Sören Auer¹,², Sebastian Dietzold ², and Thomas Riechert ²
¹ University of Pennsylvania, Department of Computer and Information
² Universität Leipzig, Institut für Informatik

ISWC 2006

OntoWiki – A Tool for Social, Semantic Collaboration

Presentation by Patrick Roos
OntoWiki Purpose

- Tool supporting agile, distributed knowledge engineering in a (semantic) web environment
- Connect existing information resources
- Collaborative ontology and knowledge base generation

→ Similar goals, issues, and challenges to SIB-data project
Recent approaches:
- Integrate RDF triples into Wiki texts in a special syntax

Obstacles:
- Usability:
  - Adding more and more syntactic possibilities counteracts ease of use for editors.
- Redundancy:
  - To allow the answering of real-time queries to the knowledge base, statements have to be stored additionally in a triple store. This introduces a redundancy complicating the implementation.
- Scalability:
  - Knowledge base changes which involve statements with different subjects will scale very bad since all corresponding Wiki texts have to be parsed and changed.
OntoWiki Strategy

- Wiki Paradigm:
  - “making it easy to correct mistakes, rather than making it hard to make them”
  - Simplest “DB” that could possibly work
- OntoWiki:
  - Simplest Knowledge Base that could possibly work
  - Simplify the presentation and acquisition of instance data from and for end users
  - Regard knowledge bases as *information maps*
Information Maps

- RDF knowledge bases
- Node (RDF resource): Web-accessible pages interlinked to related digital resources
- Divided into three parts:
  - Left sidebar: selection of content
  - Main content section
  - Right sidebar: content specific tools and complementary information
Selection options include the set of available knowledge bases, a class hierarchy browser and a full-text search.

Once a selection is made, the main content section will arrange matching content in a list view linking to individual views for individual instances.

Different views facilitated (e.g. map, calendar)
editing: WYSIWYG'esque

Conference > ISWC 2006
- title: Fifth International Semantic Web Conference
- place: Athens, Georgia, USA
- logo: ISWC 2006
- start: 2006-11-05
- end: 2006-11-09
- topics: SemanticWeb
- URL: http://iswc2006.semanticweb.org/
- rdfs:comment: The dream of the Web was to create a human communication and collaboration platform for sharing knowledge and enabling a universal space for

topics:
- SemanticWeb
- Data
- Topic: DataMining
- Topic: DataModeling
- Topic: DataVisualization
- Conference: DEXA 2006
- Conference: DMIN2006
Knowledge bases become increasingly advantageous, if once defined concepts (e.g. classes, properties, or instances) are as much reused and interlinked as possible.

- AJAX technology to interactively propose already defined concepts.
- All URI references and literals are indexed for full-text searches in the statement repository.
OntoWiki system provides a library of reusable user interface components for data editing, called widgets. Customized for usage in specific contexts by widget configurations. Basic Widgets:

- Statements: allows editing of subject, predicate, and object
- Nodes: edit literals or resources
- Resources: select and search from/for existing resources
- Literals: literal data in conjunction with a data type or a language identifier
- Widgets for specific literal data types: e.g. dates, HTML fragments.
- File widget: allows uploading of files to the OntoWiki system.
Social Collaboration

- Change tracking
- Commenting
- Rating
- Activity/Provenance
Structuring and representation of content should be employed to enhance the retrieval of information for human users

- Facet-based Browsing
  - widely deployed in the shape of tagging systems of the Web 2.0

- Semantic Search
  - full-text search for one or multiple keywords occurring in literal property values
## Semantic Search

### Search

Filter: All classes  
All properties  
Submit Query  

Search returned 6 results.

<table>
<thead>
<tr>
<th>Relevance</th>
<th>Resource</th>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td>AssistantProfessor: York Sure</td>
<td>name</td>
<td>York Sure</td>
</tr>
<tr>
<td>100%</td>
<td>InProceedings: REMINDIN': Semantic Query Routing in Peer-to-Peer Networks Based on Social Metaphors</td>
<td>address</td>
<td>New York, USA</td>
</tr>
<tr>
<td>100%</td>
<td>InProceedings: Reuse Of Problem-Solving Methods In Knowledge Engineering (short paper)</td>
<td>booktitle</td>
<td>Proceedings of the 6th Annual Workshop on Software Reuse (WiSR-6), Owego, New York, November 1-4, 1993</td>
</tr>
</tbody>
</table>
Architecture: POWL

- Web-based collaborative semantic web development platform (PHP and MySQL): [powl.sourceforge.net](https://powl.sourceforge.net)
- Parsing, storing, querying, manipulating, serving and serializing OWL knowledge bases
- Scalable
  - Based on RDBMS
- Supports
  - Versioning
  - Caching
Storage Layer (Powl Store)

- SQL compatible rel. DB supported by AdoDB
- Denormalized schema: resources and literals written in full in a table row representing an RDF statement (2-3 x faster than normalized)
- Tables used to store all information related to ontologies and their evolution

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>models</td>
<td>provides information about the models in the store</td>
</tr>
<tr>
<td>statements</td>
<td>contains all statements of models in the store</td>
</tr>
<tr>
<td>log_actions</td>
<td>holds information about editing actions on a model</td>
</tr>
<tr>
<td>log_statements</td>
<td>contains added and removed statements for every action</td>
</tr>
</tbody>
</table>
API Layer

- RDFAPI
  - Parser, serializer for different RDF serializations (XML, N3, N-Triple)
  - RDQL declarative query backend
  - Classes and methods for working with RDF models, resources and literals
  - NetAPI for publishing models on the web

The higher layered APIs, RDFSAPI and OWLAPI, extend the classes “Model”, “Resource” and “Literal” provided by RDFAPI to RDF-Schema specific classes.
- Methods exposed by the main RDFSAPI classes

<table>
<thead>
<tr>
<th>RDFSModel</th>
<th>RDFSClass</th>
</tr>
</thead>
<tbody>
<tr>
<td>add{T, C, P, I}</td>
<td>add{I, P}</td>
</tr>
<tr>
<td>count{T, C, P, I}</td>
<td>count{I, S}</td>
</tr>
<tr>
<td>find{T, R}</td>
<td>find{I}</td>
</tr>
<tr>
<td>get{T, R, C, P, I}</td>
<td>get{I}</td>
</tr>
<tr>
<td>listTop{C, P}</td>
<td>list{S, SC, P, I}</td>
</tr>
<tr>
<td>list{R, C, P, I}</td>
<td>remove{I, P}</td>
</tr>
<tr>
<td>remove{T, R, C, P, I}</td>
<td>set{S, SC, P, I}</td>
</tr>
<tr>
<td>load</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RDFSProperty</th>
<th>RDFSInstance</th>
</tr>
</thead>
<tbody>
<tr>
<td>add{D, R}</td>
<td>add{P}</td>
</tr>
<tr>
<td>add{S, SP}</td>
<td>get{C, C}</td>
</tr>
<tr>
<td>get{D, R}</td>
<td>get{P}</td>
</tr>
<tr>
<td>list{D, R}</td>
<td>list{P}</td>
</tr>
<tr>
<td>list{SP, SP}</td>
<td>list{PV}</td>
</tr>
<tr>
<td>remove{D, R}</td>
<td>set{PV}</td>
</tr>
<tr>
<td>set{D, R}</td>
<td></td>
</tr>
</tbody>
</table>
Scalability

- Only those parts of the ontology are loaded into main memory which are required to display the information requested by the user on the screen.
- Class subtrees are loaded on demand using Javascript.
- Benchmark with only MySQL's query cache optimization:

<table>
<thead>
<tr>
<th>Model</th>
<th>Triple count</th>
<th>Import time</th>
<th>Classes</th>
<th>Class hierarchy calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wordnet$^2$</td>
<td>473 528</td>
<td>624 s</td>
<td>6</td>
<td>0.30 s</td>
</tr>
<tr>
<td>NCI Cancer Ontology$^3$</td>
<td>463 878</td>
<td>597 s</td>
<td>27 652</td>
<td>0.46 s</td>
</tr>
<tr>
<td>UNSPSC$^4$</td>
<td>82 500</td>
<td>82 s</td>
<td>16 499</td>
<td>1.06 s</td>
</tr>
</tbody>
</table>
Application domain:
- Environment:
  - OntoWiki is a Web application and presumes all collaborators working in a Web environment, possibly spatially distributed.
- Usage Scenario:
  - OntoWiki focuses on knowledge engineering projects where a single, precise usage scenario is either initially not (yet) known or not (easily) definable
- Reasoning:
  - Application of reasoning services is (initially) not mission critical.
- Budget:
  - Only a small amount of financial and personnel resources are available
Conclusions

- Showed how tool support for agile, collaborative knowledge engineering scenarios can be provided
- Visual presentation of a knowledge base as an information map, social collaboration aspects as well as a semantic search strategy decrease the entrance barrier for domain experts to collaborate
- Which ideas would be good for SIB-data?