Creating and using Semantic Web information with Makna

Karsten Dello, Elena Paslaru Bontas Simperl, and Robert Tolksdorf

Freie Universität Berlin, Institut für Informatik,
AG Netzbasierte Informationssysteme, Takustr. 9, D-14195 Berlin, Germany
{dello,paslaru,tolk}@inf.fu-berlin.de
http://www.ag-nbi.de/

Abstract. Combining Wiki and Semantic Web technologies is considered by many members of the two communities as a promising alternative to current approaches for collaboratively creating and using information on the Web. The user-friendliness of the former as regarding multi-site content generation and the power of semantic technologies as w.r.t. organizing and retrieving knowledge are likely to complement one another towards a new generation of Web-based content management systems. Our system Makna (stands for *"knowledge"* in Indonesian) elaborates on this ideas by extending the Wiki engine JSPWiki with generic, easy-to-use ontology-based components for authoring, querying and browsing Semantic Web information.

1 Introduction

Combining Wiki and Semantic Web technologies is considered by many members of the two communities as a promising alternative to current approaches for collaboratively creating and retrieving information on the Web. The success of the former is primarily due to their simplicity and user-friendliness; a Wiki is a hypermedia system consisting of a collection of interconnected Web documents, which can be accessed, revised and extended by arbitrary parties with the help of a simplified hypertext syntax. However, while Wiki systems are targeted at collaborative *authoring*, they still need means to *organize and retrieve* the created content. Though its importance is widely acknowledged among Wiki solution providers, this issue is marginally addressed in current implementations, which restrict to organizing Wiki articles/pages according to a manually defined (and maintained) set of categories.

The Semantic Web provides the technological infrastructure to alleviate this situation. RDF statements can be applied to enhance the semantics of Wiki pages and of the links between them, while ontologies and associated reasoning services are a valuable extension to currently employed plain topic classifications and information retrieval capabilities. Figure 1 illustrates this idea with a simple example: the relationship between the Wiki article introducing the American actor Humphrey Bogart and the one describing his home town New York is related by the typed link `livedIn`, a property defined in a particular domain ontology.

The interconnected articles themselves are annotated with typing information, according to which they are classified as an instance of the concept `Actor` and `City`, respectively.

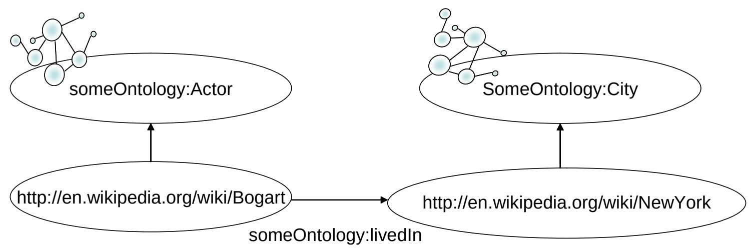

Fig. 1. Typed pages and links

Complementarily the Semantic Web can utilize Wikis as support tools in various application scenarios, the most important being probably distributed knowledge engineering and semantic content generation. The usage of Semantic Web technologies is currently inconceivable without a high level of IT expertise, with the consequence that the amount of Web information available in languages like RDF(S) and OWL is minimal compared to the dimensions of the traditional Web. In order for the Semantic Web to overcome this technical barrier of entry there is a need for tools which allow humans to contribute to the creation of Semantic Web content transparently from the underlying technologies.

In this paper we elaborate on these ideas by implementing Makna, a Wiki-based tool for distributed knowledge engineering.[1] Makna extends an existing Wiki engine (in our case the Java-based JSPWiki system)[2] with generic, easy-to-use ontology-driven components for collaboratively authoring, querying and browsing Semantic Web information. In contrast to similar attempts for combining the two fields of research, our system explicitly focuses on *the immediate and comfortable exploitation of the semantic content*, while implementing many key features of hypermedia systems targeted at supporting distributed knowledge engineering processes [4, 14, 18, 21].

The remainder of this paper is organized as follows. Section 2 specifies a set of core requirements to be fulfilled by the Semantic Wiki implementation. Building upon these we discuss the design principles and the architecture of the Makna system in Sections 3 and 4, respectively. Details on the implementation are presented in Section 5. We compare our solution with related approaches in Section 6 and conclude with a summary of future research and development directions in Section 7.

[1] `http://makna.ag-nbi.de`
[2] `http://www.jspWiki.org/`

2 Requirements analysis

A Wiki system can be understood as a collection of simplified hypertext documents which are read and edited by a community of Wiki users[5].

In relation to the rapid growth of the amount of information available within a Wiki system, its users need means to locate the relevant content, prior to accessing, creating or modifying it in any form. This task is commonly approached with the help of keyword-based information retrieval mechanisms. Mainstream Wiki implementations maintain a full-text index of their contents and provide users with the possibility of formulating unstructured queries in a simple query interface.[3]

The limitations of this simplistic approach become visible when Wiki users necessitate very specific information on a particular topic or information which is distributed over multiple pages. These use cases can not be successfully realized using keyword-based search algorithms executed on a plain set of textual documents; in turn they require heuristics which take into account the semantics of the underlying domain and of the links between individual Wiki contents. We illustrate these issues by means of two simple examples in the movies domain. A query like `Who was the actor Humphrey Bogart married to?` by Wikipedia delivers—if formulated properly—a list of movies starring the actor, as well as the Wiki article containing his biography. Though the encyclopedia contains a page dedicated to the prominent wife of Humphrey Bogart, the actress Lauren Bacall, this information can not be exploited without human intervention, in the absence of machine-processable domain knowledge and of an adequate link mechanisms between the two pages. Further on, a search for `American actors` in the same information repository returns a set of Wiki entries containing actor biographical information, but also dozens of movies or other types of articles. In order to improve the precision of such general queries, Wiki systems need to implement mechanisms for typing Wiki pages according to pre-defined schemes (such as ontologies).

To summarize, typing Wiki articles and the links interrelating them by means of ontologies enables the implementation of advanced content- and structure-oriented retrieval facilities.

Another aspect to consider is context-based navigation. Wiki systems rapidly tend to contain an impressive number of internal links.[4] If the relation between Wiki pages would be represented in a precise and formal way—and that is exactly what Semantic Web technologies provide—the Wiki engine could provide facilities to semantically navigate between meaningfully related resources, such as the articles dedicated to the aforementioned actors.

Besides domain-focused search heuristics and semantic navigation, ontology-driven technologies provide additional advantages to organizing and retrieving

[3] Often users are re-directed to a page by search engines like Google who have spidered the Wiki contents, integrating it into their own index.

[4] As stated at `http://stats.Wikimedia.org/DE/TablesWikipediaEN.htm` the English Wikipedia contained 19.3 million internal links to 922.000 articles as of November 2005. This is a rate of approximately 21 out- and in-links per article.

knowledge in a Wiki system. If Wiki contents were classified according to an ontology, this could be utilized as a commonly agreed query vocabulary, enabling users to formulate more precise and structured queries. Further on, formally represented ontologies in correlation with reasoning services are an ideal means to operationalize various quality assurance procedures, which become indispensable in any loosely coupled collaborative content authoring endeavor [1]. A third use case is the automatic link computation; given a semantic classification of Wiki articles, the system could consult the ontological content in order to automatically detect links to semantically related resources. According to a recent study by [1] the lack of adequate support for link creation and management is one of the key usability problems encountered within Wiki systems.

The aforementioned issues correspond to well-established requirements for so-called *"forth-generation hypermedia systems"*. As stated in [4, 21] advanced Web-based hypermedia systems—including Wikis—can take benefit from implementing features such as[5]

- typed annotated nodes
- typed attributed links
- computed links
- personalized links
- content- and structure-based search
- content- and structure-based navigation
- multiple view presentation

On the other hand, while extending Wiki technologies with semantics definitely contributes to the realization of advanced search and browse facilities, it also imposes several *additional* system requirements at both functionality and usability level. First, the system should provide a concept for the consistent authoring and manipulation of *semantic information*. This issue relates to the *usage* of the system ontologies, but also to the *development* of the ontologies themselves. The former primarily requires means to reference ontological primitives within content generation tasks. The latter, however, induces the need for methodological and technological support for collaborative knowledge engineering tasks. This non-trivial research question, recently tackled in approaches such as [9, 14, 18] *solely at methodological level*, is still in its infancy in the Semantic Web community. Usability requirements mainly refer to transparency and performance issues, as semantic technologies are not necessarily popular for seriously addressing any of them yet.

3 System design

Accounting for the results of the requirements analysis the realization of Makna system was influenced by two categories of design decisions, which are introduced in the remainder of this section.

[5] Hypermedia systems differentiate between *nodes* and *links*. *Nodes* denote information objects (e.g. documents, or document fragments) which are connected to each other by means of *links*.

3.1 Minimally invasive Wiki extensions

The first category of design decisions has the objective to preserve the advantages provided by conventional Wiki technology while enriching its capabilities.

The new engine should support the same usage patterns as traditional approaches. In particular this applies for the hypertext syntax employed, which need to extended with dedicated keywords. Further on, the system should provide an easy-to-use concept for the creation and management of semantic data [1]. This should allow Wiki users to annotate plain Wiki content in terms of adding, deleting or modifying RDF statements. However, as our system is not targeted at collaborative ontology engineering yet, it is not reasonable to permit *arbitrary users* to perform changes at ontological level. Without an adequate methodological support an uncontrolled ontology evolution might have considerable implications on the way Wiki data is classified and subsequently retrieved [8, 14, 17, 18, 20]. The manipulation of the employed ontologies as well as the import of semantic instance data are therefore currently limited to a particular group of users e.g. administrators (cf. Figure 3).

Technologically the minimal invasive character of Makna is reflected in the decision to build upon existing Wiki systems instead of an implementation from scratch (as for example in [2, 15, 19]). The benefits of this decision are twofold. First re-using established implementations has clear costs and quality advantages [7, 11]. Another motivation is related to the philosophy that *"the Semantic Web is not a separate Web but an extension of the current one."*[3]. Following this idea it should also be possible to turn an existing Web application like a Wiki engine into an Semantic application by plugging in the necessary extensions.

3.2 Versatile use of semantic technologies

A Semantic Wiki should provide facilities for flexibly integrate and use arbitrary Semantic Web ontologies available on the Web. This implies the possibility to refer to multiple ontologies at Wiki syntax level and their seamless usage in classification, retrieval and navigation tasks.

Wiki users should be able to comfortably access available ontologies in order to suitably annotate Wiki articles. This can be achieved by providing dedicated components to facilitate the interaction with the ontological content (cf. Section 5).

Inference is another important point to consider. Reasoning services can be applied on Wiki contents in order to enhance the retrieval capabilities of the system or to exercise consistency checking in relation to specific quality assurance procedures. However, it might be necessary to restrict this feature to a carefully defined set of ontologies, as inferencing on arbitrary ontologies on the Web could under circumstances lead to serious performance problems.

Complementarily to the integration of multiple Web ontologies the engine should provide means to import and export data formalized using Semantic Web representation languages.

Finally we consider the relations between pages which are not naturally represented as hyperlinks. This can be solved in the same way in which links between non-existent pages can be created. If it is not required that statements are exclusively formulated in the Wiki syntax the problem resolves, thus making available RDF data representable as Wiki instance.

4 System architecture

The architecture of the Makna system is depicted in Figure 2. It consists of the Wiki engine JSPWiki, extended with several components for the manipulation of semantic data, and the underlying persistent storage mechanisms. We chose JSPWiki, because it is written entirely in Java and has a clear design structure (i.e based on the Model-View-Controller-Pattern) facilitating system extensions. The Jena API was used as de facto standard Semantic Web framework for creating, managing and querying RDF data.

Additionally we used a relational database for the persistent storage of the semantic model. The persistent storage of the Wiki pages and attachments can be provided through any of the versioning storage provider modules for JSPWiki.[6]

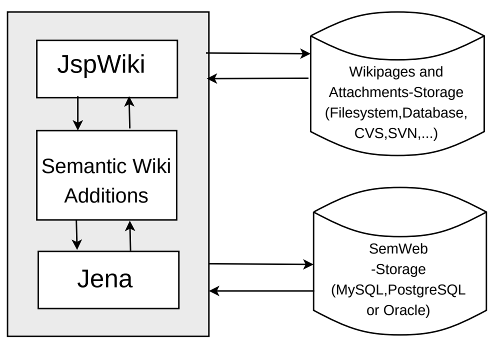

Fig. 2. Makna architecture

Due to the fact that our system currently does not include any mechanisms for collaboratively constructing ontologies, we differentiate between two types of semantic data: the ontology data and the instance data. This distinction is similar to the common terminology in Description Logics (or OWL) and corresponds to different ways of creating and manipulating the data within the system. Ontologies are expected to be imported to the Wiki instance by administrators. These can revise or extend them using external tools (such as conventional ontology editors). The instance data is the sum of

– the RDF statements formulated in Wiki syntax by the users

[6] `http://www.jspWiki.org/Wiki/PageProvider`

– the statements manually added by the user (using assistants, see below)
– and the (external) instance data imported by the administrators

This separation is reflected in the restriction that the ontology data can be modified solely in the administrator interface, while the instance data can be manipulated in diverse ways in the Web interface by arbitrary users (cf. Figure 3).

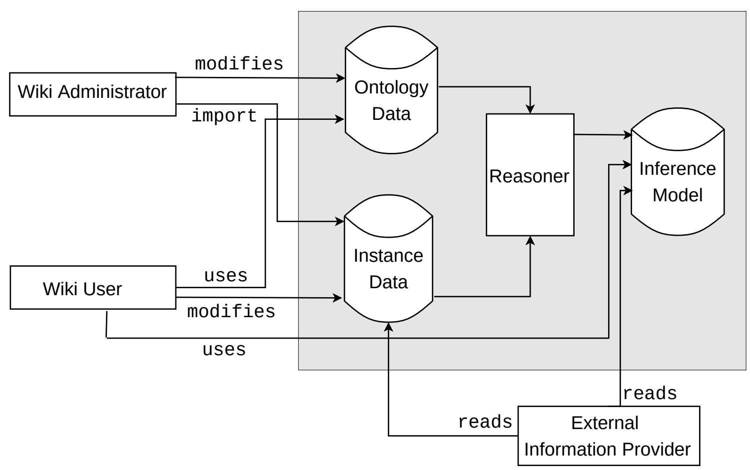

Fig. 3. Separation between ontology and instance data

The administration interface is responsible for the ontology- and configuration-related functionality:

– specify the ontology/ontologies used within the system
– import external RDF data
– define shortcuts for a more comfortable usage of ontological primitives
– configure inference engines and persistent storage systems.

The user interface embeds facilities for creating and using Semantic Web information on the basis of the imported ontologies:

– refer to ontological primitives for annotating Wiki content or defining link types
– formulate and execute content- and structure-based queries
– browse the Wiki contents on a content- or structure basis
– export semantically represented data as RDF or N3.

Details about the implementation of these features are discussed in the next section.

5 Implementation

5.1 Authoring semantic content

Wiki users are able to create semantic content (in form of RDF statements referencing pre-configured ontologies) in the classical Wiki manner. They are provided with an extended Wiki syntax and with assistant tools simplifying the interface to the ontologies employed. Further on, users can create, modify and delete RDF statements associated with Wiki pages.

Extended Wiki syntax In JSPWiki's syntax a link is represented by [<Target>] where <Target> is either an absolute URL or another page in the Wiki.

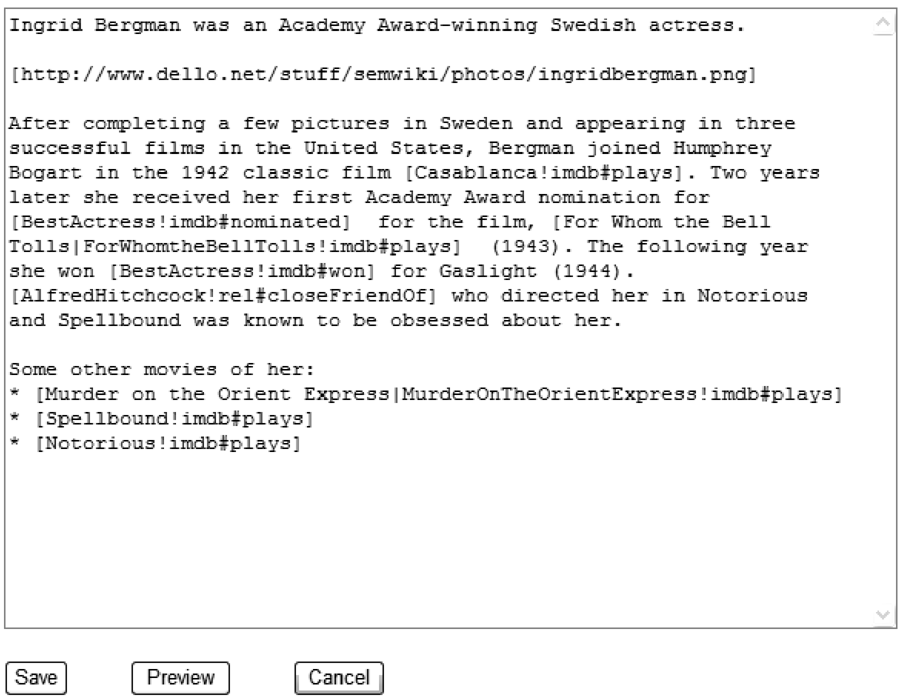

Fig. 4. Makna syntax

We extended this syntax to [<Target>!<Term>] to support semantic linking.[7] This extended link element creates a semantic statement, in which the URI of the edited page is the subject, <Term> is the predicate and <Target> is the object. Figure 4 presents the edit-page corresponding to the Wiki article on Ingrid

[7] The exclamation mark was selected because it has no other function in links in JSPWiki syntax.

Bergman, which is depicted in Figure 7 below. `<Term>` can be an URI, but more likely either a shortcut or a *namespace:predicate* combination from the resources configured in the Wiki instance. In order to support statements with literals, the range of `<Target>` was extended too. If `<Target>` is enclosed in hyphens, the system recognizes it as an literal and creates the statement accordingly.

Interactive Assistants In order to support the user formulating accurate semantic links in the proposed Wiki-syntax we integrated several interactive assistants based AJAX[6]. We will limit our description here to those two assistants which have been integrated into the edit-page of the Wiki: the predicate assistant and the page assistant. The former guides the user in finding a predicate in the configured ontologies. The latter assists the user in seeking for the names of other pages in the Wiki.

Figure 5 illustrates the functionality of the predicate assistant. The user enters the term *mail* and a drop down list with the matching predicates is automatically created. After selecting a predicate from the list, the text area below is updated with detailed information about the selected predicate.

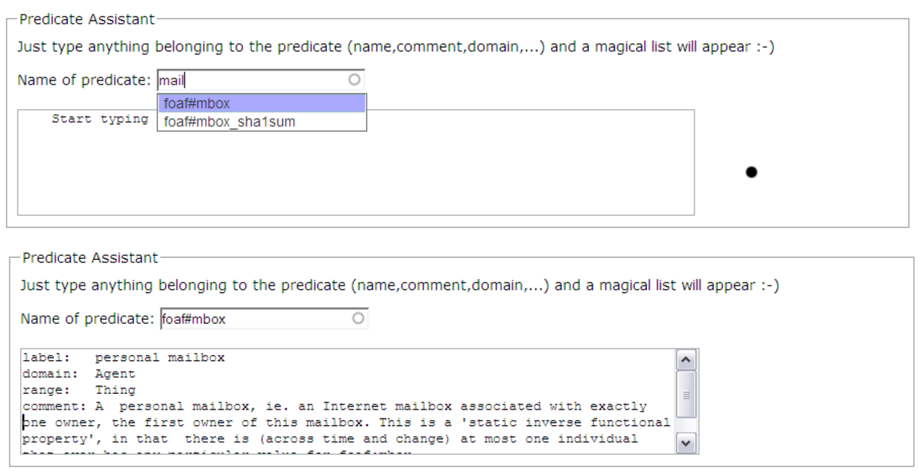

Fig. 5. Predicate assistant

As soon as the user starts typing into the input field, an asynchronous request is sent from the client-browser to the server in the background, which returns an XML-document with the matching terms. These terms are compared on a lexical basis to the natural language labels of the corresponding items (predicates and Wiki pages, respectively).

Our system also supports statements which do not appear in the Wiki pages themselves. This means that meta-statements can be expressed, and links between two non-existent pages and/or from external resources can be formulated.

The associated assistant provides a simple interface which allows users to access the content of available ontologies in order to specify subjects, predicates and objects of new statements. Figure 6 depicts this functionality: after selecting the FOAF-ontology, the assistant updates the predicate form with the corresponding information (top part of Figure 6); depending on these two parameters the assistant further computes appropriate object resources (bottom part of the same figure)

Consistency of the semantic model The consistency of the semantic model can be guaranteed by our implementation. To achieve this we implemented two functions. If a statement is submitted by the user with the syntax-external assistant (cf. Figure 6) we check its validity immediately; in case the statement is found to be invalid according to the semantic model it is rejected and the user gets notified. The same happens if the statement is found to cause inconsistencies in the model. The other function is the verification of statements which are formulated in the Wiki syntax. After the edited page is submitted the system extracts all semantic statements and checks if they are consistent with the semantic model. If statements are found which cause inconsistencies, the user is returned to the edit page, gets informed about the details of the problem detected and is asked to correct his input. By doing so we assure that the semantic model is always consistent, as there is no possibility to add statements that cause inconsistencies.

This behavior is currently being refined in order to support application scenarios with loose consistency requirements. In conjunction to the extension of our system towards advanced collaborative ontology engineering support (cf. Section 7) we are examining ways to ensure local consistency on personal ontologies, while the global shared ontology does not have to satisfy this (sometimes unfeasible) constraint.

5.2 Context-based presentation and navigation

When a call to a Wiki page is issued the Wiki engine extracts a subgraph of the semantic model which contains all statements which have the current page either as their subject or their object—no matter if they were formulated in the Wiki syntax or elsewhere.

In Figure 7 we illustrate the Wiki article about the actress Ingrid Bergman in a fictive Makna movie instance. The navigation block on the right side of the screen consists of two parts: the summary of the semantic relations of the current page (on the top) and the list of the prepared search requests for related resources (on the bottom).

The summary of the semantic relations can help the user to quickly navigate to a related topic in one step (i.e. one click). The customized links to searches for related pages are created through the following scheme: for each property of a page a search link is provided to find other resources with the same property. This is true for incoming links as well, which means that is is possible to navigate

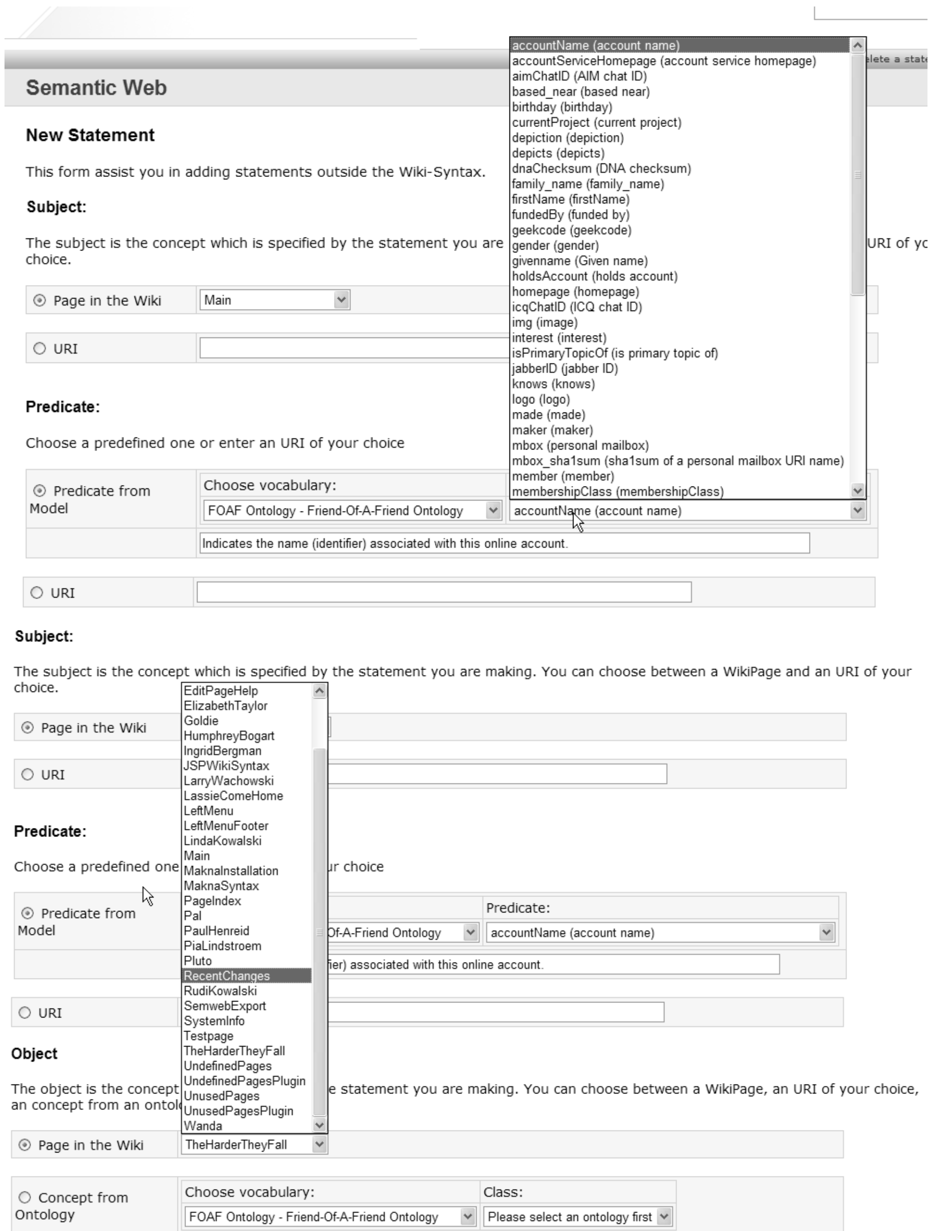

Fig. 6. Statement assistant

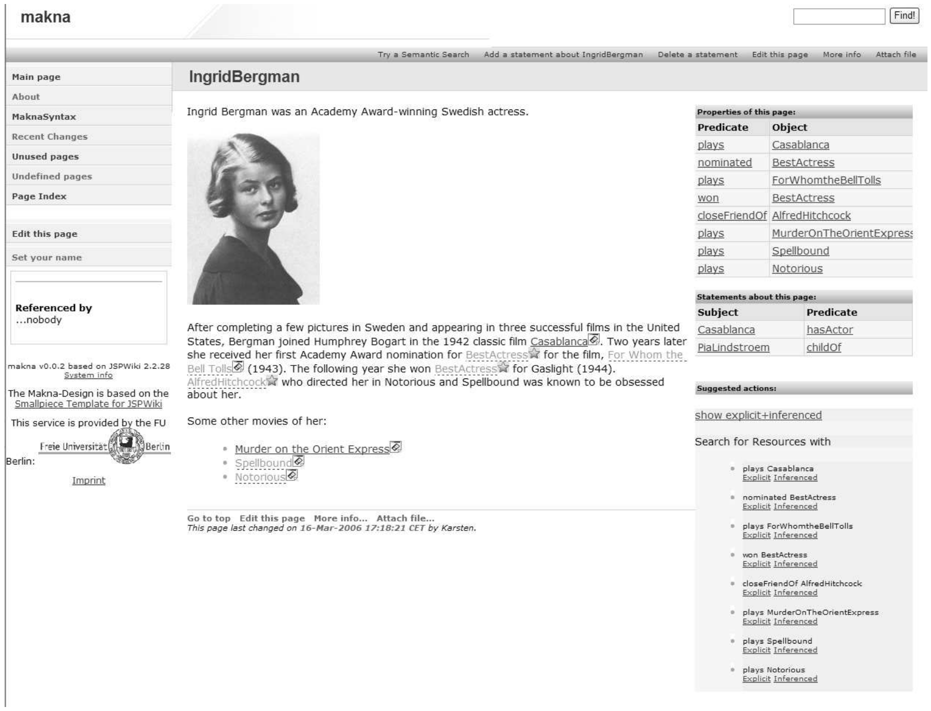

Fig. 7. Makna article on Ingrid Bergman

quickly to other resources which have the same relation to the current page. Both functions work with inference (which can for convenience be switched on and off by the user), thus enabling the user to navigate the Wiki contents conducted by semantic relations.

5.3 Content- and structure-based retrieval

Makna implements a search interface resorting to form-based search patterns, which allow users to use the ontology in formulating structure-based queries and the underlying inference engines for enabling semantic search.

We have developed several templates for comfortably formulating typical content- and structure-based query patterns. Figure 8 shows the implementation of a template returning instances of a user-defined class in a knowledge base. After choosing a vocabulary from the left drop-down list, the right list is filled up with all concepts from the chosen ontology.

6 Related Work

A multitude of promising approaches for combining Semantic Web and Wiki technologies are currently under development (cf., for example, [2, 15, 16, 19, 22]).

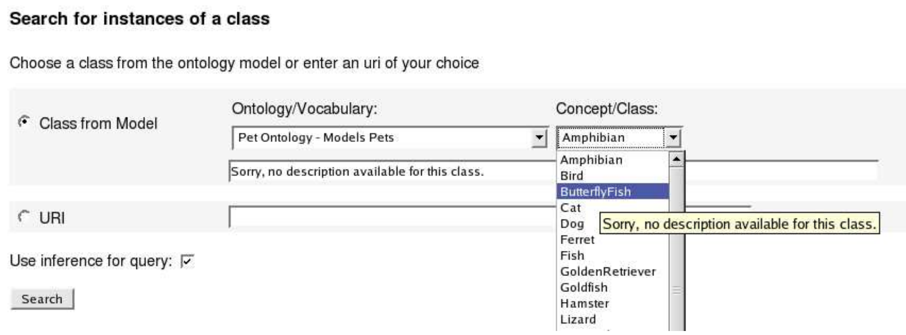

Fig. 8. Form for a simple semantic search

However, while these proposals share the declared common goal of realizing a Semantic Wiki system, a closer investigation of their—planed or implemented—features evidences that they are oriented at slightly divergent application scenarios.

Approaches such as [16, 19] develop Wiki engines which support the generation of RDF data. However, they clearly distinguish between semantic and plain Wiki contents and their usage in Semantic Web context requires technical expertise on RDF. Acknowledging for this limitations more recent proposals focus on the minimal invasive usage of semantic technologies within Wiki systems[2, 15, 22]. Makna differs from these approaches from a multitude of viewpoints. In contrast to WikSAR [2] our system is oriented at non-technical users exercising collaborative knowledge engineering. This focus motivated a different design of the user interfaces. Further on, the present implementation of Makna restricts the access to *ontological knowledge* (as opposed to *instance knowledge*) to a predefined set of Wiki users. While this might be considered as a limitation of our system as compared to other approaches (such as IkeWiki [15]), we argue that implementing such functionality without adequate process-level support might have uncontrolled consequences on the operation of the overall Wiki system. This was confirmed by recent advances in the area of distributed ontology engineering[14, 18]. The Semantic MediaWiki project has come up with a different approach [10, 22]. Their development aims at turning the hugest existing traditional Wiki-based information repository—the Wikipedia encyclopedia—into a semantic Wiki. Similar to our own requirements analysis, the authors identified the need for typed annotated links and articles in Wikipedia (cf. Section 2). The current release of the Semantic MediaWiki system shares many commonalities with the functionality of Makna. In contrast to our implementation it does not support consistency checking mechanisms and the usage of multiple ontologies. In turn, it allows for an unrestricted access to both ontological and instance data, an option which we consider disputable for arbitrary settings without an adequate process support.[8] Because Makna focuses on small to mid-size Wikis we can utilize some performance critical mechanism like real-time inference, enabling features like guaranteed consistence of the model and context-based navigation based on real-time state of the semantic model. Furthermore Makna's separation of instance and ontology data—accompanied by the user and administrator interfaces (cf. Section 4)—enables predictable response times and memory usage which are the foundation for an efficient Semantic Wiki system. Finally we mention the SemperWiki approach [12], which introduces Wikis as enabling technology for personal information management. Unlike conventional Wiki and Semantic Wiki solutions—including Makna—it is targeted at single user environments.

7 Outlook

A feasible combination of Wiki and Semantic Web technologies should preserve the key advantages of both technologies: the simplicity of Wiki systems as regarding shared content authoring, as well as the power of Semantic Web technologies w.r.t. structuring and retrieving knowledge. Building upon an analysis of the requirements induced by a collaborative knowledge engineering scenario to Wiki implementations we have introduced our concept of a Semantic Wiki engine addressing this problem.

We are currently extending the functionality of the current release of Makna w.r.t. methodological and technological support for knowledge engineering. In particular we are investigating means to automatically extract ontological structures from existing domain-focused Wiki instances (following the approach in [13]) and are implementing a component for collaborative ontology engineering based on [14].

Acknowledgements

This work has been partially supported by the EU Network of Excellence "KnowledgeWeb" (FP6-507482).

References

1. Désilets A., S. Paquet, and N. G. Vinson. Are Wikis Usable? In *Proceedings of the 1st ACM Wiki Symposium WikiSym2005*, 2005.
2. D. Aumüller. Semantic Authoring and Retrieval in a Wiki. In *Demo Session at the European Semantic Web Conference ESWC2005*, 2005.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.
4. M. Bieber, F. Vitali, H. Ashman, V. Balasubramanian, and H. Oinas-Kukkonen. Forth generation hypermedia: some missing links for the World Wide Web. *International Journal of Human-Computer Studies*, 47, 1997.

[8] However, this design decision might prove to be appropriate for the Wikipedia use case.

5. W. Cunningham and B. Leuf. *The Wiki Way. Quick Collaboration on the Web.* Addison-Wesley, 2001.
6. J. Garrett. Ajax: A New Approach to Web Applications (white paper). http://www.adaptivepath.com/publications/essays/archives/000385.php, February 2005.
7. T. Hemmann. How Knowledge Engineering Can Benefit from Software Engineering with Respect to Reuse: Towards Reusable Knowledge Models. In *Proceedings of the ERICM Workshop on Methods and Tools for Software Reuse*, pages 212–227, 1993.
8. M. Klein, A. Kiryakov, D. Ognyanov, and D. Fensel. Ontology Versioning and Change Detection on the Web. In *Proceedings of the 13th International Conference on Knowledge Engineering and Management EKAW02*, 2002.
9. K. Kotis, G. A. Vouros, and J. Padilla Alonso. HCOME: tool-supported methodology for collaboratively devising living ontologies. In *Proceedings of the 2nd International Workshop on Semantic Web and Databases SWDB2004*, 2004.
10. M. Kroetzsch, D. Vrandecic, and M. Voelkel. Wikipedia and the Semantic Web: The Missing Links. In *Proceedings of 1st International Wikipedia Conference Wikimania2005*, 2005.
11. C.W. Krueger. Software Reuse. *ACM Computing Surveys*, 24(2):131–183, June 1992.
12. E. Oren. Semperwiki: a Semantic Personal Wiki. In *Proceedings of the 1st Semantic Desktop Workshop*, 2005.
13. E. Paslaru Bontas, D. Schlangen, and T. Schrader. Creating Ontologies for Content Representation – the OntoSeed Suite. In *Proceedings of the 4th International Conference on Ontologies, Databases, and Applications of Semantics ODBASE2005*, 2005.
14. H. S. Pinto, S. Staab, and C. Tempich. DILIGENT: Towards a fine-grained methodology for Distributed, Loosely-controlled and evolving Engineering of oNTologies. In *Proceedings of the European Conference of Artificial Intelligence ECAI2004*, pages 393–397, 2004.
15. S. Schaffert. IkeWiki - A Semantic Wiki for Collaborative Knowledge Management. Technical report, Salzburg Research, 2006.
16. A. Souzis. Building a Semantic Wiki. *IEEE Intelligent Systems*, 20:87–91, 2005.
17. L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-Driven Ontology Evolution Management. In *Proceedings of the 13th European Conference on Knowledge Engineering and Management EKAW02*, 2002.
18. Y. Sure, C. Tempich, and D. Vrandecic. Ontology Engineering Methodologies. In Davies, J. and Studer, R. and Warren, P., editor, *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. Wiley, 2006.
19. R. Tazzoli and P. Castagna et al. Towards a Semantic Wiki Wiki Web. In *Poster Session at the International Semantic Web Conference ISWC2004*, 2004.
20. P. R. S. Visser, D. M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. An analysis of ontological mismatches: Heterogeneity versus interoperability. In *Proceedings of the AAAI Spring Symposium on Ontological Engineering AAAI97*, 1997.
21. F. Vitali and M. Bieber. Hypermedia on the Web: What Will It Take? *ACM Computing Surveys*, 31(4), 1999.
22. M. Völkel, M. Krötzsch, D. Vrandecic, H. Haller, and R. Studer. Semantic Wikipedia. In *Proceedings of the World Wide Web Conference WWW2006 (to appear)*, 2006.