and iterative evaluation of the policy model in a *collaborative discussion*. Starting point is either an existing policy, which needs to be revised, or a need for a new policy. A policy can be brought in either by a government agency (i.e. domain expert) or a particular stakeholder group or even by an interest group (i.e. a particular stakeholder group). The results are consistent policy models, which give particular insights into the policy case under discussion.

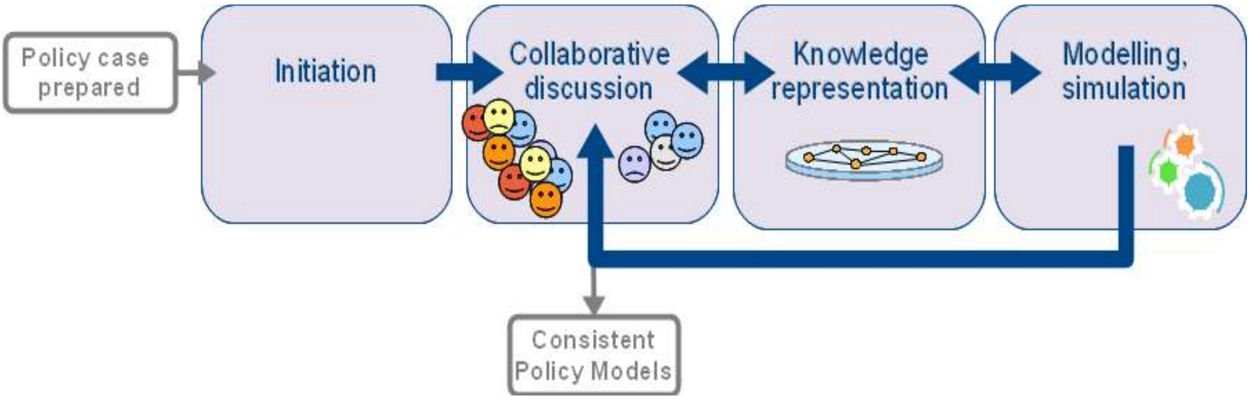

Figure 1: Main stages in the OCOPOMO policy development process

The overall policy development process of OCOPOMO is detailed in Figure 2 along the activities of policy formulation through s*cenario generation* (encompassing phase 1 – initial scenario, and phase 2 – scenario generation through stakeholders), *scenario analysis and conceptual modelling* (encompassing phase 3), t*ransformation and formal model generation* (encompassing phase 4), *simulation* (encompassing phase 5), and *evaluation and validation* (encompassing phase 6). Each phase is further explained below.

The artefacts, actors, and tools of the OCPOMO ICT toolbox along the OCOPOMO policy development process are described in subsequent sections 2.2 to 2.3. Preconditions for using the solution are presented in section 2.4.

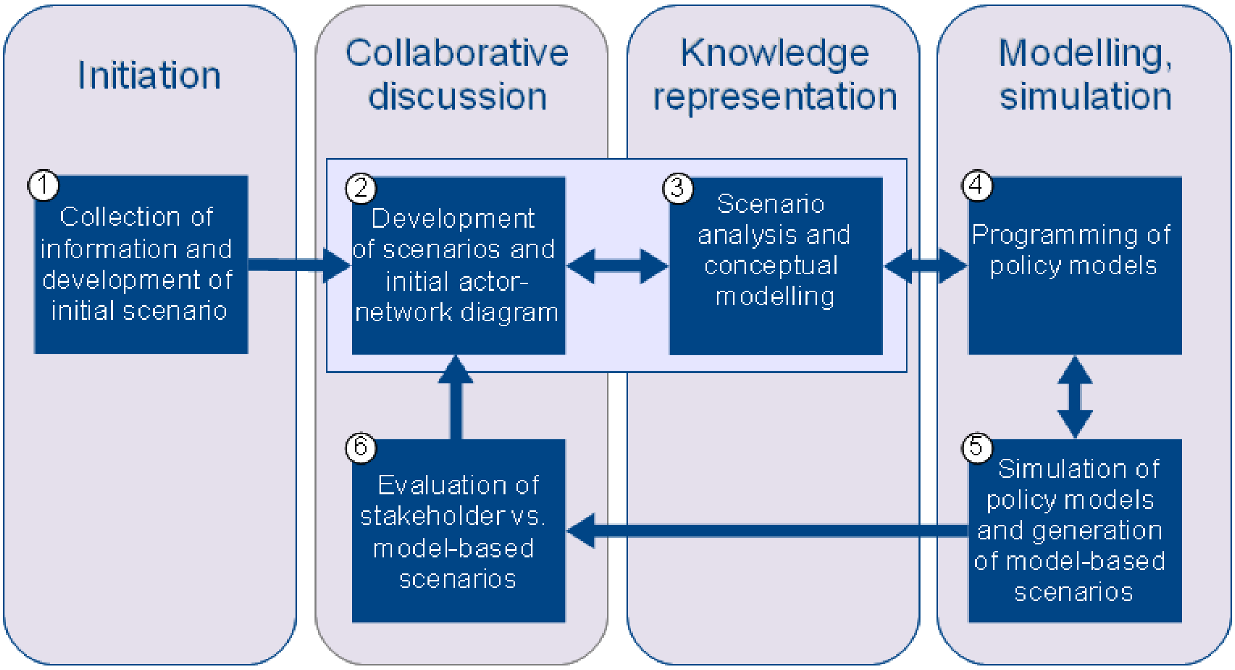

Figure 2: Six phases of the OCOPOMO policy development process

1. **Collection of information and generation of initial scenario:** Based on the policy, background information is gathered and an initial scenario is generated by facilitators and domain experts and/or policy owners.

   In OCOPOMO, scenarios are understood as narrative texts formulated in the language of participating stakeholders and domain experts (Carroll 1995). The initial scenario gives the wider stakeholders to be involved in the subsequent phase an example in regards to the particular policy case as basis for further discussions. The scenario can be provoking in order to stimulate discussions among stakeholders. Background information are documents informing about the policy under discussion, alternatives etc.

2. **Development of evidence-based stakeholder-generated scenarios:** Stakeholders involved and domain experts develop further scenarios of different kinds by describing their points of views in stories regarding the policy. Scenarios are seen as means to gather information from the view of stakeholders in addition to background documents.

   In this phase, it is recommended to initiate stakeholders' participation with a workshop in order to explain the aims of the project, the process, expectations of stakeholders etc. and to bring the participants together.

   The workshop can also serve as opportunity to develop an initial actor-network diagram together with main stakeholders and experts in order to write down important actors, things/objects, aims etc. and their relationships.

In phase 2, stakeholders and domain experts are supported by facilitators. They can collaborate online on the OCOPOMO platform in generating scenarios or in discussing scenarios. The scenario-based approach applied in OCOPOMO is detailed in section 3.1.

During phases 1 and 2, the objectives and focus of the conceptual and formal (agent-based) policy model are defined. Therefore, a document is prepared answering the questions:

- What should be modelled?
- Are the modelling objectives in line with available information or do we need more information?
- Which questions should be answered with the policy model?

Questions should be defined with decision makers (policy owners) in order to develop a policy model, which supports them in decision making (i.e. policy owners, domain experts).

3. **Scenario analysis and conceptual modelling:** To support the comprehension and understanding of a complex formal simulation model, the OCOPOMO process foresees the use of consistent conceptual models (CCDs). Such a conceptual policy model is created from the narrative and textual information base collected in phases 1 and 2, and this artefact helps to inform the formal policy models. CCDs are therefore bridging the narrative artefacts generated by stakeholders and the formal model artefacts created by policy modellers. These conceptual models are generated by policy analysts and used by policy modellers for representation of knowledge available about the policy case and to develop the policy models.

   In OCOPOMO, conceptual models are also used to trace the transformation of information from narrative text scenarios to formal policy and simulation models through the storage of the links between narrative text and conceptual model elements, which are then conveyed to the formal simulation model. This way, traceability of how information is transformed into the formal simulation model becomes visible through tool support (see also explanations in section 3.2).

   Before a CCD can be created, the policy analysts need to extensively analyse and study documents available in order to get an overview of the policy domain. An initial actor-network diagram developed with stakeholders and domain experts (cf. previous phase) can be a good starting point for further analysis.

   Then, the policy analyst starts to annotate texts to develop the conceptual model - thereby storing links between narrations and the conceptual model elements. The initial narrative scenario together with additional background documents and the document describing the objectives of the policy model give first input to the CCD developments. Additional scenarios generated and background documents uploaded in phase 2 by different stakeholders enrich the CCD models with further input.

Conceptual modelling is a collaborative process, where policy analysts bring in their experience but also come back to the stakeholders and domain experts or interact with the policy modellers with questions or discuss model elements with them. For example, if the conceptual model shows inconsistencies in the scenarios, the policy analyst might come back to scenario authors with questions (using the OCOPOMO platform as collaboration space). Variations from or extensions to the scenarios can be conceptualised by the policy analyst adding respective descriptions (called 'expert annotations') to the CCD elements. Discrepancies in stakeholder or domain expert views between scenarios are not filtered out. They are either described in one conceptual model or in different versions of the conceptual model (depending on the objective of the overall policy modelling). Then different policy models can be created to compare the simulation results.

Methodologies for scenario analysis and conceptual modelling are described in section 3.2.

4. **Programming of policy model:** Based on the CCD model, policy modellers develop the agent-based formal policy model, on which the simulation runs. In this context, the actor network is of particular importance for the development of the simulation model as it presents relevant information in regards to interdependencies of actors (e.g. an actor only sets an action based on the behaviour or impact on another actor). In OCOPOMO, multi-agent modelling (Gilbert and Troitzsch, 2005) is used. Hence, formal policy models (also called simulation models) have to cover actor descriptions, their social relationships, individual behaviour, beliefs and actions as well as rules and conditional dependencies among actions of actors. The simulation models accommodate in sum the relationships between the individual actions on the micro-level and the collective effects on the macro level to help understand interrelation and interdependencies and thereby making the system manageable. Methodologies for Transformation and Formal Policy Modelling are described in section 3.3.

5. **Simulation and generation of model-based scenarios:** Formal policy models are the starting point for running simulations of the policy case. In this phase, experts of policy modelling instantiate simulation models with particular variables and run the simulations. The results received from such simulations are textual logs, which have to be interpreted and visualised in proper format. Hence, policy modellers and policy analysts transform these textual logs from simulation runs into so-called model-based scenarios (i.e. a narrative text format) and into supportive graphical charts if respective data is an outcome of the simulation. Visualisation is needed to demonstrate how a strongly connected operation works, and which results are generated and derivable from current scenario descriptions. Visualisation is essential to provide simulation results to different users such as stakeholders or policy owners. Only through such visualisations, feedback and reasonable interaction with the stakeholders and policy owners or domain experts is possible. Such model-based scenarios and graphical charts provide a common and generally understandable format of presenting simulation outcomes. Methodologies for Formal Policy Modelling are described in section 3.4.

6. **Evaluation of evidence-based vs. model-based scenarios:** Phase 6 of the overall policy development process serves to expose the model-based scenarios (i.e. the simulation results of phase 5) to different groups (policy owners, domain experts and stakeholders). The purpose is that these groups assess, evaluate and validate the results of the simulation and therewith compare these results with the evidence-based scenarios they have generated or have in mind. Through this evaluation steps, stakeholders can reflect their positions expressed in scenarios. They may enrich their scenarios (feeding further information into phase 2 above) or may also better understand opposing positions of other stakeholders and negotiate the result to come to common agreements of a consistent policy model. A key benefit of social simulation is that aspects most probably not evident to the stakeholders through textual descriptions become visible. Methodologies for the evaluation of model outcome vs. scenario input are described in section 3.5.

OCOPOMO involves iterative implementation cycles. If the model-based scenarios of phase 6 do not sufficiently reflect the expectations of the involved stakeholders, domain experts or policy owners, these actors may revise their scenarios or may request revisions in the conceptual model (phase 3) and the simulation model (phase 4). The revision of stakeholder-generated scenarios leads to revisions of the conceptual and the simulation model. Hence, a new iteration of the process is executed. This iteration cycle is needed to ensure that, in the end, the evidence-based scenario documents are consistent with the formal policy model outcomes (i.e. the model-based scenarios).

Furthermore, OCOPOMO does not suggest a strict sequence of the process phases as outlined above. Actually, extensive interactions take place among actors of phases 2 (collaborative scenario generation) and 3 (policy analysis and generation of conceptual models) as well as among phases 3, 4 (development of simulation model) and 5 (running simulations). Hence, the development of a formal policy model is a back-and-forth and collaborative process among phases 2 through to 5. Policy analysts and policy modellers bring in their experience while also having to come back to the stakeholders and domain experts with questions or in order to discuss model elements with them. Hence, on-going interaction in OCOPOMO's collaboration space can produce revisions in the model during one implementation cycle.

The final result of the OCOPOMO policy development process are one or more consistent policy models that reflect the policy case in a formally correct policy model and that give the strategic decision makers as well as stakeholders deeper and new insights into the policy case and into alternative policy options. The policy models are themselves consistent but may be inconsistent to each other in order to represent conflicting stakeholders' opinions (and alternative policy options), if such exist.

Figure 3 presents the artefacts and their usage along the OCOPOMO policy development process as described above.

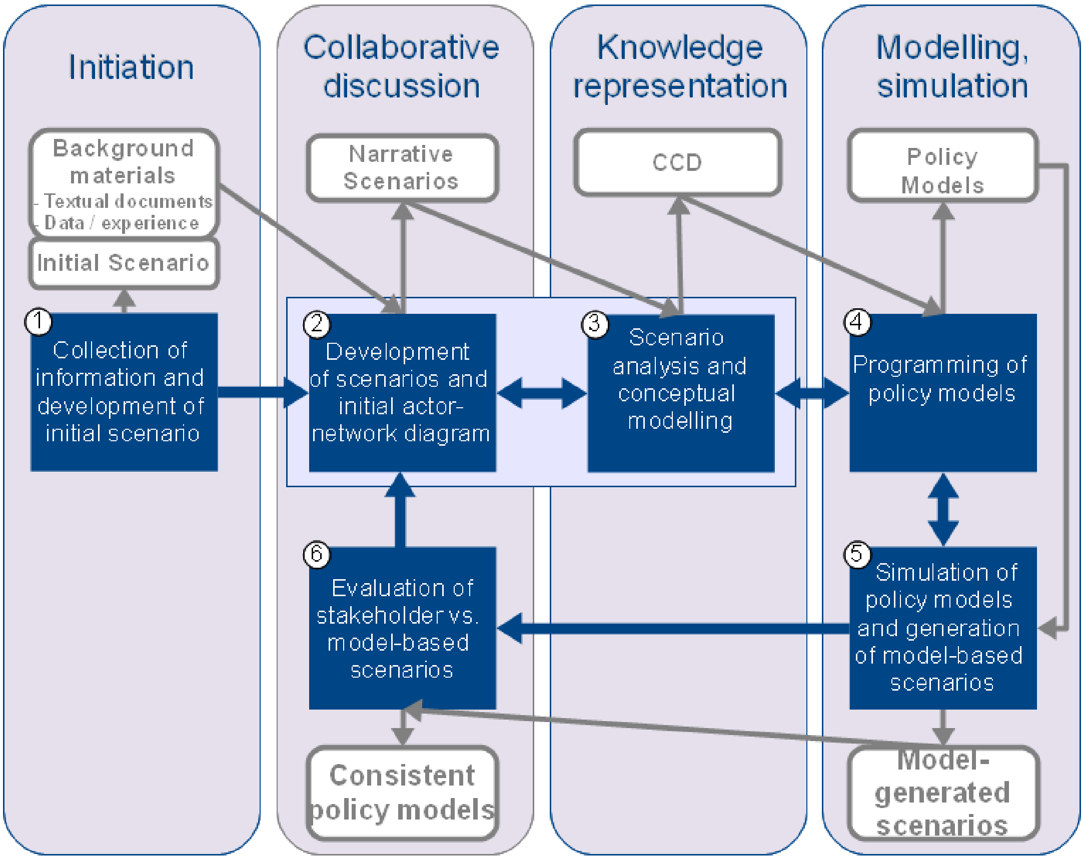

Figure 3: Artefacts along the OCOPOMO policy development process

During the stage "Initiation", existing background materials are analysed and new documents are generated from different sources (e.g. experience of policy owners). The initial scenario, which is developed during this stage, is further discussed in the "Collaborative discussion" stage. Stakeholders involved in collaboration develop further narrative scenarios. An initial actor network diagram is developed in the CCD and discussed in this stage in order to identify all relevant actors. In stage "Knowledge representation" the CCD for the policy case is/are developed (it might be that more than one CCD is developed if there are diverging opinions in the scenarios, which should be presented in different policy models). The policy model is created in the "Modelling & simulation" stage. Out from the analysis of simulation results, the policy modeller develops the model-based scenarios. These scenarios are again discussed among stakeholders. In the end, the consistent policy models emerge, which consist of narrative documents, CCDs, formal simulation models and simulation outputs (model-based scenarios in narrative text and graphics). Stakeholders and policy owners may navigate through these artefacts using the traces established via the text annotations and the links that are conveyed through the transformations.

In the subsequent sections, the concept of the actors in the policy development process and the conceptual modelling are described in more detail.

2.2 Actors (users) along the policy development process

Figure 4 visualises the main actor groups along the stages of the OCOPOMO policy development process. In OCOPOMO, the actor groups can be separated along external and internal actors to a policy development process as follows:

- *External actors are direct participants of the policy development process and users of the policy development support system,* who are intentionally involved in policy creation. They have their own preferences, ideas, or proposals of how the newly created policy should look like, and
- *Internal actors provide methodological or technical support* in the OCOPOMO collaborative policy development environment. The first are those actors called experts for policy modelling, which (i) provide support to policy owners and stakeholders in developing scenarios and collaborating in the discussions on policy aspects to be described or expressed already in stakeholder-generated scenarios; (ii) perform policy analysis and develop conceptual models; and (iii) develop formal policy models and run simulations. The latter refer to system programmers and administrators, who are responsible for technical maintenance of the system and who run the participation and simulation systems.

In subsequence, each actor type is explained in more detail:

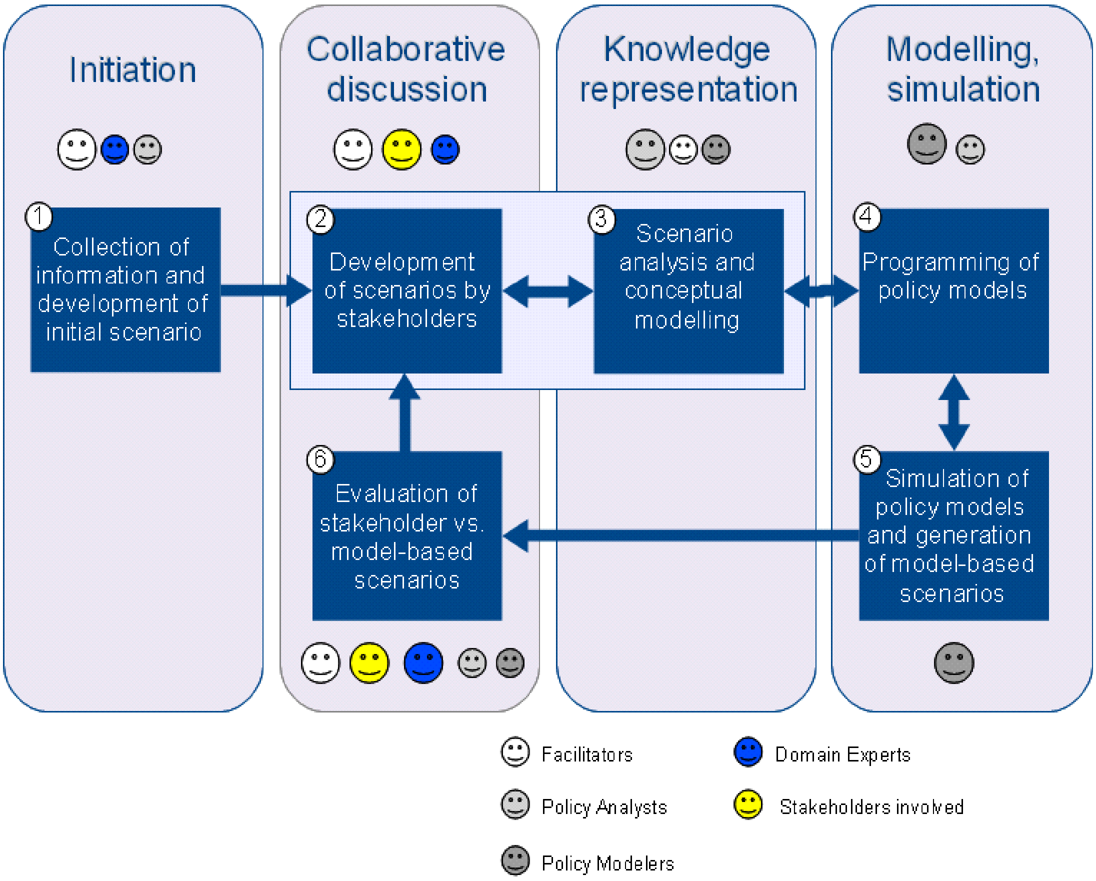

Figure 4: Internal and external actor groups along the OCOPOMO policy development process (excluding internal system administrators)

- *Domain experts* are policy owners and policy operators. They can be politicians and/or civil servants (synonymous for policy operators). They know well the policy domain and they have to develop a policy or are responsible for its generation and/or implementation.

  A politician in the project context is considered to be a decision-maker or a person that is responsible for the policy implementation under consideration. Politicians may initiate collaborative policy development (directly, or through civil servants) and may participate in the development of narrative scenarios or policy models. Politicians typically participate only in the initial phase of the collaborative policy development, and in later phases when some results of simulation are already available (especially in phase 6).

  Policy operators such as civil servants are assisting politicians and/or they provide relevant supporting materials for other participants of the policy development process. Together with politicians, they may provide an initial scenario description, which serves as a starting point for collaborative development of a new or improved policy. Policy operators may also participate in phase 2 (to lesser extent than stakeholders), phase 3 (if they have competencies in policy analysis and conceptual modelling), as well as in phase 6.

- *Stakeholders* of the respective policy domain are considered end users such as citizens, NGO's, and larger companies as well as SME's, which are willing and able to participate actively in the construction of narrative scenarios, in discussions and other information exchange of phase 2. They may have particular interests on the future policy that can be opposed to other stakeholders and domain experts. Stakeholders are also involved in phase 6, when it comes to the evaluation of the simulation outcomes and to potential revision of the scenarios developed so far in order to reach consistency of both types of scenarios (evidence-based scenarios of phase 2 and model-based scenarios of phase 5).

- *Experts for policy modelling* can be distinguished into facilitators, policy analysts and those programming and running the simulation models (policy modellers):
  - *Facilitators* are mediators who methodically control the collaboration in scenario development and evaluation. The Facilitator must speak the language of the stakeholders. He or she does not need to have skills for programming agent-based simulation models.
  - *Policy analysts* are experts that investigate scenarios and other (mostly textual) resources of phases 1 and 2. They analyse these documents and provide conceptual representations of extracted knowledge, therewith annotating the text documents. They are responsible for the qualitative analyses of narrative scenarios, which result in the construction of consistent conceptual descriptions (CCDs). The analysis includes an extraction of knowledge from discussions, comments, and simulation results in phase 6, and various materials that may support the development of scenarios (i.e. policy analysts are mostly engaged in phase 3 to develop the CCD and phase

6 to interact with the stakeholders in the model evaluation). The policy analyst does also not need to have skills for programming agent-based simulation models.

- *Policy modellers* are experts that construct formal policy models according to a given CCD. In other words, modellers derive the simulation models from an existing CCD and the underlying textual scenarios. They create the simulation environment, program the simulation models and run customisable simulations. Modellers are, subsequently, also responsible for providing simulation results to domain experts, stakeholders, facilitators and analysts for enhancing the respective scenarios accordingly. Hence, policy modellers are mostly involved in phases 4 and 5 of the OCOPOMO policy development process. The policy modeller needs to have skills in programming formal policy models (in the case of using DRAMS, skills on declarative rule-based agent modelling is a necessary[1]).

The user roles in the OCOPOMO policy development process differ from each other and, therefore, have different needs of (ICT based) support in the policy development process. For example, different knowledge of the existing policy, principles of policy formulation, of stakeholder engagement and facilitation, or technical background is needed. Accordingly, ICT support needs to support these different actors. Next section gives insight into the OCOPOMO toolset supporting the various actors / user roles in their activities.

2.3 OCOPOMO's ICT tools to support policy development

OCPOMO developed an integrated ICT toolbox to support the policy development process. Figure 5 visualises the tools along the phases of the policy development process. Subsequent explanations outline the usage of the tools (more details are provided in deliverables D 3.2 and D4.1):

Collaboration and Scenario Editing Tool (CSET) and Content Management Server (CMS): The CSET provides the main collaboration platform for the communication with the stakeholders. It allows collaborative creation of evidence-based scenarios and publishing and reviewing of the model-based scenarios. Besides of the main tool for writing and presenting of scenarios based on Wiki pages, CSET provides content management (shared Document Library), and collaboration features such as shared and personal dashboard with the overview of pending collaboration activities, online chat, discussion forum, and polling. Scenario editing is extended with the traceability navigation and browsing of the underlying CCD model. All content managed by the CSET is stored on the CMS, which provides common storage for any content generated by the various users involved in the OCOPOMO process. CMS is fully featured content repository, which supports versioning, access rights management, workflow etc. It provides standardised services to access stored content. It serves as the data interface between the CSET on one hand and CCD Tool and Simulation Analysis Tool on the other hand.

[1] If the CCD tool is used as conceptual model for a different simulation environment, according connectors and transformation support are required to be implemented. In this case, programming expertise for the chosen tool is required by policy modellers)

Consistent Conceptual Description (CCD) Tool: The CCD Tool supports Facilitators and Policy Modellers in developing a stakeholder-accessible conceptualisation of a policy model and thereby ensuring and documenting the consistency via the CCD. It supports the following main functions: a) Creating, editing and saving a consistent conceptual description (CCD) for a policy case, b) Linking the concepts of a CCD with background documents and scenarios. The CCD Tool as part of the OCOPOMO ICT toolbox has a modular design with the following components:

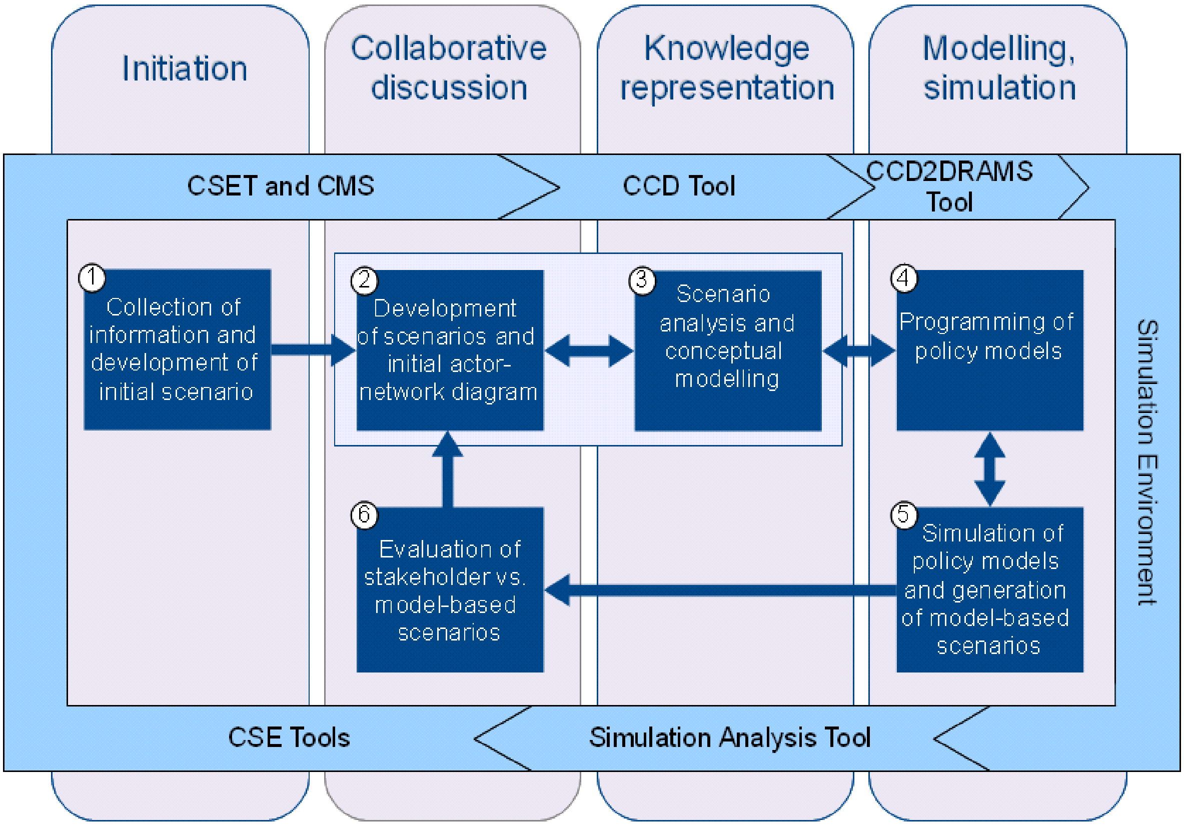

Figure 5: Tools of the integrated ICT toolbox along the policy development process

- Conceptual Description Tool: supports the creation, editing and storing of a CCD file (XML) which is representing the conceptual description of a policy case. The structure of the file is defined in the CCD meta-model.
- Annotation Tool: is used to annotate and link relevant text phrases from background documents and stakeholder scenarios with relevant elements in a CCD file.

Transformation Tool (CCD2DRAMS Tool): The CCD2DRAMS Tool supports (draft) source code generation from a CCD file for the Simulation Model in DRAMS. Thereby the CCD2DRAMS Tool supports traceability of simulation results by linking code fragments with entities of the underlying CCD file.

Simulation Environment (DRAMS): The Simulation Environment comprises parts of the OCOPOMO toolbox primarily dedicated to policy modellers. It consists of Integrated Development Environment (IDE) for editing, debugging and executing simulation models. DRAMS (Declarative Rule-based Agent Modelling System) is the rule engine software component for the OCOPOMO toolbox. It provides the necessary rule engine functionality to enable modellers in the OCOPOMO project to develop declarative agent-based simulation models. It is designed as a distributed, forward-chaining rule engine, i.e. it equips an arbitrary number of agent types with type-specific rule bases and instance-individual fact bases. DRAMS is not designed as a stand-alone simulation tool but as a framework for extending existing tools with rule engine capabilities. This implies that simulation models have to be designed as “hybrid” models, consisting of a simulation tool dependent part (for the pilot models were based on RepastJ) and the declarative rules describing the agent behaviour.

Simulation Analysis Tool: The Simulation Analysis Tool is a specialised tool that facilitates a production of model-based narrative scenarios based on obtained simulation logs. The tool provides a scenario development environment that enables policy modellers and policy analysts to browse the results of simulation runs, to use a built-in text editor to formulate scenario narratives from simulation log records, to annotate statements and text portions of the scenario by linking them to particular simulation log records, and to produce the output scenarios for publication on the Alfresco web space for further evaluation and discussion among involved stakeholders. The feature of annotation, which links scenario text fragments with respective parts of simulation logs, supports the traceability of facts and statements presented in the output narrative back to simulations, then to rules and facts in the related agent-based policy model, to entities of underlying CCD model, and finally to input evidence-based scenarios provided by stakeholders in the initial phases of the policy development process. To upload the produced output scenarios, enhanced by the traceability-enabling annotations, to the shared web space of Alfresco, the Simulation Analysis Tool invokes the Content Repository Client middleware, which handles the communication and data transfer between local Eclipse and remote Alfresco environments of the OCOPOMO ICT toolkit.

The detailed usage of these tools in the context of the policy development process is explained in the particular sections of this document. Deliverable 4.2 provides particular user manuals for each of these tools.

2.4 Embedding the OCOPOMO policy development process in a wider stakeholder engagement (or e-participation) endeavour and aspects of precaution for using the OCOPOMO solution

Successful collaborative policy development involving wider stakeholder groups require careful planning of initiatives in terms of integration into traditional policy development processes, engagement of political actors and citizens (or other stakeholders), selection of tools, maintenance and update of relevant information, and marketing the participatory policy development endeavour. This demands for the application of a holistic and systematic design approach, which addresses the various organisational and user-specific, social, technical, political, legal and procedural challenges in a comprehensive and structured way. Furthermore, the adaptation of given participation processes to enable online citizen participation is necessary. In some cases, the introduction of new participation facilities into traditional political processes may be required (Scherer and Wimmer, 2012). Proper integration of participation processes into the overall policy making life-cycle is needed to ensure the take-up of outcomes and that outcomes from one policy stage are handed over to the next stage (Macintosh and Coleman 2004, p. 18). Outcomes from online policy development have also to be thoroughly integrated with the procedures of offline participation processes and vice versa, as participation activities may occur online and offline (Macintosh and Coleman 2004, p. 10).

It is necessary to integrate the OCOPOMO policy development process in an overall initiative of online and offline engagement of stakeholders. From an investigation of procedural models for e-participation initiatives, an approach for implementation of e-participation initiatives is proposed in Scherer and Wimmer (2012) consisting of the following phases (see Figure 6):

- Phase I: initiation of the project including decision on the implementation of an e-participation initiative, and formulation of its objectives.
- Phase II: design of the initiative including design of participation processes and selection of technical tools.
- Phase III: implementation and preparation of the initiative including implementation of technical components, preparation of documents, and marketing strategy.
- Phase IV: participation phase including monitoring of political decision making.
- Phase V: evaluation of the initiative against its objectives and evaluation regarding expectations and impact achieved.

The phases are accompanied by continuous and dynamic requirements management.

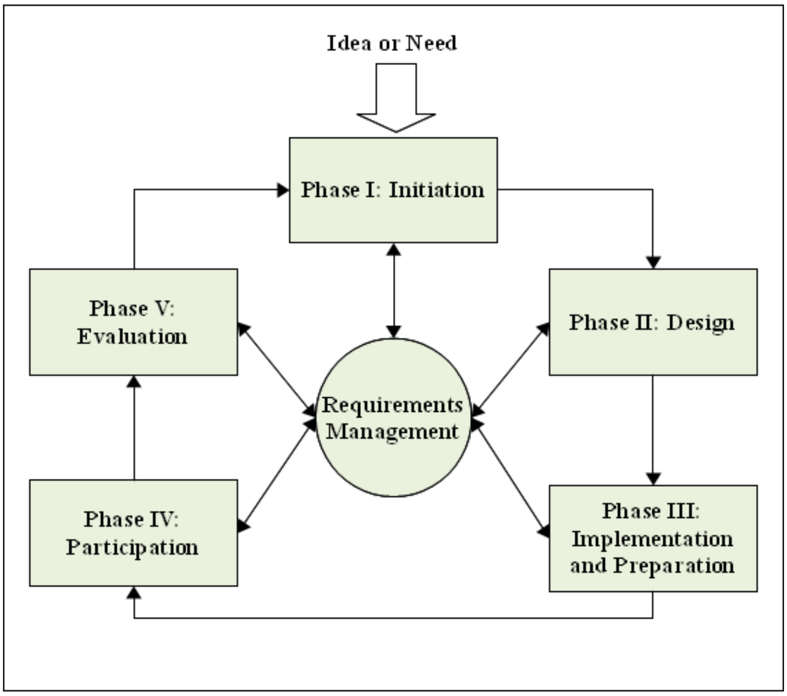

Figure 6: Phases of an e-participation initiative (Scherer and Wimmer 2012)

In this regards, the OCOPOMO policy development process as described in this deliverable is mainly part of *Phase IV Participation*. Initial phase 1 of the OCOPOMO process may be affiliated with *Phase III (the preparation part).* To set up and initiate such a wider stakeholder engagement in policy development, the proper tools and preparations have to be performed as shown in Figure 6. Preparation and post processing can be organised according to aforementioned phases:

- Phase I: Before an OCOPOMO initiative can be started, it is necessary to analyse how well the results of the OCOPOMO process i.e. the “consistent policy models” are recognised by decision makers and can influence decision making. For any stakeholder engagement initiative in collaborative policy development, it is necessary to ensure that participation of stakeholders will have an influence in the decision making process. Otherwise, stakeholders may be demotivated and frustrated for having spent their valuable time without having created any impact. The result of these reflections regarding expectation management particularly influences the structure of the initial scenario and the type of policy model (see Initiation phase of the policy development process in section 2.1). Further basic conditions as e.g. available resources, time frame are also decided roughly in this phase. The final decision about the realisation of the initiative triggers the next phase.
- Phase II: Determine how the OCOPOMO process can be fitted into the existing decision making process. Rough number and duration of iteration cycles in order to allow the discussion and refinement of results should be defined here, too. Decisions regarding if, how often and when face-to-face meetings are taking place etc. are also made.
  Results of phases I and II may result in an adaptation of the OCOPOMO policy development process and/or tools to the particular and general conditions of the initiative.
- Phase III: The OCOPOMO ICT toolbox is set up, an initial scenario is developed and background documents are collected or prepared and uploaded to the space. Stakeholders and domain experts are invited to the workspace.
- Phase IV: Participation activities take place along phases (1) 2 - 6 of the overall OCOPOMO policy development process (see section 2.1).
- Phase V: Evaluation of the stakeholder engagement initiative along the expectations and objectives of the initiative defined in Phase I.

Aspects of precaution in applying the OCOPOMO policy development process are:

Conceptual modelling is a collaborative process, where the policy analysts bring in their experience but also come back to the stakeholders and domain experts with questions or in order to discuss model elements with them. The same can be triggered by the policy modeller if he or she sees open questions. So, the development of a policy model is a back and forth process. On-going collaboration in Alfresco space can produce revisions in the model during one implementation cycle.

Organisation of workshops: A pure online collaboration is not recommended. Online collaboration cannot totally replace face-to-face meetings between stakeholders and domain experts. Face-to-face meetings in the form of workshops promise better involvement of participants into the process and its results.

The term “policy model” is a rather abstract one; an initial description of a policy case with examples of conceptual and simulation models shown during a workshop can help stakeholders and domain experts in order to get a better understanding about the meaning of a policy model.

In order to run an initiative based on the OCOPOMO policy development process, a number of resources are necessary. Estimates and details on how to calculate the number of resources and different types of expertise needed are outlined in the Exploitation Plan (D10.3).

Open participation does not necessarily mean that the OCOPOMO policy development process aims to have a mass participation. This is particularly often denied by decision makers, and not a necessarily the main aim of the OCOPOMO policy development process. In the end, it is the policy owners initiating such a stakeholder engagement process (cf. phase I of Figure 6) who decide the number of stakeholders to involve.

Technical preconditions for using the solution are not subject of this deliverable. They are further detailed in deliverable 3.1 and deliverable 4.1.

The OCOPOMO policy development process and the OCOPOMO ICT toolbox have a modular structure. That means that particular phases, aspects and tools can be used separately. This aspect is further detailed in section **Fehler! Verweisquelle konnte nicht gefunden werden.** of this deliverable.

3. Methodologies for the OCOPOMO Policy Development Process

In the subsequent sections, we outline the methodology and guidelines for the application of the method and ICT tools for the different activities along the OCOPOMO policy development process. Guidelines are provided for the following activities:

- scenario development (section 3.1),
- the development of conceptual models and text annotation (section 3.2),
- the transformation of conceptual models into rudimentary simulation models (section 3.3),
- the development of formal simulation models with DRAMS (section 3.4) and
- the simulation analysis using the Simulation Analysis Tool (section 3.5).

3.1 Scenario Development

3.1.1 Methodical steps

Scenario development is a key activity in the OCOPOMO policy development process. It consists of the following steps:

1. Preparation of initial scenarios and background documents (conferring to phase 1 in the OCOPOMO policy development process).

2. Selection of a target group of stakeholders and policy operators, and invitations to the online collaboration (conferring to preparations for phase 2 in the OCOPOMO policy development process)
3. Moderated development of evidence-based scenarios by the stakeholders and domain experts (initiated by a workshop as indicated on engagement of stakeholders – see section 2.2; conferring to phase 2 in the OCOPOMO policy development process)

Below, the course of each of the stages is presented in detail.

Ad1. Preparation of initial scenarios and background documents

For any given policy case, an initial scenario(s) along with background documents is developed by a policy owner and facilitators on the basis of available evidence. The evidence may come from interviews with domain experts, the policy owner, stakeholders, public documents, statistical reports, green and white papers or any other source that seems relevant. An initial scenario is the first version of a scenario, which is uploaded to the collaboration space for phase 2 of the OCOPOMO policy development process.

The purposes of an initial scenario are:

a) To organise the available information along the following aspects: who are the stakeholders and key actors of the policy case? what policy options does the policy owner see? What instruments might be used to come to a commonly agreed policy option? What conditions might be relevant to influence policy alternatives? and what effects might be expected from the implementation of a particular policy alternative.
b) To give stakeholders a clear and precise narrative to which they can respond by either agreeing or disagreeing with some part or all of it, or by offering alternative scenarios.

Tasks required to complete this step (all performed by policy owners or domain experts with support of facilitators):

1. Precise formulation of policy issue
2. Identification of main objectives and targeted measures
3. Preparation of initial scenario(s) and collection of background documents
4. Preparation of the OCOPOMO platform for consultations: uploading initial scenarios and background documents

Ad2. Selection of a target group of stakeholders and policy operators and invitations to the online collaboration.

Selection of a target stakeholder group is a critical step in the development of the evidence-based scenarios, as it greatly affects the final outcomes. Policy owners (strategic decision makers) or domain experts with the help of a facilitator determine the size and composition of the target group by choosing relevant experts and/or users (external stakeholders). Members of the target group of stakeholders receive invitations to participate in the online collaboration.

Tasks required to complete this step (performed by policy owners and domain experts with the help of facilitators):

1. Identification of groups of stakeholders (including domain experts) that might be influenced by policy issue
2. Decision about the scale of consultations (small, medium, large)
3. Selection of stakeholders and additional domain experts that will be invited for consultations and scenario generation
4. Preparation of introductory and informational materials for stakeholders and domain experts
5. Uploading introductory note for stakeholder engagement to the collaboration platform
6. Inviting stakeholders and domain experts to online collaboration

Ad3. Moderated development of evidence-based scenarios by the stakeholders and domain experts

The online collaboration process is initiated by a workshop, in which participation is not mandatory but highly recommended to the invited stakeholders and domain experts. Then the stakeholders and domain experts who responded positively to the received invitation read and comment on initial scenarios provided by the facilitator. After having done that, they can comment the initial scenario or generate new ones (alternative narratives or scenarios reflecting the perspectives of different individuals or groups of stakeholders).

Tasks required to complete this step:

1. Opening of the OCOPOMO platform for stakeholders giving them space for their scenarios and comments (by facilitators)
2. Commenting scenarios, uploading new background documents, developing and revising new or existing scenarios (performed by stakeholders, domain experts, policy owners and supported by facilitators)
3. Facilitating of the online collaboration (asking questions, commenting, initiating voting etc.) (facilitators)
4. Closing the phase of scenario development and informing participants (facilitator)

3.1.2 Preparing the Alfresco CSET application for scenario development

How to invite stakeholders for online collaboration?

The collaboration of stakeholders is organised online through the collaboration space – CSET, see 2.3. Overall, in the collaboration portal, users (i.e. policy owners, domain experts such as policy operators, stakeholders as well as policy analysts and policy modellers – see section 2.2 for a more detailed explanation of users) can publish evidence-based scenarios, upload background documents and review model-based scenarios and CCD models.

During the creation of the new collaboration site for the particular policy case, the facilitator can specify if the site will be public or private. For public site, everybody can view the published content and join the collaboration site and it is not required to login to view the content. It is also possible to specify, that site membership is moderated by facilitator, i.e. facilitator controls the membership by accepting or rejecting membership requests. For private site, only the invited members of the collaboration site can view or contribute to the content.

The access rights can be modified later for an existing site. For example, it is possible that everybody can view the content without the login using the guest account, but only the invited and accepted members of the collaboration site can modify content or publish the new one.

The common process for the users to be invited into the collaboration site requests the following steps:

1 Facilitator can create new user accounts for the CSET portal with the username and password credentials.
2 Facilitator invites users (i.e. the stakeholders and domain experts identified and selected by policy owners) with the given username to the selected collaboration site. User can be notified by email or on his/her personal dashboard.
3 User has to accept the invitation for the site collaboration.
4 Facilitator is notified that user accepted collaboration and became a new site member.

Invited user (i.e. stakeholder, policy owner, domain expert, facilitator, policy analyst, policy modeller) a can have assigned the following access right roles:

- *Site Manager* - can invite new members, can modify or view the content
- *Collaborator* - can create new content and checkout new version for the content published by other users
- *Contributor* - can create new content or modify its own content
- *Consumer* - can view content

Note: These roles are specific to the Alfresco online platform and must not be mixed up with the actors / users along the OCOPOMO policy development process as outlined in section 2.2. Each user in the CSET must be affiliated with a particular access right role to be enabled with certain access credentials as indicated along each role.

How to organise background materials and initial scenarios on the collaboration site?

When a user enters the collaboration site, the first displayed page is the site dashboard. The site dashboard consists of site dashlets, which organise and display information that is relevant to the site. When the site dashboard is customised, the desired dashlets can be specified; each dashlet selected displays as a pane on the dashboard. The following dashlets are available:

- *Getting Started* - The Getting Started dashlet displays helpful information for getting started in the site and provides links to perform common tasks.

- *Site Colleagues* - The Site Colleagues dashlet lists the site members (to a maximum of 100 members) and their assigned roles.
- *Site Calendar* - The Site Calendar dashlet contains a rolled-up view of events for this site.
- *Site Activities* - The Site Activities dashlet tracks the most recent activities that have been performed in this site.
- *Recently Modified Documents* - The Recently Modified Documents dashlet displays documents in this site's Document Library that have been added or edited in the past days.
- *Site Wiki* - The Wiki dashlet displays the specified wiki page. The *Configure* link on the dashlet enables the facilitator to select the desired page. Click the name of the wiki page in the dashlet header to display that page in the Wiki page component.
- *Site Links* - The Site Links dashlet displays the web links compiled by site members that are relevant to the current site. On this dashlet users can open a link, view the link details, or create a new link (click *Create Link*).
- *Site Chat* - The Site Chat dashlet allows the user to online chat with the other site members.
- *Image Preview* - The Image Preview dashlet displays a thumbnail of all images contained in the site's Document Library.
- *RSS Feed* - The RSS Feed dashlet, by default, is configured to display the Alfresco website feed.
- *Web View* - The Web View dashlet can be configured to display any website.

For each site, the dashboard appearance can be modified, the site dashlets can be selected to appear on the dashboard and the display order of the selected dashlets can be configured. For the configuration of the site dashboard, the following suggestions are provided:

- Depending on the number of configured dashlets, it is preferred to use two-column layout, with the main content area on the left hand side and banner on the right hand side. Use three-column layout only when you have many dashlets and the user has to scroll the page to see all.
- Place introductory text in the main content area. Preferred way for publishing of the web content on the site dashboard is to use the Site Wiki dashlet. Alternatively you can use Web View dashlet to display content published on the external site.
- Insert direct links to the background materials stored in the Document Library or initial scenarios Wiki pages in the introductory text.
- Configure only the important dashlets used by the majority of the site members (particular user can always customise his or her personal dashboard with additional dashlets). Many dashlets provide similar information or functionalities so you can configure just one of them for your specific needs. For example, Site Chat dashlet displays the list of site member, so it is not required to add also Site Colleagues dashlet into dashboard or Site Activities dashlet can be preferred over the Recently Modified Documents dashlet since it displays all activities on the site including the modified documents.

Besides the dashboard configuration, facilitators can customise site settings. Alfresco Share has the concept of page components, which can be added to a site as customised functionality. All CSET Tools are page components, which have rich functionality and are URL addressable. The following page components are available:

- *Wiki* - The Wiki enables users to create web pages for a collaborative website. Anyone who accesses it can contribute or modify content using a simplified mark-up language. In the OCOPOMO process, Wiki is the primary tool for collaborative editing of evidence-based scenarios and publishing of the model-based scenarios. Also the traceability tools are integrated in the Wiki page component.
- *Blog* - The Blog page component enables users to add comments, descriptions of events, and other material related to the policy case site, such as graphics or video.
- *Document Library* - The Document Library page component enables users to store and collaboratively manage any content related to a site, such as documents, media files, or graphics. In the OCOPOMO process, Document Library page component is primary used to organise background documents.
- *Calendar* - The Calendar page component enables users to schedule and track events for all sites you own or of which you are a member.
- *Links* - The Links page component enables users to maintain a list of web links related to the site.
- *Discussions* - The Discussions page component is used to post user-generated content related to a site. These often take the form of questions or comments with threaded discussions.
- *Pollings* - The Pollings page component allows users to create and manage polling and questionnaires relevant to the site.
- *Data Lists* - The Data Lists page component allows users to create and manage lists relevant to the site.

For the organisation of the initial scenarios and background documents, the following suggested conventions are given:

- Configure only the important page component required for collaboration needs. Wiki page component and Document Library page component should be always included, since they provide core functionalities for scenario editing and reviewing.
- Publish initial scenarios as Wiki pages. Consider to write a narrative introduction to the scenarios and include it on the Wiki page presented in the main content area of the site dashboard. Introduction should contain direct links to the initial scenarios.
- Publish any background materials in the Document Library. Place relevant documents, media files or graphics in dedicated folder and provides the link to the folder from the introduction Wiki page. Create any sub-folder structure to organise the content in the library.
- Use tags to categorise similar or related content within a site. A tag can be associated to any managed content including Wiki pages, documents, discussion forums, pollings, etc.

How to organise user generated content on the collaboration site?

The primary tool used by the stakeholders and domain experts to collaboratively create evidence-based scenarios is the Wiki page component. Each user can create its own Wiki page or can comment the scenarios published by others. The content of the Wiki page is formatted in the simplified mark-up language, which supports linking and embedding of the tables, charts, figures and previews of the background documents stored in the Document Library. To create or edit the Wiki content, users are using WISIWIG editor. Since each user can create his or her own Wiki page, facilitator has to keep this content organised and accessible to other users in a simple way.

Besides the narrative evidence-based scenarios published as the Wiki pages, users can upload any documents, media files or graphics as the background materials shared with others in the Document Library. It is possible to directly link or embed previews of the uploaded background materials in the text of narrative scenarios on Wiki pages.

Users can post comments to any Wiki page or folder or document in the Document Library. Alternatively, it is possible to create a multi-threaded discussion forum dedicated to the specific topic or specific scenario, which can be directly linked from the text of a scenario or from the comment associated with the folder or document.

To summarise the organisation of the user generated content on the collaboration site, the following suggestions should be taken into consideration:

- If the collaboration site is private or the users have to be invited to join collaboration on the site, prepare dedicated scenario Wiki page for the particular user. Provide a list of scenarios in the introduction Wiki page published in the main content area of the site dashboard.
- If the collaboration site is open, categorise all created scenario Wiki pages according to the discussed topic or stakeholder's point of view (for example scenarios of energy producers, NGOs, citizens, etc. in the Kosice policy case) and create a navigation Wiki page with the links to the classified scenarios. Alternatively, tags can be used to classify scenario Wiki pages.
- Create a dedicated folder for background materials uploaded by users. For sites with the moderated membership, create dedicated sub-folders for each user and provide a link to this folder from the user's scenario Wiki page. Use tags to additionally organise uploaded materials.

3.2 CCD Annotation and Modelling

3.2.1 Concept and methodological grounds

The CCD supports the transformation of narrative texts into formal policy models by conceptualising the policy domain under consideration. To ensure well-defined and structured CCD models of a policy domain, the CCD meta-model serves as a blueprint or vocabulary for describing a policy case.

The initial development of the CCD meta-model was based on literature studies of simulation and policy modelling and input from policy modellers. The concept of the meta-model in OCOPOMO is based on the definition of an ontology (Gruber, 2009), the rationale of meta-models according to (Winter, 2000) and elements of simulation models. The core elements for the CCD meta-model have further been refined based on insights from the development of the three policy cases in OCOPOMO as well as discussions with policy analysts and policy modellers in the project. As a result, the following elements have been embedded into the CCD meta-model – shown also in Figure 7 (Scherer et al 2013):

- Ontology: The ontology describes 'the single parts' of the policy domain. The following elements are defined in the CCD meta-model: Concepts with `Actors`, `Objects` (representing particular elements or sets), `Attributes` (representing properties of model elements) and `Relations` (representing relationships between model elements) relevant for the policy case. These elements allow the definition of Actor Network Diagrams.

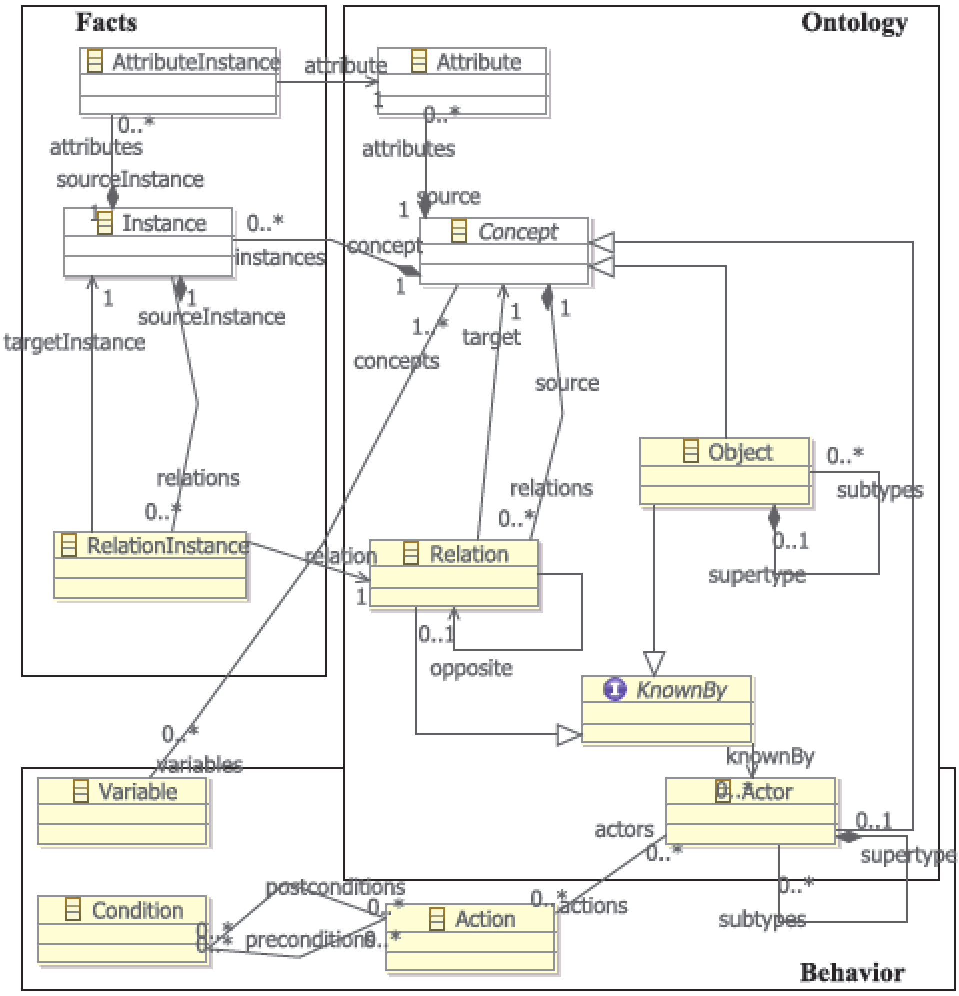

Figure 7: CCD Meta-model

- Behaviour: Dynamic aspects and behaviour are encoded as `Actions` and their interrelations. Each concrete `Action` has an `Actor`, pre-conditions and post-conditions (`Conditions`). Thus it is possible to model behaviour and interaction in one conceptual model and link them with static elements.
- Facts: The 'number and nature of inputs' (i.e. the facts) are encoded with `Instances` of elements `Actor` or `Object`. Based on `Relations` and `Attributes` defined for the underlying Concepts, concrete `RelationInstances` between `Instances` and `AttributeInstances` for `Instances` can be described.

Table 1 lists and explains the types of elements defined in the CCD meta-model. The first column shows the name of the entity. The second column gives a short description of each of the entities / elements.

Table 1: Overview of elements defined in the CCD Meta-model

CCD entity / element	Short Description
`Actor`	Represents classes of stakeholders in the policy model. These elements are transformed to agents later on in the CCD2DRAMS transformation process.
`Object`	Represents classes of things, which are relevant in the policy model such as groupings, institutions or artefacts (e.g. “House”, “Material”) as well as virtual helper objects (e.g. “Heat Delivery”)
`Relation`	A conceptual relationship between two `actors`, two `objects` or an `actor` and an `object`.
`Attribute`	*C*haracteristic or property of concepts `Actor` or `Object`. `Attribute` have a primitive data value (Integer, Double, Enum, String)
`Enum`	An enumeration as a primitive value type
`Instance`	Concrete occurrences / instantiations of an `Actor` or an `Object`. For example a concrete heat producer “TEKO” in the Kosice policy case.
`Attribute Instance`	Concrete occurrence of an `Attribute` of an `Instance`.
`Relation Instance`	Relationship between two `Instances` based on a defined conceptual `Relation` between two concepts (of type `Actor` and/or `Object`).
`Action`	Representation of tasks or actions in the policy case. `Actions` are usually executed by `actors` and pursue an objective. The OCOPOMO concept also enables modelling of natural events and consequences, e.g. “*it rains*” and as consequence “*it is wet*”; with such actions, no `actor` is affiliated.
`Condition`	Representation of a pre-condition or post-condition (meant as consequence) of an action. A `Condition` as post-condition of one `action` can be used as pre-condition for another `action`. This way, behaviour of and interaction between `actors` are modelled. `Condition` describes only one particular characteristic on an element.
`Variable`	Representation of `objects` or `actors` used as flexible declaration in one or more *`Conditions`*.

`Literal`	An enumeration value.
`File Annotation`	The CCD tool makes it possible to annotate text by selecting text in the "Annotation Editor" and dragging it on the appropriate node in the "Annotation View". By dropping it in a node, a `File Annotation` (either txt or pdf) is added to the selected node in the CCD.
`Expert Annotation`	Representation of an opinion of an expert, which is not described in documents or an argument of stakeholders. This concept is used to identify those elements in the CCD model that have been added by the policy analyst or policy modeller to enrich and complete the CCD model.

The CCD Tool supports policy analysts and policy modellers in developing a formalisation of a policy domain that is accessible to stakeholders and domain experts -- thereby ensuring and documenting its consistency. The CCD Tool supports the following main use cases:

- Creating, editing and saving a consistent conceptual description (CCD) for a policy case (see phase 3 of OCOPOMO policy development process). A CCD is an XML file. The structure of the file is defined in the CCD meta-model (see above).
- Linking the concepts of a CCD with documents (supporting again phase 3). The CCD Annotation Tool is a component of the CCD Tool. It is used to annotate and link background information (e.g. statistics, stakeholder scenarios, domain facts, background descriptions, etc.) with relevant `actors`, `objects` and `actions` documented in a CCD file. The annotations are stored directly in the CCD file. Two versions of the Annotation Tool are available - one for annotating local plain text files, and one for annotating pdf and HTML/wiki files, which are available at the OCOPOMO Collaboration Platform.

Three types of diagrams exist in the CCD tool in order to visualise a CCD model of a policy domain:

- An Actor Network Diagram visualises `actors`, `objects`, `attributes`, `relations` and their relationships.
- An Actions Diagram visualises `actions` and `conditions` and their connections in the form of pre- and post-conditions.
- An Instances Diagram visualises `instances`, `attribute instances`, `relation instances` and their relationships.

To create a CCD, the following steps are suggested, which follow the first "modelling process" steps for developing social simulation models proposed in (Doran and Gilbert, 1994, p. 10):

1 Considering what is known about the policy: Analysis of data, available information and scenarios.
2 Clarifying modelling objectives: What should be modelled, and are the objectives in line with available information? Create a CCD file, describe the modelling objectives

in the description and annotate the CCD with relevant documents in order to secure that information is sufficient to meet the objectives.

3 Choosing the aspects of the policy to be modelled:
   a Analysis of data and background information for relevant concepts (actors and objects) and their relationships. Identification and extraction of relevant phrases in relevant documents. Creation of respective elements in the CCD and linking them with text phrases.
   b Analysis of data and background information for facts. Creation of respective instance elements in the CCD and linking them with text phrases.
   c Analysis of data and background information for behaviour of actors. Creation of actions and conditions in the CCD and linking them with text phrases. Formalisation of actions and conditions by linking them with elements already defined in the CCD.

4 Running the CCD2DRAMS tool to generate the initial simulation project (see section 3.3)

Figure 8 highlights the steps 3 and 4 to extract conceptual models from narrative text using the CCD Tool and then transforming the CCD model into a formal simulation model that can be further programmed in DRAMS (see section 3.4). In the figure, the transformations are highlighted with the rounded arrows between phases 1/2 and 3 ((a) Transformation of text into CCD) and phases 3 and 4 ((b) Transformation of conceptual models into programming code) of the OCOPOMO policy modelling process.

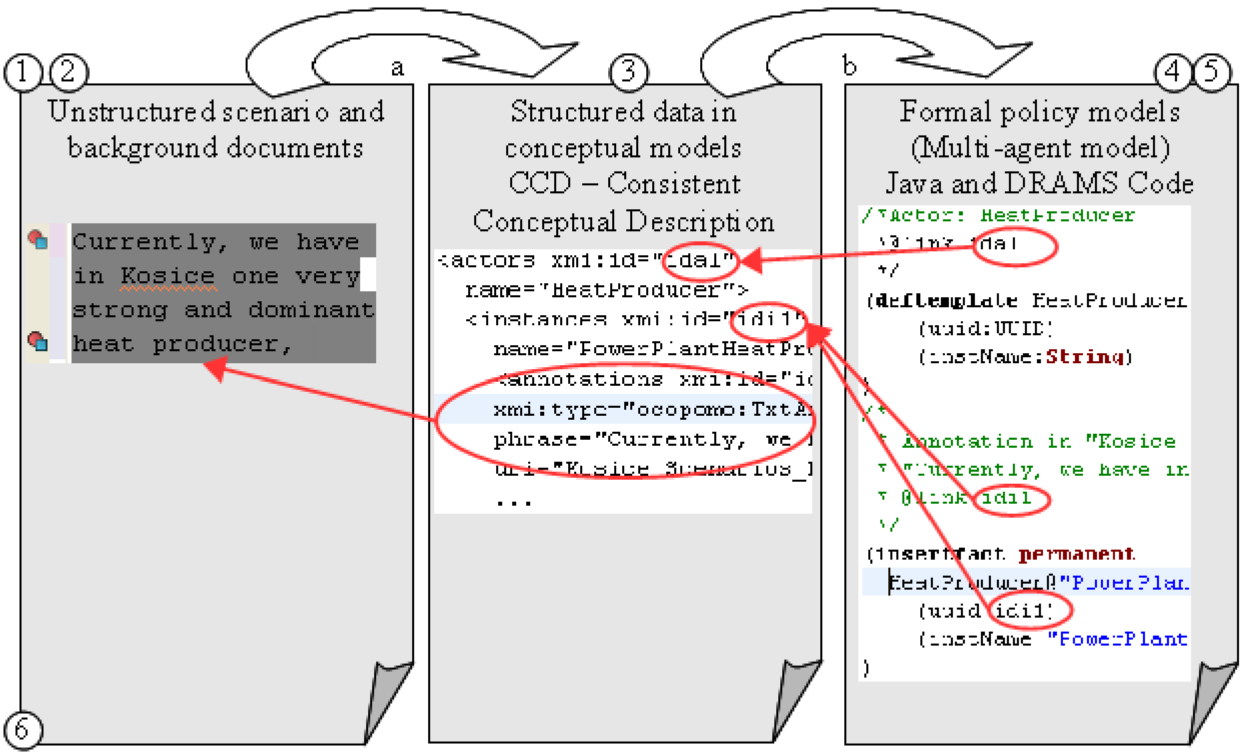

Figure 8: Transforming narrative text into conceptual model and further into simulation models thereby enabling the tracing back of parts of the simulation model to the origin phrase in a document.

Gilbert and Troitzsch (2005, p. 207) argue that iteration is a common process, when developing simulation models. Accordingly, a CCD is revised in several iterative steps until a version is finalised, which can be transformed into a formal policy model. Discussion of a CCD in phase 3 as well as evaluation in phase 6 of the OCOPOMO policy development process can be performed using the online CCD Explorer (integrated e.g. in the CSET), which allows the user (wider stakeholders, policy owners and domain experts) to comment CCD elements.

3.2.2 Exemplifying the development of a CCD and using the CCD Application through a case example of Campania

The subsequent explanation provides insights into how the conceptual model of the Campania policy case has been developed using the OCOPOMO CCD Tool. As already described in D 6.1, the Campania pilot policy case is concerned with the evaluation of public policy implementation to support research-oriented (public and/or private) spin-offs from Universities to be successful towards sustainable market penetration.

The initial scenario (see D 6.1) is conceptually divided into two (strongly interrelated) pillars: one focused on strategies to maximise participation in competitive bids provided by public R&D funding programmes and the other focused on strategies to provide a set of services and/or products for customers on the market. These two pillars, together with a few side-rules, were modelled using the CCD tool.

Hence, after an analysis of the background documents and the evidence-based scenarios generated by stakeholders, the first modelling step was the identification of the main `Actors` to be modelled in the CCD: Service Providers (SP), Promoting Agencies (PA), private Enterprises and Citizen. Most relevant `Objects` were then identified and added to the CCD, e.g. “call for proposal”, “consortium” or “service”, and annotated to the text background documents where these objects were described. The modellers should also specify `Attributes` and `Relations` regarding `Actors` and `Objects`: these properties will be useful for both the static description and the dynamic aspect of the model.

The dynamic behaviour of the `Actors` (towards `Objects` and/or other `Actors`) is controlled by `Actions`: for example in the Campania pilot, a PA will issue new calls for proposal, thus repeatedly instantiating the CCD `Object` “call for proposal”. This is in turn a pre-condition for another `Action`, performed by the SPs, which consists in setting up consortia so to apply for the call. In turn again, consortia will submit a proposal to the PA, which will decide, which proposal should be awarded according to defined criteria.

It is important to note that this chain of `Actions` comes directly from the evidence-based scenarios: the policy modeller should infer the main aspects and convert them from text to relevant CCD elements, then use `Conditions` and `Variables` to create the links within the `Actions`. A similar procedure was applied for the market-part of the model, with both the Enterprises, which are interested in buying services from the SPs.

To explain the process of constructing a CCD model, the following list of how-tos is provided:

How to use the known-by feature?

The so-called “known-by” feature can be used in a CCD to represent that a particular `Object` is only known by a number of specific `Actors`. An example from the Kosice pilot for the use of this "known-by" feature is the `Object` "Agreement", which might be only known to the two `Actors`, who made the agreement. Another example is the `Object` "HouseholdHeatingCostsInformation", which contains information about heating costs of households. As each household only knows its own energy costs, an `Instance` of "HouseholdHeatingCostsInformation" is only known to the particular `Instance` of "Household". If an `Object` has no "known-by" feature, its `instances` are globally known (i.e. by all `Actors`).

In DRAMS (see section 3.4), a global fact base exists, where all general facts are available. In addition, each agent (which confers to `Actor` in the CCD model) has a fact base, where only facts for particular agents are stored. Thus the "known-by" feature is used for transforming model aspects into respective fact bases of agents. Therefore, these features need to be modelled properly in the CCD.

How to deal with discrepancies between scenarios?

Two steps are suggested to deal with discrepancies between scenarios. First step is that the policy analyst should come back to stakeholders and domain experts in order to clarify if discrepancies are reasonable or if there are misunderstandings. If the discrepancies between the scenarios are reasonable, (second step) the policy analyst can decide to define a baseline CCD, which is then branched into different versions based on the different scenarios.

How to represent endorsements?

Endorsements are used in order to represent a positive or negative feedback of a relationship between two agents. In order to represent an endorsement in a CCD, an `Object` *“endorsed”* with three `attributes`: *“endorsement”*, *“endorsee”* and *“scheme”* should be created. An endorsement is thereby understood as a fact, which represents a positive or negative feedback of a relationship between two agents.

The first attribute of the CCD `Object` “endorsement” is a mnemonic token that should be meaningful to stakeholders. The value of the second attribute, “endorsee”, is the agent to which the endorsement is applied. IF John is an endorsee and “reliable” is the endorsement, then there will be a rule determining when John is deemed to be reliable. The “scheme” attribute refers to a means comparing different collections of endorsements. Some endorsements will increase the desirability of an agent or object and some will diminish the desirability. Some endorsements have a greater impact on overall desirability than other endorsements. There are several well established endorsement schemes. The range of possible endorsements is determined by the definition of the Endorsement `Enum` in the CCD, which is translated to a DRAMS type by the CCD2DRAMS tool as described in the next section. Every time the modeller instantiates the “endorsed” `Object`, the result is one endorsement to an `actor` in a particular scheme. E.g. the service provider Benecon in the

Campania pilot could receive an endorsement "goodManagementCapacity" for the Proposal endorsement scheme.

There is normally more than one endorsement scheme in a model, each of which relates to a different context. For example, in the Campania pilot, three endorsement schemes exist:

- Consortium endorsement scheme, related to the creation of consortia among service providers in order to apply to a call for proposal
- Proposal endorsement scheme, related to the evaluation by the funder promoting agency of submitted proposals
- Skills endorsement scheme, related to the capability of a service provider to sell services on the market

To get a clear understanding of CCD modelling, CCD model examples exist from the three OCOPOMO policy cases. The CCD models are available as open source models from work package 6 and have been described in D 6.1.

3.3 CCD2DRAMS Transformation

3.3.1 Methodological grounds

Ensuring traceability through the step-wise transformation of narrative scenario texts via the CCD into formal statements in Java and DRAMS code is an important contribution to achieve open government and to implement the Good Governance principles (i.e., improving in particular openness, transparency, participation and coherence in policy modelling).

The CCD2DRAMS component supports the policy modeller in generating both Java and DRAMS source code from a CCD file. The CCD2DRAMS Tool maintains traceability by linking the generated code fragments with concepts of the underlying CCD file using the UUID.

In order to transform a CCD (which is conform to the CCD meta-model) into an executable simulation model (see transformation (b) in Figure 8), a transformation definition is necessary. Such a transformation definition has been developed in the OCOPOMO project for the DRAMS meta-model[2]. The transformation definition was described using the "MOF 2.0 Query/View/Transformation Specification" (QVT)[3]. Since 2011, version 1.1 is available[4].

From a conceptual viewpoint, the transformation of conceptual models into simulation models is done by building analogies between elements of the CCD meta-model and elements of the simulation model (DRAMS). The CCD meta-model already differentiates between the concepts `Actor` and `Object` by the fact that an `Actor` carries out an `Action` and an `Object` does not. Thus, an `Actor` of the CCD-meta-model corresponds to an

[2] An overview of DRAMS classes is provided in Lotzmann and Meyer (2011).
[3] QVT is a standard published in 2008 by the OMG (Object Management Group) http://www.omg.org/spec/MOFM2T/1.0/, last accessed 4th February, 2013
[4] See http://www.omg.org/spec/QVT/1.1/, last accessed 4th February, 2013.

"Agent Class" in the DRAMS meta-model. Objects and Relations are transformed into fact templates; Instances are transformed into facts and Actions with their pre- and post-conditions that are transformed into rules. It is to be stressed that the two meta-models are not equally powerful. DRAMS as a formal agent-based simulation language is more expressive than the CCD meta-model. Dynamic aspects cannot completely be transferred, because in the CCD modelling the focus lays on the conceptual description of the behaviour of actors and actions are not described in the sense of "if-then-else" rules. Therefore, the CCD2DRAMS transformation can only generate rule stubs in DRAMS for Actions in CCDs.

As each CCD element has a unique identifier, the semi-automatic transformation of CCD elements into a simulation model makes it possible that each part of the simulation model can be traced back to the origin CCD element. Figure 8 exemplifies how parts of the simulation model can be traced back to the origin source phrase (see arrows). The declarative code in DRAMS --- here the definition of an instance "PowerPlantHeatProducer" of the Agent "HeatProducer" --- holds a link to the originating CCD Actor with UUID "ida1" and CCD Instance with the UUID "idi1". This CCD Instance has a text annotation which links the CCD entity to a scenario text.

In order to transform a CCD into DRAMS code, the following steps should be processed:

1 Make sure that the CCD, which is to be transformed, has a name. Otherwise the transformation will not work properly.
2 Make sure that the CCD lays in a java project. Otherwise difficulties may arise when starting a simulation afterwards.
3 Run the CCD2DRAMS tool to generate the initial simulation project.
  Check if there is a java file and a drams file for each agent type. In addition there should be a general.drams file.
4 Adapt the CCD2DRAMS generated code (see indications on How to adapt CCD2DRAMS generated code? in the next sub-section and the section 3.4 about DRAMS).

3.3.2 Application of a CCD2DRAMS transformation

Following, an example of code transferred from the CCD of the Campania pilot case is presented:

From the action "Program the evaluation time" of actor "PromotingAgency" in the CCD, the following rule stub was automatically created in the DRAMS file of the agent "PromotingAgency".

```
/*Action: Program the evaluation time
 *@link _-KqDwBO8EeKyDM5BPNPMhQ

 */
(defrule PromotingAgency::"Program the evaluation time"
//(global::callForProposal(programmeName ?programmeName))
```

```
//TODO
=>
//TODO
)
```

This CCD2DRAMS-generated `rule stub` was copied into the Protected Area and expanded by the policy modeller as follows:

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

/*Action: Program the evaluation time
 *@link _-KqDwBO8EeKyDM5BPNPMhQ

 */
(defrule PromotingAgency::"Program the evaluation time of new proposal"
	(global::callForProposal (programmeID ?programmeID) (ID ?callID) (closed
	false) (deadline ?deadline) (funder ?funder) (startDate ?callTick) (funds
	?funds))

	($SELF$ (name ?funder))
=>
	(assert deferredBy ?deadline (evaluationTime (funder ?funder) (programmeID
	?programmeID) (ID ?callID) (funds ?funds)))
)
```

It can be seen that the UUID, the rule's structure and its name were automatically generated. Yet additional clauses have to be written on both sides of the rule (details and guidelines of such programming work are provided in section 3.4).

The subsequent explanations provide how-to guidelines for using the CCD2DRAMS tool:

How to rerun the CCD2DRAMS tool if transferred code has been changed already?

It is possible to start another model transformation with the CCD2DRAMS tool, although the code generated by a previous transformation has been modified and/or extended. As CCD2DRAMS is not able to recognise modified code elements by the content (only by special tags or within protected sections), the modeller has to take care to mark the changed Java code appropriately and to move elaborated or newly added DRAMS code into the special user code section. I.e., the following steps should be considered by the policy modeller:

1. If a generated method in any Java class is modified, then the comment tag "@generated" should be changed into "@generated not". Newly added methods in Java classes and methods without the "@generated" tag will not be touched by the code generator.
2. All modified fact templates or instantiations should be copied back in the corresponding CCD Actor or Object, when possible. This way a subsequent transformation with the CCD2DRAMS tool will take into account such improvements.

3. All modified rules in DRAMS files as well as other elements, which could not be copied into the CCD, should be moved into the user code section. If an Actor/Object is modified, the modeller might also copy these modifications into the CCD before rerunning the CCD2DRAMS. This solution would be more elegant (and saves time in the long run), but is not always feasible. The modeller should check if this solution is possible. If it is not possible then we suggest to proceed with the protected area. The section starts after the comment “Start of user code Declarative-UserCode (this is a protected area)” and ends with the comment “End of user code”. After re-performing the model transformation, there might be duplicates of declarative definitions, the modified one within the user code sections and an empty one without this section. This inconsistency is automatically recognised by the DRAMS parser, and the “correct” definition is used.

How to adapt CCD2DRAMS-generated code?

Some adaptations of the CCD2DRAMS-generated code are necessary, especially regarding the transformation of CCD Actions into DRAMS rules. In fact, several features of DRAMS do not have a corresponding concept in the CCD. To fully exploit the potentialities of DRAMS, the modeller should expand each generated rule by means of the available DRAMS clauses. The procedural steps suggested are:

1. Run the CCD2DRAMS tool to generate the initial simulation project
2. Copy the rules which need to be expanded into the User-Code Protected Area, including the traceability link (UUID). Deleting the original rule is optional, but keep in mind that it will be generated again at the next CCD2DRAMS run
3. Modify the rules in the Protected Area according to the model requirements

3.4 Simulation Modelling with DRAMS

3.4.1 Methodological grounds

The aim of simulation modelling in the OCOPOMO policy development process is to build policy models and run simulations on them. In the context of the OCOPOMO process, a formal policy model is understood as a rule-based, declarative model - consisting of a number Java classes (for the Repast-based static model part) and declarative code for the rule engine DRAMS (specifying the model behaviour). The model is designed and implemented to explore the individual actions and combinations of actions that are believed to be available to governments for the purpose of achieving specific and well-formulated objectives. It enforces the specificity and clear formulation of available actions and also clear, precise and well formulated statements of the conditions, in which the actions might be taken and the consequences of those actions in the specified conditions. The model produces output in the form of print statements that describe sequences of events including decisions and the outcomes that emerge during simulation experiments with the model.

Under a simulation run initialisation, the number of time steps and the number of agents of different kinds are defined. In some cases, it will be necessary to keep the proportion of the numbers of different agent types constant in order not to bias the model.

The entire process of designing and running a simulation model, using RepastJ 3.1 as simulation platform and DRAMS as a declarative extension, can be structured into a number of steps, which depend on whether the CCD2DRAMS tool is used for code generation, or the model is built from scratch as outlined in Table 2 (see also Lotzmann and Meyer 2011).

Table 2: Steps to transform the conceptual models into DRAMS – both, using CCD2DRAMS model transformation or building the model from scratch.

<table>
<tr><th>Step</th><th>Model skeleton generated with CCD2DRAMS</th><th>Model set up from scratch</th></tr>
<tr><td>1.</td><td>The generated Repast model and .drams files (see section 3.3) should be inspected for consistency and completeness.</td><td>A Repast model class must be defined, which creates all agent instances (for which the desired agent types have to be defined, see step 2), initialises the global fact base and handles the Repast time steps by triggering the rule scheduler. An abstract model super class providing the DRAMS related code is available, thus only model related aspects have to be added.</td></tr>
<tr><td>2.</td><td>Optionally, Java classes for model or agents can be adapted or extended.</td><td>Classes for all designated agent types must be created and all necessary functionality should be implemented. Similar to the model super class, an appropriate agent super class is delivered with DRAMS.</td></tr>
<tr><td>3.</td><td>The rule definition stubs need to be elaborated by specifying the Left Hand Side (LHS) clauses (for the condition part) and the Right Hand Side (RHS) clauses (describing the action part). Usually, a number of additional declarative code definitions (e.g. type definitions for intermediate results, supporting rules) needs to be added.</td><td>For each agent type, code for the declarative model part has to be written in the related .drams files. Firstly, the fact templates have to be specified, after that the initial facts can be asserted to the fact base, and finally the rules can be written.</td></tr>
<tr><td>4.</td><td colspan="2">The declarative model part can be checked for consistency, using a visualisation of the automatically generated dependency graphs. Furthermore, testing and debugging procedures should be performed by executing the model using the Repast user interface, possibly with a reduced number of agent instances.</td></tr>
<tr><td>5.</td><td colspan="2">If the model is running as expected, then additional code for creating and storing simulation outcomes can be implemented.</td></tr>
<tr><td>6.</td><td>Productive runs of the full-scale simulation model can be carried out, again using the repast interface. Textual logs and numerical data is generated and stored according to the definitions made. XML based result files can then be analysed with the Simulation Analysis Tool.</td><td>Productive runs of the full-scale simulation model can be carried out, again using the repast interface. Textual logs and numerical data is generated and stored according to the definitions made, but no traceability information will be available.</td></tr>
</table>

3.4.2 Development of simulation models using DRAMS

To explain the application of the DRAMS formal model development, the example from the Campania pilot case is used.

After the development of a conceptual model by means of the CCD tool, the CCD2DRAMS was used to produce a skeleton of the declarative model (see section 3.3). The following methodological steps are suggested and were executed in the Campania pilot case to develop the simulation model using DRAMS:

1. The consistency of the generated model was inspected, for example by checking that all CCD Actors (Service Providers, Promoting Agencies, Enterprises and Citizens) had a corresponding Java class and a DRAMS file.
2. Java code was extended by manually adding methods, for example to implement the endorsement schemes or to create other agents (their number being set as a parameter at the beginning of every simulation run) in order to increase the number of competitors who submit proposals.
3. LHS and RHS clauses were elaborated for the CCD2DRAMS-generated rule stubs (see section 3.3.2). Also new rules were added to improve the model, for example to count the number of awarded proposals of a particular Service Provider or to produce print statements for debugging purposes.
4. The model was repeatedly executed and the DRAMS console proved itself to be a very useful debugging feature, allowing modifying and testing rules on-the-fly. Additional console windows, showing print statements with different priorities, were implemented using the related DRAMS clause (deflog).
5. DRAMS code for producing simulation outcomes was then added, both for the annotation of the model-based scenario (deflog) and the subsequent numerical data visualisation in Alfresco (defoutput).
6. Full-scale simulations were finally performed, setting 3 additional Promoting Agencies and 21 competitor Service Providers as Repast parameters. Simulation outcomes were analysed, annotated and exported to Alfresco by means of the OCOPOMO toolkit (see next section).

3.5 Simulation Analysis

3.5.1 Methodological grounds

Simulation Analysis is the phase of the OCOPOMO policy development process (cf. chapter 2), which results in model-based scenarios to be published on the Alfresco collaboration space for the stakeholders, domain experts and policy owners to discuss simulation outputs.

The simulation analysis tool thereby provides an integrated and comprehensive development environment for policy analysts and policy modellers to create the model-based scenario narrative with the graphical charts for data visualisation. Feeding the results of the simulation analysis into the Alfresco CSET enables the users to compare model-based scenarios with stakeholder-generated evidence-based scenarios (of phases 1 and 2) and to evaluate the simulation outcomes.

The main tasks related to the simulation analysis are performed by the Policy Analyst, with a support of modellers (i.e., activities related to the analysis of simulation logs and DRAMS code investigation), facilitators (i.e., output scenario publishing and formatting), and system administrators (i.e., technical maintenance and support).

After the output model-based scenario is published and properly formatted on the shared Alfresco web space, the evaluation of the scenario narrative and provenance of arguments provided in initial evidence-based scenarios are handled by involved stakeholders and domain experts and supported by the same set of collaborative Alfresco-based tools as it was used in the introductory phases of the process (see descriptions in section 3.1).

This process phase of simulation analysis and development of model-based scenarios, which is supported by the Simulation Analysis Tool and related OCOPOMO platform components, can be described in the following steps:

1) *transforming results* obtained from simulation *experiments into a narrative scenario text and visualisation of the numerical data* produced by simulation runs in a form of supportive charts included in the scenario text (by policy analyst and support by policy modellers). This step aims at the formulation of a verbal narrative that corresponds to the obtained simulation results, as well as to both types of underlying models - conceptual structures of CCD Model and formal structures of agent-based policy model that has produced the simulation logs. It implies that the development of the output scenario narrative includes a detailed analysis of obtained simulation results, which are presented to analysts in both XML and CSV Table formats in the simulation log views of the Simulation Analysis Tool. In addition, the analysis of simulation logs may require a need to browse the agent-based policy model, possibly also to run additional simulations, investigate the CCD Model, and study related input evidence-based scenarios as well. From this perspective, the creation of narrative model-based scenario is a complex task that should be accomplished iteratively, with a respect to different opinions and interpretations of various policy alternatives provided by stakeholders in their input scenarios, as well as to the facts derived from simulations of developed policy models on both conceptual and formal agent-based levels.
2) *annotating the scenario narrative to the relevant parts of simulation logs* (by policy analyst). The goal of annotations is to enable traceability from the output scenario narrative back to simulation results, rules of agent-based policy model, concepts of CCD model, and finally to initial evidence-based scenarios provided by the stakeholders, domain experts and policy owners at the beginning of the OCOPOMO process. From this perspective, the mechanism of annotations is especially important as enabler of provenance of arguments on policy alternatives, formulated by stakeholders in their input scenarios, which can then be validated in the provided output scenario narrative.
3) *publishing the resulting traceable model-based scenario and supporting files* (model-based scenario, CCD models, DRAMS, logs, unpublished background documents) to

the shared web space for further evaluation and discussion by the users (policy analyst and facilitator).

4) *Formatting and adding data visualisation charts to model-based scenarios (*by policy analyst and facilitator). The aim of this step is to enhance the scenario content originally produced in a plain text format by proper text formatting (i.e., fonts, colours, images, etc.) as well as by a visualisation of numerical outputs generated by simulation logs. These actions need to be performed after the scenario is published on the Alfresco web space, since current version of the Eclipse-based Simulation Analysis Tool supports only the plain text editing and annotation features, without a possibility to format the scenario content in the Wiki style. With respect to this limitation, it is highly recommended to finalise the output scenario by text formatting and charts inclusion before the scenario is officially announced (by a facilitator) as being ready for collaborative discussion and evaluation within the community of stakeholders and policy owners.

After the model-based scenario and relevant data a properly presented on the Alfresco space, the facilitator can trigger a discussion and possibly a collaborative enhancement of model artefacts by the involved stakeholders, domain experts and policy owners. This step is part of phase 6 of the OCOPOMO policy development process, which is already described in section 3.1.

3.5.2 Application of scenario analysis using the Scenario Analysis Tool

How to formulate the output scenario narrative?

The produced model-based scenario narrative is presented to involved stakeholders, domain experts and policy decision makers as the main output of policy modelling, which was accomplished by policy modellers locally in the Eclipse environment (CCD Tool, CCD2DRAMS tool and DRAMS). It implies that the quality of the scenario, together with its characteristics such as completeness, transparency of provided arguments, understandability, correctness, coherency, etc., are crucial factors for stakeholders to understand and accept the produced policy scenario and consequently the OCOPOMO approach as such. From this perspective it is important to formulate the output scenario narratives in accordance with obtained simulation results, developed policy models and input evidence-based scenarios, trying to minimise the influence of subjective opinions of involved policy analysis and modelling experts. For the actions related to the development of output scenario narratives we suggest the following:

- Keep the scenario narrative simple, short, and clear, and unambiguous.
- Avoid using complicated statements and/or phrases that may have several meanings.
- If possible, use the same wording as was used by stakeholders in their input narratives of evidence-based scenarios.
- Avoid using own opinions, interpretations, suggestions, etc. If it is necessary to support the output scenario by some external facts that were not included in initial evidence-based scenarios or their accompanying background documents, such documents containing the required external facts should be included as local

background documents. This requires iteration of phases 3 to 5, i.e. adjusting the CCD model with expert knowledge and new facts from background documents, adjusting the agent-based policy model, running the simulations again, and then updating the model-based scenario on these facts, which are now supported by simulation results.

- Anchor the output scenario narrative strictly to the results obtained from simulation runs. To do so, the narrative statements need to be consistently annotated with the respective records of simulation logs.
- Note that the text editor provided in the Simulation Analysis Tool does not support formatting. To highlight a text portion in the scenario narrative, use preferably the feature of annotations. Further text formatting can be done after the scenario will be published to the shared Alfresco web space.

How to annotate the scenario content?

To ensure the backward traceability of the scenario narrative to respective policy models (both conceptual and agent-based) and consequently to input evidence-based scenario narratives provided by stakeholders and domain experts, it is necessary to annotate the scenario content properly - i.e., to link key statements of the scenario narrative to the respective records of simulation results. The following suggestions are given to perform the annotation of model-based scenario narratives:

- Formulate the scenario statements simultaneously with annotations.
- In the list of simulation log records, which is available in simulation log views of the Simulation Analysis Tool in CSV Table and XML formats, identify the log records that correspond to particular actors and describe their behaviour, relationship, etc., with a certain qualitative or quantitative/numeric outcome. Refer to the underlying agent-based policy model to investigate the behaviour and relations of modelled agents. Then formulate the respective statement in the scenario narrative - for example, as a transcription of the declarative rule from the model code. Finally, highlight and link the formulated statement to the corresponding simulation log record that was invoked by this particular declarative rule.
- To ensure the backward traceability, use solely the simulation log records (or their parts in the XML format) containing UUIDs - identifiers of related CCD Model concepts.
- Note that the text editor and annotator provided within the Simulation Analysis Tool preserves existing annotations during the text editing. It means that one can type a draft statement into the scenario narrative, annotate it to proper simulation log record(s), and then update the text of the statement. The annotation(s) will be adapted accordingly.

How to publish the produced scenario narrative to Alfresco?

The publishing of produced model-based scenarios to the shared web space of Alfresco is rather straightforward. This activity is automatic since it is fully supported by the Simulation Analysis Tool. After invoking the respective command in the tool, the scenario is automatically converted to the wiki HTML format and is uploaded to the Alfresco content repository together with all the related information resources required for the traceability visualisation, namely the DRAMS source code of the agent-based model, all the simulation logs used for scenario annotations, the corresponding CCD Model, and all local background documents that were used during the conceptual modelling and/or annotation procedures. The uploaded scenario will appear on the Alfresco web space, under the list of Wiki pages.

Further actions may be required for text formatting and insertion of charts visualising the numerical outputs of simulations. Instructions for these actions are provided in the following two how-to recommendations.

How to organise content for presentation of the simulation results on the collaboration site?

The narrative model-based scenarios are presented on the Alfresco web space as Wiki pages, i.e. stakeholders and domain experts are using the same tool for editing of input evidence-based scenarios and for discussing and evaluating the simulation results. They can follow traceability links from annotations inserted in model-based scenarios and navigate to the related text fragments in evidence-based scenarios. Another traceability tool integrated in the Wiki page component is the CCD Explorer, which can be used by the stakeholders to browse underlying CCD models. The narrative annotated text of the model-based scenarios is published by the policy analyst or facilitator on the Alfresco shared web space; a facilitator then may organise the published scenarios among involved stakeholders for further evaluation and collaborative discussion. The following actions are suggested:

- Depending on the structure of the simulation model and generated simulation results, the scenario can be divided to the main Wiki page summarising an overall behaviour of all agents, and a set of sub-pages describing the simulation run from the perspective of a single agent or a group of agents. To simplify navigation for stakeholders, links may be inserted from sub-pages backward to the main scenario Wiki page.
- Provide an overview of the models and simulations in the introductory Wiki page published in the main content area of the site dashboard. The overview should contain links to the respective scenario Wiki pages.
- Use tags to additionally organise the content published for the model based scenarios.

How to visualise simulation results?

In addition to the narrative text of model-based scenarios, data logged during the simulation runs can be aggregated in tabular form and visualised in various graphical charts. Charts or tables with data can be directly included into the content formatted as a scenario Wiki page, so it is possible to mix narrative text and graphical presentation on the same page.

Table 3 shows the chart types supported.

Table 3: Chart types supported for graphical presentation of simulation outputs

Chart type	Description
Pie chart	Use pie chart to visualise numerical proportions. Displays tooltips when hovering over slices.
Scatter chart	A scatter chart is used to map correlation between sets of numbers. Displays tooltips when hovering over points.
Line chart	A line chart displays information as a series of data points connected by straight line segments. Displays tooltips when hovering over points.
Bar and Column chart	A bar chart displays horizontal rectangular bars with lengths proportional to the visualised values. Displays tips when hovering over bars. Column chart displays vertical bars.
Combo chart	A chart that lets you render each series as a different marker type from the following list: line, area, bars, candlesticks and stepped area. Displays tips when hovering over points.
Area chart	An area chart extends line chart. It displays coloured area between axis and the lines of data series. Displays tips when hovering over points.
Candlestick chart	A candlestick chart is used to show an opening and closing value overlaid on top of a total variance. Candlestick charts are often used to show stock value behaviour. In this chart, items where the opening value is less than the closing value (a gain) are drawn as filled boxes, and items where the opening value is more than the closing value (a loss) are drawn as hollow boxes.
Geo chart	A geochart is a map of a country, a continent, or a region with two modes: ● The region mode colorizes whole regions, such as countries, provinces, or states. ● The marker mode marks designated regions using bubbles that are scaled according to a value that you specify.
Motion chart	A dynamic bubble chart which allows exploring several indicators over time in interactive animated way.
Table chart	A table that can be sorted and paged. Users can sort rows by clicking on column headers.

4. Rationale and Guidelines for Stakeholder Engagement

Today's global challenges are interconnected, dynamic and complex in nature thereby having strong impact on the wellbeing of societies and economies. Dealing with complexity has become a key success factor for good governance in the 21st century. To ensure sustainable wellbeing of societies and economies, policy makers must be able to cope with unwanted side effects from environmental changes and social dynamics. Therewith, open government is a strand of development that aims to help governments to be more accessible and more responsive to demands and needs of stakeholders (citizens, businesses, etc.) along global challenges they have to cope with. Open government is an essential ingredient for democratic governance, social stability and economic development, which urges to implement key principles of good governance (see Bicking and Wimmer 2011b, Wimmer et al 2012). In implementing good governance principles, open government helps to overcome the long-lasting culture of politics of secrecy, where decisions were made without democracy and refers to public's right to know (cf. Michael 1983, Bicking and Wimmer 2011a).

Stakeholder engagement has become a key concept in current developments to exploit innovative ICT solutions towards realisation of Open Government and the principles for Good Governance. This demands in particular to engage stakeholders in the design and evaluation of models of social processes, i.e. in social policy formulation and policy analysis. Stakeholder engagement in OCOPOMO brings forth several **benefits to policy makers, domain experts and to stakeholders**:

- A more open and transparent process of policy formation and policy analysis
- Direct involvement and reflection on policy options through different stakeholder groups may bring in new ideas and, above all, can help getting to a commonly agreed policy solution
- Using the ICT toolbox, the conceptual perfect and simulation analysis help making complex policy environments more transparent and understandable
- Traceability of model outputs to model inputs helps establishing more trust in policy models
- Collaboration of a wider stakeholder group is possible through online engagement
- A higher quality of the policy models through better traceability and systematic and structured development concept
- Alternative new policy options can be discussed among stakeholders and hence reach intended addressees
- increasing capacity for innovation through wider group interactions and discussions of policy alternatives in an iterative process of policy analysis and policy formulation thereby exploiting ICT modelling means

OCOPOMO platform is dedicated to facilitate stakeholder engagement in many ways through novel forms of communication and collaboration. The main principles that underlie the OCOPOMO approach to facilitate stakeholder engagement are grounded in the good governance principles as mentioned above. In particular, these principles are:

(1) openness
(2) participation
(3) collaboration
(4) transparency
(5) coherence
(6) knowledge sharing

Besides the implementation of good governance principles along the OCOPOMO process, the development of models through policy analysts and policy modellers can always be traced by the stakeholders and policy owners through open access to the conceptual and simulation models. Hence, the facilitating staff (policy analyst, policy modellers, facilitators) is keen to ensure that data gathered from different sources are equally analysed in unbiased manner, as stakeholders and policy owners can evaluate the model outcomes and expose biased models to open discussion.

The OCOPOMO collaboration platform (CSET – see section 3.1) is designed to facilitate the stakeholder engagement using various components that, when properly utilised, can enhance new forms of collaboration among stakeholders placing them at the centre of a more open and equal relationship with policy makers. Stakeholder engagement is thereby not restricted to online interaction with stakeholders. Instead, a proper mix of online and offline interaction (face-to-face discussions in workshops, interviews, public hearings etc.) is recommended (see the vast amount of literature on e-participation and the various options on citizen engagement).

To ensure stakeholder engagement is sustainable, the whole participation endeavour is to be planned carefully and the stakeholders need to know when they can engage, what they will be expected to do and what potential impact they may create in the policy development (see the guidelines provided in section 2.4). Such aspects contribute to positive motivations of stakeholders to engage (chance to contribute to common good, possibility to shape the policy activities that influence stakeholders directly). Beyond that, efficiency of stakeholder engagement should be constantly evaluated and adjusted when needed so to ensure also trust in the process and value for money from the side of policy owners as well as stakeholders.

The following **how-tos** should help engaging stakeholders properly in the policy development and policy analysis process:

How to organise moderation?

Facilitator has many options how to monitor all activities on the collaboration site, including the changes of the content and site membership. The Site Activities dashlet tracks the most recent activities that have been performed in the site. Facilitator can configure Site Activities dashlet on site dashboard for all site members or multiple dashlets for all managed sites on his personal dashboard. Additionally, facilitator can subscribe to site RSS feed or can be notified via email digest. Another useful tool is content versioning supported for all content published on the site. Facilitator can review the modifications and display the previous versions. The facilitator and content creator can also restore content of a preceding version.

The main role of the facilitator is to actively participate in the collaboration and communicate with the stakeholders. Communication tools are integrated directly with the scenario editing tools (i.e. Wiki pages) and content sharing tools (i.e. Document Library). Comments are displayed within the context of the published content. It is also possible to use dedicated communication tools such as Chat for online communication, multi-threaded Discussion forums or Pollings, if stakeholder opinions need to be obtained in more structured way.

Besides monitoring and communication, facilitators have direct access to all content generated by stakeholders and domain experts including the published Wiki pages, background documents or comments. Facilitators can change access rights and ownership for the content or change the moderation of the membership. They can allow unauthenticated guest access to the site content or invite a new user.

How to get stakeholders involved?

The role of the facilitator is very important for the involvement of the stakeholders in the collaboration process. At first, facilitator prepares initial content on the collaboration site targeted to potential collaborators. The initial content consists of the main introductory page, which should explain the goal of the collaboration and introduce the problematic of the modelled policy case. Various groups of stakeholders then can identify their interests, which are the main motivation for collaboration. Introduction can be extended with the links to more detailed background documents containing existing case studies etc. Second important part of the initial content are scenario examples. Initial scenarios can be created from the discussion with the policy owner and/or interview discussions with pre-selected stakeholders relevant for the specific case. Facilitator is also responsible for management of the members for collaboration site in the case that site is private or has managed membership. It is also possible that stakeholders can invite others to become a member of the collaboration site (if such privileges are given to a user account). During the collaboration, facilitator is responsible for the moderation of the collaboration site, which includes active communication with the stakeholders using the provided communication tools.

How to deal with privacy concerns?

If particular issues of privacy are emerging in the policy case, discussion with the policy owner should reveal a clear policy of how to deal with such issues. The facilitator and policy owner may decide to restrict access to particular stakeholder (groups) or discuss the issue with the parties concerned. In this regards, it is important that the online collaboration platform provides means for privacy protection, e.g. restricted collaboration spaces for smaller groups or viewing user names through pseudonyms.

Overall it is important that the policy owners and facilitators provide a positive environment for stakeholders to openly writing their opinions. Otherwise, the principles of good governance are violated.

5. Guideline for transferring solution to other application domains

The OCOPOMO policy development process as well as the OCOPOMO ICT toolbox have a modular structure. That means that particular stages, aspects and tools can be used separately without having to employ the whole OCOPOMO ICT toolbox. Already the three different policy cases developed in OCOPOMO (renewable energy, knowledge transfer in economically disadvantaged regions, housing policy) show the applicability of the OCOPOMO solution in different policy contexts. Even along the implementation of these use cases, variations of methods have been applied along the policy development, hence evidencing the flexibility of the OCOPOMO approach. The following sections define extension points and interfaces, which can be used to exploit or customise the OCOPOMO process (section 5.1) or its ICT toolbox (section 5.2) in different policy domains.

5.1 Transferring the OCOPOMO policy development process to different policy domains

The OCOPOMO process and tools are primarily designed for open collaboration, where a group of stakeholders collaboratively creates scenarios for policy cases, uploads supportive background documents and discusses issues (e.g. policy options) about a policy case. This way, policy analysts and policy modellers gather relevant information for conceptually and formally modelling a policy case. The group of stakeholders subsequently discusses the results and engages in evaluating the conceptual and formal policy models. The whole process is designed to be iterative and to come to a consistent representation and analysis of policy options that are discussed among the different stakeholders, domain experts and policy owners. This process can easily be transferred to and applied in different policy domains as evidenced along the project performance (three different policy domains were studied and modelled following slightly different methods of stakeholder engagement).

As the process is not restricted to the particular methods used per phase to gather data and to engage stakeholders, it is flexible to be adapted to specific needs of individual policy development cases. For example, the development of initial scenarios can be done through direct interaction with policy owners or through collection of viewpoints via interviews of key stakeholders. The engagement of stakeholders in the scenario-building phase may embark on face-to-face workshops or interviews with stakeholders, pure online interaction or application of a mix of online and offline interactions. Key is that stakeholders are enabled to express their views and get access to different views to understand the policy context and alternative policy options expressed by other stakeholders or policy owners and domain experts.

Another key principle in OCOPOMO's policy development process is the use of conceptual models to generate a structured conceptual visualisation of particular views (who the actors are, which "objects" are subject of discussion or subject to treatment or change in the policy case, what the values and beliefs of actors are to behave and to set particular actions, etc.). This enables better understanding and through the annotation of texts enables traceability. Here, the process of OCOPOMO's policy development relies on the CCD tool and its extensions. Also here, transformation of the tools and methods to other domains is easily possible. The CCD meta-model is abstract enough to accommodate different peculiarities of policy domains. The CCD tool can also be configured to be usable with any other meta-model formulated in Ecore[5]. However in case that amendments to the meta-concept become necessary, this will require an update of the CCD tool, which will in turn have also consequences to restrict the use of the CCD2DRAMS Tool and DRAMS. Hence, the full potential of the OCOPOMO toolbox is only reached when using the whole integrated OCOPOMO ICT toolbox.

In OCOPOMO, the programming of simulation models embarks on agent-based modelling and in particular on declarative rule-based agent modelling using DRAMS. Employing the

[5] See Eclipse modelling framework: EMF / Dave Steinberg - 2. ed. . - Upper Saddle River, NJ [u.a.] : Addison-Wesley, 2009. - XXIX, 704 S. : Ill., graph. Darst. . - ISBN: 0-321-33188-5. - ISBN: 978-0-321-33188-5. - (The eclipse series)

ICT toolbox of OCOPOMO in other domains the best possible way means therefore to use the same modelling methods and tools. As the project has followed a modular approach, it is however also possible to apply either other modelling tools supporting agent-based modelling. At this point we would not argue that the OCOPOMO process or the OCOPOMO ICT toolbox would support to switch to a different modelling method as micro-modelling or econometric modelling result in very different simulation models with distinct grounds and goals of policy modelling. In consequence, OCOPOMO's policy modelling process and its respective ICT toolbox can be transferred to policy domains with the aim of applying agent-based modelling in modelling social processes. Hence, this simulation modelling paradigm cannot easily be replaced.

Based on the above reflection on applying the OCOPOMO policy development process into different policy domains, the next section provides considerations of transferring and extending the OCOPOMO ICT toolbox to other policy domains.

5.2 Transferring of ICT toolbox or individual components

5.2.1 CSET and CMS

CSET and CMS modules of the OCOPOMO ICT toolbox are designed as an extensible collaborative generic platform, which can be used independently from the other OCOPOMO modules. CSET and CMS are based on the Alfresco platform that provides good support for the interoperability standards and extensible frameworks for the development of new components. These features simplify customisation of the tools and integration into the existing knowledge sharing infrastructure. In order to extend CSET and CMS, the following extension points and interfaces can be used:

- **CSET modules** - All the components in CSET are implemented as modules (i.e. dashlets or site components), which can be deployed and configured independently. It is possible to customise existing components or to develop and integrate a new one using the Alfresco Surf and Web Scripts frameworks. Both frameworks are lightweight and scriptable, which accelerate development and simplify testing and deployment.
- **Interface between CSET and CMS** - CSET is implemented as the frontend that provides user interfaces for the content services running on the CMS server. CMS service layer is implemented using the extensible Web Scripts framework. It is possible to dynamically (i.e. without redeployment) modify existing services or provides new one. It is also possible to implement composite services, which are internally using other services. All content services are exposed through standardised REST APIs with the communication protocol based on the HTTP and JSON. Standardised interfaces allow that, besides the CSET, content can be managed by other clients implemented as web application, desktop application or as the mobile application for mobile phones or tablets.
- **Interface between CMS and CCD Tool/Simulation Analysis Tool** - Besides the content service REST APIs, CMS is compliant with the Content Management Interoperability Services standard - CMIS. CMIS is used to fetch evidence-based scenarios and background documents for the annotation in the CCD Annotation Tool and to publish model-based scenarios by the Simulation Analysis Tool. CCD

Annotation Tool currently supports annotation of documents in plain text, HTML (used for scenario Wiki pages) and PDF. Further formats can be added as extension plugins. Simulation Analysis Tool publishes model-based scenarios as HTML page with metadata embedded in the micro-format annotations. Using open formats and standards allows that besides the Alfresco based CSET and CMS modules, any collaboration platform compliant with the CMIS standard capable to publish and manage HTML content can be used for scenario editing or reviewing.

In sum, the Alfresco platform with the CSET and CMS customisations for OCOPOMO can even be replaced with other web content management systems providing the respective functionalities as exploited for the CSET and CMS components in the OCOPOMO ICT toolbox. However, programming effort will be required if these features are to be implemented in a different web CMS platform. As Alfresco and the extensions developed in OCOPOMO are available as open source components and as these are well integrated with the subsequent components, we recommend embarking on this toolset for the open collaboration and scenario development / scenario analysis when applying the OCOPOMO policy development process in a next policy case. This will also facilitate the use and exploitation of subsequent components of the ICT toolbox (CCD tool, DRAMS, Scenario Analysis Tool).

5.2.2 CCD and CCD2DRAMS Tools

The CCD tool is a particular development of OCOPOMO embarking on the Eclipse Modelling Framework. It enables systematic text analysis and generation of a structured knowledge representation of a complex policy domain. The building of the conceptual model is a manual process involving policy analysts and policy modellers. Many conceptual modelling tools do exist to support the construction of model artefacts in various domains and for particular purposes. In general, conceptual models represent an aspect of a real world or of a domain of consideration. Thereby, models are usually dedicated to a specific viewpoint of an aspect (see e.g. the different viewpoints of enterprise architecture models: processes, data, organisational structures, networks of actors etc. (Zachman, 1987)). In the case of OCOPOMO, the CCD tool embarks on the concept of knowledge representation in terms of an ontology (Gruber, 1993). The ontology helps to understand which actors are involved in a policy domain, what elements are treated, what actions are carried out by actors based on particular endorsements, beliefs or conditions, etc. I.e. the ontology conceptualises a policy domain. The CCD tool helps to construct such a knowledge representation. It furthermore helps to store provenance information by providing annotation features and linking the textual input to particular model elements. Through the representation of endorsement schemes and actions based on conditions, the CCD tool differs from the typical ontology modelling (using e.g. Protegé[6] or other ontology modelling tools) in the way that the resulting conceptual models do represent dynamic aspects (i.e. social behaviour and actions) of actors. Hence, the CCD tool supports policy modellers to first construct conceptual policy models before programming them into agent-based models.

[6] http://protege.stanford.edu/ (last access 20 May 2013)

The CCD tool can be applied to conceptual policy modelling independent of particular policy domains as has been proven in OCOPOMO already through three different policy cases. The tool has currently a transformation support towards DRAMS policy modelling – the CCD2DRAMS tool. If a different modelling tool will be used for programming the simulation models, the conceptual models of the CCD tool need currently to be programmed manually in the simulation environment as code generation features other than CCD2DRAMS do not exist at present. However, CCD tool is flexible as it provides the knowledge representation in standard XML code. Hence, if a different simulation modelling tool is to be used in a new policy development case, such a CCD2DRAMS equivalent component could be implemented that transfers the CCD concepts into the respective simulation environment (e.g. CCD2Repast, CCD2NetLogo, CCD2SKIN, etc.). Having the CCD2DRAMS transformation tool and the mapping concept available from OCOPOMO, such implementation seems not to be a big effort provided that the resulting simulation model environment is based on a structured meta-model similar to the DRAMS meta-model.

The CCD Explorer is an applet that can be integrated into different web-based environments to explore CCD models. Currently, it is integrated in the Alfresco environment as a plug-in. Like the CCD tool, this component can be easily used in different policy domains, provided that the OCOPOMO CCD tool or a corresponding CCD model is to be explored on a web-based environment.

To translate a CCD model into a local language (or into another language), a new model can be developed fully in the local language using the CCD Tool - the interface of CCD Tool being still English. Another option is to use some helper tools with the following steps for simpler translation (e.g. in case the model is developed in English):

a) at first, some prerequisites have to be ensured:

1. Installed Java RE 6 (http://www.oracle.com/technetwork/java/javase/downloads/index.html)
2. Installed SaxonHE9 (http://sourceforge.net/projects/saxon/files/Saxon-HE/9.4/SaxonHE9-4-0-6J.zip/download)
3. Download and unpack http://ocopomo.ekf.tuke.sk/svn/ocopomoprj/trunk/eclipse/updatesite/TranslationXsltTemplates.zip

b) subsequently, perform the translation with the following steps:

1. Convert CCD file to csv file:
   java -jar saxon9he.jar -s:<input ccd file> -xsl:ccd2csv.xslt -o:<output csv file>
2. Translate csv file to desired language (use supporting tools such as MS Excel and goolge translator)
3. Convert translated csv file to xml format:
   java -jar saxon9he.jar -it:main -xsl:csv2xml.xslt -o:<output xml file> translated=<translated csv file> encoding=<encoding of csv file i.e.: Windows-1250, UTF-8>
4. Convert original CCD file to translated CCD:
   java -jar saxon9he.jar -s:<original input ccd file> -xsl:ccdAndXml2ccd.xslt -o:<output translated CCD file> translated=<xml translation from step 3>

c) Localised CCD files can be published to Alfresco Data Repository via process described in Simulation Analysis Tool - User Manual (see deliverable 4.2)

To sum up, the CCD tool and its respective extensions are particular developments of OCOPOMO to support a better understanding and knowledge representation of a given policy domain through conceptual modelling, visualisation and exploration of CCD models via web environments as well as generation of basic simulation models in DRAMS. CCD Tool is fully supporting traceability through annotation features enabling text annotation, pdf annotation and annotation of wiki texts. This feature is a particular innovation in OCOPOMO. The tools can be fully applied to different policy domains, provided the use of the OCOPOMO ICT toolbox. Even though the replacement of Alfresco as the front end to enable stakeholder interaction is possible, the integration with the simulation environment DRAMS is much tighter through the CCD2DRAMS transformation tool. In consequence, the CCD tools have much greater potential of use and of providing the full benefits of the OCOPOMO solution, if employed along the OCOPOMO policy development process and integrated with the full set of OCOPOMO tools. For extensions to new simulation environments, programming effort will be needed to provide code generation and tracing features in the simulation environment.

5.2.3 DRAMS

Since DRAMS is designed as a self-contained Java software framework implementing a declarative rule engine, it can basically be used in a broad range of applications. Although it would be possible to use DRAMS as a rule engine outside of the multi-agent simulation environment to implement e.g. business rules, the actual strength of the algorithms driving the DRAMS engine is the optimisation for distributed interrelated rule engines, working on rapidly changing fact bases. As these are typical conditions in agent-based models, DRAMS performs well in such environments, while it will not reach the efficiency of rule engines developed for expert systems, e.g. based on the RETE algorithm.

Thus, the most important means by which DRAMS can be adapted to specific applications are the Model and Agent interfaces. Implementing the Model interface makes DRAMS cooperate with any Java-based simulation tool. A default implementation exists for RepastJ 3.1, but concrete model classes for e.g. Repast Simphony[7] or Mason[8] [ref] can be realised with little effort. Special implementations of the Agent interface might be needed for certain simulation tools, but these are usually more dependent on the objective of the model.

Large potential for opening DRAMS towards other application domains also provides the integrated plugin interface. Different kinds of plugins allow to extend the functionality of the DRAMS core in various ways, e.g. for adding (or replacing) user interface parts, components for storing or analysing simulation outcomes, and even new clauses to be used within rules can be added by plugins. Currently, the entire GUI, all output writing facilities and the Model Explorer are realised as DRAMS plugins.

[7] http://repast.sourceforge.net/repast_simphony.html (last access 20 May 2013)
[8] http://cs.gmu.edu/~eclab/projects/mason/ (last access 20 May 2013)

As with the other tools, DRAMS provides its full potential with the traceability feature when deployed with the other OCOPOMO tools in an integrated environment. If used as single component, provenance information cannot be conveyed from texts and conceptual models, hence this feature would be lost unless a conceptual modelling tool is implemented that is competing with CCD Tool and CCD2DRAMS transformation.

5.2.4 Scenario analysis tool

The scenario analysis tool consists of the viewer of the simulation run traces (log) and an editor for developing and editing narrative model-based scenarios based on the simulation log. The policy modeller and/or policy analyst inspects the traces and writes text of the scenario, which is annotated with the links to the traces. During the publishing of the scenario text, annotated links are resolved to the list of relevant CCD concepts and links to the relevant part of the evidence-based scenarios. The following interfaces can be used to extend Scenario analysis tool or to change the process of editing or publishing of the model-based scenarios:

- **Simulation traces** are contained in the **XML file** for which a specific schema was defined. Part of the DRAMS extensions is the plugin, which is used for logging of trace records into the XML file. Each logged record contains links to relevant CCD concepts. The Scenario analysis tool can be extended to process outcomes (XML-shaped or other formats) also from other rule engines or simulation environments.
- **Model-based scenarios** are published at the **HTML Wiki page** using the CMS REST API. HTML content is annotated using the micro-format annotations with CCD concepts and links to the evidence-based scenarios. Instead of the CMS REST API for publishing of Wiki pages in Alfresco Share, Scenario analysis tool can be extended and publish the model-based scenarios using the CMIS standard. In this way, text of the model-based scenario can be published as part of any web page (in different form, for example as a Blog page) managed by any web content management system compliant with the CMIS standard.

The Simulation analysis tool provides its full potential when used within the integrated OCOPOMO toolbox. The tool can be applied in different policy domains as has already been proven along the three policy development cases of the OCOPOMO pilots.

As with any concept and tool, these are developed for particular purposes. Hence, the next chapter outlines potential limitations of the solutions.

6. Potential Limitations of the OCOPOMO policy development process and its ICT toolbox

Potential limitations of the OCOPOMO solution are outlined along the main activities carried out in the OCOPOMO policy development process. Therewith, limitations of the process and of the ICT tools supporting the process steps are outlined in the subsequent sections.

6.1 Scenario Generation

When stakeholders develop their narrative representations and stories about policy issues, potential limitations of the OCOPOMO solution may be of different nature, as outlined below:

First, limited access to internet or lack of ICT literacy may exclude important stakeholders and domain experts from the process. This may lead to lack of some viewpoints in the policy analysis and might result in inappropriate simulation results subsequently. OCOPOMO therefore foresees offline and online interaction with stakeholders. Stakeholders with lacking ICT skills may be interrogated along interviews or may participate in workshops with stakeholders and domain experts. In that case facilitators will need to write down the scenarios based on information gathered in interviews, get authorisation from the offline stakeholders and subsequently and publish the material to the platform so other stakeholders may comment.

Second, some key stakeholders may be very busy and/or may not be willing to fully participate in online collaboration. The gather input from these stakeholders, interviews may be a means to collect relevant viewpoints and policy considerations. Similar to the offline case, facilitators have a key role to ensure information is collected from such stakeholders. The procedure is as in the offline case. The ideal situation would be that such stakeholders acknowledge the benefit of the OCOPOMO toolbox and engage in the process because they see added value in the results developed through using the OCOPOMO toolbox. The project may have to accept that such key stakeholders engage in certain stages but not along the whole process.

Third, mistrust between people may hinder communication and affect the output of the scenario generation process. For example, people from different domains may not want to share their knowledge, companies may not want to reveal their strategies, stakeholders that just met online may not feel comfortable to discuss some sensitive issues with others etc. The facilitator has to identify such obstacles and counteract by strengthening stakeholders' need for inclusion and the overall principles of good governance implemented through the OCOPOMO solution. Still, the OCOPOMO collaboration space even enables smaller groups to have their own restricted workspace so that they can express their views even though not exposed at first hand to other stakeholders. It is the facilitator and policy owners to decide whether such closed spaces are provided to particular groups of stakeholders. Important for the success of the OCOPOMO policy development is that the relevant stakeholders engage in providing their views. The CCD model will later on bring such restricted inputs together and expose the models to the whole audience of stakeholders and domain experts, as all relevant information feeding into the conceptual and later on the simulation models are annotated and can be traced back through the traceability mechanism. This feature supports the good governance principles and need therefore to be clearly communicated to the stakeholders when such concerns emerge.

Fourth, stakeholders may have insufficient knowledge to understand or to judge presented opinions. In that case, the online and offline discussions among stakeholders become crucial. The facilitator has to ensure that the communication ethics are kept and that there are not violating and offending statements among stakeholders. It is the facilitators' responsibility to intervene immediately if such indications of misconduct become noticeable in the CSET. If necessary, the facilitator needs to intervene among stakeholders and needs to modify text statements provided by stakeholders that contain such offending statements.

Fifth, the OCOPOMO solution accommodates both, smaller stakeholder groups as well as wider stakeholder engagement (e.g. involving the general public). It is the policy owners and the facilitators to discuss how wide the stakeholder engagement should reach and who is to be involved. The facilitator has a certain responsibility to ensure that no key stakeholder is left out from participation even though the policy owner might have objections or concerns to involve a particular actor. The facilitators together with the policy analysts and policy owners have to ensure to gather relevant input from different stakeholder groups that are relevant to the policy case. Stakeholder theory may help to identify relevant stakeholder groups. Of course, the interaction with the policy owner is crucial when identifying the relevant stakeholders. If certain tensions are noticed by the facilitator, the policy owner needs to be made aware of the risks of biased views or of lacking particular views of critical stakeholders.

6.2 CCD Annotation and Modelling

The CCD meta-model gives the conceptual modeller (policy analyst) freedom in the design of a CCD. The result is that different policy analysts can interpret things in different ways and can concentrate on different aspects of the underlying policy case. This freedom makes it difficult to compare distinct CCDs originating from different conceptual modellers. CCD models for different types and domains of policy models (meant as specifications for distinct policy domains) should be developed to support the conceptual modellers in developing CCDs to be more comparable.

The CCDs modelled in the OCOPOMO policy development process can be large in the sense of having many elements (vertices in graph theory) and relations among those nodes (edges in graph theory). Graph theoretic concepts of structure and paths can be applied to CCDs to compare different model designs. Whether by means of graph theory or other means, features are required to optimise the management and overview of elements in the CCD.

A representation of the CCD for non-expert stakeholders is another strand of further research to support policy modellers to speed up the implementation of a simulation model. Steps in this direction have already been taken by allowing for the filtering of visible CCD elements in the three diagrams.

Current CCD model representations for non-experts are complex. Usability of features and functions will be work of future research, too. Support for policy modellers may be further optimised in order to speed up the implementation of a simulation model.

The structure of actions and conditions is not strictly defined. This was chosen as an approach to make the arrangement of actions and conditions flexible and fast. With a more enhanced solution for automatic code transformation, this flexibility is no longer practicable.

A more formal representation of action flows should be chosen with the possibility to express logics.

The experience from OCOPOMO's policy modelling was that the development process is forward and backward at the same time. The CCD facilitates this process by structuring and conceptualising the policy domain and the aim of the resulting policy model before starting with the development of the policy model itself. However, a CCD needs to be extended and changed a number of times during the development of a policy model because simulation runs of the policy model produce a number of open questions, which need further investigation. The support of this forward and backward development process needs further investigation to be optimised. This is in particular important in order to support policy modellers to speed up the implementation of a simulation model.

6.3 CCD2DRAMS Transformation

The CCD2DRAMS transformation currently only supports transformation of CCD models towards the DRAMS simulation environment. As already explained in section 5.2.2, further transformation tools would be needed if different simulation environments are to be deployed for the agent-based modelling.

Within the OCOPOMO toolbox, a further step for better integration would be the implementation of support of the transformation in the opposite direction - i.e. from DRAMS to CCD. This would allow that changes in the simulation model by a policy modeller are transformed directly in the CCD. Rather than a model-to-text transformation approach, a model-to-model approach should be implemented, which allows the execution in both directions. This would require to select a corresponding transformation engine and to assemble a meta-model for DRAMS simulation models so that it can be processed more easily as one single, monolithic model.

Another supportive measure would be to see the CCD as a higher abstraction level of the simulation model in DRAMS, and to reflect this understanding in the meta-models. To achieve such, a common meta-model for technology-neutral policy models would be useful, which enables to have different views on a policy case. A CCD would then represent a more abstract view on the policy case - i.e. by taking bird's eye view representing e.g. agents only with respect to the occurring agent types and instances, while the implementation details of the DRAMS model are shown when zooming into the programming details. This would allow a smarter transformation between the two models, which would indeed represent the same facts, yet in different description languages and at different abstraction levels (conceptual, simulation programme). Such a solution would also resolve the backward consistency of models (from DRAMS to CCD), which is currently a manual step to be performed.

6.4 DRAMS Simulation Modelling

Although developing, debugging, running and analysing of simulation models is possible with the features currently provided by DRAMS, a number of measures would make performing these steps more efficient and more comfortable.

The most important point in this regard is the full integration of DRAMS in the Eclipse IDE. Currently only the editing of DRAMS code is supported by an Eclipse feature. Analysing and debugging models, testing of code fragments and supervising simulation runs require GUI components separated from Eclipse. The goal of an integrated DRAMS development environment could be achieved with a software “adaptor”, implemented on the base of the DRAMS plugin interface, and at the same time realising an Eclipse feature, which consists of a set of Eclipse plugins for the different UI parts.

Another set of issues affect (and also constrain) the design of simulation models. This is on the one hand the inherent (due to the data-driven rule scheduling algorithm) simulation time sub-structure (tasks) and the associated effect of multiply firing rules within the same (top-level) time step, which can be amended by an improved rule scheduling algorithm in the DRAMS core that is capable of scheduling the rule evaluation in a more “intelligent” order. On the other hand, the provision of additional features like meta-rules or backward-chaining would introduce new application fields for DRAMS.

There is also some potential for optimising DRAMS in order to make the execution of simulation models faster and to reduce the hardware requirements. The execution speed will get a boost by the above-mentioned improved rule scheduler. Another aim is to reduce memory consumption of (large) models. This could be achieved e.g. by implementation of a fact disposal policy, which intelligently cleans fact bases by removing facts that are no longer used or obsolete. At present, this can be (and has been) implemented either in the Repast Java code or by rules in DRAMS that retract specified facts asserted more than a specified number of time steps previously.

6.5 Simulation Analysis

The simulation analysis is based on a tool development that is more independent than other tools of the OCOPOMO ICT toolbox. Like the other tools, the Simulation Analysis Tool can be deployed in different policy domains. It embarks on a simulation log and provides formulation of a model-based scenario therewith annotating the log entries of the simulation outcome. The tool is used by policy modellers and policy analysts to interpret simulation outcomes for stakeholders to enable better understanding. It supports developing a textual narrative that helps stakeholders to understand what the simulation results are. The tool can be connected to web content management systems to publish the narrative with the log annotation and underlying conceptual models such as CCD models.

Some constraints or limitations of the Simulation Analysis tool can be mentioned as follows:

- The WYSIWYG editor in the Simulation Analysis Tool currently only supports plain text editing. The formatting of model-based scenarios needs to be done in the web content management system (CSET). An extension of the tool could enable to develop output scenarios in Wiki HTML format already on the Eclipse side, including formatting, images, tables, etc.
- The current version of the Simulation Analysis Tool also does currently not support to define the charts of simulation outcomes directly. The policy analyst has to publish the scenario without charts to Alfresco, and only after the scenario is published, the

charts can be included into the scenario Wiki content. An update of this Simulation Analysis Tool could enable to already include the charts and other images when defining the scenario narrative using the Wiki HTML format as indicated before.

- Data for the visualisation is uploaded into the CMS during the publishing of the scenarios. Data is thereby converted to tabular form and exposed as the dataset for the Google Visualisation API. Currently the policy analyst or facilitator has to manually copy the URL of dataset into the Wiki page editor form during the inserting of the chart. Also, it is possible to filter and aggregate data from one dataset, which corresponds to one simulation trace from one run. However it is not possible to join multiple datasets, i.e. data from multiple simulation runs. To visualise data combined from multiple runs, data can be exported and analysed in external tools. For this case, OCOPOMO charts support visualisation of CSV and spread sheet files published in Google Docs.

7. Added Value of the OCOPOMO developments for different stakeholder groups

The subsequent explanations provide insights into the added value and benefits of the OCOPOMO policy development process and ICT toolbox from the viewpoint of distinct stakeholder groups: Policy makers, wider stakeholders, policy analysts and policy modellers.

7.1 Policy Makers

For policy makers, the added value of the OCOPOMO policy development process and the ICT toolbox can be recognised in an overall improvement of the quality of policy making. From the perspective of decision makers it is very important to be able to clearly understand and interpret the outcomes of simulation models usually created by external experts. OCOPOMO's platform answers this need giving policy makers, as well as other stakeholders, an opportunity to trace the simulation output, through simulation logs and CCD, back to the stakeholder input and background documents. This is vital to counteract the notion of computer simulations being a "black box". Also, providing traces as to how stakeholders' and domain experts' knowledge and views on policy options have informed formal policy models and in turn have generated particular outcomes of formal simulations provides insight into internal workings and logics of simulation models that have not been possible before. This way, policy makers as well as stakeholders of the policy domain are equipped with a much better understanding of complex policy issues and their interdependencies in particular related to social behaviour of actors of the policy domain. This in turn results in better informed decision making and better understanding of potential impacts of alternative policy options.

Also for the policy makers it is important to deepen their understanding of the particular policy topic from different perspectives represented by all affected parties and to know if planned changes are important or redundant from the citizen point of view. OCOPOMO platform is designed to enhance the consultation process facilitating it online and hence reducing the consequences of spatial distance and time limitations (people can participate from various location and at different time) allowing for more convenient data gathering.

As policy makers may not necessarily engage along the whole OCOPOMO process of policy development, the tools developed, the development of model-based scenarios interpreting simulation outcomes and the traceability feature enable policy makers to quickly grasp an overall picture of the policy development process and of the stakeholder views and positions in a policy case.

In particular, the current request for open government with the principles of collaboration, openness and participation is supported to a large extent through the OCOPOMO policy development process and its respective ICT toolbox.

7.2 Stakeholders

For stakeholders, the benefits listed for policy makers are equally central. Another added value of the OCOPOMO platform is that it gives an opportunity to participate in an online collaboration leading to e.g. changes in their surroundings or implementation of new policy solutions. Also the online collaboration process is designed in a way that it allows for iterations and discussions among stakeholders on policy options, different perspectives and interests.

The iterative process enables stakeholders to comment simulation results and to request revisions if scenarios of stakeholders have not been mapped properly or the simulation outcomes produce insights that lead to revisions in the views and considerations of stakeholders.

7.3 Policy Analysts and Policy Modellers

For policy analysts and policy modellers, a main added value and benefit of the CCD tool (with the underlying CCD meta-model) is to enable non-expert policy modellers to extract aspects of the policy domain under consideration, to conceptualise and structure the domain. With that, a CCD supports policy modellers in getting an overview of the policy context and available documents.

An understandable conceptual model of a policy domain can further support expert policy modellers to generate initial code fragments for the simulation model through the CCD2DRAMS transformation tool. The implementation of the transformation makes it possible to generate from the CCD fundamental aspects of a simulation model in DRAMS, which are each provided with traceability information. Thus the CCD Tool helps to structure and speed the development of the simulation model.

As described in Deliverable D6.1, a key added value for the stakeholders and the policy modellers is the concept of traceability, i.e. being able to trace an agent description, a fact description or rules firing in the simulation runs back to the descriptions from where these have been derived (the evidences).

8. Conclusion

In this deliverable, guidelines for how to use and apply the overall OCOPOMO process and its respective ICT toolbox in particular policy contexts are provided. The guideline explains at first the overall OCOPOMO policy development process and respective methods used in individual phase. Therewith, the different actor groups and their role in the policy development are outlined. Subsequently the different tools along the main activities in the OCOPOMO
policy development process are briefly described in terms of methodical grounds and regarding how-to guideline for their application in policy analysis and development. The deliverable also reflects whether the solution can be transferred and deployed in different policy contexts. It outlines potential limitations and need for further research and development to extend and improve the solutions, and it reflects the added value of the solutions for the three main target users: policy makers, stakeholders and policy analysts / policy modellers.

The OCOPOMO tools have been developed on the basis of open source solutions. Hence, the tools can be used in different policy contexts. The deliverable outlines that the highest benefit of the OCOPOMO toolbox is granted when the whole ICT toolbox is deployed to support the OCOPOMO policy development process. However, the tools can also be used independently – with some restrictions that e.g. traceability is not supported if not all tools are deployed in their full scale in a new policy modelling endeavour. This deliverable outlines some pros and cons of individual use of components and potential extensions or adaptations to different tool contexts.

There is need for further development if the solution is to be enriched with more modelling tools for agent-based modelling. Yet, the solution as such is ready to use in different policy contexts. The deliverable 10.3 provides insights into exploitation plans and licence models for the solutions developed.

Overall, the solution developed – both, for the policy development process and for the ICT support – provides innovative features to implement open government and good governance principles in policy development therewith engaging wider stakeholder groups. It is to be stressed that the tools and policy development process can basically be applied by anybody who wants to explore a policy context with different policy options in a structured way using agent-based policy modelling. However, the skills and capacities to develop a conceptual policy model and henceforth a formal simulation model must not be underestimated. For this, the OCOPOMO project team intends to offer consultancy service for new policy development cases using the OCOPOMO toolbox and process. Further details on this are provided in the exploitation plan as documented in D 10.3.

References

Albrecht, S., et al (2008). eParticipation - Electronic Participation of Citizens and the Business Community in eGovernment. Study for Federal Ministry of the Interior (Germany), see: http://-www.ifib.de/-publikationsdateien/-study_e-participation_engl.pdf

Bicking, M., Wimmer, M.A. (2011a). Concept to Integrate Open Collaboration in Technology Roadmapping: Stakeholder Involvement in Strategic E-Government Planning. In: Proceedings of the 44th Annual Hawaii international Conference on System Sciences (44-HICCS). IEEE Press, New York

M. Bicking, M. A. Wimmer (2011b). A Scenario-Based Approach towards Open Collaboration for Policy Modelling. In M. Janssen, H. J. Scholl, M. A. Wimmer, Y.-H. Tan (Eds.). Electronic Government. 10th IFIP WG 8.5 International Conference, EGOV 2011. LNCS # 6846, Springer Verlag: Heidelberg et al, 2011, pp. 223-234

Carroll J (1995) Scenario-based design: envisioning work and technology in system development. John Wiley and Sons

Doran J, Gilbert N (1994) Simulating societies: an introduction. In: Gilbert N, Doron J

Gilbert N, Troitzsch KG (2005) Simulation for the Social Scientist. Berkshire: Open University Press.

Gruber, Thomas R. (1993). A translation approach to portable ontology specifications. In: Knowledge Acquisition 5 (2): 199–220.

Lotzmann, U and Meyer, R (2011). DRAMS - A Declarative Rule-based Agent Modelling System In: 25th European Conference on Modelling and Simulation, pp. 77-83.

Macintosh, A. and Coleman, S. (2004). Promise and Problems of E-Democracy, Challenges of online citizen engagement. Technical report, OECD.p.18

Michael, J. (1983): The Politics of Secrecy: Confidential Government and the Public's Right to Know. In: REVIEWS. Public Administration, vol. 61, no. 2, pp. 225--236. Wiley

Scherer, S.; Wimmer, M A. (2012): E-participation and enterprise architecture frameworks: An analysis. In: Information Polity. Volume 17, Nr. 2. pp. 147-161

Scherer, S.; Wimmer, M A, Markisic, S. (2013): Bridging Narrative Scenario Texts and Formal Policy Modeling through Conceptual Policy Modeling: The Consistent Conceptual Description Tool. In: Artificial Intelligence and Law, 2013 (forthcoming)

M. A. Wimmer, et al (2011). Open Collaboration in Policy Development: Concept and Architecture to integrate scenario development and formal policy modelling. In: Y. Charalabidis, S. Koussouris (Eds.). Empowering Open and Collaborative Governance. Technologies and Methods for Online Citizen Engagement in Public Policy Making. Springer Verlag: Berlin / Heidelberg, 2012, pp. 199 – 219

Winter, A. (2000). Referenz-Meta-schema für visuelle Modellierungssprachen. Deutscher Universitäts-Verlag, Oktober 2000.

J. A. Zachman (1987). A framework for information systems architecture. IBM Systems Journal, 26(3):277-293