

Sabrina Scherer, Maria Wimmer, Ulf Lotzmann, Scott Moss and Daniele Pinotti (2015)

Evidence Based and Conceptual Model Driven Approach for Agent-Based Policy Modelling

Journal of Artificial Societies and Social Simulation 18 (3) 14<http://jasss.soc.surrey.ac.uk/18/3/14.html>

Received: 31-Mar-2014 Accepted: 15-Apr-2015 Published: 30-Jun-2015

 Abstract

Agent-based policy modelling is an application of agent-based social simulation. In this contribution it is applied to strategic policy making in the public sector. Open government principles relevant in this domain demand solutions that trace the origins of modelling decisions from narrative texts (background documents and stakeholder scenarios) through the whole policy modelling process up to the simulation results. With the help of such traces, decisions made on the basis of such simulation results are more transparent and comprehensible. This paper presents a conceptual model-driven approach developed and implemented in the OCOPOMO project. The approach ensures traceability by integrating technologies for agent-based social simulation, semantic web and model-driven development. Narrative texts are transferred into Consistent Conceptual Description (CCD) models. Those CCD models are transferred semi-automatically into formal policy models implemented in the DRAMS (Declarative Rule-based Agent Modelling System) language. These formal policy models are further elaborated (i.e. the policy modeller has still full flexibility in programming the model), and runnable simulation models are programmed. From the simulation logs, model-based scenarios are generated to interpret and support a better understanding of simulation results. The model-based scenarios are textual narratives with charts summarising the output produced by the simulation runs. Thereby passages in these texts are linked with documents containing original narrative scenarios. These traces are realised via the CCD models. A well-elaborated policy modelling process and a software toolbox support the approach. A pilot case exemplifies the application of the process and the toolbox. Evaluation results from the OCOPOMO project show benefits as well as limitations of the approach. We also reflect how the process and toolbox can be transferred into other application domains.

Keywords:

Model-Driven Development, Agent-Based Policy Models, Annotation of Policy Models, Conceptual Models, Social Simulation Models, Provenance Information

 Introduction

- 1.1 Consultation in public policy development has a rich and longstanding history. Increasingly, agent-based models are being used both to guide and to formalise the consultative processes (Becu et al. 2008; Bommel et al. 2014; Downing et al. 2001; Moss 2002; Robinson et al. 2007; Valbuena et al. 2010). The OCOPOMO¹¹ project extended the use of agent-based policy models by embedding the modelling in a software toolbox that supports direct inputs of text-based scenarios and comments by the public as well as electronic documents in a variety of formats.
- 1.2 The aim of the project was to involve stakeholders in the policy¹² formation process thereby being participatory, consensus oriented, transparent, and as inclusive as possible in order to implement key good governance principles¹³. These inputs were used to develop both a structure of the model – stakeholders and relationships amongst them – and rule bases capturing stakeholders' behaviour and social interactions. The structure design was developed as an actor-network diagram and the rule base designs as process diagrams showing the conditions in which rules will fire and the consequences of the firings. The resulting model design constitutes a Consistent Conceptual Description (CCD) of the policy case (Scherer et al. 2013). Each conceptual model element – entities, relationships, rules and their contents – can and should have one or more hyperlinks to relevant text phrases in documentary evidence. Participants in the consultative process as well as those conducting the consultations, officials and the model developers contribute to this documentary evidence. Policy analysts transform the CCD automatically to basic programming code with hyperlinks from the CCD elements to corresponding elements in the code. The structure of the simulation model is encoded in Java using the Repast¹⁴ library and the rule base design into definitions and rule stubs in DRAMS – Declarative Rule-based Agent Modelling System (Lotzmann & Meyer 2011). Expert policy modellers then produce the code for the rule stubs, which maintain the links back to the CCD and the documentary evidence from which the CCD has been formulated. After running the resulting policy simulation models, model-based scenarios are produced from the simulation log. The model-based scenarios are again narrative documents supported with visualisations, which provide an interpretation of the simulation outputs to stakeholders in order to give further insights into the policy case. Through the whole process, the supporting software toolbox maintains links between the background of the CCD, the programming code, the numerical and textual outputs produced during simulation runs and narratives based on those outputs. These links ensure that the exact extent and nature of the evidence base for the resulting scenarios are transparent to users.
- 1.3 In this paper, we present the evidence-based approach for agent-based policy modelling answering the demand for transparent social models as argued by Waldherr and Wijermans (2013). The achievements of this framework are
 - Increased transparency of policy models in order to make them more "trustworthy"
 - Increased transparency of the modelling process in order to give guidelines to "beginners"
 - Involvement of stakeholders and other domain experts in scoping the policy case and hence in model design
 - Use of evidence-based (rather than theory-driven) models
 - Application of effective methods from software engineering (conceptual modelling, graphical design languages, model driven architecture) to the social simulation domain
- 1.4 The overall framework for evidence-based and conceptual model-driven agent-based policy modelling consists of a policy development process describing the main phases and steps of the approach and the OCOPOMO software toolbox – a selection of technical tools developed with the aim of supporting participants and modellers through the different phases of policy development.
- 1.5 The following sections describe the OCOPOMO approach. The next section reviews related work. The third section introduces the framework for an evidence-based approach to agent-based policy modelling, i.e. the policy development process and the software toolbox. The demonstration of its effectiveness is based on a pilot study evaluating a policy of public support for research-oriented spin-offs from Universities. This policy intends to achieve sustainable tourism and cultural industries in the Campania region of Italy, which includes Naples, Pompeii, Herculaneum and Mt. Vesuvius. Accordingly, the fourth section briefly outlines the scope of the Campania example, while the subsequent sections five to seven detail the phases 3 to 5 of the policy development process, which are at focus in this paper. The framework in section three and each of the phases in sections five to seven are presented along the following aspects¹⁵:
 - Objectives
 - Approach
 - Inputs
 - Steps
 - Outputs
 - Exemplification from the Campania case
- 1.6 Finally, benefits, limitations and the potential for transferring the solution into other application domains are discussed in section eight.

 Related work

- 2.1 The framework applies model-driven development of agent-based simulation models known from software engineering (Mellor et al. 2003; Völter et al. 2013). Gilbert and Troitzsch (2005) propose the use of UML¹⁶ diagrams for designing agent-based simulation models in social science: class hierarchies in order to present objects and agents with attributes; sequence and activity diagrams to present dynamics. Bommel et al. (2014) propose the usage of UML for communication with stakeholders. However, UML is not widely used in agent-based modelling (Bersini 2012). A reason might be that some scholars see 'pure' UML as an insufficient basis for modelling agents and agent-based systems (Bauer et al. 2001; Pavón et al. 2006). Bauer et al. (2001, p. 93) propose e.g. AgentUML as an UML extension. However, the policy modellers in the OCOPOMO project originated from research disciplines such as psychology or social sciences. Even if they are able to program excellent simulation models, it often turned out that the approaches preferred by them were different from those that computer scientists, software engineers or information architects would choose. In particular the object-oriented paradigm of UML was not adopted by them (DRAMS – the language used for programming the simulation models – is declarative). The policy modellers suggested an approach that should be adapted to their specific needs. In addition, policy modellers in the project were very sceptical towards an engineering-based approach, because experiences with different approaches limited them in modelling without providing significant or, at least in their view, sufficient benefits. For this reason, we adopted the approach of our social simulation modellers together with them in order to preserve some freedom in modelling issues with the CCD meta-model. Concept and software should be adapted to their needs with the aim to develop a conceptual design-based approach for policy modelling supporting traceability of design decisions. However from a technical point of view, the CCD meta-model is relying on Ecore¹⁷, which is concerned with modelling aspects in UML (Steinberg, Budinsky, Merks, & Paternostro 2008, p. 16).
- 2.2 Many approaches exist, which support the visual development of agent-based simulation models from prescriptive models and model-driven development of agent-based simulation models as e.g. the Eclipse Agent Modelling Platform¹⁸ (AMP), Repast Symphony¹⁹, INGENIAS (Hassan et al. 2009), ELMS (Environment Description Language for Multi-agent Simulation, Okuyama et al. 2005), or MAIA (Modelling Agent systems based on Institutional Analysis, Ghorbani et al. 2013). They focus on the development of runnable simulation models. These approaches can be seen as rather complex due to the required level of detail

for producing complete models for code transformation. They are not supposed to be used as communicative basis for discussions with stakeholders. In difference, the OCOPOMO approach aims in the first step to develop a policy model, which provides the main concepts of a policy and is understandable for stakeholders and domain experts. A CCD is not a formal simulation model but supports the semi-automatic transformation into one. The reason is that the project team saw the descriptive character of the conceptual model as more important than automatic code transformation (a semi-automatic transformation is accepted).

- 2.3 Pignotti et al. (2013) describe an approach that supports the linking of model elements with documents. In difference, it does – however – not include such a detailed conceptualisation as it is provided with the CCD. In addition, the concept of Pignotti et al. does not regard stakeholder participation in the policy modelling process.
- 2.4 The achievements described in the introductory section make the OCOPOMO process distinct from previous and more conventional approaches to policy formation in particular, and social analysis more generally. The key differences are the reliance on direct evidence to the exclusion of any more general social theory and the effect of this reliance on the degree of complication of the models. We offer no apology for ignoring social theory because, in the first place, we have found no cases, in which any models, theoretical or otherwise, have successfully forecast the impacts of any social policies and no cases, in which social theories have been constrained by evidence to anything like the same extent as OCOPOMO models.
- 2.5 Because of the reliance of the OCOPOMO process on evidence, which can be both voluminous and wide-ranging, the CCD can become exceptionally complicated. As the CCD is used to generate the elements of the formal models, this complication is reflected in the degree of complication of the models themselves. In principle no limit other than computational resources to the number of agents of any given type in a model exists. This feature of the process is also unconventional in relation to the social and methodological norms of the social sciences, where the models should be simple enough to be readily understood. However, in developing the OCOPOMO process, we took the view that the purpose was not to understand models but, rather, to understand and assess various social policy alternatives and, more generally, particular social processes. We are not aware of any evidence to suggest that the simplest models can capture all of the essential aspects of complicated social processes. We therefore allowed the degree of complication of the CCD (and therefore the simulation models) to be determined by the views and the needs of the stakeholders. It was because the resulting models could be quite complicated that we concentrated on ensuring both the transparency of the models and facilities for exploring the models piecemeal. The textual and numerical outputs from the simulation experiments are recast into natural-language narratives (the so-called model-based scenarios). As with any reports of actual events in mass media or scholarly histories, best practice requires the formulation of the reports to be comprehensible to a wide audience. In this case, the narratives meeting this criterion also have hyperlinks back to the CCD (which can be viewed graphically and focused on relevant groups of rules or stakeholders or other social entities) and through the CCD to the documentary evidence. As a result, stakeholders and other users of the OCOPOMO software toolbox can form an initial overview of the model-based scenario and then explore particular elements of that scenario to build up their own understanding of how the various entities and relationships suggested by the evidence lead of the simulated outcomes. Not all, and perhaps not many, of the stakeholders will want to understand the whole model and the whole social process it simulates. How much of the relevant processes they want to explore is up to them. We see our role as one of facilitation rather than advocacy or suggestion.
- 2.6 The next section presents the approach chosen in the OCOPOMO project and how it is tackling aforementioned challenges with the framework for an evidence-based and conceptual model-driven approach in agent-based policy modelling.



Framework for an evidence-based and conceptual model-driven approach in agent-based policy modelling

Objectives

- 3.1 The objectives of the framework are (as outlined in the introductory section):
 - To increase transparency of policy models in order to make them more "trustworthy"
 - To increase transparency of the modelling process in order to give guidelines to "beginners"
 - To involve stakeholders and other domain experts in model design
 - To use evidence-based (rather than theory-driven) models
 - To apply effective methods from software engineering (conceptual modelling, graphical design languages, model-driven architecture) to the social simulation domain

Approach

- 3.2 The starting point of the policy development process is either an existing policy, which needs to be revised, or a need for a new policy. A policy can be specified either by a government agency or an NGO or other particular stakeholder groups. Stakeholders and domain experts provide information in the form of accounts of actual events or stories or electronic documents such as web pages or PDF documents. These documents can be textual or statistical or any combination of the two. These documents constitute the evidence base for any OCOPOMO process.
- 3.3 To support comprehension and understanding of a complicated policy simulation model, the OCOPOMO process entails the use of Consistent Conceptual Descriptions (CCDs). From a technical point of view, a CCD means a file describing a policy case using the CCD meta model that defines a set of model elements. These conceptual descriptions are consistent both in the sense of formal logic (it is not possible to prove the same statement both true and false) and in the sense that they cohere with the evidence base described above. Conceptual model elements are linked with text phrases and statistics in the evidence documents. CCDs are therefore bridging the evidence base and elements of the simulation models developed by policy modellers. By semi-automatic transformation to the programming code of policy simulation models and its simulation outputs, the software toolbox maintains the provenance of outcomes from the simulated policy implementations. This framework uses multi-agent modelling; a modelling paradigm described e.g. in Gilbert and Troitzsch (2005). Hence, policy simulation models have to capture 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 the relationships between the individual actions on the micro-level and the collective effects on the macro-level. The outcome is a set of narrative texts describing the consequences of policies in a manner that are formally consistent with the evidence base. They aim to give the strategic decision makers as well as other stakeholders deeper and new insights into the effects of proposed policies and to inform the development of alternative policy options. Of course, the policy models based on the particular perceptions of different stakeholders are likely to be contradictory although each should be internally consistent. A key benefit of social simulation is that aspects are manifest in simulations that might not be evident from the textual descriptions by the stakeholders.
- 3.4 The software toolbox supporting the phases of the OCOPOMO process consists of the following tools: The CSE tool is a web content management system^[10] providing access by stakeholder to scenario editing and communication facilities. It integrates a CCD explorer, which allows browsing CCDs. Further, the software toolbox includes the CCD tool to create and edit CCDs, the CCD2DRAMS tool to transform a CCD into a DRAMS model, the simulation environment to program and simulate DRAMS models, and a simulation analysis tool to create model-based scenarios from the results gained along simulation runs. All tools except the CSE tool are available as Eclipse^[11] plug-ins. Hence policy analysts and modellers can use them in one environment. Interfaces have been programmed to import documents available in the CSE to Eclipse, and export model-based scenarios to the CSE.
- 3.5 Wimmer et al. (2012) detail the overall approach. The software toolbox supporting the process and models can be downloaded on the project website^[12].

Inputs

- 3.6 The inputs to the framework are
 - An existing policy, which needs to be revised, or a need for a new policy
 - Any background documents relevant in the frame of the policy including quantitative empirical data if available

Steps (Phases of the evidence-based approach)

- 3.7 Figure 1 details the overall policy development process with its phases, the artefacts generated in each phase and the tools of the software toolbox supporting the activities.

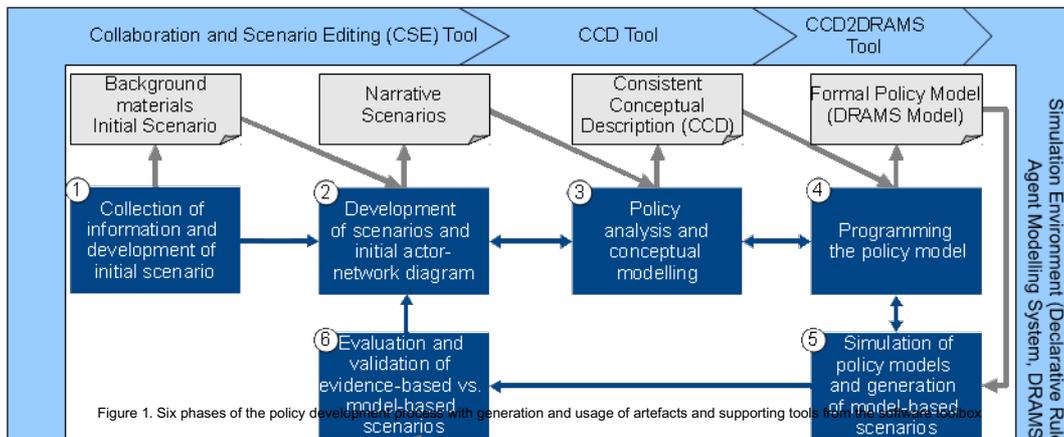


Figure 1. Six phases of the policy development process with generation and usage of artefacts and supporting tools of model-based scenarios

3.8 The process involves the following phases:

1. *Collection of information and development of initial scenario:* Policy analysts gather background information on the policy. Facilitators, domain experts and policy owners develop an initial scenario. The initial scenario gives an example in regards to the particular policy case as a basis for further discussions. The scenario can be provoking in order to stimulate discussions among stakeholders. The software toolbox supports this phase with the Collaboration and Scenario Editing (CSE) tool that provides web content management features for editing and storing documents online and for discussions among stakeholders, and with policy analysts.
2. *Development of scenarios and initial actor-network diagram:* The CSE tool supports the facilitator in this Simulation Analysis Tool functionality and a discussion forum to support stakeholders in collaboratively developing scenarios. The policy analyst and/or facilitator designs an initial actor-network diagram together with stakeholders.

During phases 1 and 2, facilitators support the process and focus of the conceptual and formal policy model together with decision makers (policy owners) using the toolbox for the policy model, which supports them in decision making.

3. *Policy analysis and conceptual modelling:* The policy analyst creates a conceptual policy model (i.e. a CCD model) from the narrative and textual information base collected in phases 1 and 2 with the CCD tool. Storage of links of conceptual model elements to narrative texts allows tracing the transformation of information from narrative text scenarios to formal policy and simulation models. Policy modellers use these conceptual models for representation of knowledge available about the policy case and to develop the policy models.
4. *Programming the policy model:* Based on the CCD model, policy modellers develop the agent-based formal policy model in DRAMS with the simulation environment, on which the simulation runs. The CCD2DRAMS tool provides semi-automatic transformation of a CCD into DRAMS. Formal policy models are the starting point for running simulations of the policy case.
5. *Simulation of policy models and generation of model-based scenarios:* Policy modelling experts instantiate simulation models with particular variables and run the simulations. The results received from such simulations are textual logs. Policy modellers have to interpret and visualise them in a proper format. Hence, they transform these textual logs from simulation runs into so-called model-based scenarios (i.e. narrative texts) and into supportive graphical charts if respective data is a simulation outcome.
6. *Evaluation and validation of evidence-based vs. model-based scenarios:* Phase 6 serves to expose the model-based scenarios (i.e. results of phase 5) to different groups (policy owners, domain experts and wider stakeholders). The purpose is that these groups assess, evaluate and validate the simulation results and therewith compare these results with the evidence-based scenarios they have generated or have in mind. Through these 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.

3.9 The approach involves iterative implementation cycles. If the model-based scenarios from 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). Hence, a new process iteration is executed. This iteration cycle ensures that, in the end, the stakeholder scenarios are consistent with the formal policy model outcomes (i.e. the model-based scenarios). In addition, the simulation and scenario analyses contribute to new insights, which might result in revisions of the scenarios or demand for further background information that should be considered. Hence, we do not suggest a strict sequence of the process phases. Figure 1 outlines this with the double arrows.

Outputs

- Annotated narrative text scenarios generated by stakeholders and policy owners to reflect different views and aspects of a policy case
- Annotated background documents supporting evidence of stakeholders' views in their scenarios as well as overall understanding of the policy case
- Conceptual policy models (CCD models) describing the main elements such as actors, key objects in the policy case, actions and conditions for actions, and carrying the links to the text annotations in scenarios and background documents
- Formal agent-based policy models in the form of DRAMS models
- Simulation logs and print statements
- Model-based scenarios interpreting simulation logs and print statements

3.10 Sections five to seven detail phase 3 *Policy analysis and conceptual modelling*, phase 4 *Programming the policy model* and phase 5 *Simulation of policy models and generation of model-based scenarios* of the overall policy development process as subject of the paper. The next section presents the example used to explain these phases and application of components of the software toolbox.

An Example: the Campania case

4.1 The OCOPOMO project encompassed three pilot cases; each case tackling a specific policy issue:

- Competence centres for knowledge transfer in Campania region, Italy
- Heating in Kosice self-governing region, Slovakia
- Housing facilities in London, United Kingdom

4.2 In each case, the framework presented in section three has been applied for developing policy models¹³. In the context of the paper, we are focussing on the competence centres for knowledge transfer in Campania region, Italy.

4.3 The policy to be developed in the Campania pilot case aims at establishing competence centres for knowledge transfer in order to support the development of industrial clusters within the Campania region (Europe area project). The formation of networks between private and public bodies is assumed as the main way to stimulate innovation in Small and Medium Enterprises (SME) and extend the dissemination of the results of Research & Development (R&D) results. The need to create intermediary institutions that were neither business enterprises nor government agencies, but which form an integral part of the local/regional system of innovation, was recognised in Campania. The competence centres represent a networking model with the objective of bringing together the research capacities of regional actors such as research institutes, universities, laboratories, etc. that were previously unconnected. In the OCOPOMO project, the focus is on financing new and sustainable activities that take advantage of economic resources and points of scientific and technological excellence. The objective of the pilot case was simulation of different activities to assess policies intended to provide finance for new and sustainable activities through a combination of public and private incomes.

4.4 In the next three sections, this policy case is used to demonstrate the conceptual modelling and simulation as well as the traceability supported by the software toolbox of OCOPOMO.

Policy analysis and conceptual modelling (Phase 3)

Objectives

5.1 The objectives of the *policy analysis and conceptual modelling* phase are

- To analyse the information available on the policy case
- To develop a conceptual model of the policy case based on information available
- To annotate information sources in order to establish provenance

Approach

5.2 Policy analysts develop a conceptual policy model in the form of a CCD. Policy modellers use this CCD model for representation of knowledge available about the policy case and to develop the policy simulation model. Conceptual models base on the CCD meta-model embedding the following elements (Scherer et al. 2013, pp. 463–467):

- Structure: Description of the basic elements of the policy domain. Concepts describing 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. Actor Network Diagrams visualise Actors, Objects and their relationships.

- Behaviour: Actions and their interrelations encode dynamic aspects. 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 basic elements. Action Diagrams visualise Actions and Conditions and their relationships.
- Facts: Instances of elements Actor or Object encode the 'number and nature of inputs' (i.e. the facts). Instance Diagrams visualise Instances and their relationships.
- Provenance: Annotations maintain the provenance of a model element by linking related text phrases (either txt or pdf) or describing decisions made by the policy analyst e.g. to enrich and complete the CCD model.

- 5.3 Each element in a CCD can be identified via its UUID^[14]. The CCD tool^[15] supports the policy analysis and conceptual modelling phase. Scherer et al. (2013) provide an overview of the CCD meta-model (vocabulary) and CCD tool features along the pilot case *Heating in Kosice self-governing region, Slovakia*.

Inputs

- 5.4 The inputs to the *Policy analysis and conceptual modelling* phase are different evidence documents as
- An existing policy, which needs to be revised, or a need for a new policy
 - Any background documents relevant in the frame of the policy including quantitative empirical data if available
 - Initial actor-network diagram describing relevant stakeholders for the policy
 - Initial scenario describing the policy
 - Narrative scenarios developed by stakeholders
 - (Model-based scenarios in a cycle iteration)

Steps

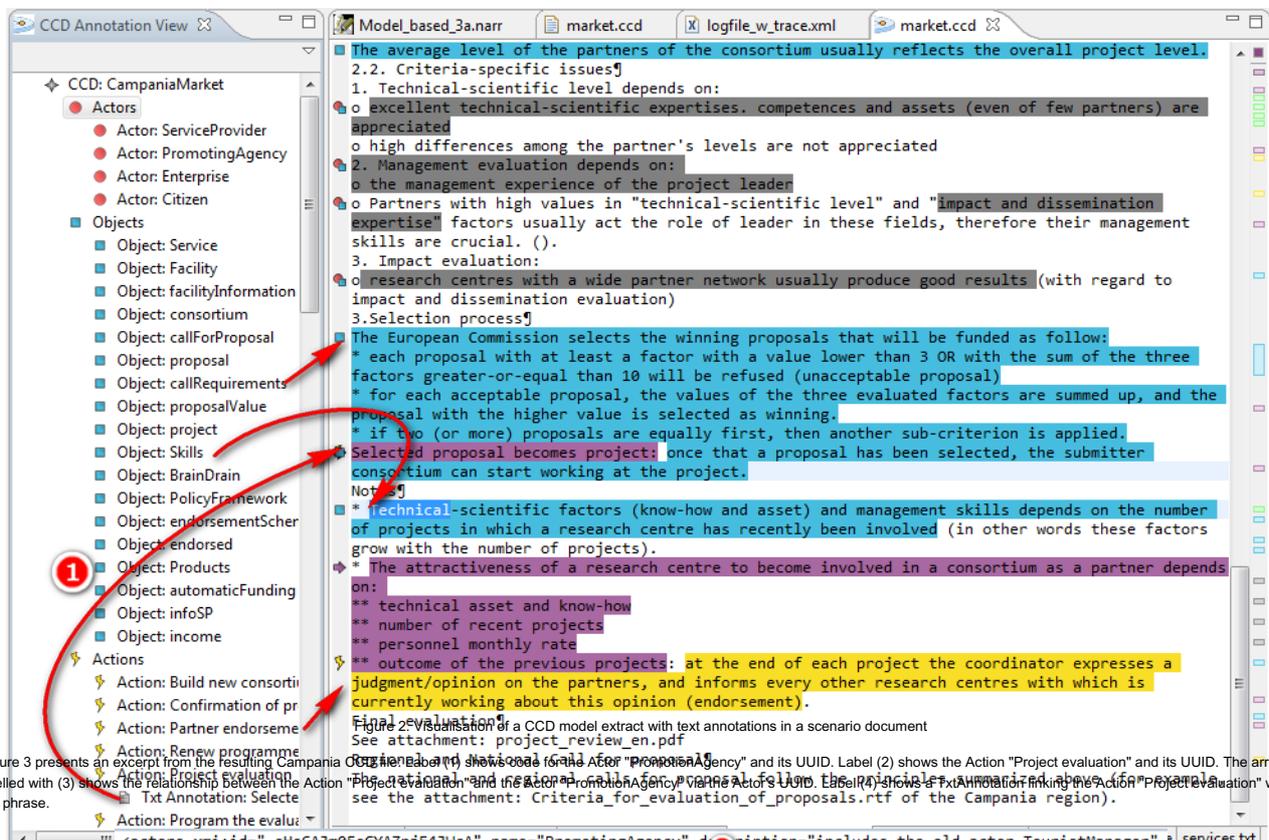
- 5.5 The policy analyst should apply the following steps to develop a CCD. The steps follow the *first modelling process steps for developing social simulation models* proposed in Doran and Gilbert (1994, p. 10):
1. Considering what is known about the policy: Analysis of data, available information and scenarios. If discrepancies between scenarios exist, 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, the policy analyst can decide to define a baseline CCD, which is then branched into different versions based on the different scenarios.
 2. Clarifying modelling objectives: Key questions to be clarified are: What should be modelled, and are the objectives in line with available information? The policy analyst should create a CCD file, describe the modelling objectives in the description and annotate the CCD with relevant documents in order to ensure that information is sufficient to meet the objectives.
 3. Choosing policy aspects to be modelled: Analysis of data and background information for relevant elements and their relationships. The policy analyst should extract relevant phrases of relevant documents and create respective elements in the CCD that are linked with text phrases. Elements can be transformed as follows
 - a. Concepts are represented with Actors or Objects, depending on whether they have an active role or not.
 - b. Facts are represented with Instances of Actors or Objects.
 - c. Relationships are represented with Relations on the Concept level and Instance level.
 - d. Characteristics are represented with Attributes on the Concept level and Instance level.
 - e. Behaviour of actors is represented with Actions and Conditions. The policy analyst needs to link these elements with those already defined in the CCD.
 - f. Annotations are represented with TxtAnnotations, PdfAnnotations, or ExpertAnnotations.
- 5.6 Gilbert and Troitzsch (2005, p. 207) argue that iteration is a common process when developing simulation models. Accordingly, the policy analyst revises a CCD in several iterative steps until a version is finalised. Conceptual modelling is also 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, i.e. the CSE tool as online collaboration environment). Variations from or extensions to the scenarios can be conceptualised by the policy analyst adding respective descriptions (called 'expert annotations') to the CCD elements. It is not the task of the policy analyst to filter out discrepancies in stakeholder or domain expert views between scenarios. The policy analyst needs to decide to describe them in one conceptual model or in different versions of the conceptual model (depending on the objective of the overall policy modelling). In the latter case, the policy modeller can create different policy models to compare simulation results in the next phase.

Outputs

- 5.7 The outputs of the *Policy analysis and conceptual modelling* phase are:
- One or more CCDs describing the policy case
 - Annotated documents

An example

- 5.8 First, policy analysts imported inputs generated in collaboration with stakeholders and background documents from the CSE tool into the CCD tool. Further, policy analysts annotated these narrative texts and elaborated model elements. Figure 2 presents an extract of the results. It highlights the relations between annotated text fragments and elements of the CCD. On the left, it visualises an extract of the CCD model as a tree with its elements. On the right, it visualises the annotation editor of the CCD tool with an annotated text fragment. The user selects text passages in the annotation editor. By drag and drop, the user can create new elements or add annotations to an existing element in the tree. All coloured text phrases in the annotation editor present annotations. Each colour represents a different element type. It is possible to annotate one text phrase with different elements. The arrow labelled with (1) shows a TxtAnnotation element that links the parent element, in this case the Action "Project evaluation", with the text. This way, policy analysts linked other elements as well (see a few further arrows as examples).



5.9 Figure 3 presents an excerpt from the resulting Campania CCD file. The red circle (1) shows the Actor "PromotingAgency" and its UUID. Label (2) shows the Action "Project evaluation" and its UUID. The arrow labelled with (3) shows the relationship between the Action "Project evaluation" and the related text phrase.

```

<actors xmi:id="sHsCAJm0EeGYAZnjF4JwCA" name="PromotingAgency" description="includes the old actor TouristManager" =
<annotation xmi:type="org.ocopomo.annotation:TxtAnnotation" xmi:id="yXobALqxEEGfI5AVKZldYA" phrase="promoting ag
<instances xmi:id="N0WfAJm3EeGYAZnjF4JwCA" name="CampaniaRegion">
  <annotations xmi:type="org.ocopomo.annotation:TxtAnnotation" xmi:id="Iy4vwLqYEeGfI5AVKZldYA" phrase="the Campan
  <relations xmi:id="sDVVwLxLEeG0xNf8XGArKw" targetInstance="_CEozYLxHEeG0xNf8XGArKw" relation="VGMQILxBEeG0xNf8
  <relations xmi:id="6BnAMp9EeG8ZeJKercPTQ" targetInstance="_NRoMcm3EeGYAZnjF4JwCA" relation="_0V-TEcp9EeG8ZeJK
</instances>
<instances xmi:id="_ObsJmZm3EeGYAZnjF4JwCA" name="HeritageMinistry">
  <annotations xmi:type="org.ocopomo.annotation:TxtAnnotation" xmi:id="gcnxobqxEEGfI5AVKZldYA" phrase="Ministry o
  <relations xmi:id="_n2jYoJnjEeGOK_Z60L_wMQ" targetInstance="_NRoMcm3EeGYAZnjF4JwCA" relation="eS4_8Jm1EeGYAZnj
  <annotations xmi:type="org.ocopomo.annotation:TxtAnnotation" xmi:id="_A3c0ILq7EeGfI5AVKZldYA" phrase="manages
</relations>
<instances>
<instances xmi:id="_CCTGcLnmEeG16NsQLi0T7Q" name="EuropeanCommission" description="Plays the role of PA in Heritag
  <annotations xmi:type="org.ocopomo.annotation:TxtAnnotation" xmi:id="_8kVVvQLqxEEGfI5AVKZldYA" phrase="European C
</instances>
<relations xmi:id="eS4_8Jm1EeGYAZnjF4JwCA" name="manage" target="_7znIUJm0EeGYAZnjF4JwCA"> <annotations xmi:type=
</relations>
<relations xmi:id="_N9JugJnfEeGOK_Z60L_wMQ" name="observe" target="_1EUyQJnUEeGxT_31eiTFcw"> <annotations xmi:type
</relations>
<relations xmi:id="_fppAJ2VEeGEd5K_GPnmSQ" name="evaluates" target="_xopPwJ2UEeGEd5K_GPnmSQ"> <annotations xmi:ty
</relations>
...
</actors>
<actions xmi:id="4iDQ8BFVEeKyDM5BPNMhQ" name="Project evaluation"
  description="Promoting agency evaluates submitted proposals and awards the best"
  preconditions="0BMpGfBCEeKyDM5BPNMhQ"
  postconditions="_UuClkBFWEeKyDM5BPNMhQ"
  actors="sHsCAJm0EeGYAZnjF4JwCA">
  <annotations xmi:type="org.ocopomo.annotation:TxtAnnotation" xmi:id="fEAzgfBFEeKyDM5BPNMhQ" phrase="Selected pro
  uri="Public funding in the heritage sector .txt" start="6911" length="33"/>
</actions>

```

Figure 3. Excerpt of the Campania CCD file in XML

- 5.10 The Campania CCD model describes strategies implemented by a regional centre of competence (i.e. a consortium of research centres) labelled as "service provider". It outlines the dynamics associated with i) competitive calls for proposals issued by public bodies at European, national and regional level and ii) free-market services sold to promoting agencies or private enterprises. The Campania CCD model contains two sub-models:
- The *competitive bids sub-model* focuses on strategies implemented by regional centres of competence to maximize participation in competitive bids provided by public R&D funding programmes. In this model part, public organisations (such as the Campania region or the Ministry of Education) issue calls for proposals, service providers set themselves up in consortia and submit proposals to answer the call. The promoting agency selects the best proposals, which become a financed project. Service providers increase their skills on the basis of funded projects.
 - The *service and products delivering sub-model* focuses on provisioning a set of services and/or products for customers on the market by regional centres of competence (service providers). In this model part, providers sell services to both enterprises and promoting agencies. The latter having a double-sided interaction with the providers (e.g. on both the bids and the market side). The skills of a service provider influence the effectiveness of its services.
- 5.11 Both sub-models generate incomes and endorsements for service providers (i.e. Regional centres of competence). In the following, we focus on the *competitive bids sub-model*.
- 5.12 Figure 4 shows the general structure of the *competitive bids sub-model* created in the Actor Network Diagram. The model contains two main actors: the funding agencies that promote and evaluate calls for proposals ("PromotingAgency") and the service providers ("ServiceProvider"), which set up consortia in order to participate in such a call. The diagram represents main Actors with a red dot, Objects with blue

Programming the policy model (Phase 4)

Objectives

- 6.1 The objectives of the *Programming the policy model* phase are
- To transform a CCD into a runnable simulation model programmed in Java and DRAMS
 - To develop and refine the resulting policy model

Approach

- 6.2 A policy simulation model in the OCOPOMO process implements the behaviour of agents as rule-based, declarative entities created and initialised by a number of Java classes transformed automatically by the CCD2DRAMS tool for the Repast-based implementation of the agents as defined by the actor-network diagram. As described in more detail below, the CCD2DRAMS tool also produces DRAMS code fully specifying initial facts, permitted clause definitions and various terms and tokens. The tool also generates rule stubs, which contain the rule names from the Actions in the process diagram as well as conditions in which the rule will fire and further conditions generated by the rule. A DRAMS specification can be found in the OCOPOMO project documentation^[16].
- 6.3 The policy modeller elaborates the rule stubs to produce complete rules and also to generate print statements later used for the development of narratives describing the scenarios generated during simulation runs. These narratives enable stakeholders and analysts to explore the individual actions and combinations of actions that are believed to be available to policy owners (e. g. governments) for the purpose of achieving specific and well-formulated objectives. The model 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.
- 6.4 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 of OCOPOMO to policy modelling. As we have seen, the CCD2DRAMS component supports the policy modeller in generating both Java and DRAMS source code from a CCD file. It maintains provenance by linking the generated code fragments to concepts of the underlying CCD file using their UUIDs. From a conceptual viewpoint, the transformation of conceptual models into formal policy models is done by building analogies between elements of the CCD meta-model and elements of the DRAMS meta-model. 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 "Agent Class" in the DRAMS meta-model. The CCD2DRAMS tool transforms Objects and Relations into fact templates^[17], CCD Instances into facts^[18] and CCD Actions with their pre- and post-conditions into rules^[19]. The tool also generates required Java code to set up and run a Repast model. 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 of the CCD cannot completely be transformed automatically into DRAMS. In the CCD modelling, the focus lays on the conceptual description of actors' behaviour. It does not describe Actions in the sense of "if-then-else" rules. Therefore, the CCD2DRAMS transformation can only generate rule stubs in DRAMS from Actions in CCDs.
- 6.5 At a minimum, the policy modeller has to fill in the rule stubs generated by the CCD2DRAMS tool with DRAMS code. It is also possible to elaborate the Java code produced by the tool to allow for efficient computations called by clauses in the DRAMS rules. Any of the usual devices used by developers of agent-based models (such as the use of random numbers to determine when particular behaviours will be manifest) can also be incorporated into the simulation models.

Inputs

- 6.6 The inputs to the *Programming the policy model* phase are
- A CCD file
 - Background documents from former phase to inform the policy modeller

Steps

- 6.7 Table 1 describes the steps for programming the policy model – both, using the CCD2DRAMS model transformation or building the model from scratch.

Table 1: Steps to transform the conceptual models into DRAMS – both, using CCD2DRAMS model transformation or building the model from scratch.

Step	Model skeleton generated with CCD2DRAMS	Model set up from scratch
1.	Running the CCD2DRAMS tool to generate the initial simulation project. The generated Repast model and .drams files should be inspected for consistency and completeness.	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 model-related aspects have to be added.
2.	Optionally, Java classes for model or agents can be adapted or extended.	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.
3.	The rule definition stubs need to be elaborated by specifying the Left Hand Side clauses (for the condition part) and the Right Hand Side clauses (describing the action part). Usually, additional declarative code definitions (e.g. type definitions for intermediate results, supporting rules) need to be added.	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 the rules can be written.
4.	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.	
5.	If the model is running as expected, then additional code for creating and storing simulation outcomes can be implemented.	
6.	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 to generate model-based scenarios.	Results can be analysed manually. No traceability information will be available.

Outputs

- 6.8 The outputs of the *Programming the policy model* phase are one or more policy models on which simulations can be started.

An example

- 6.9 The CCD2DRAMS tool was used to produce a skeleton of the simulation model. According to the CCD, the formal policy model contains two main agent classes: the funding agencies that promote and evaluate calls for proposals ("PromotingAgency") and the service providers ("ServiceProvider") which set up consortia in order to participate in such a call. The entire process has been modelled, including allocation of funds for the call, the setting up of the consortium, development of a proposal, submission and evaluation of results. The model also captures free-market services, whereby service providers are able to sell market services to either promoting agencies or private enterprises.
- 6.10 Figure 6 presents an extract of code generated from the Campania CCD model via the CCD2DRAMS tool. From the Actor "PromotionAgency" in the CCD, the corresponding Agent was created with a Java class and a DRAMS file. The arrow labelled with (1) shows that provenance of the Java class, i.e. the Actor "PromotionAgency", is maintained via this Actor's UUID. From the action "Project evaluation" with Actor "PromotionAgency" in the CCD (cf. Figure 3), a rule stub was automatically created in the DRAMS file of the agent "PromotionAgency". This CCD2DRAMS-generated rule stub was expanded by the policy modeller. The UUID (@link (see the arrow labelled with (2)), the rule's structure and its name were generated, while additional clauses were written on both rule sides by the policy modeller. Figure 6 also shows maintenance of provenance between Java and DRAMS source code (on the upper right, arrows labelled with (1) and (2)) via CCD (on the left) to the original text phrase (on the lower right, the arrow labelled with (3)).

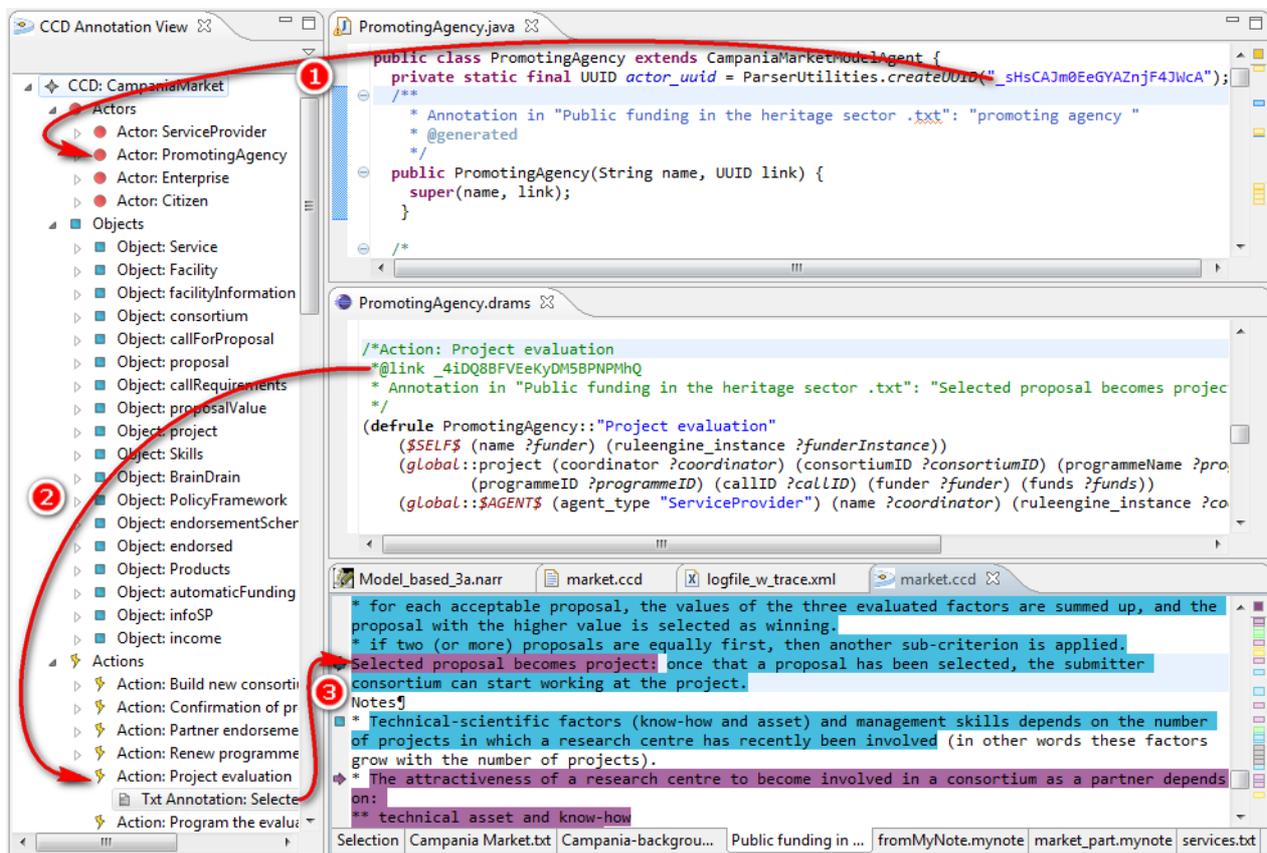


Figure 6. Fragments of CCD and programming code, which link a rule with CCD action and text phrase

6.11 The model is able to simulate agents' behaviour over a time period of many years, starting from the situation depicted in the CCD. The simulation runs generated simulation output files, so that policy modellers could produce model-based scenarios for stakeholders. This phase is explained in the next section.

Simulation of policy models and generation of model-based scenarios (Phase 5)

Objectives

- To run simulations on formal policy models
- To provide input for refining formal policy models
- To create model-based scenarios in order to expose simulation results to stakeholders and policy owners

Approach

- 7.1 Running simulations on a policy model produces output in the form of print statements and log files. They describe sequences of events including causes, decisions and the outcomes that emerge during simulation experiments with the model. Under a simulation run initialisation, the policy modellers have to define the number of time steps and different agents 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.
- 7.2 With the help of the scenario analysis tool, the policy modellers transfer the formal modelling results into a narrative model-based scenario storing the provenance of simulation results through the conceptual model. Visualisation of simulation results is needed to demonstrate how operation works, and which results are generated and derivable from current scenario descriptions. Visualisation is further essential to provide simulation results to different users such as stakeholders or policy owners.
- 7.3 The visualisations significantly facilitate feedback and interaction with the stakeholders and policy owners or domain experts. Such model-based scenarios and graphical charts provide a common and generally understandable format for the presentation of simulation outcomes.

Inputs

- 7.4 The inputs of the *Simulation of policy models and generation of model-based scenarios* phase are policy simulation models developed in the previous phase.

Steps

- 7.5 The steps of the *Simulation of policy models and generation of model-based scenarios* phase are:

1. The policy modeller repeatedly executes the simulation model. DRAMA allows the modification and testing of rules on the fly. Print statements with variables to be instantiated can be included anywhere in a rule. The modellers can direct these print statements to any console windows that have been defined by them, or to log files with different data formats (TXT, CSV, and XML). Some statements are used for debugging in the usual way. Others are used to report specific actions by individual agents and the reasons for them.
2. Once the policy modeller has completed a simulation, the policy modeller and/or analyst develop a natural-language narrative from textual logs and graphs and visualisations from numerical outputs. The passages in the narrative and the visualisations are each annotated with links to one or more textual and/or numerical outputs from the simulation.
3. In general, policy modellers will increasingly elaborate a sequence of simulations (and perhaps complicated) simulation models. Having analysed and annotated the textual and numerical outputs from the simulation, the policy modeller exports the model-based scenario(s) to the CSE tool with support of the Scenario Analysis Tool.

Outputs

- 7.6 The outputs of the *Simulation of policy models and generation of model-based scenarios* phase are

- Print statements and log files describing sequences of events including decisions and the outcomes that emerge during simulation experiments with the model
- Model-based scenarios i.e. narrative scenario texts and visualisations of numerical data.

An example

7.7 Figure 7 visualises how provenance of simulation outputs is ensured during this phase. It shows on the figure's upper part, the "Project evaluation" rule in the "PromotingAgency" agents DRAMS file. Below this, it shows a log file, which has been produced during a simulation run. Each entry in the log file presents a task in a time step (tick) and holds information on the causes of this task. The latter is done via the UUID of fact templates and rules (see the arrow labelled with (1)), which caused the result described in the info tag (<info> ... </info>). Tasks result in a print statement as visualised with the arrow labelled with (2). Print statements inform the policy modeller in a textual form on the outcomes of the simulation run.

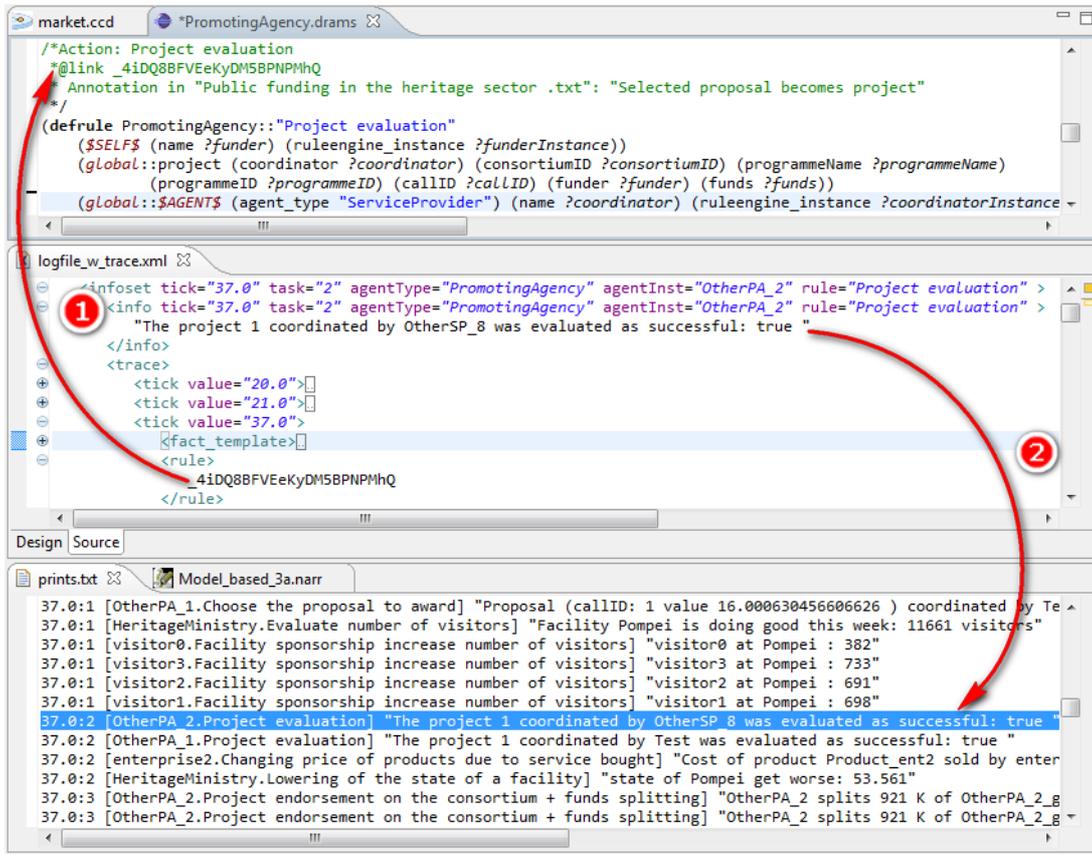


Figure 7. Extract of a simulation output of the Campania case model and links to DRAMS code

7.8 From the simulation results in log files and print statements, policy modellers used the simulation analysis tool to produce three model-based scenarios for the Campania pilot by aggregating simulation outputs from three different simulation runs. The third model-based scenario is shown in Figure 8 to demonstrate the use of this tool. The policy modeller annotated the marked text phrases with entries in the log file to document their provenance.

In this simulation the financial outcomes of research centres operating in Campania and the outcomes of other research centres outside the region are presented.

At the beginning of the simulation there are **mainly automatic funds** decreasing over time. During the simulation regional research centres apply for public grants and intends to sell services on the market.

Over time few bids fant are sy awarded grants are successful, but on the other hand **Regional research centres are able to sell successfully services on the market and their turnover increases even though automatic funding decrease**.

Compared to several research centres outside Campania, research centres operating in Campania become self-sustainable over time and do not rely on automatic funding for their survival, but rely mainly on market revenues and competitive bids. On the other hand research centres outside Campania are not equally able to both generate incomes from the market and attract grants.

From the perspective of the Regional authority, in this simulation the Regional competitiveness (measured as the variation over time of income generated by Campania research centres vs. the total income of all research centres) **increases**.

Figure 8. Example for a model-based scenario, with text annotations highlighted

7.9 The simulation analysis tool stores model-based scenarios with a *.narr file (narr for narrative). This tool supports the policy modeller by linking text phrases to the original entries in the log file (see the arrow labelled with (1) in Figure 9). Via the UUIDs of the relevant fact templates and rules, the log entries are already linked with the CCD (see arrows from log file to the CCD in Figure 9) and via annotations in the CCD to the original text phrases as shown above in Figure 2 and Figure 3.

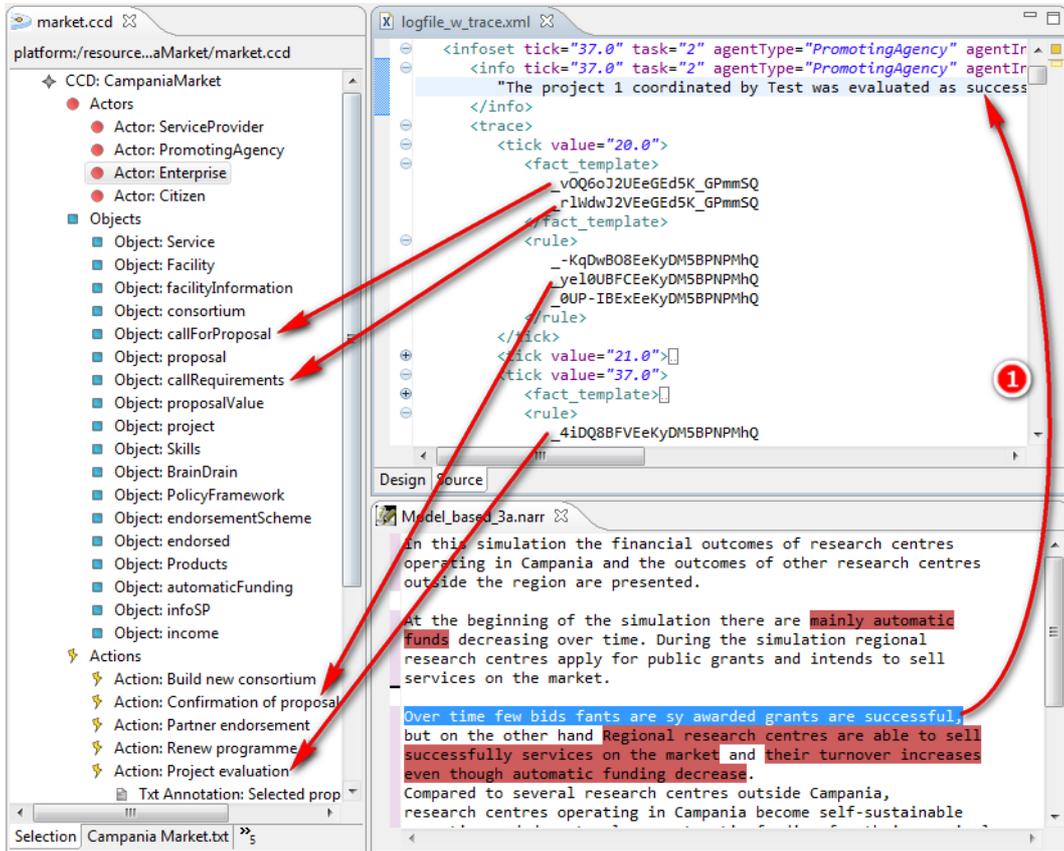
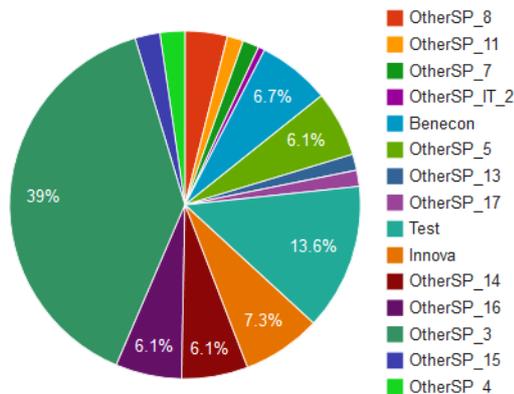


Figure 9. Model-based scenario linking to entries of the log file and to CCD

7.10 Figure 10 presents the model-based scenario in the form of a screenshot from the CSE tool. In this tool, stakeholders (domain experts, decision makers, participants) can view all model-based scenarios. The grey text parts in the *Model-based Scenario 3 Campania* link the relevant phrases of initial scenarios or background documents as described above (see the arrow to the window *Annotations* on the right visualising the first annotation in the text on the left). So the provenance of information presented in the model-based scenarios is made transparent for stakeholders. Charts, automatically generated through the tool from simulation logs (if relevant numerical data is available), are further means to visualise simulation results to stakeholders.

In this simulation the financial outcomes of research centres operating in Campania and the outcomes of other research centres outside the region are presented. At the beginning of the simulation there are mainly automatic funds decreasing over time. During the simulation regional research centres apply for public grants and intends to sell services on the market.

Incomes percentage year 1



Over time few bids for publicly awarded grants are successful, but on the other hand Regional research centres are able to sell successfully services on the market and their turnover increases even though automatic funding decrease. Compared to several research centres outside Campania, research centres operating in Campania become self-sustainable over time and do not rely on automatic funding for their survival, but rely mainly on market revenues and competitive bids. On the other hand research centres outside Campania are not equally able to both generate incomes from the market and attract grants. From the perspective of the Regional authority, in this simulation the Regional competitiveness (measured as the variation over time of income generated by Campania research centres vs. the total income of all research centres) increases.

Annotations

org.ocopomo.annotation.TxtAnnotation
 Dependence on public funds for innovation
 org.ocopomo.annotation.TxtAnnotation
 Selected proposal becomes project
 org.ocopomo.annotation.TxtAnnotation
 Proposals must fulfill all of the eligibility criteria if they are retained for evaluation. These criteria are rigorously defined. The following eligibility criteria apply to all proposals under a call. * Receipt of proposal by the Promoting Institution before the deadline date and time established in the applicable * Minimum conditions (such as number of participants), as referred to in the call for proposals. * Sometimes further requirements, e.g. a constraint on nationality of the partners. If it becomes clear before, after the evaluation phase that one or more of the eligibility criteria have not been fulfilled, the proposal is declared ineligible by the Commission, and is withdrawn from examination
 org.ocopomo.annotation.TxtAnnotation
 once that a proposal has been selected, the submitting consortium can start working at the project.
 org.ocopomo.annotation.TxtAnnotation
 try to build some proposals
 org.ocopomo.annotation.TxtAnnotation
 Partners' experience in projects development is evaluated
 org.ocopomo.annotation.TxtAnnotation
 Each call for proposal is characterized by a set of relevant features:
 org.ocopomo.annotation.TxtAnnotation
 The European Commission selects the winning proposal will be funded as follows: * each proposal with at least a value lower than 3 OR with the sum of the three greater-or-equal than 10 will be refused (unacceptable proposal) * for each acceptable proposal, the values evaluated factors are summed up, and the proposal with the higher value is selected as winning. * if two (or more) are equally first, then another sub-criterion is applied
 org.ocopomo.annotation.TxtAnnotation
 consortium building process
 org.ocopomo.annotation.TxtAnnotation
 The average level of the partners of the consortium reflects the overall project level.
 tab.viewwcd

Figure 10. Model-based scenario as presented to stakeholders – screenshot from the CSE tool

Reflecting benefits, limitations and transferability of the approach

- 8.1 The combined advantage of the evidence-based approach with the conceptual model is that it enables stakeholders to understand the reasons why the model produces what it does and, at the same time, to evaluate the plausibility of those outputs. In difference to the approaches presented in section two on related work that are supporting model-driven development of agent-based models, the following key value-added of the approach can be highlighted:
- The CCD meta-model is less complex because it does not contain elements for modelling environments, agent groups etc. Instead the two concepts Agent and Object can be used. An environment could e.g. be modelled as an object (if necessary) in which agents can be placed. A group of agents could be modelled as a single agent type.
 - The CCD meta-model and tool support traceability of modelling decisions.
 - The policy development process foresees the involvement of wider stakeholder groups in providing the evidence inputs as well as in validating conceptual models and simulation outcomes. Traces to provenance of information help stakeholders to better understand results of simulations and interrelations of aspects of the policy case.
- 8.2 In this section, we first discuss the benefits for different stakeholder groups; then, potential limitations of the approach are reflected, and finally the transformation of the solution to other domains is investigated. The insights regarding benefits and limitations of the solution have been gathered through an evaluation of the OCOPOMO process and its software toolbox along the project, where different stakeholders, policy modellers and policy analysts have been interrogated. The evaluation was based on an evaluation framework, which emerged from the consideration of analysing the process and the toolbox from views of different actors. For each reasonable combination of dimensions, it was considered what should be evaluated and what the key evaluation criteria were. On this basis, expected results^[20] were defined and the evaluation methods (questionnaires and interviews) were selected. Questions were then developed for different groups in the process: stakeholders, policy modellers, policy analysts and facilitators. The results of the evaluation are available in a technical report of the project^[21].
- Benefits for different stakeholder groups
- 8.3 The subsequent explanations provide insights into the added value and benefits of the policy development process and software toolbox from the viewpoint of distinct user groups: Policy makers and stakeholders on the one hand, as well as policy analysts and policy modellers on the other hand.
- Policy makers and stakeholders*
- 8.4 Of utmost importance for decision makers is to be able to clearly understand and interpret the outcomes of simulation models. The framework answers this need by giving 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 informing 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 and stakeholders 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. Further, the framework 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.
- 8.5 As policy makers and stakeholders may not necessarily engage in the whole process, 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.
- 8.6 In particular, the framework supports the current request for open government with the principles of collaboration, openness and participation to a large extent through the policy development process and its respective software toolbox. The iterative process enables stakeholders to comment simulation results and to request revisions if stakeholders' scenarios have not been mapped properly. Or the simulation outcomes produce insights that lead to revisions in the views and considerations of stakeholders.

Policy analysts and policy modellers

- 8.7 For policy analysts and policy modellers, a main added value and benefit of the CCD tool 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. It is also possible for policy modellers to discuss or elaborate conceptual policy models with stakeholders by using different visualisations.
- 8.8 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 and Java. All these elements contain traceability information. Thus the CCD tool helps to structure and speed the development of the simulation model.
- 8.9 A key added value for the stakeholders and the policy modellers is the traceability concept, 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). Therewith the overall framework supports the communication function of simulation models corresponding to Waldherr and Wijermans (2013) and trust in simulation results corresponding to Pignotti et al. (2013). And not to underestimate, traceability enhances the quality of simulation models by basing modelling decisions on evidences and by visualising these through the links maintained along the whole policy development process of OCOPOMO.

Potential limitations of the policy development framework

- 8.10 This section outlines potential limitations of the process and the software toolbox along the activities in the policy development process that were presented in this paper: CCD annotation and modelling, CCD2DRAMS transformation, and DRAMS simulation and modelling.

CCD annotation and modelling

- 8.11 The CCD meta-model gives the conceptual modeller (policy analyst) freedom in the design of a conceptual model. 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 modellers. CCD models for different types and domains (meant as specifications for distinct policy domains) should be developed to support conceptual modellers in developing CCDs to be more comparable.
- 8.12 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 logic.
- 8.13 CCDs generated in the policy development process can be large in the sense of having many elements (vertices in graph theory (Bondy & Murty 1976)) 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 by other means, features are required to optimise the management and overview of elements in the CCD.
- 8.14 The current CCD model representations are complex for non-experts. Improving the usability of features and functions is future work of our research. Support for policy modellers may be further optimised in order to speed up the implementation of a simulation model.

CCD2DRAMS transformation

- 8.15 At present, the CCD2DRAMS tool supports transformation of CCD models to the DRAMS simulation environment. While the value-add of conceptual modelling of the policy case with provenance information is high, the transformation needs yet to be extended to connect different agent-based modelling environments with the CCD tool. This is another research work to be focused on in the near future.
- 8.16 The experience from policy simulation modelling was that the development process is forward and backward chaining 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 produce 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 in speeding up the implementation of a simulation model. Transformation from DRAMS to CCD would allow that changes in the simulation model by a policy modeller be reflected automatically 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 selection of a corresponding transformation engine and assembling a DRAMS meta-model.
- 8.17 Another supportive measure would be to see a CCD model as a higher abstraction level of a DRAMS simulation model, and to reflect this understanding in the meta-models. To achieve this, 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 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 two model types, which would indeed represent the same facts, yet in different description languages and at different abstraction levels. Such a solution would also resolve the backward consistency of models (from DRAMS to CCD), which is currently a manual step to be performed.

DRAMS simulation modelling

- 8.18 Although DRAMS allows developing, debugging, running and analysing simulation models, a number of measures would make performing these steps more efficient and more comfortable. One important point in this regard is the full integration of DRAMS in the Eclipse IDE. Currently, DRAMS code editing is supported by an Eclipse feature. Analysing and debugging models, testing of code fragments and supervising simulation runs require GUI components separated from Eclipse. 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. This can be amended by an improved rule scheduling algorithm in the DRAMS core 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 implementing a fact disposal policy, which intelligently cleans fact bases by removing unused or obsolete facts. 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.

Transferring the OCOPOMO solution to other application domains

- 8.19 The policy development process and the software toolbox of OCOPOMO have a modular structure. That means, particular stages, aspects and tools can be used separately without having to employ the whole software toolbox. Already the three different policy cases developed (renewable energy, knowledge transfer in economically disadvantaged regions, housing policy) show the applicability of the solution in different policy contexts. Even along the implementation of these pilot cases, variations of methods have been applied along the policy development, hence evidencing the flexibility of the framework.
- 8.20 The process of policy development relies on the CCD tool and its extensions. The use of the OCOPOMO tools and methods to other domains is easily possible. The CCD meta-model is abstract enough to accommodate different peculiarities of policy domains. The approach was e.g. applied in combination with qualitative text analysis using MaxQDA^[22] in the GLODERS Project^[23] to model the collapse of a criminal network (Neumann & Lotzmann 2014; Neumann 2015). The CCD tool can also be configured to be usable with any other meta-model formulated in Ecore^[24]. However, if 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.
- 8.21 In OCOPOMO, the programming of simulation models is a process of agent-based modelling and in particular declarative rule-based agent modelling using DRAMS. Employing the software toolbox 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 possible to apply other modelling tools supporting agent-based modelling. As already mentioned before, the 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 transferring the CCD concepts into the respective simulation environment (e.g. CCD2Repast, CCD2NetLogo^[25], etc.). We would not argue that the process or the software toolbox would support a different modelling method such as micro-modelling or econometric modelling because of the enormous difference in technique and goals. In consequence, the policy modelling process and its respective software toolbox can be transferred to policy domains with the aim of applying agent-based modelling in simulating social processes. Hence, this simulation modelling paradigm cannot easily be replaced.
- 8.22 Since DRAMS is designed as a self-contained Java software framework implementing a declarative rule engine, it can be used in a broad range of applications. Although it would be possible to use DRAMS as a rule engine outside 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 based on the RETE algorithm^[26] which are optimised for static databases.
- 8.23 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 Symphony or Mason^[27] can be realised with little effort. Special implementations of the agent interface might be needed for certain simulation tools, yet these are usually more dependent on the objective of the model.
- 8.24 The integrated plugin interface provides a large potential for opening DRAMS towards other application domains. 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.
- 8.25 As with the other tools, DRAMS provides its full potential with the traceability feature when deployed with the CCD and CCD2DRAMS 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 competing with CCD tool and CCD2DRAMS transformation is implemented.

Acknowledgements

The authors acknowledge the contributions of and express their gratitude to the OCOPOMO project partners for the numerous discussions on the framework. The CCD tool, CCD2DRAMS tool, and DRAMS, CSE tool and Scenario Analysis Tool were developed in the OCOPOMO project; CCD tool and CCD2DRAMS tool by the OCOPOMO team @ University of Koblenz-Landau (UKL); DRAMS by project teams @ UKL,

Notes

- ¹Open Collaboration for Policy Modeling, <http://www.ocopomo.eu/> (last access 2nd October 2014)
- ²By policy, we refer to strategic areas of complex decision-making with various stakeholders having potentially diverging interests. An extensive discussion of the understanding of policy and of public policy can be found in Anderson (2014).
- ³European Commission (2001), European Governance, A white paper, COM(2001) 428 final, Brussels, <http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52001DC0428&rid=2>. Archived at: <http://www.webcitation.org/6XUHqB5Zh>
- ⁴<http://repast.sourceforge.net/> (last access 2nd October 2014)
- ⁵This structure follows the presentation of the Architecture Development Method of The Open Group Architecture Framework (TOGAF®) in TOGAF Version 9 (2009). TOGAF Series. The Open Group. Van Haren Publishing, Zaltbommel, www.vanharen.net.
- ⁶Unified Modelling Language. An introduction of the UML modelling concept is available in Holt (2004).
- ⁷<http://download.eclipse.org/modeling/emf/emf/javadoc/2.9.0/org/eclipse/emf/ecore/package-summary.html#details> (last access 2nd October 2014)
- ⁸<http://eclipse.org/amp> (last access 2nd October 2014)
- ⁹http://repast.sourceforge.net/repast_simphony.php (last access 2nd October 2014)
- ¹⁰CSE tool uses and extends Alfresco Community Edition (<http://www.alfresco.com/products/community>, last access 6th October, 2014).
- ¹¹See <http://eclipse.org/> (last access 2nd October 2014)
- ¹²<http://www.ocopomo.eu/results/software-and-models/software-and-model-artefacts> (last access 3rd October 2014)
- ¹³The models (CCD models, policy models, model-based scenarios) for all three pilot cases are available at <http://www.ocopomo.eu/results/software-and-models/software-and-model-artefacts/eclipse-based-tools-and-simulation-models> (last access 3rd October 2014).
- ¹⁴Universally Unique Identifiers – see <http://pubs.opengroup.org/onlinepubs/9629399/apdx.htm> (last access 2nd October 2014) for more details
- ¹⁵See system documentation of the CCD tool available at http://www.ocopomo.eu/results/public-deliverables/system-documentation/d4.2-sd-2-system-documentation-of-ccd-tool-and-annotation-extensions.pdf/at_download/file and CCD tool user manual available at http://www.ocopomo.eu/results/public-deliverables/user-manuals/d4.2-b-user-manual-on-ccd-tools.pdf/at_download/file (last access 30th September 2014)
- ¹⁶See system documentation of DRAMS available at http://www.ocopomo.eu/results/public-deliverables/system-documentation/d4.2-sd-3-system-documentation-of-drams-tools.pdf/at_download/file, and DRAMS user manual available at http://www.ocopomo.eu/results/public-deliverables/user-manuals/d4.2-c-user-manual-on-policy-modelling-and-simulation-tools.pdf/at_download/file (last access 30th September 2014)
- ¹⁷A fact template is the specification of "the number and types of data slots for a specific fact" (Lotzmann & Meyer 2011).
- ¹⁸A fact consists of "a number of definable data slots and some administrative information (time of creation, entity that created the fact, durability)" (Lotzmann & Meyer 2011).
- ¹⁹A rule contains "a condition part (called left-hand side) and an action part (called right-hand side)" (Lotzmann & Meyer 2011).
- ²⁰An example of an expected metric for evaluating the usability of the CSE tool was: "The general usability and the usability and usefulness of features is recognised as good."
- ²¹<http://www.ocopomo.eu/results/public-deliverables/d-7.1-evaluation-strategy-trial-design-and-evaluation-results> (last access 2nd October 2014)
- ²²MaxQDA is a qualitative data analysis software (<http://www.maxqda.com/>, last access 2nd October 2014)
- ²³<http://www.golders.eu> (last access 2nd October 2014)
- ²⁴Steinberg et al. (2008) provide a good overview of meta-modelling in Ecore.
- ²⁵<http://ccl.northwestern.edu/netlogo/> (last access 5th October 2014) and Tisue and Wilensky (2004).
- ²⁶The RETE algorithm is e.g. detailed in Forgy (1982).
- ²⁷<http://cs.gmu.edu/~eclab/projects/mason/> (last access 2nd October 2014)

References

- ANDERSON, J. E. (2014). *Public policymaking: An introduction* (8th ed.). Cengage Learning.
- BAUER, B., Müller, J. P., & Odell, J. (2001). Agent UML: A Formalism for Specifying Multiagent Software Systems. *International Journal of Software Engineering and Knowledge Engineering*, 11(3), 91–103. [doi:10.1142/S0218194001000517]
- BECU, N., Neef, A., Schreinemachers, P., & Sangkapitux, C. (2008). Participatory computer simulation to support collective decision-making: Potential and limits of stakeholder involvement. *Land Use Policy*, 25(4), 498–509. [doi:10.1016/j.landusepol.2007.11.002]
- BERSINI, H. (2012). UML for ABM. *Journal of Artificial Societies and Social Simulation*, 15(1) 9: <http://jasss.soc.surrey.ac.uk/15/1/9.html>
- BOMMEL, P., Dieguez, F., Bartaburu, D., Duarte, E., Montes, E., Pereira Machin, M., Corral, J., et al. (2014). A Further Step Towards Participatory Modelling. Fostering Stakeholder Involvement in Designing Models by Using Executable UML. *Journal of Artificial Societies and Social Simulation*, 17(1) 6: <http://jasss.soc.surrey.ac.uk/17/1/6.html>
- BONDY, J. A., & Murty, U. S. R. (1976). *Graph theory with applications*. Macmillan London. [doi:10.1007/978-1-349-03521-2]
- DORAN, J., & Gilbert, N. (1994). Simulating societies: an introduction. In N. Gilbert & J. Doron (Eds.), *Simulating societies - The computer simulation of social phenomena*. UCL Press.
- DOWNING, T. E., Moss, S., & Pahl-Wostl, C. (2001). Understanding climate policy using participatory agent-based social simulation. *Multi-Agent-Based Simulation* (pp. 198–213). Springer.
- FORGY, C. L. (1982). Rete: A fast algorithm for the many pattern/match problem. *Artificial Intelligence*, 19(1), 17–37. [doi:10.1016/0004-3702(82)90020-0]
- GHORBANI, A., Bots, P., Dignum, V., & Dijkema, G. (2013). MAIA: a Framework for Developing Agent-Based Social Simulations. *Journal of Artificial Societies and Social Simulation*, 16(2), 9: <http://jasss.soc.surrey.ac.uk/16/2/9.html>
- GILBERT, N., & Troitzsch, K. G. (2005). *Simulation for the Social Scientist* (2nd ed.). Open University Press.
- HASSAN, S., Fuentes-Fernández, R., Galán, J. M., López-Paredes, A., & Pavón, J. (2009). Reducing the Modeling Gap: On the use of metamodels in agent-based simulation. *6th Conference of the European Social Simulation Association (ESSA 2009)* (pp. 1–13).
- HOLT, J. (2004). *UML for Systems Engineering: watching the wheels* IET Professional Applications of Computing (2nd ed.). Institution of Engineering and Technology. [doi:10.1049/pbpc004e]
- LOTZMANN, U., & Meyer, R. (2011). DRAMS - a declarative rule-based agent modelling system. In T. Burczynski, J. Kłodziej, A. Byrski, & M. Carvalho (Eds.), *Proceedings of 25th European Conference on Modelling and Simulation*. European Council for Modelling and Simulation. [doi:10.7148/2011-0077-0083]
- MELLOR, S. J., Clark, T., & Futagami, T. (2003). Model-driven development: guest editors' introduction. *IEEE software*, 20(5), 14–18. [doi:10.1109/MS.2003.1231145]
- MOSS, S. (2002). Policy analysis from first principles. *Proceedings of the National Academy of Sciences*, 99(suppl 3), 7267–7274. [doi:10.1073/pnas.092080699]
- NEUMANN, M. (2015). Grounded Simulation. *Journal of Artificial Societies and Social Simulation*, 18(1) 9: <http://jasss.soc.surrey.ac.uk/18/1/9.html>

- NEUMANN, M., & Lotzmann, U. (2014). Modelling The Collapse Of A Criminal Network. In F. Squazzoni, F. Baronio, C. Archetti, & M. Castellani (Eds.), *28th European Conference on Modelling and Simulation, ECMS 2014, Brescia, Italy, May 27–30, 2014* (pp. 765–771). European Council for Modeling and Simulation. [doi:10.7148/2014-0765]
- OKUYAMA, F. Y., Bordini, R. H., & Rocha Costa, A. C. da. (2005). ELMS: An environment description language for multi-agent simulation. *Environments for Multi-Agent Systems*, 91–108. [doi:10.1007/978-3-540-32259-7_5]
- PAVÓN, J., Gómez-Sanz, J., & Fuentes, R. (2006). Model Driven Development of Multi-Agent Systems. In A. Rensink & J. Warmer (Eds.), *Model Driven Architecture Foundations and Applications*, Lecture Notes in Computer Science (Vol. 4066, pp. 284–298). Springer Berlin / Heidelberg. [doi:10.1007/11787044_22]
- PIGNOTTI, E., Polhill, G., & Edwards, P. (2013). Using Provenance to Analyse Agent-based Simulations. *Proceedings of the Joint EDBT/CDT 2013 Workshops* EDBT '13 (pp. 319–322). Genoa, Italy: ACM.
- ROBINSON, D. T., Brown, D. G., Parker, D. C., Schreinemachers, P., Janssen, M. A., Huigen, M., Wittmer, H., et al. (2007). Comparison of empirical methods for building agent-based models in land use science. *Journal of Land Use Science*, 2(1), 31–55. [doi:10.1080/17474230701201349]
- SCHERER, S., Wimmer, M. A., & Markisic, S. (2013). Bridging narrative scenario texts and formal policy modeling through conceptual policy modeling. *Artificial Intelligence and Law*, 21(4), 455–484. [doi:10.1007/s10506-013-9142-2]
- STEINBERG, D., Budinsky, F., Merks, E., & Paternostro, M. (2008). *EMF: eclipse modeling framework*. (E. Gamma, L. Nackman, & J. Wiegand, Eds.)the eclipse series (2nd ed.). Addison-Wesley.
- TISUE, S., & Wilensky, U. (2004). NetLogo: Design and implementation of a multi-agent modeling environment. In C. M. Macal, D. Sallach, & M. J. North (Eds.), *Proceedings of the Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence, Chicago, IL* (pp. 161–184).
- VALBUENA, D., Verburg, P. H., Veldkamp, A., Bregt, A. K., & Ligtenberg, A. (2010). Effects of farmers' decisions on the landscape structure of a Dutch rural region: An agent-based approach. *Landscape and Urban Planning*, 97(2), 98–110. [doi:10.1016/j.landurbplan.2010.05.001]
- VÓLTER, M., Stahl, T., Bettin, J., Haase, A., & Helsen, S. (2013). *Model-driven software development: technology, engineering, management*. John Wiley & Sons.
- WALDHERR, A., & Wijermans, N. (2013). Communicating Social Simulation Models to Sceptical Minds. *Journal of Artificial Societies and Social Simulation*, 16(4), 13: <http://jasss.soc.surrey.ac.uk/16/4/13.html>
- WIMMER, M. A., Scherer, S., Moss, S., & Bicking, M. (2012). Method and tools to support stakeholder engagement in policy development: the OCOPOMO project. *International Journal of Electronic Government Research*, 8(3), 98–119. [doi:10.4018/ijegr.2012070106]