

Modeling in a Wiki with MoKi: Reference Architecture, Implementation, and Usages

Chiara Ghidini, Marco Rospocher, and Luciano Serafini

Fondazione Bruno Kessler - Centro per la Ricerca Scientifica e Tecnologica (FBK-irst),

Via Sommarive 18 Povo, I-38123, Trento, Italy

e-mail: {ghidini, rosposcher, serafini}@fbk.eu

Abstract—The success of wikis for collaborative knowledge construction is triggering the development of a number of tools for collaborative conceptual modeling based on them. In this paper, we present a reference architecture for wiki-based collaborative conceptual modeling tools. The characteristics of our reference architecture are: (i) the use of wiki pages to describe semantic terms and organizational mechanisms of a conceptual modeling language; (ii) the organization of wiki pages in an unstructured part and a structured part; and (iii) a multi-mode access to the pages. We also present a completely revised version of the MoKi tool fully compliant with the presented reference architecture. A detailed description of some usages of MoKi in different application contexts is also reported.

Keywords-Conceptual Modeling; Collaborative Modeling; Semantic Wikis; Ontology Modeling; Process Modeling.

I. INTRODUCTION

In this paper, we present the wiki-based collaborative conceptual modeling tool MoKi, its reference conceptual architecture, the specific current implementation and usages in real-world scenarios. The paper extends the work presented at the “Fourth International Conference on Information, Process, and Knowledge Management (eKNOW 2012)” [1]. More in detail,

- the description of the reference conceptual architecture presented in Section III has been extended with a more detailed and formal description of how to represent conceptual modeling languages in a wiki, with the aim of providing a general conceptual architecture for collaborative conceptual modeling tools which encompasses MoKi;
- the description of the MoKi tool presented in Section V has been extended with more examples and the description of additional functionalities; and finally,
- an entire new section describing the usage of MoKi in different application contexts has been added (Section VI).

The idea of building wiki-based modeling tools has emerged in recent years as a promising approach towards the collaborative construction and visualization of conceptual models. In fact, from the success of Wikipedia onwards, wikis have been increasingly adopted as tools for collecting, sharing and managing knowledge, both in the case of domain specific knowledge (e.g., in enterprises) and in the case

of encyclopedic knowledge. Moreover, recent projects such as DBpedia [2], YAGO [3], and Semantic Media Wiki (SMW) [4] have empowered traditional wikis with the capability of publishing their (usually) unstructured text and multimedia content in a structured, RDF-based, format. This has enabled users to employ better search, browse, and share facilities, and has extended the power of wikis transforming them from tools for the collaborative creation and management of content, to tools for the collaborative creation and management of (on-line) data and knowledge bases. This, in turn, has prompted the idea of building wiki-based tools for the collaborative construction and visualisation of conceptual models (see, e.g., the Halo extension and SMW+ [5], MoKi [6], and Ontowiki [7]), and has suggested the usage of the wiki philosophy in tools which are not directly built on top of wikis (e.g., Senso Comune [8], Freebase [9], and PoolParty [10]).

Despite this great amount of work, building a wiki-based tool for the modeling of a specific domain remains a challenging task, as the basic features of wikis must be used in a way that effectively support the construction of good quality conceptual models. The development of a clear reference architecture, where the focus is placed on identifying the key constructs and abstractions rather than on the technical characteristics of the tools themselves, would provide a significant contribution to meet this challenge. In this paper we address this task, taking into account the following needs:

- **Generality.** Until now, the work in the area of wiki-based modeling tools has mainly focused on the development of instruments targeted to specific conceptual models: thesauri, ontologies, RDF content, organizational / workflows, and so on. While this has contributed to show the potential of wikis, it has also delayed the emergence of a wiki-based paradigm for conceptual modeling. Defining a general paradigm for different conceptual modeling languages is a crucial step as it enables the use of similar abstractions and features for different types of models (e.g., an ontology or a workflow). This becomes especially important when users need to build, share, browse scenarios composed of different models. Think, for instance, to the case of an enterprise which needs to model an ontology

of competences together with the processes that need them, or to the case of a user who needs to browse a workflow together with the taxonomy used to annotate it. The reference architecture must aim at understanding how the features of wikis can be used to represent the building blocks of a general conceptual modeling language, before tailoring them to the needs of a particular one.

- **Collaboration.** A crucial step in building good quality conceptual models is the involvement of domain experts in the modeling process. As argued in [11], traditional methodologies and tools are based on the idea that knowledge engineers drive the modeling process. This often creates an extra layer of indirectness which makes the task of producing and revising conceptual models too rigid and complex, e.g., for the needs of business enterprises. In addition, the leading role of knowledge engineers can hamper the model construction as the domain experts (and domain knowledge) may become secondary to the process of efficient knowledge modeling, especially when domain experts have no understanding of the languages and tools used to build the conceptual models. The reference architecture must aim at understanding how the features of wikis can be used to support a well-balanced collaboration between domain experts and knowledge engineers in modeling process.

The contribution of this paper is manifold. First, we present a reference architecture for wiki-based conceptual modeling tools which satisfies the two needs described above. The distinctive characteristics of our architecture are: (i) the use of wiki pages to mimic the basic building blocks of conceptual modeling languages, namely semantic terms and structuring mechanisms; (ii) the organization of wiki pages for semantic terms in an unstructured part (for unstructured content) and a structured part (for structured content); and (iii) a multi-mode access to the pages to facilitate the usage by domain experts and knowledge engineers. Second, we illustrate an implementation of this architecture in a completely revised version of MoKi [6]. This description aims at showing the feasibility of the proposed architecture by means of a practical realization. Third we report on the usage of MoKi in different application contexts. Again this description aims at showing the sustainability of the proposed architecture by means of concrete usage in real application contexts.

The novelty of our work can be found at different levels: at a *foundational* level, this paper provides the first architectural model for wiki-based conceptual modeling tools, which can be used to implement tools for different conceptual modeling languages in a uniform manner; at an *architectural* level, it introduces the idea of multi-mode access to pages to support easy usage both by domain experts and knowledge

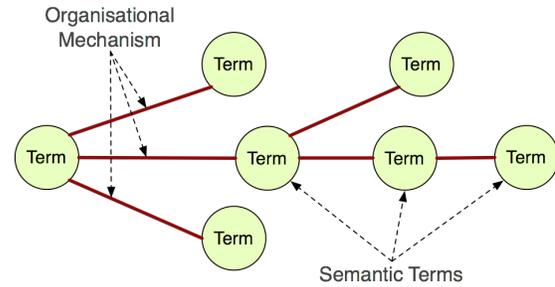


Figure 1. Conceptual Modeling Languages.

engineers; at the *implementation* level, MoKi provides the first attempt to build a single tool for different conceptual modeling languages able to support the collaboration of domain experts and knowledge engineers through the usage of a multi-mode access to knowledge.

The paper is structured as follows: we start from an analysis of conceptual modeling languages (Section II) and we proceed by defining an architecture which satisfies the needs of generality and collaboration (Sections III and IV). We then provide a description of MoKi (Section V) and some application contexts in which it has been used (Section VI), concluding with a comparison between the proposed architecture and state of the art tools for wiki-based conceptual modeling (Section VII).

II. CONCEPTUAL MODELING

Conceptual modeling (aka semantic modeling) has been researched and used in several areas of Computer Science and Engineering, such as Database / Knowledge Representation, Software Engineering, and Artificial Intelligence, often with different usages, characterizations, and terminologies. According to [12] and [13], we can say that conceptual models provide a description of knowledge based on the so-called associationist viewpoint, where knowledge is organized in terms of: (i) nodes that represent concepts, and (ii) associations (or, links) that represent relationships between them. In particular, [13] provides a characterization of Conceptual Modeling Languages (CMLs) in terms of their two main building blocks, also illustrated in Figure 1:

- 1) **Semantic terms:** these are the concepts built into the conceptual model. Semantic terms are used to describe different types of concepts, such as Entities, Activities, Agents, Goals, and so on, depending on the CLM used; and
- 2) **Organizational mechanisms** (also called Abstraction Mechanisms in [13]): these are primitive mechanisms for structuring the model along different dimensions. Examples of abstraction mechanisms are: generalization (often referred to as *isA*), aggregation (*partOf*), classification (*instanceOf*), contextualization / modularization, and so on.

The different uses of Conceptual Models in the diverse areas of Computer Science and Engineering had important consequences on the development of specific CMLs. If the models are used mainly by people, e.g., to capture, organize and communicate high level knowledge, then the CML notation may be semi-formal or even informal, as in the case of Concept Maps, where no, or an extremely informal, semantics is usually associated to the diagrams. On the contrary, if the models need to be as less ambiguous as possible, and/or they need to be algorithmically exploited by computers to provide services such as consistency analysis or query answering, then the notation needs to correspond to a precise formal semantics, as in the case of OWL ontologies. In between these extreme cases are “semi-formal” CMLs. An example is the Business Process Modeling Notation [14], which provides a very detailed and specific syntactic notation with a semi-formal semantics.

III. CONCEPTUAL MODELING IN WIKI PAGES

The first challenge for wiki-based modeling tools is to be able to represent the two basic building blocks of conceptual modeling languages, namely semantic terms and organizational mechanisms. In this section we provide a definition of Conceptual Modeling Wiki (CMW) which uses the notion of wiki pages to represent these building blocks.

Let Σ be an alphabet in some CML C , and Γ be the set of organizational mechanisms in C . A *Conceptual Modeling Wiki (CMW)* \mathcal{W} for Σ and Γ is a tuple (P, SP, p, a) where P is a non empty set of wiki pages, SP is a non empty set of special pages, $p: \Sigma \rightarrow P$, and $a: \Gamma \rightarrow SP$. We say that:

- $P \cup SP$ is the set of pages of the wiki \mathcal{W} ;
- $p(\alpha)$ is the wiki page associated to the semantic term $\alpha \in \Sigma$;
- $a(\beta)$ is the set of (one or more) wiki special pages associated to the organizational mechanism $\beta \in \Gamma$.

A pictorial representation of a conceptual modeling wiki is given in Figure 2. In a nutshell, a wiki is composed of a set $P \cup SP$ of pages, where each (regular) page in P is used to describe semantic terms in the model, and each special page in SP is used to display a functionality which enables the browsing / editing of the overall organization of the conceptual model according to a specific organizational mechanism. For instance, if we consider a CML which contains semantic terms for concepts, instances, and roles, and two organizational mechanisms such as generalization and aggregation, then we need a wiki able to associate a regular wiki page to each semantic term of type concept, instance, and role, plus two special pages which enable to visualize (edit) the overall model organised according to the generalization and the aggregation/decomposition dimensions respectively.

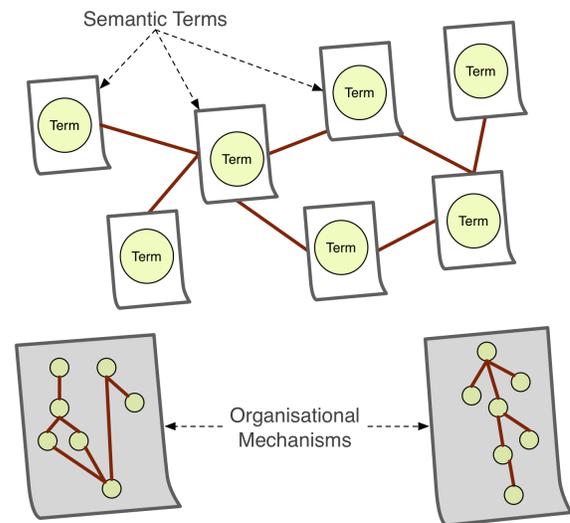


Figure 2. Representing a conceptual model in a wiki.

A. Building Wiki Pages for Terms

The idea of associating a wiki page to each semantic term, contained in the definition of Conceptual Modeling Wiki, is adopted by most of the state of the art wiki-based tools used to represent and manage knowledge (see Section VII). Nevertheless, this first idea needs to be refined and expanded if we aim at providing tools able to exploit in full the wiki potential and to make all the actors of the modeling team collaborate towards the creation, modification and exploitation of knowledge.

An important characteristic of wiki-based tools is their capability to deal with both *structured* and *unstructured* content. Assume, for instance, that we have to describe the term “Mountain”. We can describe it in a “wikipedia style”, by using text and pictures, as for instance is done in the “Mountain” page in Wikipedia [15], or we can provide more structured descriptions, in the style of Freebase, Ontowiki or of a Wikipedia Infobox. In this paper we argue that both types of content are essential in a process of conceptual modeling, and that a wiki page for a semantic term should be composed of two parts: the *unstructured part* and the *structured part*, as depicted in Figure 3. The first, unstructured part contains the rich and often exhaustive descriptions of knowledge which is better suited to humans and is built using linguistic and pictorial instruments. While some guidelines can be provided to organize the unstructured part, asking for instance for definitions, descriptions of the main characteristics, samples individuals (prototypes), a gallery of pictures, related/relevant documents, and so on, the content of this part of the page has a high degree of freedom. The second, structured part is instead the one which is used to provide the portion of knowledge which will be directly encoded in the CML. Differently from the unstructured part,

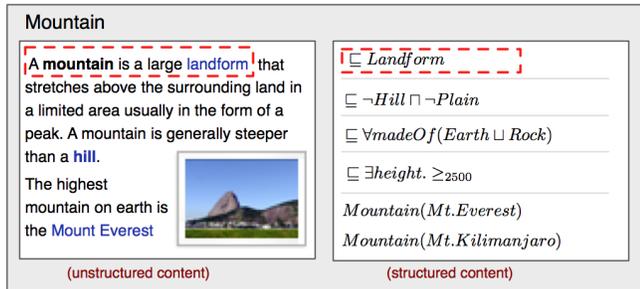


Figure 3. Wiki page for semantic terms.

which is expressed using natural language and multimedia content, the structured part of the page can have different formats, according to the CML used. Examples are: simple statements which describe the attributes of the semantic term being described; a list of inclusions axioms defining a concept in OWL (as in Figure 3); diagrams expressed in a workflow (business process) oriented language, and so on.

More formally, let $\alpha \in \Sigma$ be a semantic term of some CML C . We propose that a page $p(\alpha)$ for α in a wiki \mathcal{W} is a pair (u, s) , where u is a regular wiki string, that possibly contains links to other wiki pages of the wiki \mathcal{W} , and s is a description of that term in C using the alphabet Σ .

The advantage of storing the unstructured and structured descriptions in the same tool is twofold. First, the informal descriptions are usually used both to provide the initial description upon which the formal model is built, and to document the elements of the model, e.g., for future access and revisions. Storing the unstructured and structured descriptions in the same tool can facilitate the interplay between these parts, e.g., by adding alignment functionalities. Second, domain experts, who usually create, describe, and review knowledge at a rather informal/human intelligible level, may find the unstructured part their preferred portion of page where to describe knowledge. Instead, knowledge engineers should be mainly focused on the descriptions contained in the structured part. Nevertheless, by using the same tool and accessing the same pages they can be notified of what the others are focused at. Moreover, the discussion facilities of wikis, together with special fields for notes and comments, can be used by both roles to discuss and collaborate on specific parts of the model.

Note that, while a complete alignment between the unstructured and structured parts of a wiki page is not achievable, and most likely not even appropriate, as the rich nature of the unstructured representation is often not meant to be entirely transferred in a formal representation, it is easy to observe that specific portions of the unstructured part can provide descriptions upon which a certain piece of the structured representation is based, or can provide documentation which justifies or explains parts of the structured description (see, e.g., the two sentences surrounded by dotted

lines in Figure 3). Manual or semi-automatic functionalities to interlink the content contained in the unstructured and structured descriptions should therefore be provided in a CMW to support the interplay between the unstructured and structured knowledge contained in the wiki.

IV. SUPPORTING MULTI-MODE ACCESS TO CONCEPTUAL MODELS

The organization of a page in an unstructured and structured part is a second important step in defining the architecture of a conceptual modeling wiki, but may not be enough in the case of complex CMLs, such as the ones based on logical formalisms (e.g., OWL [16]) or very complex notations (e.g., BPMN [14]). In this case the structured part of the page will contain very precise, and often logic based, descriptions of a term, preventing domain experts from accessing the domain knowledge encoded in the conceptual model.

To overcome this problem, we propose to separate the content of the page from the functionalities used to view and edit it. Hereafter we call these functionalities *access modes*. The idea of this novel characteristic of wiki-based tools for conceptual modeling is to associate different access modes to each part of the page, as depicted in Figure 4, to enable a multi-mode access to the content stored in the page. In the example of the wiki page for "Mountain", introduced in the previous section and depicted in Figure 4, the unstructured content is stored in a regular wiki string and the structured content is stored in OWL. Therefore, the access mode to the unstructured part can be provided by means of the regular view/edit facilities of wikis, while the access to the structured content can be provided by means of two different modes: one based on a translation of the OWL content in, e.g., DL axioms or in the Manchester OWL syntax, and another based on a structured, but semi-formal rendering of the OWL content in a pre-defined template as the one depicted at the bottom of Figure 4. In this way the knowledge engineers can formally describe the semantic term "Mountain" in the chosen CML by using a highly formal access mode, while the domain experts can access a simplified version of the same content using a different, simpler, mode.

More formally, the *multi-mode access* of a page (u, s) for a semantic term α is a pair $(\{uam_1, \dots, uam_n\}, \{sam_1, \dots, sam_k\})$, $n, k \geq 1$, where each uam_i is an access mode for u and each sam_j is an access mode for s .

As we can see from the definition above we can potentially define a number of different access modes for each part of the page, which can be based on the different existing approaches towards representation of content and knowledge. Examples are: different access modes which represent the OWL structured content using different syntax, controlled natural languages, or graphical representations. Analogously

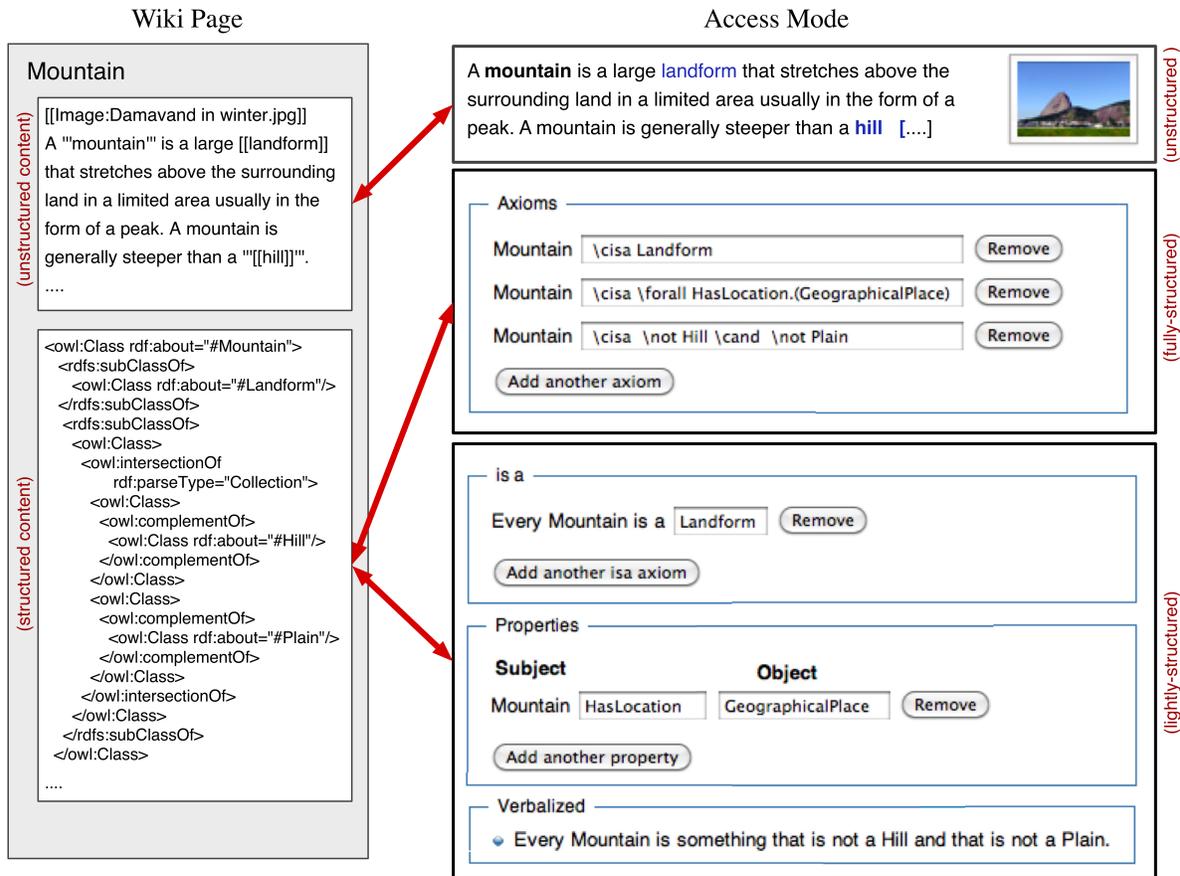


Figure 4. Multi-mode access to a wiki page for semantic terms.

we can have different templates which render the structured content at a different levels of complexity. Nevertheless, we believe that CMW tools for highly structured CMLs should be based on (at least) three different access modes:

- a *unstructured access mode* to view/edit the unstructured content;
- a *fully-structured access mode* to view/edit the complete structured content; and
- a *lightly-structured access mode* to view/edit (part of) the structured content via simple templates.

We propose these three modes only for highly structured CMLs as the distinction between fully-structured and lightly-structured access modes may become unclear in case of simple CMLs with informal semantics such as concept maps. In these cases the fully structured representation is often simple enough to be directly accessible also by domain experts.

The advantage of providing two distinct modalities to access the structured content of a wiki page lies in the ability of providing an access to the conceptual model to both domain experts and knowledge engineers. In this way domain

experts can not only have access to the knowledge inserted by knowledge engineers, but can also comment or directly modify part of it. An important aspect of the implementation of a CMW is therefore the design of appropriate access modes, which can be based on templates whose formats depend upon the CML used and also upon the degree of complexity handled by the domain experts. Examples of templates which can be used to provide a lightly-structured access mode are: (possibly simplified) verbalizations of OWL statements; simple flow diagrams which represent the main steps of a workflow (business process); matrixes which provide a diagrammatic representation of binary roles; and so on. Another important aspect in the implementation of a CMW is the interaction between the structured content s and the lightly-structured access mode. Differently from the unstructured access mode and fully-structured access mode where the content shown/edited within the access mode can be considered a one-to-one syntactic variant of the content stored in the page, this is not the case for the lightly-structured access mode. In fact, the content stored in the structured part may be too expressive or complex to be directly represented in the lightly-structured access mode. In

this case, functionalities must be provided to “translate” the structured content of the page in the simplified representation in the lightly-structured access mode, and vice-versa.

V. CONCEPTUAL MODELING WITH MoKi

MoKi [17] is a collaborative MediaWiki-based [18] tool for modeling ontological and procedural knowledge in an integrated manner. MoKi uses OWL (Description Logics) and BPMN as the reference CMLs for ontological and procedural knowledge respectively, and associates semantic terms of the two CMLs to wiki pages containing both unstructured and structured information, accessible using different access modes.

In this section, we present a completely revised version of MoKi, which extends the first release of the tool (see [6]) to be fully compliant with the architecture illustrated in Sections III–IV. The main changes w.r.t. [6] are:

- (i) a redesign of the content organization of the MoKi page, which now comprises an unstructured part and a structured part. This extends and replaces the simple representational languages used in [6], and enables to model rich semantic terms using expressive ontology and complex business process CMLs; and
- (ii) a new support for multi-mode access to the page content which implements the three different access modes described in Section IV. This extends and replaces the single template-based access mode provided in [6].

A. The MoKi page for a semantic term

Being a tool supporting the description of ontological and procedural knowledge according to OWL and BPMN, the types of semantic terms relevant for MoKi are *concepts*, *properties*, and *individuals* in the ontology, and *process* (in MoKi we use the term “process” as a synonym for (complex or simple) activity) in the process model. Each term belonging to one of these types is therefore associated to a MoKi page which, coherently with the discussion in Section III-A, is composed of an unstructured part and a structured part.

The unstructured part: This part contains text written following the standard MediaWiki markup format: in particular, it can contain plain text, possibly enriched by formatting information, links to other MoKi pages or to external resources, uploaded images, and so on. The format of this part of the page is the same for all the different semantic terms.

The structured part: This part, which is delimited by specific tags to separate it from the unstructured text, contains knowledge stored according to the CML adopted. In the current implementation, the structured part of a page describing an ontology term contains a RDF/XML serialization of a set of OWL statements formalising the term, while, similarly, the structured part of a page describing a BPMN process contains a serialization of the process diagram in the JavaScript Object Notation (JSON).

B. Supporting multi-mode access in MoKi

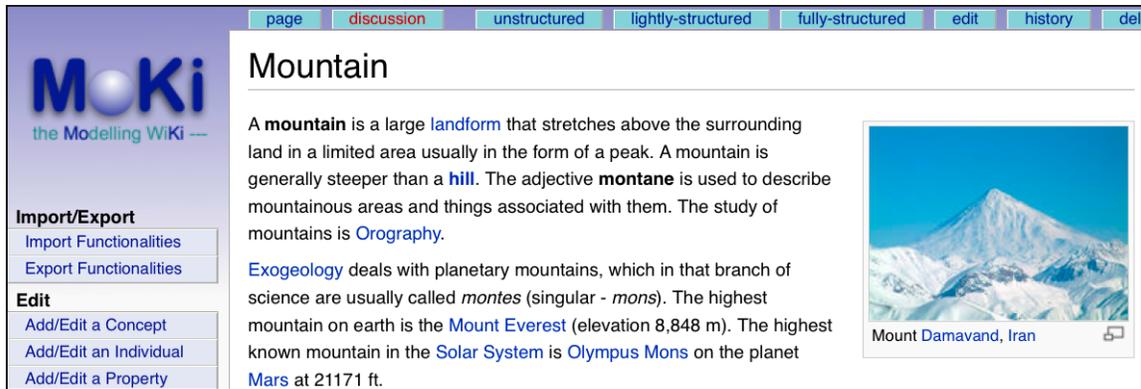
Users can access the ontological and procedural knowledge contained in MoKi using the three different access modes described in Section IV: one mode, the *unstructured access mode*, to access the unstructured part of a MoKi page, and two different modes, the *fully-structured access mode* and the *lightly-structured access mode*, to access the structured part.

The unstructured access mode: This access mode allows the user to edit/view the content of the unstructured part of the MoKi page of a semantic term. The editing/viewing of this part occurs in the standard MediaWiki way. Figure 5(a) shows the unstructured access mode of a portion of MoKi page describing the concept of “Mountain” (the content of the page in the figure is an excerpt taken from Wikipedia [15]).

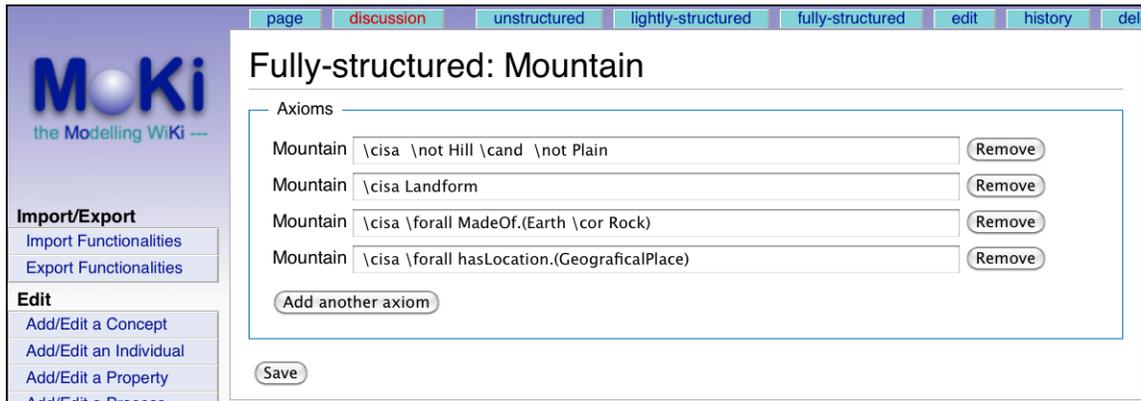
The fully-structured access mode: This access mode allows the user to edit/view the content of the structured part of a MoKi page using the full expressivity of the chosen CML. For ontological knowledge the fully-structured access mode allows the user to view/edit formal statements (axioms) describing the term associated to the page. Axioms are written according to the *latex2owl* syntax [19], an intuitive latex-style format for writing ontologies using a text-editor, format which can be automatically translated into (an RDF/XML serialization of) OWL. The *latex2owl* syntax was chosen because of its resemblance to the DL syntax; however, the approach illustrated here can be easily used to support a fully-structured access mode based on other OWL syntaxes such as the Manchester OWL syntax [20]. The user can easily edit the list of axioms in a form based interface, as the one shown in Figure 5(b). When saving the page, all axioms in the page are translated in OWL by the *latex2owl* tool, and the resulting code is stored in the structured part of the page. Conversely, when loading the page, the *owl2latex* tool translates the OWL code into statements adherent to the *latex2owl* syntax.

For procedural knowledge we have implemented an access mode that allows the user to edit the BPMN process diagram described in the page as shown in Figure 6(a). In particular we have tightly integrated in MoKi the Oryx editor [21], a full-fledged business process editor that allows to create processes according to several modeling languages, including BPMN.

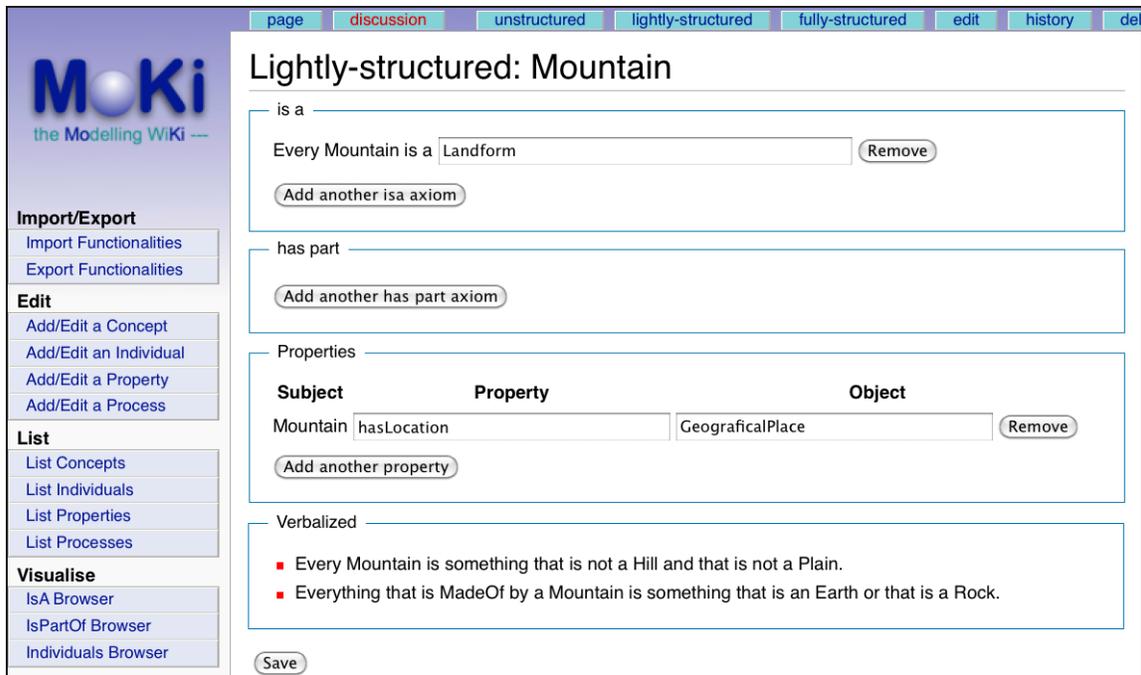
The lightly-structured access mode: As described in Section IV the purpose of this access mode is to allow users with limited knowledge engineering skills, to edit/view the content of the structured part of the MoKi page in a simplified and less formal way. For ontological knowledge the lightly-structured access mode is provided through a form made of two components, as depicted in Figure 5(c). In the top half part the user can view and edit simple statements which can be easily converted to/from OWL statements. For instance, in the case of concepts the user can edit statements



(a) Unstructured access mode.

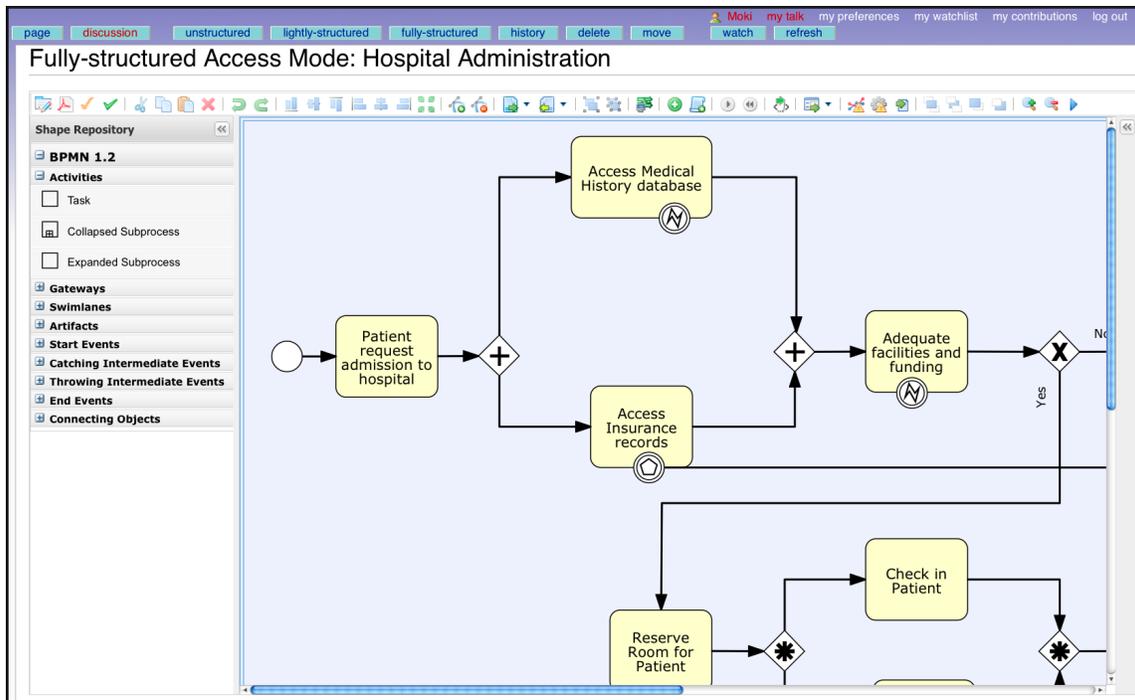


(b) Fully-structured access mode.

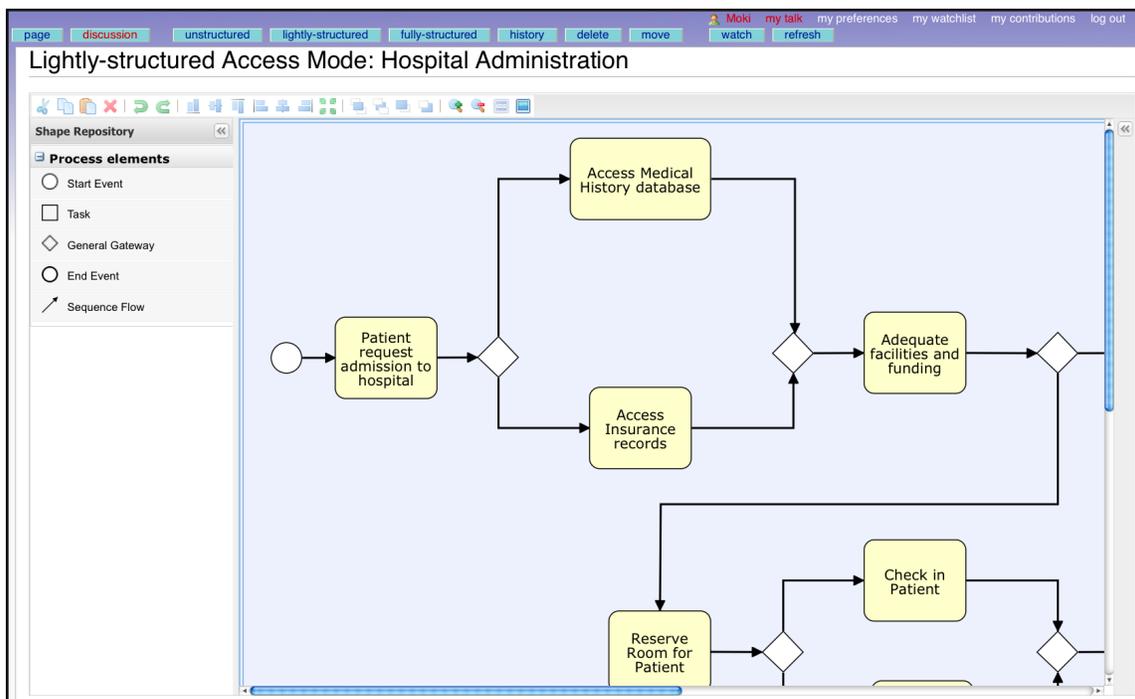


(c) Lightly-structured access mode.

Figure 5. Multi-mode access to the page of concept *Mountain*.



(a) Fully-structured access mode.



(b) Lightly-structured access mode.

Figure 6. Fully-structured access mode and lightly-structured access mode to the page of a process about Hospital Administration.

of the form “Every *subject* is a *object*”, “Every *subject* has as part a *object*”, or, more generally, statement of the forms (*subject*, *property*, *object*), which correspond to the

latex2owl statements “*subject* \cisa *object*”, “*subject* \cisa \exists exists hasPart.*object*”, and “*subject* \cisa \forall forall *property*.(*object*)” respectively. Analogous forms are provided

for properties and individuals. If the OWL version of any of these statements is already contained in the structured part of the page, then the corresponding fields are pre-filled with the appropriate content. Similarly, when any of these simple statements is modified in the lightly-structured access mode, the changes are propagated to the content of the structural part of the page. The bottom half of the form provides a description of those OWL statements which cannot be intuitively translated/edited as simple statements as the ones in the top half of the page. In the current implementation, this part contains the translation of those statements in Attempto Controlled English, provided by the OWL 2 Verbalizer [22]. The purpose of this bottom half of the form is to give the domain experts a flavour of the complex statements that a knowledge engineer has formalized. If a domain expert is doubtful about some of the statements, he/she can mark them and ask for a clarification using, e.g., the MediaWiki Discussion functionality.

For procedural knowledge we have implemented an access mode based on a light-weight graphical process editor such as the one shown in Figure 6(b). This editor shows only the basic workflow of the activity, and the main elements of the process such as start and end events, plus the sub-processes it can contain, hiding the details and complexity typical of BPMN diagrams. The challenges we are currently addressing are how to visualise in a simplified way a complex BPMN process stored in the structured part, and how to update the content of the structured part according to the changes performed in the lightly-structured access mode.

C. Organizational mechanism pages in MoKi

Organizational mechanism pages are MoKi special pages dynamically created from the (structured) content of the semantic term pages. Differently from wiki pages for terms, which are mainly constructed using textual representations, the organizational mechanism rely also on graphical forms of representation, which include graphical browsing and editing facilities. For ontological knowledge the organizational mechanism pages allow to explore the generalization and part/subparts decomposition hierarchies of ontology concepts, as well as the classification of the ontology individuals. In particular, MoKi provides two kinds of organizational mechanism pages. In the tabular-based one, the user can access a table listing every concept (resp. individual) of the ontology together with the concepts of which it is a specialization and the concepts in which it decomposes according to the part of relation (resp. the concepts to which the individual belongs to). In the graphical-based one, a tree-like view shows the hierarchy of concepts according to either the subclass (see Figure 7) or the part-of relation, or the membership of individuals to concepts. Drag and drop editing facilities are also provided to rearrange the tree. For procedural knowledge, the current organizational mechanism page provides an overview of the activity/sub-

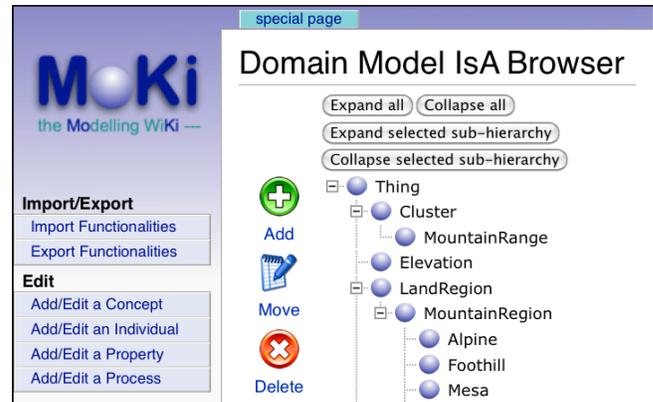


Figure 7. Example of organizational mechanism (generalization) page: the IsA Browser.

activity decomposition mechanism by means of a tree-based view as well as a table listing every process defined in MoKi together with the processes in which it decomposes. Already planned work aims at including a workflow-based representation of the before/after abstraction mechanism, which, in the current version, is limited to the description of the sub-process which represent how a complex activity is structured, as depicted in Figure 6.

VI. USAGES OF MoKi

MoKi has been successfully applied in several application contexts. Next, we report some of them, describing the specific tasks for which MoKi was used, as well as some of the findings and insights we observed. Detailed users experiments and evaluations are reported in [23], [24].

A. APOSDLE Project

The APOSDLE EU project [25] developed a software platform and tools to support the process of *learning@work*, that is learning within the context of the immediate work of a user and within her current work environment. To deliver a user with context-sensitive learning material or suggestions, tailored to her specific needs, the APOSDLE system needs to know not only the profile of the user, but also about various aspects of the sphere in which the user is acting, spanning from the specific domain of knowledge in which the user is working, the tasks (aka processes) the user can perform, the learning goals she can have, and also the material available to compose adequate learning suggestions.

An early version of MoKi [26] have been successfully applied to develop integrated models of ontology and process for the purpose of initialising and serving as the knowledge back-end of the APOSDLE platform. Six different domains were considered: Information and Consulting on Industrial Property Rights (94 concepts and 2 properties; 13 processes), Electromagnetism Simulation (115 concepts and 21 properties; 13 processes), Innovation and Knowledge Management

(146 domain concepts and 5 properties; 31 processes), Requirements Engineering (the RESCUE methodology) (78 concepts and 2 properties; 77 processes), Statistical Data Analysis (69 concepts and 2 properties; 10 processes) and Information Technology Infrastructure Library (100 concepts and 2 properties; no processes).

The modelling activities involved people with different modelling skills and levels of expertise of the application domains, and located in different places all over Europe. The evaluation of the version of MoKi used in the project showed that the users highly appreciated the form-based interface of MoKi, and the fact that they were able to participate in the creation of the models without having to know any particular syntax or deep knowledge engineering skills. Thus, MoKi was perceived as an adequate tool to actively involve domain experts in the modelling process. People with some knowledge engineering skills found MoKi as comfortable to use as other state-of-the-art modelling tools, and MoKi helped the users in structuring and formalizing their knowledge in a simple, intuitive and efficient manner. Particularly appreciated have been the functionalities, in particular the graphical ones, which allow to browse the models along the organizational abstractions.

B. PESCaDO Project

Citizens are increasingly aware of the influence of environmental and meteorological conditions on the quality of their life. One of the consequences of this awareness is the demand for high quality environmental information and decision support that is tailored (i.e., personalized) to ones specific context and background (e.g., health conditions, travel preferences). Personalized environmental information may need to cover a variety of aspects (e.g., meteorology, air quality, pollen) and take into account a number of specific personal attributes of the user (e.g., health, age, allergies), as well as the intended use of the information. The goal of the PESCaDO EU project [27] is to develop a multilingual web-service platform providing personalized environmental information and decision support.

The backbone of the PESCaDO platform [28] is an environmental ontology-based knowledge base where all the information relevant for a user request are dynamically instantiated. The ontology formalizes a variety of aspects related to the application context: environmental data, data sources, user requests, user profiles, warnings and recommendations triggered by environmental conditions, and so on. MoKi is being exploited by some knowledge engineers in the consortium to support the construction and revision of this ontology, also by exploiting some automatic ontology concepts extraction techniques offered by the tool [29], [30] (functionalities that may also be exploited for terminologically evaluating ontologies [31]). The current version of the ontology consists of 241 concepts, 151 object properties, 43 datatype properties, and 672 individuals.

C. Organic.Edunet and Organic.Lingua Projects

Organic.Edunet [32] was a EU project that aimed to facilitate access, usage and exploitation of digital educational content related to Organic Agriculture (OA) and Agroecology. It deployed a multilingual online federation of learning repositories, populated with quality content from various content producers. In addition, it deployed a multilingual online environment (the Organic.Edunet portal) that facilitate end-users search, retrieval, access and use of the content in the learning repositories.

MoKi was used by a team of knowledge engineers and domain experts to collaboratively build and revise the Organic Agriculture and Agroecology Ontology (61 concepts, 30 properties, and 222 individuals) at the core of the Organic.Edunet portal. The experience was perceived as positive enough by the user to favour the adoption of MoKi as the central modelling tool in the follow-up project, Organic.Lingua.

Organic.Lingua [33] is a EU project aiming to enhance the existing Organic.Edunet Web portal with educational content on Organic Agriculture (OA) and Agroecology (AE), introducing automated multi-lingual services that will further support the uptake of the portal from its targeted audiences, facilitate the multilingual features of the portal, and further extend its geographical and linguistic coverage.

The version of MoKi that has been customized for Organic.Lingua addresses the challenges posed by the project, that is to manage the multilingual aspects of the ontology used to tag the resources deployed on the portal, and the internationalization of the tool itself. Therefore, the Organic.Lingua MoKi implements features that permit to manage: the translations of each ontology entity name and description, the discussions about the changes that have to be carried out on the ontology, and the translations of the interface labels.

D. ProDe Project

ProDe [34] is an Italian inter-regional project with the aim of defining a national reference model for the management of electronic documentation (dematerialized document) in the Public Administration. This reference model follows an archival science perspective, and can be used for the identification of guidelines and functions needed to safely store, classify, manage, and retrieve, electronic documents produced within the PA in an archival system.

A customized version of MoKi [35] supported teams of users (both domain experts and knowledge engineers involved in the project) in the construction of the reference model, which consisted of an *ontological* part, formalizing document management and organizational aspects, i.e., document archiving-related aspects and the offices and profiles involved in the dematerialization, as well as a *process* part, describing the activities which produce (manage, consume) documents.

A quantitative and qualitative evaluation of the usage of MoKi in ProDe was performed [24]. Users perceived the tool as more than easy to use, and positively rated the overall usefulness of the tool for the collaborative modeling of documents and processes.

E. ProMo Project

ProMo [36] is an industrial project (FESR) founded by the province of Trento that aims at the development of a platform that supports the collaborative modelling of the processes, the structure, the actors, and the artifacts of a complex organization like a Public Administration, the grounding of the objects in this abstract model to the technological layer of the organization (services execution platform), and the support for monitoring the process execution.

MoKi is currently used by a team knowledge engineers to formally describe some exemplar processes of the Public Administration, as for instance the procedure for registering new citizens in a town hall. These processes are enriched with entities defined in a domain ontology, defined in MoKi as well, that provides the domain knowledge semantic of the processes elements, in line with the semantic annotation of business processes approach presented in [37].

F. OncoCure and eOnco Projects

The general aim of the OncoCure project [38], [39] was to use innovative ICT-based methods and models for clinical governance in oncology, by designing and developing a system for supporting and controlling the best evidence-based oncological care process. The system is based on electronic guidelines and recommendations, to be integrated with clinical information systems that manage the oncological patients. The main goal was the design and development of a prescriptive decision support system (DSS) for clinicians during the care process, based on the execution of AIOM (Associazione Italiana di Oncologia Medica) guidelines for breast cancer formalized in the Asbru, a plan-specification language for defining clinical protocols.

Although MoKi as presented here is tailored to the development of ontologies and business processes, a preliminary and customized version of the tool that supports the modelling of clinical protocols in the ASBRU modelling language is described in [40]. This version of MoKi, called *Clip-MoKi*, provides support for modelling the key elements of an Asbru model (e.g. plans, parameters, <http://www.asbrusoft.com/>) as wiki pages, and for exploring the models created according to the organizational mechanisms for structuring knowledge provided by the language (e.g. the plan/plan children decomposition).

The eOnco project [41] had the main objective of supporting knowledge intensive management of cure process in Oncology. One of the aspects of the project concerned the elicitation and formal representation of the activities

done by the nurses in a ward, to investigate for instance the bottlenecks, performance-wise, of the nurses work.

A multi-disciplinary team composed of sociologists and knowledge engineers participated to the formalization of these activities in MoKi, which resulted in 10 BPMN processes consisting of 140 activities.

VII. RELATED WORK

To the best of our knowledge, there are no works in the literature that explicitly address the problem of defining a reference architectural model for wiki-based conceptual modeling tools. Partially related to the content of this paper is the work in [42], which investigates *concept* modeling approaches applied in semantic wikis, that is design patterns to organize data in wikis (e.g. RDF Modeling, Relational Modeling, Rule Modeling).

Focusing on tools, wiki systems and semantic wikis have been mainly applied to support collaborative creation and sharing of ontological knowledge. *AceWiki* [43] was developed in the context of logic verbalization, that is, the effort to verbalise formal logic statements into English statements and vice-versa. *AceWiki* is based on Attempto Controlled English (ACE), which allows users expressing their knowledge in near natural language (i.e. natural language with some restrictions). *Semantic MediaWiki+* [5], which includes the Halo Extension, is a further extension on Semantic MediaWiki with a focus on enhanced usability for semantic features. Especially, it supports the annotation of whole pages and parts of text, and offers “knowledge gardening” functionalities, that is maintenance scripts at the semantic level, with the aim to detect inconsistent annotations, near-duplicate entries etc. *IkeWiki* [44] supports the semantic annotation of pages and semantic links between pages. Annotations are used for context-specific presentation of pages, advanced querying, consistency verification or drawing conclusions. *OntoWiki* [7] seems to focus slightly more directly on the creation of a semantic knowledge base, and offers widgets to edit/author single elements/pages and whole statements (subject, predicate, object). Finally, a proposal of modeling straightforward workflows using Semantic MediaWiki is implemented in the *Semantic Result Formats* extension [45].

We have compared the tools mentioned above, plus the previous and current versions of MoKi, against the distinctive characteristics of our reference architecture. The results are displayed in the Table I. The columns of the table refer to the capability of: (i) associating a page to a semantic term (*one page/one term*); (ii) browsing / overviewing the model according to the some organizational mechanism (*overview*); (iii) describing a semantic term using both unstructured and structured content (*unstructured/structured*); (iv) accessing content in a multi-mode manner (*multi-mode*); and (v) defining models according to two or more (substantially different) CMLs (*multiple CMLs*).

Table I
COMPARISON OF STATE-OF-THE-ART MODELLING WIKIS.

	one page/ one term	overview	unstructured/ structured	multi-mode	multiple CMLs
AceWiki	✓				
SMW+	✓	✓	✓		
IkeWiki	✓		✓		
OntoWiki	✓	✓	✓		
Semantic Result Formats	✓	✓	✓		
MoKi v.1	✓	✓	✓		✓
MoKi v.2	✓	✓	✓	✓	✓

As we can see from Table I, the proposed architectural model takes into account typical characteristics of wiki based-tools for conceptual modeling, pointed out by the first three columns of the table, and enriches them with two novel aspects, namely the multi-mode access to pages and the focus on multiple CMLs.

VIII. CONCLUSIONS

In this paper, we have presented a reference architectural model for wiki-based conceptual modeling tools grounded on three distinctive characteristics; (i) the use of wiki pages to mimic the basic building blocks of conceptual modeling languages; (ii) the structuring of wiki pages for semantic terms in an unstructured part and a structured part; and (iii) a multi-mode access to the pages to support easy usage both by domain experts and knowledge engineers. We have also described a fully revised version of MoKi which complies with the proposed architectural model. Several application contexts in which MoKi has been successfully applied have also been presented.

In our future work, we aim at improving the support for process modeling, in particular in providing an extensive automatic support for aligning the fully-structured access mode and lightly-structured access mode for procedural knowledge. One of the key aspects on which we are currently working is on enhancing the support for collaboration between people who model at different levels of abstraction: in particular, we are implementing facilities to highlight changes across the different access modes, to make domain experts aware of the changes introduced by knowledge engineers and vice-versa.

ACKNOWLEDGEMENTS

APOSDLE was funded under the FP6 ICT Support Program of the EU Commission (Grant Agreement Number FP6-IST-2004-027023). PESCaDO is funded under the ICT Support Program of the EU Commission (Grant Agreement Number FP7-248594). Organic.Lingua is funded under the FP7 ICT Support Program of the EU Commission (Grant Agreement Number 270999). ProDe is an Italian inter-regional project. OncoCure was funded by the Fondazione

Caritro of Trento. ProMo is funded by the Operational Programme FESR 2007-2013 of the Province of Trento.

REFERENCES

- [1] C. Ghidini, M. Rospocher, and L. Serafini, "Conceptual modeling in wikis: a reference architecture and a tool," in *The Fourth International Conference on Information, Process, and Knowledge Management (eKNOW2012), Valencia, Spain, 2012*, pp. 128–135. [Online]. Available: http://www.thinkmind.org/download.php?articleid=eknow_2012_6_10_60015
- [2] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "Dbpedia - a crystallization point for the web of data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.
- [3] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago - a core of semantic knowledge," in *16th international World Wide Web conference (WWW 2007)*, 2007.
- [4] M. Krötzsch, D. Vrandečić, M. Völkel, H. Haller, and R. Studer, "Semantic wikipedia," *Journal of Web Semantics*, vol. 5, pp. 251–261, 2007.
- [5] D. Hansch and H.-P. Schnurr, "Practical applications of semantic mediawiki in commercial environments - case study: semantic-based project management," in *3rd European Semantic Technology Conference (ESTC2009)*, 2009.
- [6] C. Ghidini, B. Kump, S. Lindstaedt, N. Mahbub, V. Pammer, M. Rospocher, and L. Serafini, "MoKi: The Enterprise Modelling Wiki," in *Proceedings of ESWC 2009*, ser. LNCS, vol. 5554. Springer, 2009, pp. 831–835, demo Session.
- [7] S. Auer, S. Dietzold, and T. Riechert, "Ontowiki - a tool for social, semantic collaboration," in *Proceedings of the 5th International Semantic Web Conference, Nov 5th-9th, Athens, GA, USA*, vol. 4273. Springer, 2006, pp. 736–749.
- [8] A. Oltramari and G. Vetere, "Lexicon and ontology interplay in senso comune," in *Proceedings of OntoLex 2008 (Hosted by Sixth international conference on Language Resources and Evaluation)*, Marrakech (Morocco), 2008.

- [9] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proc. of the 2008 ACM SIGMOD international conference on Management of data*. New York, ACM, 2008, pp. 1247–1250. [Online]. Available: <http://ids.snu.ac.kr/w/images/9/98/SC17.pdf>
- [10] T. Schandl and A. Blumauer, "Poolparty: Skos thesaurus management utilizing linked data," in *The Semantic Web: Research and Applications*, ser. LNCS, vol. 6089, 2010, pp. 421–425.
- [11] V. Dimitrova, R. Denaux, G. Hart, C. Dolbear, I. Holt, and A. G. Cohn, "Involving domain experts in authoring owl ontologies," in *Proceedings of the 7th Int. Semantic Web Conference (ISWC 2008)*, ser. LNCS, vol. 5318/2010. Springer Berlin / Heidelberg, 2008, pp. 1–16.
- [12] M. Hammer and D. McLeod, "Database description with sdm: A semantic database model," *ACM Trans. Database Syst.*, vol. 6, no. 3, pp. 351–386, 1981.
- [13] J. Mylopoulos, "Information modeling in the time of the revolution," *Information Systems*, vol. 23, no. 3-4, June 1998.
- [14] OMG, "Business process modeling notation, v1.1," www.omg.org/spec/BPMN/1.1/PDF.
- [15] Wikipedia, "Mountain — Wikipedia, the free encyclopedia," 2012, [Online; accessed 09-July-2012]. [Online]. Available: <http://en.wikipedia.org/w/index.php?title=Mountain&oldid=500787709>
- [16] M. K. Smith, C. Welty, and D. L. McGuinness, "Owl web ontology language guide," W3C Recommendation, 10 February 2004.
- [17] DKM Unit, "MoKi: the MOdelling wiKI," [Online; accessed 09-July-2012]. [Online]. Available: <https://moki.fbk.eu/>
- [18] Wikimedia Foundation, "Mediawiki," <http://www.mediawiki.org>.
- [19] D. Unit, "Latex2owl," 2012, [Online; accessed 09-July-2012]. [Online]. Available: <http://dkm.fbk.eu/index.php/Latex2owl>
- [20] CO-ODE, "The manchester owl syntax," 2012, [Online; accessed 09-July-2012]. [Online]. Available: http://www.co-ode.org/resources/reference/manchester_syntax/
- [21] University of Potsdam, "Oryx - web-based graphical business process editor," 2012, [Online; accessed 09-July-2012]. [Online]. Available: <http://code.google.com/p/oryx-editor/>
- [22] K. Kaljurand and N. Fuchs, "Verbalizing OWL in Attempto Controlled English," in *Proceedings of Third International Workshop on OWL: Experiences and Directions (OWLED 2007)*, 2007.
- [23] C. Di Francescomarino, C. Ghidini, and M. Rospocher, "Evaluating wiki-enhanced ontology authoring," in *The 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW2012)*, Galway, Ireland, 2012.
- [24] C. Casagni, C. Di Francescomarino, M. Dragoni, L. Fiorentini, L. Franci, M. Gerosa, C. Ghidini, F. Rizzoli, M. Rospocher, A. Rovella, L. Serafini, S. Sparaco, and A. Tabarroni, "Wiki-based conceptual modeling: An experience with the public administration," in *The Semantic Web – ISWC 2011*, ser. LNCS. Springer, 2011, vol. 7032, pp. 17–32.
- [25] APOSDLE Consortium, "Advanced Process-Oriented Self-Directed Learning Environment (APOSDLE)," 2006-2010, [Online; accessed 09-July-2012]. [Online]. Available: <http://www.aposdle.org/>
- [26] C. Ghidini, M. Rospocher, L. Serafini, B. Kump, V. Pammer, A. Faatz, A. Zinnen, J. Guss, and S. Lindstaedt, "Collaborative knowledge engineering via semantic mediawiki," in *International Conference on Semantic Systems (I-SEMANTICS '08)*, S. Auer, S. Schaffert, and T. Pellegrini, Eds., Graz, Austria, september 3–5 2008, pp. 134–141.
- [27] PESCaDO Consortium, "Personalized Environmental Service Configuration and Delivery Orchestration (PESCaDO)," 2010-2012, [Online; accessed 09-July-2012]. [Online]. Available: <http://www.pescado-project.eu/>
- [28] L. Wanner, S. Vrochidis, S. Tonelli, J. Moßgraber, H. Bosch, A. Karppinen, M. Myllynen, M. Rospocher, N. Bouayad-Agha, U. Bügel, G. Casamayor, T. Ertl, I. Kompatsiaris, T. Koskentalo, S. Mille, A. Moutzidou, E. Pianta, H. Saggion, L. Serafini, and V. Tarvainen, "Building an environmental information system for personalized content delivery," in *Proceedings of the 9th IFIP WG 5.11 International Symposium on Environmental Software Systems - Frameworks of eEnvironment (ISESS 2011)*, Brno, Czech Republic, June 27-29, 2011, ser. IFIP Advances In Information And Communication Technology, vol. 359. Springer, 2011, pp. 169–176. [Online]. Available: http://link.springer.com/content/pdf/10.1007%2F978-3-642-22285-6_19
- [29] M. Rospocher, S. Tonelli, L. Serafini, and E. Pianta, "Key-concept extraction for ontology engineering," in *Knowledge Engineering and Knowledge Management: 18th International Conference on Knowledge Engineering and Knowledge Management - EKAW 2012, Galway, Ireland, 2012*, vol. 7603. Springer Berlin Heidelberg, 2012, pp. 413–416. [Online]. Available: http://link.springer.com/content/pdf/10.1007%2F978-3-642-33876-2_36
- [30] S. Tonelli, M. Rospocher, E. Pianta, and L. Serafini, "Boosting collaborative ontology building with key-concept extraction," in *Proceedings of 5th IEEE International Conference on Semantic Computing (September 18-21, 2011 - Palo Alto, USA)*, 2011.
- [31] M. Rospocher, S. Tonelli, L. Serafini, and E. Pianta, "Corpus-based terminological evaluation of ontologies," *Applied Ontology*, vol. 7, no. 4, pp. 429–448, 2012. [Online]. Available: <http://iospress.metapress.com/content/3356r67874680284/>
- [32] Organic.Edunet Consortium, "Organic.Edunet: A multilingual Web portal for sustainable agriculture & environmental education," 2007-2010, [Online; accessed 09-July-2012]. [Online]. Available: <http://www.organic-edunet.eu/>

- [33] Organic.Lingua Consortium, "Organic.Lingua - Demonstrating the potential of a multilingual Web portal for Sustainable Agricultural & Environmental Education," 2011-2014, [Online; accessed 09-July-2012]. [Online]. Available: <http://www.organic-lingua.eu/>
- [34] ProDe Consortium, "Progetto Dematerializzazione (ProDe)," 2010-2012, [Online; accessed 09-July-2012]. [Online]. Available: <http://www.progettoprode.it/>
- [35] C. Di Francescomarino, M. Dragoni, M. Gerosa, C. Ghidini, M. Rospocher, and M. Trainotti, "Achieving interoperability through semantic technologies in the public administration," in *The Semantic Web: Research and Applications*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2012, vol. 7295, pp. 793–807.
- [36] ProMo Consortium, "ProMo - Un Approccio Collaborativo e Agile per la Modellazione e il Monitoraggio di Processi di Business Basati su Servizi," 2012-2013, [Online; accessed 09-July-2012]. [Online]. Available: <http://dkm.fbk.eu/index.php/ProMo>
- [37] C. Di Francescomarino, C. Ghidini, M. Rospocher, L. Serafini, and P. Tonella, "Reasoning on semantically annotated processes," in *Proceedings of the 6th International Conference on Service Oriented Computing (ICSOC'08)*, Sydney, Australia, 2008.
- [38] OncoCure Consortium, "OncoCure - Information and Communication Technologies for Clinical Governance in Oncology," 2007-2009, [Online; accessed 09-July-2012]. [Online]. Available: <http://www.cvast.tuwien.ac.at/projects/OncoCure/>
- [39] C. Eccher, A. Ferro, A. Seyfang, M. Rospocher, and S. Miksch, "Modeling clinical protocols using semantic MediaWiki: the case of the Oncocure project," in *ECAI workshop on Knowledge Management for Healthcare Processes (K4HeLP)*, 2008.
- [40] M. Rospocher, C. Eccher, C. Ghidini, R. Hasan, A. Seyfang, A. Ferro, and S. Miksch, "Collaborative encoding of asbru clinical protocols," in *Electronic Healthcare*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, M. Szomszor and P. Kostkova, Eds. Springer Berlin Heidelberg, 2012, vol. 69, pp. 135–143. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23635-8_17
- [41] eOnco Consortium, "eOnco Project," 2009-2011, [Online; accessed 09-July-2012]. [Online]. Available: <https://dkm.fbk.eu/index.php/Projects>
- [42] J. Bao, L. Ding, D. L. McGuinness, and J. Hendler, "Towards social webtops using semantic wiki," in *International Semantic Web Conference (Posters & Demos)*, ser. CEUR Workshop Proceedings, C. Bizer and A. Joshi, Eds., vol. 401. CEUR-WS.org, 2008. [Online]. Available: <http://dblp.uni-trier.de/db/conf/semweb/iswc2008p.html#BaoDMH08>
- [43] T. Kuhn, "AceWiki: A Natural and Expressive Semantic Wiki," in *Proceedings of Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges*, 2008.
- [44] S. Schaffert, "Ikewiki: A semantic wiki for collaborative knowledge management," in *1st Int. Ws. on Semantic Technologies in Collaborative Applications (STICA'06)*. [Online]. Available: http://www.wastl.net/download/paper/schaffert06_ikewiki.pdf
- [45] F. Dengler, S. Lamparter, M. Hefke, and A. Abecker, "Collaborative process development using semantic mediawiki," in *5th Conference of Professional Knowledge Management. Solothurn, Switzerland, 2009*.