Design of a System Architecture for Support of Collaborative Policy Modelling Processes

P. Butka*, M. Mach**, K. Furdik***, J. Genci**
* Technical University of Kosice, Faculty of Economics, Kosice, Slovakia
** Technical University of Kosice, Faculty of Electrical Engineering and Informatics, Kosice, Slovakia
*** Intersoft, a.s., Kosice, Slovakia
peter.butka@tuke.sk, marian.mach@tuke.sk, karol.furdik@intersoft.sk, jan.genci@tuke.sk

Abstract—The paper describes a design of system architecture for support of collaborative policy modelling processes in electronic governance. The policy modelling processes, as complex and knowledge intensive activities, combine collaborative creation and analysis of narrative scenarios with agent-based simulation. The proposed architecture will support collaborative approach to the policy modelling (i.e. enhance the modelling processes by collective intelligence). A scope of the proposed system will be specified and platform architecture components, including some practical details regarding their implementation, will be described.

I. INTRODUCTION

One of the advanced R&D topics in the area of electronic governance is policy modelling supported by a wide collaboration of the policy stakeholders by means of Information and Communication Technologies (ICT). The open collaboration aims to support an active and qualified engagement of citizens in governmental decision-making processes. This paper is focused on design, development and provision of ICT tools that are capable, via open collaboration and discussion, to gather inputs from and opinions of relevant stakeholders and transform them into shared knowledge. The ICT toolkit supporting the policy modelling processes described above is expected to perform model-based simulations, integrating all relevant variables, parameters, interferences, and scenarios necessary to forecast potential outcomes and impacts of the proposed policy measures [1]. This approach is also trying to eliminate drawbacks of the macro-economic modelling, which, especially in the context of the global economic and financial crisis, was not able properly recognise and understand warning signals, and reflect them by decision makers in an adaptation of policies on global or local levels [2].

From technological perspective, the proposed solutions of collaborative policy modelling are usually built on already known and proven technologies, which typically include rule-based or agent-based modelling platforms [3], tools and interfaces for visualised social simulations, knowledge representation models, and web-based e-Participation tools incorporated into e-Governance frameworks. The innovative aspects and challenges of the proposed approach lay in an integration and functional interconnection of these technological frameworks into a comprehensive platform supporting an evidence-based decision making process. In this paper, we present architecture of the software platform supporting collaborative policy modelling processes in the context of the aforementioned objectives. The focus is given to the aspects of integration of e-Participation techniques, methods for scenario creation and analysis, formal modelling and simulation, supported by a proper technological framework, including a description of platform architecture components and specific implementation details related to practical aspects of the system integration and information provision.

The research presented is carried out within a running FP7 project 248128 “Open Collaboration for Policy Modelling” (OCOPOMO). Several projects related to governance and (collaborative) policy modelling is being implemented within the FP7 ICT Programme. A support action project CROSSROAD (http://crossroad-eu.net) aims to build a roadmap for ICT research in the field of governance and policy modelling, to be supported by the results derived from other FP7 projects in the area of eGovernment and Policy Making (where OCOPOMO also belongs to), like COCKPIT Project (http://www.cockpit-project.eu), modelling of policy modelling processes), PADGETS project (http://www.padgets.eu, complex social Web2.0 solutions for governance), +SPACES project (http://www.positivespaces.eu/, virtual spaces for policy making), etc. In OCOPOMO, starting point is the analysis of collaborative policy modelling processes of two pilot applications. Based on this analysis, a design of suitable platform architecture, corresponding to the needs of the analysed processes, was proposed. This platform will be implemented in the next stage of the project.

II. POLICY MODELLING PROCESSES - SCOPE OF THE PROPOSED SYSTEM

The process of policy modelling specifically addressed by the OCOPOMO project is based on narrative scenarios and related formal policy models that are constructed and modified collaboratively, by various stakeholder groups using e-participation tools for information exchange and mutual communication. Envisioned functionality of the platform was proposed based on information from different sources – an initial project proposal ; state of the art analysis in the area of e-Participation, formal modelling and simulation tools, scenario creation and analysis, integration aspects, standards, etc.; analysis of policy development processes as described by the project user partners (regional governments in Slovakia and Italy), theoretical background of the aforementioned research areas; as well as collaboratively designed ideas for improvement of the current processes. The functional
description was further elaborated and specified based on the user requirements generated by the project partners. All this resulted in an architecture design, identification of proper technologies, and validation of the developed architectural structure, which is in detail described in [4].

In general, the proposed platform should support iterative and collaborative creation of policy models and respective scenarios. To design architecture of a software platform that is capable to provide the required functionality, first a 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 artefacts that are exchanged within the supported phases of the policy creation process, as it is shown in Fig.1. Three basic types of information artefacts were identified: Scenario, Simulation model, and CCD (Consistent Conceptual Descriptions).

**Scenario** is the main artefact created 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 the discussed policy topic. It may cover an existing status, or structured text of a perceived view or understanding of policy proposal. The scenario is a narrative, unstructured or structured text of a perceived view or understanding of the discussed policy topic. It may cover an existing status, or structured text of a perceived view or understanding of policy proposal. The scenario can also be defined as a form of knowledge representation schema that captures descriptions and perceptions of the stakeholders in a structured, formal, and machine-readable way. The simulation model can be run to produce an output, which can be interpreted as a "proof of hypothesis" (while the hypothesis is defined by the model and the inputs).

**Simulation model** is a simplified logical representation of objects, phenomena, and processes in a domain of interest. Typically, the model consists of parts (elements, logical units), interactions between the parts, and inputs of a defined quantity and nature. The simulation model can be run to produce an output, which can be interpreted as a "proof of hypothesis" (while the hypothesis is defined by the model and the inputs).

The **Consistent Conceptual Description (CCD)** is a knowledge representation schema that captures descriptions and perceptions of the stakeholders in a structured, formal, and machine-readable way. For example, the CCD can be constructed by means of topic maps, ontology, and qualitative data analysis with knowledge structures like social networks, rule-depency graphs, etc. The CCD plays a role of an intermediary structure between the scenarios and simulation models, allowing forward and backward transformations from scenarios to models and vice-versa.

The information resources are exchanged in the phases A – D of the policy creation process:

- **Phase A: Initiation.** A dedicated user, usually a decision-maker, prepares a policy case to be discussed and further developed. The policy case is provided to other stakeholders in a form of textual description and other background materials. This phase results in an initial scenario, which is directly derived from the policy case description.

- **Phase B: Enabling interactions between stakeholders (users).** In a collaborative discussion, stakeholders may provide various background data, opinions or arguments, personal experience and documents. Likewise, evidence-based user generated scenarios are developed.

- **Phase C: Capturing and representing knowledge.** CCD schemas are elaborated based on the user generated scenarios, inputs of the background experience and documents.

- **Phase D: Simulation model construction.** The knowledge accumulated within the CCD is transformed into simulation models. Users may run the simulation models and generate model-based scenarios on the output. These scenarios can then serve as supportive data/experience materials and/or documents in further discussion on the policy case.

The process is iterative and open-ended. Alternative policy proposals, consisting of narrative scenarios and related simulation models, are produced and discussed in the collaborative environment. Further analysis of the arguments and contributions of policy stakeholders participating in the social network may help to identify the most probable, most appropriate, or even unexpected, negative policy trends. Context of the system is defined by external entities that can interact with it by sending or receiving data. The external entities correspond to the prospective users of the platform, i.e. stakeholders that are involved in the process of collaborative policy development. Two main groups of 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 the policy creation and have their own preferences, ideas, proposals what the newly created policy should look like, include stakeholders (user roles) such as a politician (or any responsible decision-maker), civil servant (assistant of decision-maker and/or provider of relevant supporting materials), and other stakeholders (citizens, NGOs, interest groups, who actively participate in construction of the narrative scenarios, discussions, supportive data/experience materials and/or documents in further discussion on the policy case).

The second group of users, which provides a methodological or technical support for the direct participants, includes the user roles like facilitator, analyst or modeller. The facilitator is a mediator, which methodologically administers the collaboration space (e.g. maintains the development process, providing initial text descriptions or supporting materials, invites stakeholders, assigns user accounts, controls iteration cycles, etc.). The
The modeller is an expert that constructs formal policy models according to the given knowledge. The model provides constructed models for the participants.

III. THE PROPOSED PLATFORM ARCHITECTURE

The proposed platform functionality in the identified context and scope needs to be adequately supported by a proper technological framework. Therefore, specific technology identification is described at the beginning of this section, followed by information model, high-level integration architecture and description of particular components.

A. Technology identification

In order to implement the platform functionality, first a proper technology had to be identified before starting the design of architecture details. Integration means and supporting platforms depend on the underlying architecture of the solution. The three-tier architecture [5], frequently employed for web-based applications, implies integration on data, middle-ware (business logic), and presentation levels. The integration of the presentation layer and content, which is of particular importance for web applications, is handled by portal technologies, content repositories, or complex CMS (or Enterprise CMS - ECM). One of the leaders in advanced CMS technologies is Alfresco (http://www.alfresco.com), which provides powerful collaborative environment, document flow capabilities, and portal-based features in its Alfresco Share front-end interface.

In the field of e-Participation, in recent years an increasing activity resulted in a development of several specialised solutions such as, for example Gov2Demos (http://www.gov2demooss.org) or Discourse Machine (http://www.discourse-machine.de). However, the e-Participation tools such as on-line forms, polling/voting systems, chat rooms or discussion forums are often included in existing CMS (e.g. Alfresco, Joomla, Drupal, Plone, TYPO3, etc.), usually enhanced by additional functionality of document flow, user administration and many advanced web publishing features. In addition, the CMS-based solution provides means for easier integration with external tools. This aspect of integration, together with a suitable methodological design of participation offerings, was identified in [6] as a critical success factor for e-Participation and a significant importance was assigned to it.

Tools for the scenario analysis should support extraction of relevant information and parameters from narrative texts. The QDA method of qualitative data analysis might be applicable to support the collaborative policy modelling approach. Formal modelling is focused on transformation of narrative descriptions to agent-based policy models that enable running simulations of the policy alternatives. Dynamic behaviour of software agents in these models is expressed by declarative rules that capture possible relationships described by stakeholders in the underlying scenario. From this perspective Repast 3.1 (http://repast.sourceforge.net) could be a good solution, since it provides agent-based simulation environment with sufficient functionality that can be customised and integrated into the platform in a straightforward way. Within the OCOPOMO project, Repast (as a simulation platform) will be supported by rule engine, so-called “declarative, rule-based, agent modelling system” (DRAMS), developed specifically for the needs of the OCOPOMO project.

B. Basic data model and information blocks

Based on the initial analysis of information resources, the data objects and structures were identified together with the means of storage, maintenance, and distribution of information through the system architecture using the methodology described in [7]. The design of information structures was driven by analysis of the functional requirements on the platform, technology frameworks, and selected integration infrastructure for data access, security, and content management. According to this analysis, six types of information blocks (IB) were identified for data structures required by various parts of the platform. Each of these information blocks can be further subdivided into a set of data objects, elementary units of the data architecture, as depicted in Fig.2.

The CMS-IB (Content and semantics management) represents data structures and resources that handle the collaborative process of policy modelling in the workspace. It includes social network environment, predefined workflow and document flow sequences, data objects for storage and manipulation with semantically enhanced textual content of information artefacts (e.g. textual data analysis rules, underlying semantic knowledge model, etc.). The ePart-IB (e-Participation objects and tools) consists of data objects that facilitate sharing information materials, active communication and information exchange between stakeholders in the collaborative environment. Generic e-Participation object encapsulates basic properties for particular types of consequent data objects as Document, Discussion, etc. The SM-IB (Narrative scenarios and related CCD) covers data objects and resources for manipulating with the narrative scenarios and related CCD structures, which are used for transformation the scenarios into simulation models. Scenario has its textual content, may have a context defined by links to other artefacts, and can be included into a workflow. The scenario can be transformed to a simulation model and modified by a simulation output. The NS-IB (Narrative scenarios and related CCD) consists of data objects related to creation and modification of simulation models with agent, rules,
simulation, and model output objects. The DR-IB (Data repository) represents an integrated data storage place for the whole system, including databases and file-based repositories, respective data connectors to the repositories, as well as global settings and platform configuration data. The UMS-IB (User management and security) provides data objects and structures for user management, authentication and authorisation for the overall platform with user roles, profiles and preferences, access rights, etc. The specified data objects are created, manipulated, exchanged and maintained by functional components, action units of the platform, which are described in next subsection.

C. Structure of Platform Functional Components

In accordance with the technological analysis, functional architecture of the platform is proposed in a structure of three main layers (Fig.3), with particular sets of functional components (so-called managers).

The Tools layer provides components that are responsible for maintenance of particular tools within the system, together with the respective user interfaces. This layer is structured into three modules, namely:

- The Communication subsystem – covers communication, collaboration and cooperation-based features of the platform. The managers of this module will support components of other subsystems in aspects of communication and information exchange, i.e. in e-Participation features that strengthen the collaborative manner of the policy modelling process.

- The Scenario subsystem - provides functions for generating and analysing narrative scenarios of policy proposals. It includes document management and text analysis mechanisms, context enhancements and semantic annotations.

- The Simulation subsystem – provides functionality, which is important for modelers in order to create, update, visualise and execute simulation models, transformation of CCD schemas to simulation models consisting of agents, rules, axioms, constraints, as well as customising created models by setting up inputs or other parameters, invoking the models into running simulations, and providing results of the simulations.

The Core layer is dedicated to processing all the data exchanged inside the system. It supports the Tool layer with any business logic related to information resources, metadata and processes that may be required by the scenario, communication, or simulation subsystems. Functionality of a broader scope than an individual tool, namely federated search, system-wide notification, initiation and maintenance of processes, and management of user profiles, is also provided by the respective Core components by means of business logic and user interface.

The Data layer provides infrastructure for persistent storage, management, secure access, sharing and versioning of particular content of any required type. It may include relational databases for structured data, XML-based semi-structured data for knowledge representations and semantic annotations, file systems for textual or multimedia documents, various indexes enabling the retrieval of data, as well as storage of system properties and global settings.

The presented layers can be mapped onto a standard three-tier structure of client-server applications. The Core managers together with subsystems’ managers of the Tool layer correspond to the middle tier (usually called business logic tier). User interfaces of the subsystems’ and Core managers belong to the upper tier (i.e. presentation tier), and managers of the Data layer correspond to the data tier. This approach is based on the assumption that the practical solution will be built on the integrated platform provided by the Alfresco framework. It enables a composition and functional integration of the particular managers, i.e. their business logic, user interfaces and underlying data structures, into a single web-based application. In accordance with this approach, the information flow was specified as a part of the architecture design. It represents means of information and data exchange between the subsystems and managers of the Tool, Core, and Data layers, which are transformed into the presentation, business logic, and data tiers. The schema of information flow, presented in Fig.4, was constructed as a mapping of the designed information blocks into the structure of functional modules with respect to the three-tier architecture provided by the Alfresco framework.

In more details, the particular managers were described by their context (connection to other managers), basic functionality, supported use cases (from the user and system point of view), and their APIs for other components. Due to limitation on the length of this paper, only a simple description of the managers will be provided in the next paragraphs. In most of the components the User Manager is used in order to obtain access rights for actions as an application of the role-based access control, so for the sake of space limitation we will not mention it in every component. Similarly, search is supported by the Search Manager components and many components will use combined or particular searches using this manager. Whenever process-specific information is needed, the Process Manager is used for accessing such data. Furthermore, for every manager, which has some user interface or functionality shared in the collaboration space,
connection to the **Collaboration Space Manager** is important, but as it is also straightforward, we will omit this information hereinafter.

The **Discussion Forums Manager** is responsible for providing discussion forum functionality (through the collaboration space) within the platform. Expected functionality includes creation, editing and deleting of discussion forums, threads (topics) and messages with rating/tag and notification functionality reused. The rating functionality is reused for discussion evaluation (relevance feedback) and analysis, where the Polling and Rating Manager is helpful. In case of need for notification support (e.g. email notification, RSS, etc.) the Notification Manager is reused. For storage and retrieval the Content Manager is used for content repository functions.

The **Chat Manager** is responsible for providing chat functionality (through the collaboration space) within the platform. Expected functionality is creation of (context-specific) chats, save history of chat as a document, adding messages to chat, create discussion forum related to chat. The Document Manager is used for saving the chat’s history as a document. Users are also enabled to create discussions related to the chat, where communication can continue, using API of the previous manager.

The **Calendar Manager** is responsible for providing shared calendar functionality (through the collaboration space) within the platform. Expected functionality includes adding, editing and removing calendar events (which are process or context specific) in a shared calendar, as well as reminder functionality (reused notification). In case of need for notification support (e.g. reminder functionality) the Notification Manager is reused. For storage and retrieval of manager-specific data the Content Manager is used.

The **Polling and Rating Manager** is responsible for providing polling and (in a more general way) rating functionality (through the collaboration space) within the platform. Expected functionality includes creation and managing of polls, rating API and rating user interfaces for different type of objects available for other components (e.g. discussion forums). The rating functionality is mostly planned for the Discussion Forums Manager. For storage and retrieval of manager-specific data the Content Manager is used. Notification functionality, provided by the Notification Manager, is used for announcing of polls and their results.

The **Document Manager** is responsible for providing content management functionality (through the collaboration space) within the platform, mostly for scenario creation phase of the process and linking of conceptual models with information from real documents. Expected functionality includes creation, opening, deleting and tagging of documents, inserting different resources into the system (as documents of different formats and/or links to resources), and versioning of documents. One of the main functions is the support of scenario generation and analysis, therefore the Annotation Manager is connected to this manager. For storage, retrieval, versioning and notification of data the Content, Version and Notification managers are used.

The **Annotation Manager** (Annotation tool) is responsible for transformation from unstructured text (scenarios) to structured information (CCD - Consistent Conceptual Description). Expected functionality includes opening scenario (document), scenario analysis using annotation tool, highlighting the text and creation of specific objects (annotations), as well as creation of relations between the objects and their groups. An output of the analysis from the annotation tool is then an important part of the conceptual description of the current understanding of the problem area. Relevant objects are also managed within the Concept Manager (manager of concepts in CCD) and the Link Manager (linking utility for linkage of the CCD elements, data, models, simulation, etc.). For storage and retrieval of manager-specific data the Content Manager is used.

The **Rule Manager** is mostly responsible for creation of fact templates, facts and rules from the CCD, which are then part of simulation models. Expected functionality is to provide user interface for evidence-based rules/agents creation and backward understanding of current modelling status. Relevant objects are reused and managed also within the Concept Manager and the Link Manager. For storage and retrieval of manager-specific data the Content Manager is used, versioning support is provided by the Version Manager.

The **Simulation Manager** is responsible for running simulation models and providing results of the simulations. Expected functionality includes import and revision of simulation models, running simulations (in different modes), evidence-based inspection of models from rules to the CCD and documents according to simulation results (using linking utility), and creation of results (text-based, statistics, etc.). The simulation models can be imported and revised with the help of the Rule Manager and evidence-based inspection using the Concept Manager and the Link Manager. For storage and retrieval of manager-specific data the Content Manager is used, versioning support for models is provided by the Version Manager. The Document Manager is used also for storage of documents created within simulations (results – text-based, statistics, etc.).

The **Search Manager** is component which provides (federated) search within the platform or partial searches in different resources. This manager provides widely shared functionality within the platform – search for different type of data, metadata, content objects, etc. It is possible to combine more searches into one output or use partial searches within components. The Content Manager supports storage and retrieval of manager-specific data.

The **Collaboration Space Manager** is responsible for managing collaboration space where all communication aspects (sharing of documents, forums, chat, polls, etc.) are connected and shared together using one shared space. Expected functionality includes creation of a collaboration space (CS), management of members in the CS, adding objects into the space (by specific tools), opening objects within the space, preference-based starting view of the CS, search in the CS, and notification features within the space (e.g. RSS, hints, news, etc.). For storage and retrieval of manager-specific data the Content Manager is used. Notification functionality, provided by the Notification Manager, is used within collaboration space for publishing news, hints and providing RSS feed(s).

The **Notification Manager** is responsible for notification services within the platform, where any necessary information (by other components of the collaboration space) could be provided using selected channel(s) like
news, hints, newsletter, but mostly RSS feeds and email. Expected functionality is preparation of notification message, selection of channel(s), and execution of notification service.

The Process Manager is responsible for managing workflow of the whole process for collaborative policy modelling (from initiation through scenario building and analysis to modelling and simulation). It is important for controlling the process current status, changing steps, sharing data regarding process (using the Collaboration Space Manager), notification of changes (using the Notification Manager), reuse/sharing of process-specific data and information in all other components. For storage, retrieval and versioning of manager-specific data the Content and the Version Manager are used.

The Concept Manager is responsible for managing structured information about the currently modelled problem - the CCD, which leads to simulation models. Basic functionality includes creation of the CCD elements, updating structure, definition of structure of data, as well as storage, retrieval, visualisation and versioning them. The Annotation Manager uses concept creation for explicit identification of structure in scenarios and data. For storage and retrieval of manager-specific data the Content Manager is used. The Version Manager is used for support of versioning functionality for concepts in the CCD. The Link Manager and the Rule Manager are directly connected to this structure information, therefore are also able to communicate with this manager. The concepts are available to the Simulation Manager for evidence-based analysis of simulations.

The Link Manager is responsible for managing links between information about the currently modelled problem from the evidence-based and CCD elements to rules, agents and simulation models (and their results). Basic functionality includes creation of links and retrieving of them for explanations in simulation analysis (starting from the Simulation Manager) and policy modelling updates. The Annotation Manager is needed for connection of concepts within the structured data (as a part of the evidence-based linkage). Content management and versioning are realized as in previous manager.

The User Manager is responsible for managing users, roles, profiles and their access rights within the platform. This part of the system has several functions like creation of new users, managing their roles and profiles, and managing access rights to other components.

The Content Manager is responsible for managing specific content (like conceptual descriptions, metadata for search, specific types of documents, links between conceptual elements, etc.) within the platform. It means that the manager is data-level component for managing content storage and retrieval from the content repository for different types of data. Therefore many components are using the content repository for storing and retrieving of such specific data like links, concepts, metadata, searches, process-specific data, etc. The Version Manager is used for the support of versioning for content items.

The Version Manager is responsible for versioning different data resources within the platform. The main responsibility is to support the Content Manager and its content repository with versioning support. Moreover, several components are able to use versioning directly for their elements like process-specific data (the Process Manager), documents (the Document Manager), concepts and links (the Concept Manager, the Link Manager), and simulation models (the Simulation and the Rule Manager).

IV. CONCLUSION

The presented architecture design is resulted from the requirements on the collaborative policy modelling processes. An important part of the requirements was implied by the pilot applications of collaborative policy modelling to be carried out within the OCOPOMO project. After a detailed analysis, categorisation and prioritisation of the requirements, the scope and context of the system were identified by means of user interactions and exchanged information artefacts. Processes of collaborative creation of narrative scenarios, extraction of knowledge from the scenarios, construction of formal agent-based policy models, iterative simulation, evaluation, and validation of policy alternatives were specified. The system will be implemented and tested on pilot applications in Slovakia (policy of exploiting renewable energy resources) and in Italy (optimal allocation of EU structural funds in the Campania region). 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, contract No. 248128. The authors would also like to acknowledge the contributions of and express their gratitude to all the OCOPOMO project partners, especially prof. Maria Wimmer, prof. Scott Moss and Melanie Bicking, PhD. The work presented in the paper was also supported by the Slovak Grant Agency of the Ministry of Education and Academy of Science of the Slovak Republic within the 1/0042/10 Project “Methods for identification, annotation, search, access and composition of services using semantic metadata in support of selected process types”.

REFERENCES