Design and Integration of ICT Tools for OCOPOMO Policy Modelling Approach


* Technical University of Kosice, Faculty of Economics, Kosice, Slovakia
** Intersoft, a.s., Kosice, Slovakia
*** Technical University of Kosice, Department of Cybernetics and Artificial Intelligence, Kosice, Slovakia

peter.bednar@tuke.sk, peter.butka@tuke.sk, karol.furdik@intersoft.sk, marian.mach@tuke.sk, psmatana@gmail.com

Abstract—This paper describes design and integration of several tools for ICT-based support of knowledge intensive activities within the process of collaborative policy modelling proposed by the project OCOPOMO. This process combines collaborative creation and analysis of narrative scenarios with agent-based simulation models. The policy model for a given domain is created iteratively using cooperation of several stakeholder groups (decision makers, analysts, companies, civic societies, and the general public). Integration of scenario generation module, annotation module, model preparing phases and simulation tool (with the visualization tools) makes the whole OCOPOMO platform. The main focus of the paper is on the presentation of design and integration aspects of the architecture. Therefore, we describe software components, functionality, as well as some integration details.

I. INTRODUCTION

Information and communication technologies (ICT) are intensively used in the area of e-Government in order to support innovation within the paradigm of knowledge-driven economy, as well as to help governance in support of an active and qualified engagement of citizens in their decision-making activities. One of the main functions of advanced e-Government solutions is to provide effective co-operation and information exchange between involved parties, governmental bodies, citizens and businesses, during the policy initiation, development, implementation, monitoring and evaluation. Involvement of a wider public in the process of policy creation and evolution is important and can be of special value.

Therefore, according also to FP7 ICT Programme initiative related to policy modelling, any toolkit should be able to perform simulations integrating all possible variables, parameters, relations, and scenarios necessary to forecast potential outcomes and impacts of proposed policy measures [1]. Hence future trends in e-Government research and applications should extend the paradigm of collaborative policy creation. This process supported by statistical methods and agent-based social simulation [2] can provide an environment for achieving goals of advanced e-Government applications. The area of collaborative policy modelling is quite diverse, multidisciplinary and complex, including also Web 2.0/Web 3.0 concepts of social semantic web. The proposed solutions are usually built on already known and proven technologies, which typically include rule-based or agent-based modelling platforms [3], tools and interfaces for visualized social simulations, knowledge representation models, and web-based collaborative frameworks. The innovative aspects and challenges lay in an integration and functional interconnection of these technological frameworks into one comprehensive platform.

The design of a software platform and methodology providing an environment for modelling policies in a collaborative manner is in the focus of the European R&D project OCOPOMO (Open COLlaboration for Policy MOdelling). OCOPOMO is co-funded by the European Commission under the 7th Framework Programme. It is coordinated by the University of Koblenz-Landau, the project consortium consists of 10 partners from 5 European countries. The 3-year project (started in 2010) is tested on pilot applications in Italy, Slovakia, and UK.

The remaining part of the paper is structured as follows: Section 2 describes relevant approaches and related projects to OCOPOMO. Section 3 provides description of tools (modules) within the OCOPOMO platform applied to policy modelling process. Section 4 presents some integration details and description of architecture from this point of view, which completes the picture of OCOPOMO platform in its usage of ICT tools for support of collaborative policy modelling processes.

II. RELEVANT APPROACHES AND RELATED PROJECTS

The collaborative policy modelling applied in the field of e-Government can be seen as an interdisciplinary concept, supported by several mutually interrelated topics. It includes different policy modelling approaches, scenario-based development of foresights, and collaborative environments for e-Participation and e-Governance applications. The policy modelling is particularly applied on general modelling and analysis of economy at the macro level. A more recent approach is based on micro theoretical relationships. The analysis and modelling is performed in a respect to the behavior of ‘representative agents’ – independent actors that are taking decisions to maximize their utilities over an infinite time horizon. These two approaches to policy modelling are complementary in approaches to handling and processing the data. While the first approach is top-down, the second approach employs bottom-up approach with data gathered from observations of individual agents. Each of the interaction types can be modeled by a set of rules that may emerge from an average behavior of agents. Simple rules of social influence among individuals may
lead to the emergence of complex patterns of public opinion [2]. In a policy modelling system applied for e-Governance, it is especially important to capture these complex patterns on a more global level and recognize the tendencies that may lead to strategic planning.

The technique of narrative scenarios is an advanced and commonly taken method for representing a definition and exploration of futures in a domain. The scenario-building procedure aims at generating different perspectives of the future to gain more insight into possible opportunities and threats. This technique allows better and more effective exploration of alternative trajectories of a certain domain beyond short-term forecasting [4]. Agent-based approach to the policy modelling is highly dependent on a possibility of open and free information exchange between participating actors. Individual goals, interests and preferences [5] are confronted with actual status of the policy model, expressed in a set of relevant narrative scenarios. Active participation of an actor in the process of policy modelling can lead to a self-organization with consensus-based policy model as its collaborative result [6].

The interaction between policy modelling actors requires a proper suite of collaborative tools enabling presentation of stakeholders’ ideas, discussions and negotiations between actors, voting on open issues and decision making. In this sense, e-Participation is used for involvement of citizens in their own governance [7], whereas the whole policy modelling solution follows the principles of e-Governance to facilitate effective decision making and improvement of public policies [8].

The OCOPOMO project aims at integration of relevant specialized ICT platforms into a single e-Governance toolkit that will enable a collaborative policy modelling for decision support of governmental representatives. From agent-based policy modelling tools, built on the platform of multi-agent systems (MAS), probably the most popular open-source software platforms for developing MAS are JADE and Repast. A survey and evaluation of available agent-based simulation tools can be found in [9]. Content management systems can be used for maintenance of shared policy models, narrative scenarios and supporting documents, with the examples like Alfresco, Apache Lenya, Drupal, Jackrabbit, Plone, etc. Collaboration platforms support communication between participants and collaborative work upon shared artifacts, examples of some solutions are Alfresco Share, Open-Xchange, eGroupWare, OpenGroupware.org, etc.

Several projects related to governance and policy modelling are implemented within FP7 ICT Programme:

- CROSSOVER (http://www.crossover-project.eu/) - a support action project that aims at recognizing trends in emerging area of ICT supportive tools for policy modelling and e-Government.
- COCKPIT (http://www.cockpit-project.eu/) - to define a new Governance model which actively engages and empowers citizens in public service delivery decision making process.
- PADGETS (http://www.padgets.eu) - aims at bringing together two well established domains, the mash-up architectural and developmental approach of Web 2.0, for creating web applications (gadgets) and modelling/simulation methodologies in analysis of complex system behavior.
- IMPACT (http://www.policy-impact.eu/) - aims at computational models of argumentation about policy issues and produces new data-mining techniques, advanced interfaces and visualizations.
- +SPACES (http://www.positivespaces.eu/) - aims at policy making simulation in virtual spaces using existing virtual worlds as the basics for modelling real world behavior.
- UBIPOPOL (http://www.ubipol.eu) - this project develops ubiquitous platform that allows citizens to be involved in policy making processes regardless their current locations and time using context aware knowledge provision.
- WEGOV (http://www.wegov-project.eu/) - aims at development of a toolset that allows full advantage to be taken of a wide range of existing and well established social networking sites (Facebook, Twitter, etc.) to engage citizens in two-way dialogs as a part of governance and policy making processes.
- e-Policy (http://www.epolicy-project.eu/node) - aims at support of policy makers in their decision process across a multi-disciplinary effort aimed at engineering the policy making life-cycle according to individual aspects, extracted from the web in order to find social impacts through opinion mining techniques. OCOPOMO is represented there in the Advisory and Dissemination Board by the coordinator from University of Koblenz (in order to achieve cross-project relationships and cooperation).
- FuturICT (http://www.futurict.ethz.ch/FuturIcT) is a project developed with the EU FET Flagship initiative (instrument for huge projects for tackling global challenges in the 21st century). The project overview tackles some similar issues as OCOPOMO, especially agent-based modelling used for preparing long-term strategies. The project goals are rather ambitious, i.e., to implement Living Earth Simulator expected to run simulations with 10 billions of agents. One of the OCOPOMO partners, University of Warsaw, co-organized last large conference of the FuturICT project partners in Warsaw (Nov 2011).

III. OCOPOMO PLATFORM - SYSTEM FOR SUPPORT OF COLLABORATIVE POLICY MODELLING PROCESSES

The process of policy modelling specifically addressed by the OCOPOMO analysis is based on narrative scenarios and related formal policy models that are constructed and modified collaboratively, by various groups of involved persons that use tools for information exchange and communication. Envisioned functionality of the platform was combined from different sources, like initial proposal of the project, state of the art analysis related to different areas important for such system, analysis of the available specific processes (from user partners and stakeholders), theoretical background of research areas and ideas for improvement of the current processes.

In general, a proposed platform should support the iterative and collaborative creation of policy models and respective scenarios. To design the architecture of a software platform that is capable to provide the required functionality, the scope of such system needs to be specified as a basis for further development. The scope of
the proposed system can be described by a set of information artifacts that are exchanged in several supported phases of the policy creation process. Three basic types of information artifacts were identified:

- **Scenario** - main artifact manipulated in the process of policy development that contains a description of the policy proposal. The scenario is a narrative, unstructured or structured text of a perceived view or understanding of a policy topic under discussion. It may cover an existing world status, mental models of stakeholders or an output of future simulations. Three types of scenario are available:
  - Initial scenario - provided to stimulate policy modelling process
  - Evidence-based user generated scenario - collaboratively generated by the users, result of discussions, exchanges of opinions, etc.
  - Model generated scenario - result of simulation model transformed into human readable format (narrative text, tables, graphs, etc.)

- **Simulation model** - simplified representation of objects, phenomena, and processes in a domain of interest. Typically, the model consists of parts, interactions between the parts, and inputs of a defined quantity and nature, mostly represented by the agents, environment, and their interactions. The simulation model can be activated to produce an output by simulation runs.

- **Consistent Conceptual Description (CCD)** - is a knowledge representation schema that captures descriptions and perceptions of the users (stakeholders) in a structured, formal, and machine-readable way. CCD can be constructed by means of knowledge-based structures (e.g., ontology, rule-dependency graphs, etc.) and plays a role of an intermediary structure between scenarios and simulation models, allowing forward and backward linkage between them.

The core objective of OCOPOMO is to ensure conceptual consistency between the scenarios created by stakeholders, background information on the case, and the corresponding simulation models developed by policy modelers. For this aim the following process, consisting of the six steps, has been developed in OCOPOMO [10]:

1. Definition of an initial scenario and collection of background information
2. Development of evidence-based, stakeholder generated scenarios
3. Development of conceptual models
4. Programming of a policy model
5. Simulation and generation of model-based scenarios
6. Evaluation of evidence-based vs. model-based scenarios

The first decomposition of architecture and its design (described in [11]) mapped user requirements into functional blocks, which were developed according to the well-known architecture design methodology by Rozanski and Woods [12], and proposed a structure of three main layers: tools, system core, and data storage. The Tools layer provides components that are responsible for maintenance of particular tools within the system, together with the respective user interfaces, with managers (components) for different collaboration functions, document management, annotation, modelling and simulation. The Core layer is dedicated to processing all the data exchanged inside the system in order to support tools with specific business logic, like federated search, system-wide notification, initiation and maintenance of processes, management of user profiles, etc. The Data layer provides an infrastructure for persistent storage, management, secure access, sharing and versioning of particular content of any required type, with relational databases and other data source management.

After definition of functional components in general architecture, we applied it to the aforementioned policy modelling process and mapped functional blocks to particular steps in order to optimize the support of the OCOPOMO method, with the more precise distinction between managers and their usage. For this reason, the OCOPOMO managers were divided into the following four installation modules, according to provided functionality and deployment environments:

- Collaboration and Scenario Editing tools (CSET)
- Content Management Server/System (CMS)
- Consistent Conceptual Description Tool (CCD Tool)
- Simulation environment (SE)

Fig. 1 visualizes the steps of the OCOPOMO method described above and the modules supporting these steps. In Tab.1 we provide mapping of particular managers from functional architecture into installation modules. The main characteristics of particular modules are described bellow.

![Figure 1. Modules within OCOPOMO process (derived from [10])](image)

**Table 1. Mapping of managers to installation modules**

<table>
<thead>
<tr>
<th>Manager</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation Manager</td>
<td>CCD Tool (Annotation Tool Component)</td>
</tr>
<tr>
<td>Calendar Manager</td>
<td>CSET Calendar</td>
</tr>
<tr>
<td>Chat Manager</td>
<td>CSET Dashboard/Chat</td>
</tr>
<tr>
<td>Collaboration Space Manager</td>
<td>CSET (site management)</td>
</tr>
<tr>
<td>Concept Manager</td>
<td>CCD Tool (Annotation Tool Component)</td>
</tr>
<tr>
<td>Content Manager</td>
<td>CMS</td>
</tr>
<tr>
<td>Discussion Forums Manager</td>
<td>CSET Discussion Forum</td>
</tr>
<tr>
<td>Document Manager</td>
<td>CSET Documents and Wiki pages</td>
</tr>
<tr>
<td>Link Manager</td>
<td>CCD Tool (Annotation Tool Component)</td>
</tr>
<tr>
<td>Notification Manager</td>
<td>CMS Alfresco Content Repository (events), CMS Content and Collaboration services (activities)</td>
</tr>
</tbody>
</table>

---

- **Scenario** - main artifact manipulated in the process of policy development that contains a description of the policy proposal. The scenario is a narrative, unstructured or structured text of a perceived view or understanding of a policy topic under discussion. It may cover an existing world status, mental models of stakeholders or an output of future simulations. Three types of scenario are available:

  - Initial scenario - provided to stimulate policy modelling process
  - Evidence-based user generated scenario - collaboratively generated by the users, result of discussions, exchanges of opinions, etc.
  - Model generated scenario - result of simulation model transformed into human readable format (narrative text, tables, graphs, etc.)

- **Simulation model** - simplified representation of objects, phenomena, and processes in a domain of interest. Typically, the model consists of parts, interactions between the parts, and inputs of a defined quantity and nature, mostly represented by the agents, environment, and their interactions. The simulation model can be activated to produce an output by simulation runs.

- **Consistent Conceptual Description (CCD)** - is a knowledge representation schema that captures descriptions and perceptions of the users (stakeholders) in a structured, formal, and machine-readable way. CCD can be constructed by means of knowledge-based structures (e.g., ontology, rule-dependency graphs, etc.) and plays a role of an intermediary structure between scenarios and simulation models, allowing forward and backward linkage between them.

The core objective of OCOPOMO is to ensure conceptual consistency between the scenarios created by stakeholders, background information on the case, and the corresponding simulation models developed by policy modelers. For this aim the following process, consisting of the six steps, has been developed in OCOPOMO [10]:

1. Definition of an initial scenario and collection of background information
2. Development of evidence-based, stakeholder generated scenarios
3. Development of conceptual models
4. Programming of a policy model
5. Simulation and generation of model-based scenarios
6. Evaluation of evidence-based vs. model-based scenarios

The first decomposition of architecture and its design (described in [11]) mapped user requirements into functional blocks, which were developed according to the well-known architecture design methodology by Rozanski and Woods [12], and proposed a structure of three main layers: tools, system core, and data storage. The Tools layer provides components that are responsible for maintenance of particular tools within the system, together with the respective user interfaces, with managers (components) for different collaboration functions, document management, annotation, modelling and simulation. The Core layer is dedicated to processing all the data exchanged inside the system in order to support tools with specific business logic, like federated search, system-wide notification, initiation and maintenance of processes, management of user profiles, etc. The Data layer provides an infrastructure for persistent storage, management, secure access, sharing and versioning of particular content of any required type, with relational databases and other data source management.

After definition of functional components in general architecture, we applied it to the aforementioned policy modelling process and mapped functional blocks to particular steps in order to optimize the support of the OCOPOMO method, with the more precise distinction between managers and their usage. For this reason, the OCOPOMO managers were divided into the following four installation modules, according to provided functionality and deployment environments:

- Collaboration and Scenario Editing tools (CSET)
- Content Management Server/System (CMS)
- Consistent Conceptual Description Tool (CCD Tool)
- Simulation environment (SE)

Fig. 1 visualizes the steps of the OCOPOMO method described above and the modules supporting these steps. In Tab.1 we provide mapping of particular managers from functional architecture into installation modules. The main characteristics of particular modules are described bellow.

![Figure 1. Modules within OCOPOMO process (derived from [10])](image)

**Table 1. Mapping of managers to installation modules**

<table>
<thead>
<tr>
<th>Manager</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation Manager</td>
<td>CCD Tool (Annotation Tool Component)</td>
</tr>
<tr>
<td>Calendar Manager</td>
<td>CSET Calendar</td>
</tr>
<tr>
<td>Chat Manager</td>
<td>CSET Dashboard/Chat</td>
</tr>
<tr>
<td>Collaboration Space Manager</td>
<td>CSET (site management)</td>
</tr>
<tr>
<td>Concept Manager</td>
<td>CCD Tool (Annotation Tool Component)</td>
</tr>
<tr>
<td>Content Manager</td>
<td>CMS</td>
</tr>
<tr>
<td>Discussion Forums Manager</td>
<td>CSET Discussion Forum</td>
</tr>
<tr>
<td>Document Manager</td>
<td>CSET Documents and Wiki pages</td>
</tr>
<tr>
<td>Link Manager</td>
<td>CCD Tool (Annotation Tool Component)</td>
</tr>
<tr>
<td>Notification Manager</td>
<td>CMS Alfresco Content Repository (events), CMS Content and Collaboration services (activities)</td>
</tr>
</tbody>
</table>

---
The main technology for the implementation of site components is the Spring Surf framework (http://www.springsource.org/extensions/se-surf). It is an Alfresco extension of Spring MVC (Model-View-Controller architectural pattern in Spring framework) implemented in JavaScript. Developer can use JavaScript API for access to the Alfresco and OCOPOMO REST services in order to implement application logic, which will prepare a data model for view, process actions or forms submitted by the user. The views in Surf are generated using the FreeMarker templates.

### CMS - Content Management Server - all the content produced during the collaboration work on policy modelling scenarios is stored in the OCOPOMO Content Management Server (see Fig.3.), which is based on the Alfresco Repository. The managed content includes all information artifacts produced by the collaboration tools, i.e. documents, wiki pages, discussion forum messages, calendar events, chat history, polling posts, etc. The Alfresco Content Repository stores content in a relational database and manages full-text index. Additionally, it provides interfaces for the functionalities like definition of the content structure, query of the content and metadata, access control, versioning, working version management, events, rules, import/export, etc.

The content is modeled in the repository as a node construct, which associates metadata and the binary or textual content. Each node has an assigned content type which defines which metadata properties should be defined for that node. The data type of the property can be Boolean, decimal, string, date-time or reference to another node. Additionally, the node type can constrain value range and cardinality of the properties (e.g., for decimal properties, minimal/maximal value can be specified).

The node types are organized in a simple inheritance hierarchy (constraints of the super-type are inherited for the sub-type). Alfresco repository supports aspect oriented modelling, where constrains for the nodes can be defined in the aspect, which can be dynamically added or removed to the node definition (multiple aspects can be assigned to one node). The nodes are organized using hierarchical parent/child relation. Each node type can specify what types can be children of that node (if any). On the top of the hierarchy there is a single root node, which represents the whole repository. It is possible to navigate in the hierarchy and to list nodes using absolute or relative path.

Besides the content services, the core functionalities are also provided as a set of Content and Collaboration Services embedded in the Alfresco application server, which can be divided into the following groups:

- **Content services** – services like tagging (arbitrary user-generated tags or defined classification schemes), content transformation (conversion of the content from one format to another), and metadata extractions (synchronization of document and node metadata properties).
- **Control services** - with workflow processes which can involve human interactions through allocation of tasks (implemented using integrated JBPM engine).
- **Collaboration services** - with services for Persons (user profile management), Activities (continuous personalized feed of activities performed by other users or by the OCOPOMO system), Document library, Wiki pages, Discussion forums, Calendar

### Collaboration and Scenario Editing tools

<table>
<thead>
<tr>
<th>Polling and Rating Manager</th>
<th>CSET Polling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Manager</td>
<td>CMS Content and Collaboration services (workflow)</td>
</tr>
<tr>
<td>Rule Manager</td>
<td>CCD Tool (Conceptual Description Tool and Transformation Tool Component)</td>
</tr>
<tr>
<td>Search Manager</td>
<td>CMS Alfresco Content Repository (query)</td>
</tr>
<tr>
<td>Simulation Manager</td>
<td>SE</td>
</tr>
<tr>
<td>User Manager</td>
<td>CMS Alfresco Content Repository (access control), CMS Content and Collaboration services (persons)</td>
</tr>
<tr>
<td>Version Manager</td>
<td>CMS Alfresco Content Repository (versioning)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alfresco Content Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCOPOMO Content Management Server</td>
</tr>
<tr>
<td>CMS REST API</td>
</tr>
<tr>
<td>OCOPOMO Services extensions</td>
</tr>
</tbody>
</table>

Figure 2. Collaboration and Scenario Editing tools

Figure 3. Content Management server
and Polling services – provide application logic and data layer backend for CSET module.

The functionalities provided by the services are accessible by clients through REST APIs. Alternatively, client developer can use standardized CMIS services. The CMIS (Content Management Interoperability Services) standard defines repository and language agnostic content model and a set of content management services, which cover querying, versioning, renditions and access rights. The CMIS defines two protocols to access the CMIS compliant repository: one is REST-like protocol based on the extensions of the AtomPub format, and the other is SOAP binding.

The next modules are mainly developed by other partners of project, while the above given modules were implemented by the paper's authors. Therefore, we will only provide some short description of next tools. Just annotation tool (a subpart of CCD tool) will be defined in more details, as it was implemented by several authors of this paper.

**CCD Tool** – in order to find a connection between narrative texts of stakeholder-generated scenarios (evidenced through background documents of the policy to be discussed) and formal policy models (generating model-based scenarios), the OCOPOMO approach introduces the Consistent Conceptual Description (CCD) as "modelling middle-layer". A CCD enables different stakeholders in the OCOPOMO process to better understand the policy context and to support semi-automatic transformation of text fragments into formal statements and agent descriptions. Thereby, the CCD is exposed to be consistent, conceptual and descriptive ontology-based definition of agents, their relationships, functions, behavior rules, inputs, and outputs [13]. The main parts of CCD tool, which is implemented as an Eclipse IDE plug-in, are modelling tool (for creation of CCD meta-model, which is used for annotation), annotation tool (which is used for annotation of scenarios for the creation of case-specific CCD models), and transformation tool (used for transformation of case-based CCD model into code stubs of simulation model). The information is linked from scenario text and background documents, through CCD model, into simulation code (agents and their rules).

**SE** - simulation environment - comprises parts of the toolbox primarily dedicated to policy modelers. It consists of IDE for editing, debugging and executing policy simulation models, and a facility for inspecting, analyzing and commenting simulation outcomes (Simulation Analysis Tool). Although it should be possible to employ these tools in a stand-alone way, the intended way of their use is in an integrated manner as a part of the OCOPOMO method, e.g., IDE imports model code stubs generated by the CCD Transformation Tool, policy modeler extends and specifies more precisely the model code, simulation outcome data is transferred to the Simulation Analysis Tool (where policy modelers can comment, analyze and edit the data, thus creating the model-based scenarios), and then model-based scenarios are fed back to the CSET. In order to achieve better integration, the IDE of SE is designed as an extension of the Eclipse platform, which is also basis of selected simulation backend (Repast) with the rule engine developed within the project (which is called DRAMS).

Text Annotation Tool (as a part of CCD Tool) is also implemented as Eclipse Plug-in using EMF (Eclipse Modeling Framework) and GMF (Graphical Modeling Framework). The PDF annotation feature is developed using the JPedal library and is connected with Alfresco CMS by means of the CMIS interface. Fig.4 shows screenshot of the Annotation tool. The user interface of the PDF-HTML Eclipse plug-in, developed as the second version of the Annotation tool, is presented there. It enables the annotation of PDF documents and HTML wiki pages that are stored in a remote Alfresco installation. The usage of Alfresco Repository is provided by the standard CMIS services.

**IV. INTEGRATION DETAILS**

In general, the OCOPOMO process includes two distinctive sets of functionality, which are consumed and performed by the respective groups of users. The context of the system is defined by external entities that can interact with it by sending or receiving the data.

The external entities correspond to the prospective users of the OCOPOMO system, i.e., the stakeholders that are involved in the process of collaborative policy development. Two main groups of system users can be distinguished according to the level of intentional involvement in the policy creation: direct participants and supportive actors. The direct participants, who are intentionally involved in policy creation and have their own preferences, ideas, or proposals of how the newly created policy should look like, include the user roles such as politician (decision makers responsible for the policy implementation), civil servant (assistants of politicians and/or providers of relevant supporting materials for other participants of the policy development process), and stakeholder (end users such as citizens, NGO’s and SME’s, that actively participate in the construction of narrative scenarios, discussions, information exchange and other phases of the collaborative policy development).

The second group of users, which provides a methodological or technical support for the direct participants and is most important for distinction between sets of functionality (with influence on integration of platform), includes the user roles like facilitator, analyst, or modeler. Facilitator is a mediator, who
methodologically controls the collaboration working space, maintains the collaborative scenario development (e.g. by providing initial text descriptions or supporting materials), invites stakeholders of relevant interest groups, assigns user accounts, contacts analysts and modelers to provide respective models, controls iterations of narrative scenarios and publishes agreed policy descriptions. Analyst is an expert that investigates scenarios and other (mostly textual) resources, performs qualitative analyses of narrative scenarios, discussions, comments, simulation results, etc., and provides a formal representation of extracted knowledge. Modeler is an expert that constructs formal policy models according to a given knowledge representation extracted from narrative scenarios and other resources, creates the simulation environment and provides the constructed models for participants, which can run customizable simulations. Modelers are also responsible for maintenance of simulation results and their provision to analysts for enhancing the respective scenarios accordingly.

Integrated OCOPOMO platform wants to support all potential users with their respective tools, but with specific user interface, where it is applicable and needed. Therefore we have two distinctive functions – web application based on Alfresco and Eclipse plug-ins. The collaborative discussion on scenarios, exchange of information, ideas, and opinions between involved stakeholders, and production of user-generated scenarios implies a web-based collaborative environment that covers the functionality of CMS and CSET installation modules. This should stay as simple as possible and collaborative. On the other side, the expert-like processing of produced scenario alternatives towards runnable policy simulations (including means of text annotation, CCD modelling, development of rule-based policy models, and an environment for running simulations), requires a rich toolbox that integrates CCD Tool and SE modules. Based on this distinction, the structure of integrating the installation modules to unified OCOPOMO platform was proposed as it is depicted in Fig. 5.

Figure 5. Basic structure of OCOPOMO integrated platform

V. CONCLUSION

The presented design and integration of tools is composed to support policy modelling processes identified within the OCOPOMO project. Processes of collaborative creation of narrative scenarios, annotation of the scenarios, construction of formal agent-based policy models, iterative simulation, and evaluation were also specified. In this paper we have introduced architecture, tools and integration details, which are able to fulfill such processes. The designed architecture was implemented and it is or will be tested during the pilot applications within year 2012. The system will be tested on pilot applications in Slovakia (policy for exploiting renewable energy resources, Kosice), in Italy (optimal allocation of EU structural funds, Campania), and in UK (housing policy, London.). More information about the project OCOPOMO can be found at http://www.ocopomo.eu.

ACKNOWLEDGMENT

The OCOPOMO project is co-funded by the European Commission within the 7th Framework Programme, theme ICT-2009.7.3, contract No. 248128. The authors would also like to acknowledge the contributions of and express their gratitude to OCOPOMO project partners, especially prof. Scott Moss, prof. Maria Wimmer, Paul Ormerod, Sabrina Scherer, Melanie Bicking, and Ulf Lotzmann.

REFERENCES