Conceptual Modeling in Wikis: a Reference Architecture and a Tool

Chiara Ghidini, Marco Rospocher, 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,rospocher,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 organisational 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 describe MoKi, a conceptual modeling wiki for ontologies and business processes fully compliant with the presented reference architecture.**

***Keywords*-Conceptual Modeling, Collaborative Modeling, Semantic Wikis, Ontology Modeling, Process Modeling**

I. INTRODUCTION

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. While traditional wikis allow to enter unstructured text and multimedia content directed to other human users, and not in a format apt to be understood by computers, recent projects, such as DBpedia [1], YAGO [2], and Semantic Media Wiki (SMW) [3] have empowered traditional wikis with the capability of publishing their 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+ [4], MoKi [5], and Ontowiki [6]). and has suggested the usage of the wiki philosophy in tools which are not directly built on top of wikis (e.g., Senso Comune [7], Freebase [8], and PoolParty [9]).

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. 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, 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 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 scenarios composed of different models. *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 [10], 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.*

The contribution of this paper is twofold. 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

 ISBN: 978-1-61208-181-6

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 both by domain experts and knowledge engineers. Second, we illustrate an implementation of this architecture in MoKi, a wiki for modeling ontologies and business processes. This implementation aims at showing the feasibility of the architecture by means of a practical realization.

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 engineers; at the *implementation* level, MoKi provides 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 (§II) and we proceed by defining an architecture which satisfies the needs of generality and collaboration (§III and §IV). We then provide a description of MoKi (§V) and we conclude with a comparison between the proposed architecture and state of the art tools for wiki-based conceptual modeling (§VI).

II. CONCEPTUAL MODELING

Conceptual modeling (aka semantic modeling) has been researched into and used in several areas of Computer Science and Engineering often with different usages, characterizations, and terminologies. According to [11] and [12], 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, [12] 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. They are used to describe different types of concepts, such as Entities, Activities, Agents, Goals, and so on, depending on the CLM used; and
2) **Organisational mechanisms**: these are primitive mechanisms for structuring the model along different dimensions. Examples of organisational mechanisms (also called *abstraction mechanisms* in [12]) are generalization (often referred to as *isA*), aggregation (*partOf*), classification (*instanceOf*), contextualisation / modularization, and so on.

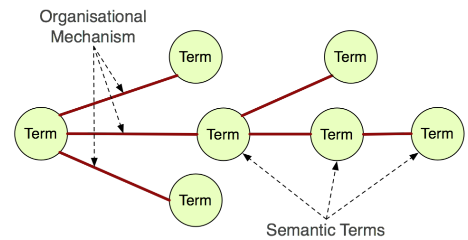

Figure 1. Conceptual Modeling Languages.

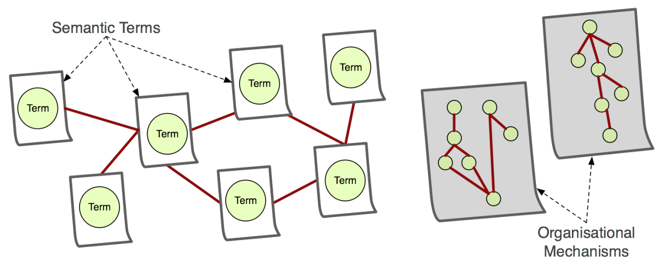

Figure 2. Representing a conceptual model in a wiki.

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 extremely informal semantics (in some cases even none) is usually associated to the diagrams. On the contrary, if the models need to be as less ambiguous as possible, 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 there are "semi-formal" CMLs: an example is the Business Process Modeling Notation [13], 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 organisational mechanisms. In this section we introduce the notion of Conceptual Modeling Wiki (CMW) which uses wiki pages to represent these building blocks.

A pictorial representation of a CMW is given in Figure 2. A CMW 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 organisational mechanism. For instance, if

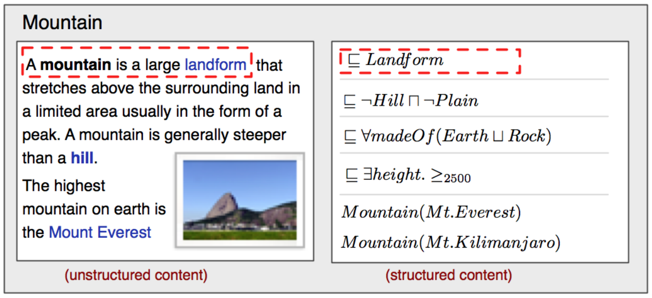

Figure 3. Wiki page for semantic terms.

we consider a CML having as semantic terms concepts, instances, and roles, and two organisational mechanisms such as generalisation 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 generalisation and the aggregation/decomposition dimensions respectively.

A. Building Wiki Pages for Terms

The idea of associating a wiki page to each semantic term is adopted by most of the state of the art wiki-based tools used to represent and manage knowledge (see §VI). 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 at http://en.wikipedia.org/wiki/Mountain, 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 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, 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.

The advantage of storing the unstructured and structured descriptions within 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 organisation of a page in an unstructured and structured part is a second important step in defining the architecture of a CMW, but may not be enough in the case of complex CMLs, such as the ones based on logical formalisms (e.g., OWL [14]) or very complex notations (e.g., BPMN [13]). In this case the structured part of the page will contain very precise (often logic based) description 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.

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 (structured) knowledge. Examples are: different access modes which represent the OWL structured content using different syntax, controlled natural languages, or graphical representations. Analogously 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 or complex 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.

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 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 is a collaborative, MediaWiki-based [15], 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 any instantiation of the 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 an implementation of MoKi (see also [5]), fully compliant with the architecture illustrated in §III–IV. A running installation of MoKi can be tested on-line at https://moki.fbk.eu/moki/tryitout2.0.

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* (we use this term 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 §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.

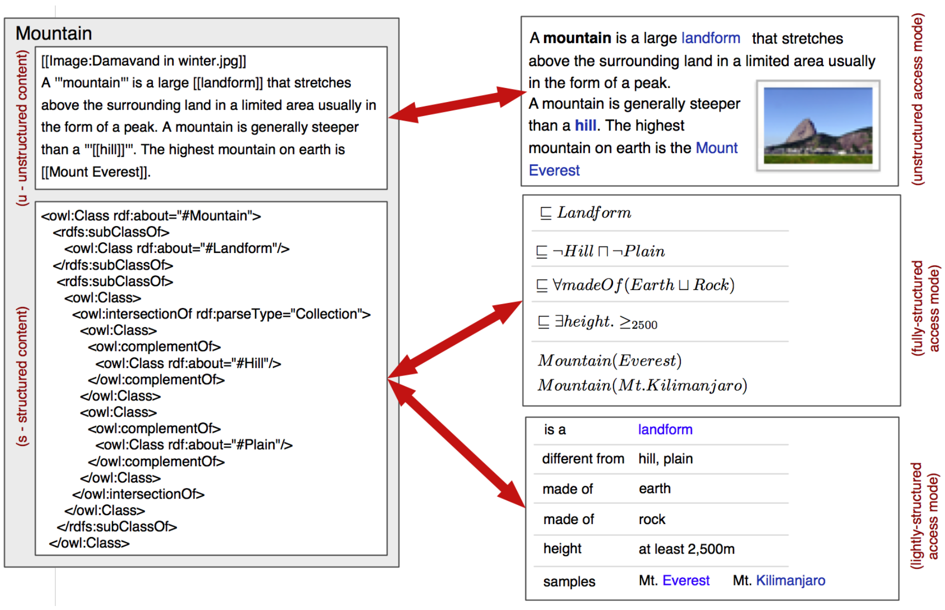

Figure 4. Multi-mode access to a wiki page for semantic terms.

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 serialisation of a set of OWL statements formalising the term, while, similarly, the structured part of a page describing a BPMN process contains an XML serialisation of the JSON object representing the process diagram.

B. *Multi-mode access in* MoKi

Users can access the ontological and procedural knowledge contained in MoKi using the three different access modes described in §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.

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, an intuitive latex-style format for writing ontologies using a text-editor, format which can be automatically translated into (an RDF/XML serialisation of) OWL. The latex2owl syntax was chosen because of its resemblance to the DL syntax, but the tool can be adapted to handle fully-structured access mode based on other OWL syntaxes such as the Manchester OWL syntax. The user can easily edit the list of axioms in a form based interface, as the one shown in the top part of Figure 5. 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 the top part of Figure 6. In particular we have tightly integrated in MoKi the Oryx editor [16], a full-fledged editor that allows to create processes according to several modeling languages, including BPMN.

The lightly-structured access mode: As described in §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 the bottom part of Figure 5. 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 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 hasPart.*object*", and "*subject* \cisa

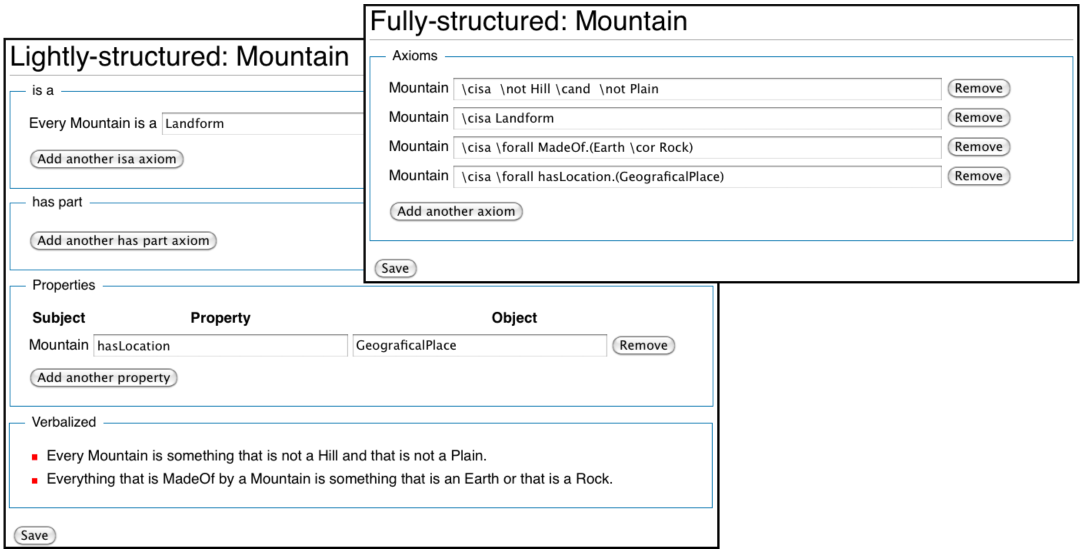

Figure 5. Fully-structured access mode and lightly-structured access mode to the page of concept *Mountain*.

\forall *property*.(*object*)". 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 [17]. 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 the Oryx editor (see the bottom part of Figure 6) which shows only the basic workflow of the activity, the main elements of the process such as start and end events and the (sub-)processes it can contain, hiding the details and complexity typical of BPMN diagrams.

C. *Organisational mechanisms in* MoKi

Organisational 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 organisational mechanism rely also on graphical forms of representation, which include graphical browsing and editing facilities. For ontological knowledge the organisational mechanism pages allow to explore and edit the generalisation and part/subparts decomposition hierarchies of ontology concepts, as well as the classification of the ontology individuals. For procedural knowledge, the current organisational mechanism pages provide an overview of the activity/sub-activity decomposition, and 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. 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.

Focusing on tools, wiki systems and semantic wikis have been mainly applied to support collaborative creation and sharing of ontological knowledge. *AceWiki* [18] was developed in the context of logic verbalisation, 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+* [4], 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* [19] supports

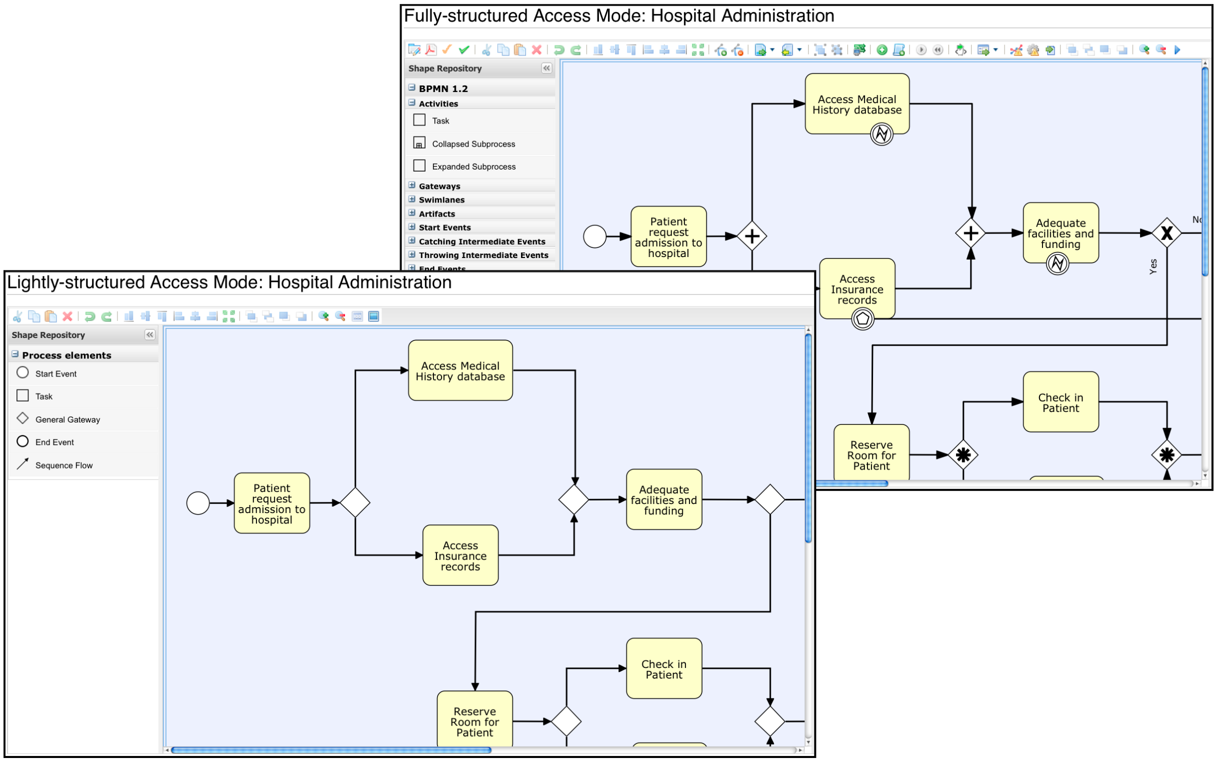

Figure 6. Fully-structured access mode and lightly-structured access mode of a process.

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* [6] 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 workflows using Semantic MediaWiki is implemented in the *Semantic Result Formats* extension [20].

We have compared the tools mentioned above, together with the current and previous versions of MoKi (a preliminary version of the tool was presented in [21]), against the distinctive characteristics of our reference architecture. The results are displayed in Table I, where the columns 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 organisational 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*).

As we can see from the table, 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 CLMs.

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

	1 page/ 1 term	overview	unstruct./ struct.	multi-mode	multiple CMLs
AceWiki	X				
SMW+	X	X	X		
IkeWiki	X		X		
OntoWiki	X	X	X		
Sem. Res. Form.	X	X	X		
MoKi v.1	X	X	X		X
MoKi v.2	X	X	X	X	X

VII. CONCLUDING REMARKS

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 an implementation of MoKi fully compliant with the proposed architectural model.

A customized version of MoKi described in this paper is being successfully used by domain experts in five Italian regions within the ProDE National project to develop models of documental flows in five different sectors of the Public Administration (PA). The models are composed of a process model, describing the flow of activities carried out in a PA sector, and of a domain ontology, describing the documents generated / used by the different activities, and the roles performing the different activities. A qualitative evaluation on the entire modeling process has been performed [22], and the results shows that the domain experts perceived the tool as more than *easy to use*, and *useful* for the collaborative modeling of documents and processes. A customized version of the tool, called *CliP*-MoKi, has also been applied to model clinical protocols encoded in the ASBRU language [23].

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. 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.

REFERENCES

[1] 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 WWW*, vol. 7, no. 3, pp. 154–165, 2009.

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago - a core of semantic knowledge," in *16th international World Wide Web conference (WWW 2007)*, 2007.

[3] M. Krötzsch, D. Vrandecic, M. Völkel, H. Haller, and R. Studer, "Semantic wikipedia," *Journal of Web Semantics*, vol. 5, pp. 251–261, 2007.

[4] 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.

[5] C. Ghidini, M. Rospocher, and L. Serafini, "MoKi: A Wiki-Based Conceptual Modeling Tool," in *Proc. of ISWC 2010, Posters and Demonstrations Track*, Shanghai, China, 2010.

[6] 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.

[7] A. Oltramari and G. Vetere, "Lexicon and ontology interplay in senso comune," in *Proceedings of OntoLex 2008*, Marrakech (Morocco), 2008.

[8] 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.

[9] 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.

[10] V. Dimitrova, R. Denaux, G. Hart, C. Dolbear, I. Holt, and A. G. Cohn, "Involving domain experts in authoring owl ontologies," in *Proc. of ISWC 2008*, ser. LNCS, vol. 5318. Springer Berlin / Heidelberg, 2008, pp. 1–16.

[11] 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.

[12] J. Mylopoulos, "Information modeling in the time of the revolution," *Information Systems*, vol. 23, no. 3-4, June 1998.

[13] OMG, "Business process modeling notation, v1.1," www.omg.org/spec/BPMN/1.1/PDF, January 2008.

[14] M. K. Smith, C. Welty, and D. L. McGuinness, "Owl web ontology language guide," W3C Recommendation, 2004.

[15] Wikimedia Foundation, "Mediawiki," http://www.mediawiki.org, last Accessed on 6 Nov 2011.

[16] The Oryx Project, "The oryx editor," http://bpt.hpi.uni-potsdam.de/Oryx/, last Accessed on 6 Nov 2011.

[17] K. Kaljurand and N. E. Fuchs, "Verbalizing owl in attempto controlled english," in *Proceedings of Third International Workshop on OWL: Experiences and Directions, Innsbruck, Austria (6th–7th June 2007)*, vol. 258, 2007.

[18] T. Kuhn, "AceWiki: A Natural and Expressive Semantic Wiki," in *Proceedings of Semantic Web User Interaction at CHI 2008: Exploring HCI Challenges*, 2008.

[19] S. Schaffert, "Ikewiki: A semantic wiki for collaborative knowledge management," in *1st Int. Ws. on Semantic Technologies in Collaborative Applications (STICA'06).*

[20] 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.*

[21] M. Rospocher, C. Ghidini, V. Pammer, L. Serafini, and S. Lindstaedt, "Moki: the modelling wiki," in *SemWiki2009 - Fourth Workshop on Semantic Wikis*. CEUR-WS.org, 2009, pp. 113–127.

[22] 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, 2011, vol. 7032, pp. 17–32.

[23] 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.