An Ontology Driven Web Site and its Application in the Archaeological Context

Dipartimento di Informatica, Sistemistica e Comunicazione
Università di Milano-Bicocca

bandini@disco.unimib.it
ale.m@disco.unimib.it
matteo.palmonari@disco.unimib.it
andrea.bonomi@disco.unimib.it

tel. 02 6448 7835
tel. 02 6448 7879
Main motivation: growing need of semantics

- Advanced information systems have a growing need of semantics
  - To improve existing functionalities and services (e.g. browsing or searching for digital contents in web portals)
  - To support interoperability of heterogeneous repositories
  - To enable advanced functionalities in innovative architectures (SOC and Semantic WS)

- Ontologies and Semantic Web languages and tools are growingly investigated and adopted as means of providing information systems with the required level of semantics
From data to ontology driven web sites

From the initial hand-made HTML pages

... to data driven web sites

... to methodologies supporting design of data driven web sites (e.g. Web ML).

Why not ontology driven web sites?
Some problems and opportunities

- Some problems:
  - **Knowledge** (even in “basic” terminological forms) **must be acquired**
  - For most practical applications, **ontologies** can not be seen as static entities, they rather **change over time**

The need of proper instruments supporting the **acquisition and maintenance of ontologies** from an end-user perspective has often been overlooked...

- ... especially considering end-users that are not experts in Semantic Web technologies!
A sample scenario of application

- A publicly accessible repository of information and documents about cultural resources of Central Asia
  - Partners from different countries must be able to access the system and contribute...
  - Users have different skills and competences, but *not in Semantic Web technologies*...
  - Due to the nature of the project, information, documents but also knowledge will be *incrementally added*
  - Users *must be involved* in the definition and management of the *domain ontology*...
  - ... but the system must “help” them!
A sample scenario of application

- The domain ontology can be useful
  - to support users that are members of the “community” in annotating information and documents
  - to support internal mechanisms improving the navigation experience (semantic querying, semantic browsing)
    - e.g. “find all papers about Samarkand region”
  - to supply simple visitors with a comprehensive view of the treated topics
ArcheoServer.it

• In 2005, the chair of Prehistory and Protohistory - University of Milan, the Department of Informatics, Systems and Communication - University of Milan-Bicocca and the Department of Archaeology - University of Bologna, have started a long-term project for the creation of a set of Web-oriented services aimed at supporting the sharing of knowledge on prehistory and protohistory in Italy.

• The main objective is the creation of a Web portal which will provide a collaborative platform for the exchange of scientific information among the communities of Italian archaeology researchers.
Main functions

- **e-Library**: Digital Library helping users to find information about scientific publications.
- **Ceramic types Db**: Online database of ceramic type collections.
- **Web GIS**: Web GIS of prehistorical sites in Italy.
- **Collaborative editing platform**: Collaborative editing platform for Archaeology researcher and students.
- **Sites Db**: Online database of prehistorical sites in Italy.
Navigation Details

Web GIS

e-Library

Search publications

Publication

Places / publications

Show details

Show publication ceramic types

Sites / publications

Sites / Web GIS

Sites / Search

Show details

Show site ceramic types

Search publications

Ceramic types

Search

Show types distribution

Show publications

Show site details

Ceramic type
A Navigation Example

Ceramic types

Search form

Show query results

Results

Show details

Details

Web GIS

Export data

Show ceramic types distribution in the Web GIS
e-Library

- Digital library: collections of bibliographical information + effective mechanism for the retrieval

- The main requirement is to give the community itself the possibility of autonomously managing the contents by means of simple editing tools

- Bibliographic data will be entered manually by the students, while archaeology professors and researchers will supervise the work

In most cases, archaeologists have just low-level technical competence
Application Scenario

- In order to effectively describe contents beyond a keyword based approach, and to support **effective forms of information retrieval**, users must have available a **domain ontology** whose elements can be selected as **indicators of the topics treated** in the publications.

Example

Searching for: “Sicily, Bronze Age”

The system will retrieve the article “The early bronze age Village of Mursia” annotated as “Mursia”, “Early Bronze Age”
(A piece of the) Domain ontology

- The ontology has to be designed by archaeology professors and researchers with the aid of knowledge engineers.
- The e-Library provides on-line tool to maintain and evolve the ontology.
Content Description Approach

- The ontology T-Box has to be designed by archaeology professors and researchers with the aid of knowledge engineers.
- The e-Library supports the maintenance on the domain ontology A-Box component.
- Publications description is done by Archaeology students associating A-Box elements to the publications.
- End user can view ontology and bibliographic data
NavEditOW

- The e-Library is based on **NavEditOW**, a system for the visualization, navigation, querying, updating and maintenance of ontologies through the web.

- In order to support **navigation**, **editing** and **query** of ontologies for users with little or no knowledge of formal languages in which they are represented, a number of features should be implemented.

- Ontology visualization is essential for the user to explore the available information and it also helps non-expert users to refine their search requirements, should they start with no specific requirement in mind.

[Image of NavEditOW interface]

http://www.lintar.disco.unimib.it/NavEditOW
NavEditOW
Navigation & Editing of Ontologies on the Web

- NavEditOW is a framework supporting the design and realization of web based systems supporting the navigation, editing and querying of knowledge bases structured according to Semantic Web related models and technologies
  - NavEditOW strongly emphasizes the usefulness of ontology visualization in supporting these functionalities
- Systems based on NavEditOW are aimed at supporting users not having any competence on Semantic Web languages and instruments
- Therefore… functionalities supporting designers and especially end-users of the final application are needed: focus on…
  - Navigation
  - Editing
  - Query
- OWL DL based ontologies
System Architecture

- **Ontology Storage**
  - owl (T-Box, A-Box)
  - SWRL rules
  - Reasoner + SWRL Engine
  - SPARQL Engine

- **Ontology Update**
  - Remote SPARQL Client
  - External Reasoning Service
  - Ontology Storage

- **Ontology Update**
  - A-Box Editor
  - Semantic Query Interface
  - Semantic Navigation Interface
  - Web Interface

- **SPARQL**
  - SPARQL Engine
  - Semantic Framework Adapter
  - Jena
  - Persistent Storage

- **User Interface**
  - SPARQL queries
  - results
System Architecture Details

The presentation layer is a Web-based user-interface. The aim of the persistence layer is to store the ontology. The semantic framework layer includes Semantic Query Interface and Semantic Navigation Interface. SPARQL is used for reasoning and querying. External Reasoning Service can be used for remote SPARQL queries. Persistent Storage is used to store the ontology.
Ontology visualization is essential for the user to explore the available information and it also helps non-expert users to refine their search requirements, should they start with no specific requirement in mind.

With the navigation interface the users can view ontology individuals and their properties and browse properties via hyperlinks.

http://www.lintar.disco.unimib.it/space/Cultural+Resource+Management/NavEditOW
Visualization formal point of view

- 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`).
Our approach is to use a tree navigation and to visualize the details of the selected item only.

The navigation tree is a hierarchy of classes, and individuals connected by subClassOf, instanceOf and other domain-specific properties.
A common problem with ontology is a flat visualization of individuals.

For certain classes of relations, their formal properties can provide structure to the individual organization.

Example

- **tree-like properties** (not symmetric and inverse functional relations) 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).

- **total and partial orders** kind of relations exploited for the visualization are relations defining total orders on individuals (e.g. $isFollowedBy$).
Structured visualization of individuals (2)

- OWL (RDF Schema, RDF) allows to represent only binary relations.

- \( n \)-ary relations \((n>2)\) are generally reified into concepts with \((n\text{ properties})\)

- Example:

  The \textit{reifiedProperty} annotation property is used to mark the classes that are reified relationships. A reified relationships is a class that connects a subject with an object, so that we could attach additional attributes to the relationship. For example, if we want to define a “\textit{roleInGroup}” property that connect an actor, a group, a role and a time interval to this relationship, we have to introduce a class (e.g. \textit{ReifiedRoleInGroup}).

- Individuals of such classes are not true domain entities, but basically “db-style” tuples and need a special treatment for the visualization.
Editing

- The NavEditOW allows the users to create, edit and remove individuals of the ontology, their properties and labels (A-Box editing).

- Contextual editing:
  - users can also create new individuals related to an existent one exploiting the tree-like properties (e.g. partOf)
  - users are supported in the selection of the property filler bases on the semantics (co-domain, range restrictions)

![Editing interface](http://www.archeoserver.it/elibrary.owl#LuogoGeografico11816609:
Lombardia > Varese
Istanza di Luogo Argomento LuogoGeografico
create new instance | elimina

Varese
label: Varese
nomeLuogo: VA

<table>
<thead>
<tr>
<th>Bronzo Medio</th>
</tr>
</thead>
<tbody>
<tr>
<td>label: Bronzo Medio</td>
</tr>
</tbody>
</table>

SUCCESSIVO A

<table>
<thead>
<tr>
<th>Bronzo Antico</th>
<th>PeriodoStorico</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Protostoria</td>
</tr>
<tr>
<td></td>
<td>Preistoria</td>
</tr>
<tr>
<td>valori effettivi</td>
<td>valori disponibili</td>
</tr>
</tbody>
</table>

PRECEDENTE A

<table>
<thead>
<tr>
<th>Bronzo Recent</th>
<th>PeriodoStorico</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Protostoria</td>
</tr>
<tr>
<td></td>
<td>Preistoria</td>
</tr>
<tr>
<td>valori effettivi</td>
<td>valori disponibili</td>
</tr>
</tbody>
</table>

Datatype property: the datatype editor allows editing a literal values, displayed as a text input box.

Object property: Cardinality and range restrictions for properties used to support users while editing.

Querying

Web-based Query interface

- SPARQL Query Interface
- Predefined Queries
- Custom Query Form

Protégé like interface...

Three query interfaces:

- SPARQL Query Interface
- Predefined Queries
- Custom Query Form

user

Web-based Query interface

NavEditOW

semantic framework
(query, inference)

SPARQL

Sesame, Jena, ...

http://www.archeserver/ArchLibrary.ow`a`l rdf:type owl:Ontology
this:Palafitta rdf:type this:TipologiaStruttura
this:Templo rdf:type this:TipologiaStruttura
this:AmbienteDiCottura rdf:type this:TipologiaStruttura
this:Ambiente rdf:type this:TipologiaStruttura
this:Tomba rdf:type this:TipologiaStruttura
this:Capanna rdf:type this:TipologiaStruttura
this:OfficinaLithica rdf:type this:TipologiaStruttura
this:TipologiaSito owl:disjointWith this:TipologiaStruttura
this:TipologiaRapporto owl:disjointWith this:TipologiaStruttura
this:TipologiaPubblicazione owl:disjointWith this:TipologiaStruttura
this:OrganizzazioneSociale owl:disjointWith this:TipologiaStruttura
node12op4u79k1 owl:inValuesFrom this:TipologiaStruttura
tipologiaStruttura rdf:range this:TipologiaStruttura
tipologiaStruttura rdf:type owl:Class
this:TipologiaSito rdf:type owl:Class
Predefined Parametric Queries

• The Predefined Queries Interface is based on a predefined set of queries.

• The interface guides the user in the construction of a query by means of a wizard which enables the generation of query expressions.

• Every predefined queries is composed of a description in natural language and a SPARQL query with (eventually) free parameter.

1) Query selection
2) Parameters filling
3) Query execution and results
Custom Ontology Navigation-based Query

• In some application, it could be useful to define customized form.

• For example, in the ArcheoServer eLibrary (digital library), the publications are annotated with topic selected from a domain ontology in order to support effective forms of information retrieval.

• The query form allows searching for papers characterized by the “traditional” bibliographics fields and topics chosen from the domain ontology.

http://www.archeoserver.it/eLibrary/
In the ArcheoServer eLibrary, the NavEditOW is integrated within a “traditional” database-based digital library.

It allows the content editors to associate a ontology-based description to the publications and the end-user searching for papers characterized by a specific topics.

http://www.archeoserver.it/eLibrary/
Wiki

- The NavEditOW is able to render WikiText used as value of datatype properties.

- It is possible to use NavEditOW a semantic wiki engine that has an underlying ontological model of the knowledge described in the pages.

WikiText is a markup language that offers a simplified alternative to HTML and is used to write pages in wiki websites such as Wikipedia.
NavEditOW’s ontology annotation for the visualization

- The visualization is driven by a set of annotation.
- Annotation properties on the OWL ontologies are exploited to present the contents (OWL allows classes, properties, individuals and the ontology itself to be annotated)
Conclusions and Future Works

- NavEditOW is currently adopted and in use in two projects (Archeoserver and SilkRoDE)
  - both as an element for annotation of digital contents of an e-Library to support semantic queries
  - and as the main component of a web-based architecture entirely based on an ontology

- Future works
  - Comprehensive evaluation of the tool (expressiveness and adequacy, also with reference to performances)
  - Investigation of several methodological issues

http://www.archeoserver.it
http://www.silkrode.org
Conclusions and Future Works

NavEditOW v2

Extended editing capabilities

User Interface

Database back-end

Modular Architecture (plug-in)

Open standard

Efficient triple store

Versioning

Security

Versioning

Improved query form

Open standard

Web 2.0

Customization

Usability

User Interface
Useful links

ArcheoServer: http://www.archeoserver.it/
SilkRoDE Web Portal: http://www.silkrode.org/
NavEditOW: http://www.lintar.disco.unimib.it/NavEditOW