Integrating a Wiki in an Ontology Driven Web Site: Approach, Architecture and Application in the Archaeological Domain

Andrea Bonomi, Alessandro Mosca, Matteo Palmonari, Giuseppe Vizzari

Department of Computer Science, Systems and Communication (DISCo)
University of Milan - Bicocca
{andrea.bonomi, ale.m, matteo.palmonari, giuseppe.vizzari}@disco.unimib.it

Abstract. This paper describes an approach to the design and implementation of ontology driven dynamic web sites combining ontologies and wiki technologies. The core of the architectural solution proposed is completely based on ontologies rather than on more traditional forms of persistent data storage facilities, such as relational databases. This approach provides a flexible support to the design and implementation of web portals in which navigation schemes are not entirely predetermined but are instead influenced by actual relationships among the contents of the ontology, that are used to generate web pages as well as hyperlinks. A wiki technology is integrated with this approach in order to create pages not directly derived by elements of the ontology, but also to enrich the textual contents with suitable formatting, images and hyperlinks. The application of this approach to the realization of a web portal is also described; the portal is devoted to enable a number of scientific communities to share archaeological knowledge about the Silk Road domain.

1 Introduction

The introduction of technologies enabling the development of data driven web sites represented an extremely important step in the process that lead the initial version of the Web to its current state of pervasive diffusion and impressive growth rate. Data driven web sites are able to generate web contents and pages according to the information persistently stored in a suitable module, such as a relational Data Base Management System. This represents the last tier of an architecture encompassing a middle tier that is able to interpret suitable templates for web pages that are instantiated according to actual data stored in the data tier, and that are eventually passed to the first tier responsible of the visualization of the page (i.e. the client tier, a common web browser). Traditionally these templates have been realized by integrating modules developed in common programming languages with a web application (e.g. Common Gateway Interface or Java Servlet technologies), or embedding "programming languages–like" control structures and abstractions in traditional web pages through scripting languages (e.g. PHP, ASP, JSP). Some of the most relevant abstractions related to a data driven web site, and determining its organization and function are thus implemented and expressed in terms of a database logical schema and specific programming language control structures embedded in page templates.

Even if this approach surely represents a huge improvement with respect to static web sites, that are essentially repositories of HTML documents and embedded multimedia contents, a set of different research efforts have been carried out in an attempt to supply higher level abstractions to support the design and implementation of web based advanced information systems and applications. Some of these approaches, for instance [1, 2], are based on traditional data conceptual models for site contents and extend the scope of the modelling activity to aspects of relevance in the web context, such as navigation and presentation. In this vein, this paper presents an approach that provides instead the adoption of an *ontological* rather that data tier, building on experiences and results of research in the Semantic Web [3] area to provide new abstractions and instruments for the structuring and management of information supporting dynamic web pages composition. In particular, the basic idea is to exploit an explicit and formal conceptualization of concepts related to the web site domain, as well as specific aspects related to web sites in general, to structure data and information required to generate contents, to specify the navigation among them and to generate an effective presentation. The ontological approach provides a uniform and expressive framework for the representation and management of these different aspects. In particular, the ontology can encompass documents, and one particular type of document can be represented by a *wikipage*. This allows endow this ontology driven approach to the definition, design and realization of web sites with a more traditional and established form of definition of contents of web pages.

The following section will elaborate the research context in which this work is set, briefly introducing relevant related works, while Section 3 introduces the architecture and the various functionalities offered by NaVEditOW [4], the framework on which this approach is based. A case study in which the approach has been applied for the development of a portal organizing archaeological information and documents will then be introduced. Conclusions and future developments will end the paper.

2 Ontology Driven Web Sites and Wikis

In his proposal for a global hypertext [5], Tim Berners-Lee proposed a "gateway program" to generate hypertext view of existing data source. He has imagined a simply generic gateway with a limited, perhaps read-only, access on a database that allow to display it as a hypertext and navigate through the data.

Within a short period, the Internet and World Wide Web have become ubiquitous and today Internet is full of data-driven Web sites: yellow pages, e-Commerce sites, digital libraries are only the most common examples. But also, forum, blog, wiki, video and photo sharing website, hotel booking, online auction website, online community, Web-based email client and Web mapping service are all data-driven Web sites.

A data-driven Web site is much easier to maintain than a static Web site: most content changes require no change to the pages. Instead, changes are made to the data source and this source could be enterprise or organization database. Sharing a common data source, the Web application can be easily integrated within information system.

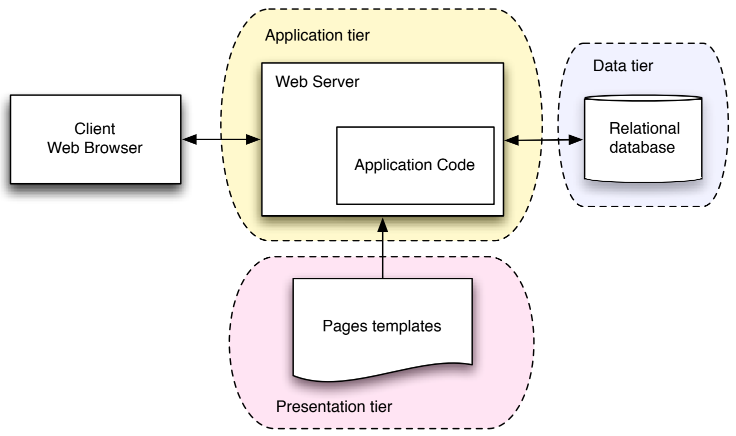


Fig. 1. On the top schema, the architecture of a traditional data-driven Web site.

So, for example, in a manufacture company when a new product is added to the enterprise database, it could be automatically displayed on the corporate Web site products catalog.

One of the most common architecture used in developing of dynamic data driven web sites, is shown in Figure 1. In particular, the persistent data storage is generally delegated to a relational database management system. The web server generally hosts dynamic pages including server side scripts necessary to query the data tier and collect the information required to compose pages related to the contents of the database. These dynamic pages represent a sort of *template* for the web pages that must be generated according to the stored data, specifying different aspects ranging from the queries that must be submitted to the database to retrieve them, to the kind of links that must be created to support the navigation inside the data.

An evolution of this approach is the adoption of the the MVC (Model View Controller) [6] architectural pattern. In a web based MVC application, the model is the domain-specific representation of the application information, the view layer is responsible to renders the model into user interface element (HTML pages) and the Controller processes and responds to the users actions and can invoke data changes.

Generally, the data is stored in a relational database and the model is represented by a database schema. In our opinion, it is difficult to represent all the relations that could be present in a complex domain (such the archaeology domain) with a database schema. In many cases, is difficult to translate some aspects of a conceptual schema (such the generalization) into the relational model. Some kind of extra-relational constraint are not representable within the database schema and requests database store procedure or external application code. There are also problems to manage the database schema, for example, the is no way to check the schema consistency.

The proposed approach provides the adoption of an ontology [7], rather than a relational database, as data layer (a schema of such architecture is shown in Figure 1) and to use a Wiki engine in order to simplified content authoring and management func-

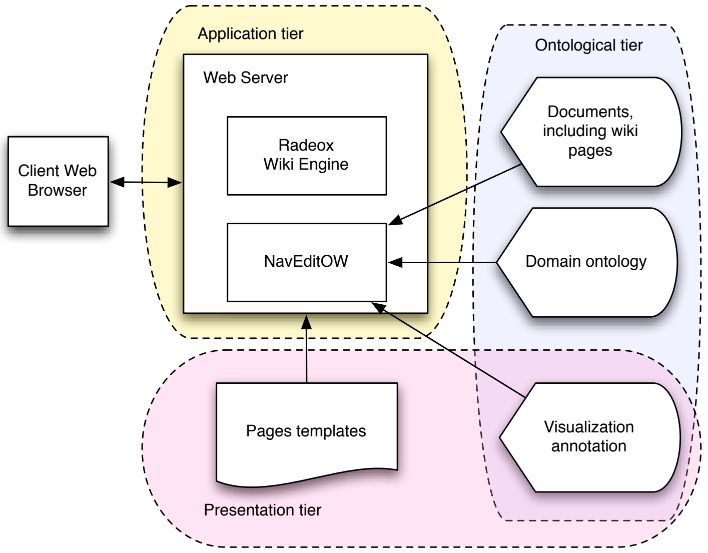


Fig. 2. The overall architecture of the proposed system, integrating a wiki engine for rendering specific documents stored in the ontological tier.

tionality. In particular, we employed NaVEditOW [4], a system for navigating, editing and querying ontologies through the web, as a means to access and manage the ontology, and we integrate the Wiki technology through the adoption of Radeox [8], that is an Open Source Wiki engine written in the Java language. An interesting feature of this engine is that it could be easy extended with custom macros that can support, for instance, the inclusion in a wiki page of a table of all instances of a given class. The overall resulting architecture is shown in Figure 2.

The main motivations of this architectural modification are related to a more *comprehensive exploitation* of the explicit relationships among the concepts described in the ontology, and a *simpler integration* of this kind of architecture with instruments and systems developed in the Semantic Web context.

Other approaches of ontology-driven Web site architectures can be found in the literature: in [9], it is described an application, OntoWeaver, that uses an ontology to provide a comprehensive support for the design and management of data-driven Web sites. The described approach uses site ontologies to enable a declarative representation of all the aspects of a Web sites: in particular, a domain ontology is used as an abstract of the back-end data sources and a user ontology models information about the Web site users. In [10] it is described a similar approach in which data from various sources are converted into RDF-annotated format (based on a domain ontology), composed together and used to generate a browsable Web site.

A large number of approaches for combining Semantic Web and Wiki technologies are currently under development; some relevant examples are Makna [11], IkeWiki [12], Rhizome [13]) and Semantic MediaWiki [14]. The above cited approaches and the related systems are all examples of Wikis expanded to encapsulate and exploit a well–defined semantics to enhance their functionalities. It is not the aim of this section to introduce a structured and comprehensive discussion of these approaches and systems (an interesting discussion on this line of work can be found in [15]), but it is important to note that the described approach follows a totally different line of work: we are more focused on supporting a collaborative effort towards the definition of domain ontologies and their exploitation in web based systems than in supporting the enhancement of wikis through the usage of semantic web technologies. The NavEditOW system and this effort can be intended as an attempt to bring part of the spirit of wiki technologies and the social aspects of the Web 2.0 in the semantic web context. The remainder of the paper will introduce the NavEditOW system and its application to the realization of a web portal supporting a community of archaeologists working on Cultural Heritage of Central Asia.

3 NavEditOW

In this section we present a system for web based navigation, querying and updating of ontological KBs. The presented software allows exploring the concepts and their relational dependencies as well as the instances by means of hyper-links; moreover, it provides a front-end to query the repository with the SPARQL[1] query language.

NavEditOW is an environment for navigating, querying and A-Box[2] editing of OWL[3] (Web Ontology Language) ontologies through a web-based interface.

With respect to ontology *navigation*, since individuals play a fundamental source of knowledge for people accessing an ontology, A-Box navigation should be supported. From this perspective, it is important to support not only navigation of concept hierarchies defined by *isA* relations, but also other forms of ordering on the individuals domain. As a first example, locations can be linked through a *partOf* relation and it should be possible to group locations under the location of which they are all subparts (e.g. browsing all countries of which Europe is composed of starting from Europe); as a second example consider a number of historical periods ordered according to a relations such as *followedBy*: it should be possible to exploit this relation to sort such individuals from the first to the last one.

With respect to *editing*, although T-box[4] maintenance require a certain knowledge about ontological formalisms, A-Box editing should be supported taking into account: (i) cardinality and range restrictions defined in the T-Box need to be respected; (ii) ranges of properties and individuals stored in the ontologies can be also exploited to drive and suggest instance update. Moreover, contextual editing, that is, the editing of the A-box while browsing the ontology, should be supported.

[1] SPARQL (SPARQL Protocol and RDF Query Language) is an RDF query language standardized by RDF Data Access Working Group of the World Wide Web Consortium. For more information, see http://www.w3.org/TR/rdf-sparql-query/

[2] The A-Box is the "assertion component" of a knowledge base.

[3] http://www.w3.org/TR/owl-features/

[4] The T-Box is the "terminological component" of a knowledge base.

Although end-users may not be familiar with *query* languages, the possibility to perform expressive queries should be supported. From the one hand a language as much as similar to well-known query languages for relational databases language should be preferred. On the other hand, interfaces enabling non expert users querying the ontology should be developed (e.g. query forms).

In the following paragraphs, we presents more details about each of these tree basic functionalities and the application architecture.

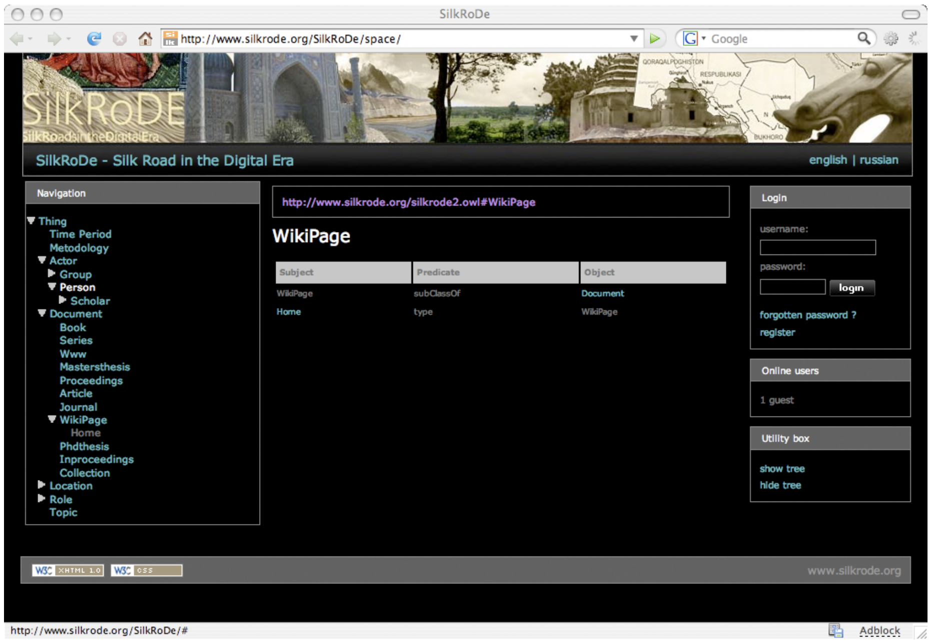
3.1 Navigation

With the ontology navigation interface the users can view ontology individuals and their properties and browse properties via hyperlinks. Browsing the ontology is essential for the user in order to explore the available information and it also helps non-expert users to refine their search requirements, when they start with no specific requirement in mind [16]. The hierarchical organization of the different concepts and individuals of the ontology is graphically represented as a dynamic tree. The aim of the navigation tree is to explore the ontology, view classes and instances, discover the relation between them. The tree does not represent only the a hierarchy of classes connected with *isA* binary relations (like the navigation tree of Protégé), but represents also also tree-like connections of individuals for domain dependent classes of properties (e.g. *partOf*, *isLocatedIn*, and so on). From a formal point of view, ontological relations supporting tree-like visualization (tree-like properties) are those represented as properties not symmetric and whose inverse is functional (therefore identifying directed acyclic graphs). These properties link directly an individual with its "father" and are particularly relevant with respect to mereological relations (e.g. *partOf*, *composedOf*), and to relations defining hierarchical spatial and temporal structures (e.g. representing the unfolding of historical periods). Another kind of relations exploited for the visualization are relations defining total orders on individuals (e.g. *isFollowedBy*).

The root of the navigation tree is the OWL class *Thing*, and the rest of the tree is organized as follows: under the root node, there are the top-level classes (i.e. direct subclasses of Thing); each class can be expanded to show its subclass hierarchy and its individual members; individual-to-individual tree connections are defined according to a number of selected tree-like properties (e.g. *partOf*); finally if total order relations are selected, they are exploited to order individuals within a given level of the tree. In order to distinguish between classes and individuals, the former are represented in petrol blue while the latter are shown in shocking pink. An example of a navigation tree is presented in the Figure 3. In this example *Geographical Region* is a class and *Cental Asia*, *Uzbekistan*, etc... are its instances. This individuals, in turn, are connected each other by the *partOf_directly* property.

3.2 Editing

The application allows the users to create, edit and remove individuals of the ontology, their properties and, in particular, their labels. In fact, To ensure multi-languages support, it's possible to define several labels in different languages for every individual.



http://www.silkrode.org/silkrode2.owl#WikiPage

WikiPage

Subject	Predicate	Object
WikiPage	subClassOf	Document
Home	type	WikiPage

Fig. 3. A screenshot of the NavEditOW system in the archaeological case study.

The properties of each classes are defined in the T-Box. Two types of properties are distinguished: *object property* is a binary relation between two individuals and *datatype property* is a binary relation between an individual and a literal (a primitive type, like string or number). Cardinality and range restrictions for properties are used to support users while editing. For example, in an archeological ontology, the class *TypologyOfArchaeologicalObject* has the property *builtOf*. This property has no cardinality restriction (so it can have zero, one ore more values) but *Material* is specified as range (co-domain). For instance, *Sword* is an instance of *TypologyOfArchaeologicalObject* and has the property *builtOf Metal*, where *Metal* is an instance of *Material*.

There are a datatype and an object editor in the framework: the datatype editor allows editing a literal values, displayed as a text input box, object allows defining the property values presenting the user a tree for selecting the values; the individuals displayed in the tree are only those that are valid for the property range.

Literal values can be not only plains strings but also Wiki text. This allows the users to insert long formatted text with link in the generated pages. For example, we can have the 'Uzbekistan' individual in our ontology. 'Uzbekistan', as an instance of Country, has some properties such population, capital city, part-of, currency, history, foreign relations, etc. Same (e.g. population) are plain literals (numbers, dates, strings), others are references to other ontology instances (e.g. the filer of the property 'capital city' is Tashkent which is an instance of the class City). History of foreign relations are Wiki Text, so they can be formatted, contains images an hyperlinks. Wiki Text can also contains 'macro' that allows to include in the text information from the ontology or results of SPARQL queries.

NavEditOW adopts an editing policy. Knowledge engineers interacting with domain experts, are supposed to create the ontology and edit the Tbox with standard design-time editing tools such as Protégé [17]. They are supposed to test its quality with reasoners such as Racer [18] or Fact++ [19] in order to check if the Tbox (i) is consistent and (ii) it captures the intended meaning (by concept hierarchy inference). End-users can edit the Abox; they can insert new instances, asserting that they are member of a concept of the ontology, and assert relation statements; this means, that they can edit the textual descriptions contained in the ontology. The assumption behind this editing policy is that end users are not familiar with formal semantics. First, learning to use concept constructor and ontology axioms is difficult for end users, if one goes beyond simple Tbox updates, such as the insertion of a new concept as subconcept of a concept in the ontology. Second, every axiom has logical implications that are difficult to control, especially for people non expert in formal semantics.

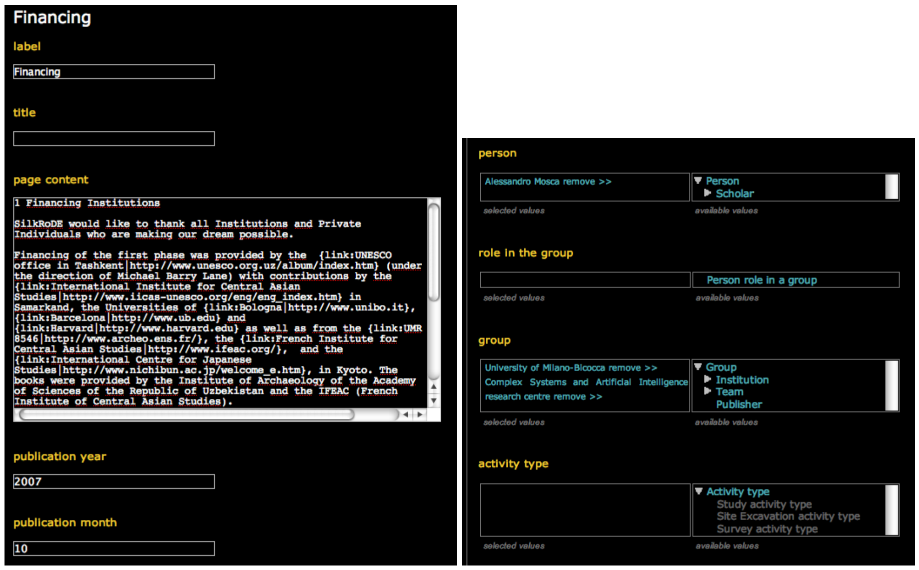
3.3 Querying

The first implemented query interface is the SPARQL query form in which users can write query in the SPARQL language, display results in paginated tabular form and navigate through results via hyperlinks. This interfaces is very flexible because the users can write arbitrary queries but is not suitable for end users. Another kind of query interfaces is based on a predefined set of queries. Every predefined queries is composed of a description in natural language, a SPARQL query with eventually free parameters and a list of parameters. Every parameters have a label, a type and eventually a restriction on the valid values (e.g. a parameter can be filled only with instances of a specific class). For this interface, users can select a query by its description, fill the query parameters and execute it. The results are presented as the results of the other query form. The queries can be inserted in the Wiki Text through a macro, for example the following macro shows in the resulting page all the provinces of the Uzbekistan:

```
{sparql | SELECT ?x
WHERE { ?x silkrode:partOf silkrode:Uzbekistan .
?x rdf:type silkrode:Province }}
```

3.4 Application architecture

From an architectural point of view, the functionalities (ontology editing, navigation and querying) of the user interfaces are based on the Application API, as shown in Figure 5. The main purpose of this API is to support the manipulation and querying of the ontology through the standard SPARQL query language and through a set of specific adapters, shielding the user from the underlying semantic framework. A plug-in interface, in fact, makes the application independent from the adopted specific semantic framework (as long as SPARQL is supported). A different adapter for every semantic



Financing

label

Financing

title

page content

```
1 Financing Institutions

SilkRoDE would like to thank all Institutions and Private
Individuals who are making our dream possible.

Financing of the first phase was provided by the  {link:UNESCO
office in Tashkent|http://www.unesco.org.uz/album/index.htm} (under
the direction of Michael Barry Lane) with contributions by the
{link:International Institute for Central Asian
Studies|http://www.iicas-unesco.org/eng/eng_index.htm} in
Samarkand, the Universities of {link:Bologna|http://www.unibo.it},
{link:Barcelona|http://www.ub.edu} and
{link:Harvard|http://www.harvard.edu} as well as from the {link:UMR
8546|http://www.archeo.ens.fr/}, the {link:French Institute for
Central Asian Studies|http://www.ifeac.org/},  and the
{link:International Centre for Japanese
Studies|http://www.nichibun.ac.jp/welcome_e.htm}, in Kyoto. The
books were provided by the Institute of Archaeology of the Academy
of Sciences of the Republic of Uzbekistan and the IFEAC (French
Institute of Central Asian Studies).
```

publication year

2007

publication month

10

person

selected values	available values
Alessandro Mosca remove >>	▼ Person ▶ Scholar

role in the group

selected values	available values
	Person role in a group

group

selected values	available values
University of Milano-Bicocca remove >> Complex Systems and Artificial Intelligence research centre remove >>	▼ Group ▶ Institution ▶ Team Publisher

activity type

selected values	available values
	▼ Activity type Study activity type Site Excavation activity type Survey activity type

Fig. 4. Example of datatype properties editing with Wiki Text (on the left) and object properties editing (on the right).

framework is needed because SPARQL is only a query language and does not offers any data manipulation statements (e.g. INSERT, UPDATE, DELETE).

The adapter API supports the manipulation and query of the RDF graphs in two different ways: *frame-centric* and *statement-centric*. The former view is similar to the object-oriented paradigm: every resource is viewed as an object and properties as attribute. This view is used for ontology navigation and resource manipulation. *Statement-centric* is a lower level view in which the graph is represented as a set of triples. Each triple contains three components: subject, predicate and object.

Currently two semantic framework adapters have been implemented: the first one is a wrapper for the Jena Semantic Web Toolkit, the other for the Sesame Framework. The former is an open-source Semantic Web Toolkit[5] aimed at supporting the development of applications that use the Semantic Web information models an languages [20]. We have initially adopted this framework since it matched our requirements, it is widely used within the Semantic Web research community and well documented. This first adapter implementation works well with small ontologies, but fails with larger ones since Jena is not suitable to manage an huge amount of data (a performance evaluation of several frameworks suitable for large OWL ontologies is presented in [21]). For this reason, we choose to develop a new adapter for the Sesame Framework[6] [22]. Sesame provides a number of functionalities for handling (querying and manipulating) RDF graphs. It also supports various types of storage facilities and inference mechanisms.

[5] http://jena.sourceforge.net/
[6] http://www.openrdf.org

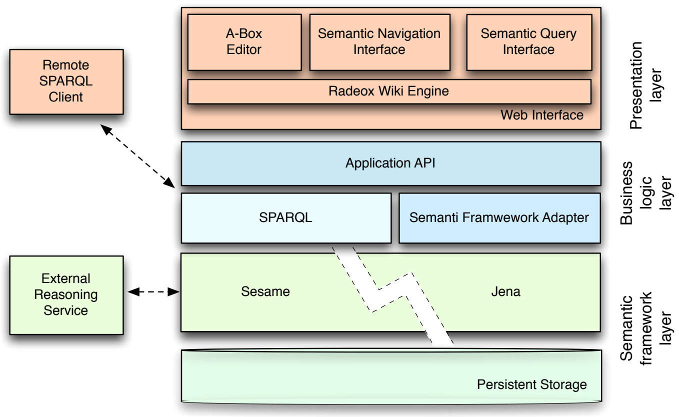


Fig. 5. The NavEditOW architecture

The default implementation supports inferencing and querying on RDF Schema but lacks a specific support for OWL.

4 The SilkRoDE Case Study

SilkRoDE (Silk Roads in the Digital Era) is a project that aims to collect, structure and diffuse all knowledge concerning the Cultural Heritage of Central Asia, including not only archaeological material, sites and historical monuments but also data from fields such as geography, sociology and ethnography. It is an open and evolutive project, which functions as an intelligent network linking all interested institutions, research groups and scholars, that worked and working in Central Asia.

As a first step, SilkRoDE aims to create a Wiki, used on a daily basis by all specialists of Central Asia, interested members of the general public and those involved in Cultural Management. The creation of this Wiki will be possible thanks to a collaboration between specialists from the Humanities and from Computer Sciences in a very close multidisciplinary context. One of the keys to the success of the SilkRoDE project is the decision of all participating scholars, institutions and research groups to work together as Equal Partners. This does not mean that all resources are simply pooled together but that each resource is clearly associated to the authors and funding agencies that enabled its creation. SilkRoDE thus aims to be a Network rather than a new institution. In this perspective, the NavEditOW system represents the general platform adopted by the Project to organize and manage the various different contents of SilkRoDE Wiki. The main aim is to share information and knowledge, and all partners should thus be enabled to employ the system to publish, connect personal data and information, and to identify other groups or Institutions working or interested in same themes. This need for a collaborative and collective participation in data and information collection phases lead us to choose a Wiki–based approach to develop the SilkRoDE portal.

The introduced approach and the NavEditOW system were adopted to represent and organize basic information about institution, research groups and scholars having active researches in Central Asia, as well as information about relevant bibliography and important cultural and archaeological sites. The platform will be integrated with webGIS: the possible connection between specific entities of the SilkRoDE ontology and georeferred entities stored in a GIS will support the visualization of spatial position and distribution of various entities in dynamically generated maps.

The ontological approach provided the required expressiveness and flexibility even in these starting phases of the project, in particular in order to support rich forms of navigation among stored contents. In particular, this approach provided the possibility of representing and managing relationships like "is-a" and "part-of" without flattening the related entities in a single table or splitting them in different tables, as would have been necessary adopting a traditional relational database. For instance institutions, research groups and individual scholars are all actors of the SilkRoDE ontology, in other words, they are individuals belonging to classes that are related to the *Actor* class by an "is-a" relation. It is now possible to define a generic relationship binding an archaeological site to an instance of the *Actor* class or one of its subclasses, without the need to define different relationships. This flexibility in defining and establishing relationships among individuals will support further types of analysis aimed, for instance, at identifying possible connections among actors that are working on similar or related research issues, or geographic areas, or adopting similar methodologies.

5 Conclusions and Future Developments

The paper has described an ontology driven approach to the modeling, design and implementation of dynamic web sites. In particular, we aimed at simplifying the realization of web based systems that exploit and give access to a shared ontology, but these systems should also look like traditional web sites and support simple forms of navigation. To this aim, we endowed the system with a wiki engine and we included documents, and in particular wiki pages, in the ontological tier, to support a simple form of editing of pages that give the site an ordinary structure and appearance, that can be enhanced by means of the exploitation of the underlying domain ontology.

The motivations of this effort, as well as related work and the research context were introduced, and the NavEditOW framework was described, in terms of provided functionalities and architecture. A case study providing the application of the introduced approach and framework was also presented.

Future works are mainly aimed at extending the range of the represented and managed concepts, with particular reference to the various topics that can be used to characterize relevant scientific publications, in an effort similar to the one described in [23]. Moreover, in the medium term, the project will consider the possibility to realize specific wrappers able to to export contents complying to the CIDOC Conceptual Reference Model [24], so as to achieve a high level of interoperability with this relevant standard for cultural heritage information organization.

With respect to the editing policy, it will be tested the possibility to enable end-users to insert new concepts and the respective subclass-relations. Moreover the more extended use of reasoning capabilities will be investigated.

References

1. Atzeni, P., Nostro, P.D.: T-Araneus: Management of Temporal Data-Intensive Web Sites. In Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E., eds.: EDBT. Volume 2992 of Lecture Notes in Computer Science., Springer (2004) 862–864
2. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): a Modeling Language for Designing Web Sites. Computer Networks **33**(1-6) (2000) 137–157
3. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284**(5) (2005) 34–43
4. Bonomi, A., Mosca, A., Palmonari, M., Vizzari, G.: NavEditOW - a System for Navigating, Editing and Querying Ontologies through the Web. In Apolloni, B., Howlett, R.J., Jain, L.C., eds.: KES (3). Volume 4694 of Lecture Notes in Computer Science., Springer (2007) 686–694
5. Berners-Lee, T.: Information Management: a Proposal (1989)
6. Leff, A., Rayfield, J.T.: Web-Application Development Using the Model/View/Controller Design Pattern. In: EDOC, IEEE Computer Society (2001) 118–127
7. Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies **43**(5-6) (1995) 907–928
8. Jugel, M.L., Schmidt, S.J.: The Radeox Wiki Render Engine. In Riehle, D., Noble, J., eds.: Int. Sym. Wikis, ACM (2006) 33–36
9. Lei, Y., Motta, E., Domingue, J.: Modelling Data-Intensive Web Sites with Ontoweaver. In Grundspenkis, J., Kirikova, M., eds.: CAiSE Workshops (1), Faculty of Computer Science and Information Technology, Riga Technical University, Riga, Latvia (2004) 106–121
10. Jin, Y., Decker, S., Wiederhold, G.: Ontowebber: Model-Driven Ontology-Based Web Site Management. In Cruz, I.F., Decker, S., Euzenat, J., McGuinness, D.L., eds.: SWWS. (2001) 529–547
11. Dello, K., Simperl, E.P.B., Tolksdorf, R.: Creating and Using Semantic Web Information with Makna. In Völkel, M., Schaffert, S., eds.: SemWiki. Volume 206 of CEUR Workshop Proceedings., CEUR-WS.org (2006)
12. Schaffert, S.: IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. In: WETICE, IEEE Computer Society (2006) 388–396
13. Souzis, A.: Building a Semantic Wiki. IEEE Intelligent Systems **20**(5) (2005) 87–91
14. Krötzsch, M., Vrandecic, D., Völkel, M.: Semantic MediaWiki. In Cruz, I.F., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L., eds.: International Semantic Web Conference. Volume 4273 of Lecture Notes in Computer Science., Springer (2006) 935–942
15. Krötzsch, M., Schaffert, S., Vrandecic, D.: Reasoning in Semantic Wikis. In Antoniou, G., Aßmann, U., Baroglio, C., Decker, S., Henze, N., Patranjan, P.L., Tolksdorf, R., eds.: Reasoning Web. Volume 4636 of Lecture Notes in Computer Science., Springer (2007) 310–329
16. Ram, S., Shankaranarayanan, G.: Modeling and Navigation of Large Information Spaces: a Semantics Based Approach. In: 32nd Annual Hawaii International Conference on System Sciences (HICSS-32), 5-8 January, 1999, Maui, Hawaii, Track 6: Modeling Technologies and Intelligent Systems., IEEE Computer Society (1999)

17. Gennari, J.H., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The Evolution of Protégé: an Environment for Knowledge-Based Systems Development. Int. J. Hum.-Comput. Stud. **58**(1) (2003) 89–123
18. Haarslev, V., Möller, R.: Racer: a Core Inference Engine for the Semantic Web. In Sure, Y., Corcho, Ó., eds.: EON. Volume 87 of CEUR Workshop Proceedings., CEUR-WS.org (2003)
19. Tsarkov, D., Horrocks, I.: Fact++ Description Logic Reasoner: System Description. In Furbach, U., Shankar, N., eds.: IJCAR. Volume 4130 of Lecture Notes in Computer Science., Springer (2006) 292–297
20. McBride, B.: Jena: a Semantic Web Toolkit. IEEE Internet Computing **6**(6) (2002) 55–59
21. Guo, Y., Pan, Z., Heflin, J.: An Evaluation of Knowledge Base Systems for Large OWL Datasets. In McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: International Semantic Web Conference. Volume 3298 of Lecture Notes in Computer Science., Springer (2004) 274–288
22. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: an Architecture for Storing and Querying RDF Data and Schema Information. In Fensel, D., Hendler, J.A., Lieberman, H., Wahlster, W., eds.: Spinning the Semantic Web, MIT Press (2003) 197–222
23. Bonomi, A., Mantegari, G., Vizzari, G.: A Framework for Ontological Description of Archaeological Scientific Publications. In Tumarello, G., Bouquet, P., Signore, O., eds.: Semantic Web Applications and Perspectives 2006, Proceedings of SWAP 2006, the 3rd Italian Semantic Web Workshop, Pisa, Italy. Volume 201 of CEUR Workshop Proceedings. (2006)
24. Doerr, M.: The CIDOC CRM, an Ontological Approach to Schema Heterogeneity. In Kalfoglou, Y., Schorlemmer, W.M., Sheth, A.P., Staab, S., Uschold, M., eds.: Semantic Interoperability and Integration. Volume 04391 of Dagstuhl Seminar Proceedings., IBFI, Schloss Dagstuhl, Germany (2005)