Semantic Utilities and E-Learning

Tomáš Gregar, Tomáš Pitner, Miroslav Warchil, Jan Mudrák,
Ondřej Zezula, Jakub Talaš, Igor Žemský, Zdeněk Lévek
Faculty of Informatics, Masaryk University, Botanická 68a, Brno, Czechia
(xgregar,tomp}@fi.muni.cz,(warmir,marduk,172856,talas,143407)@mail.muni.cz

Keywords: e-learning, semantics, owl, fresnel, visualization, annotation

Abstract: Machines knowing content of the documents, i.e. the idea of semantic web, could help with transmitting the knowledge. The lack of user-friendly tools for acquiring, storing and utilizing document semantics inflicts very rare usage of semantics among non-specialized users. Semantic-web research concerns with theoretical issues. We designed the project SELE (Semantic e-learning) to cope with some of these issues. This project allows user to create open source light-weight tools for gathering semantics from real usage, usable in further research, and to use these tools (and also their development) in the learning. This paper describes a structure of the framework, their basic processes and developed tools and modules.

Introduction

Semantic web (Berners-Lee, 2001) remains still the subject of academic discussions rather than usable technology. Its need of metadata leads to unrealistic requirements on the users or excessive requirements on resources and time due to incorporation of sophisticated automatic methods. Real-life applications incorporated much simpler methods for describing the document content – tagging and folksonomies (Li et al., 2008). That is Web 2.0 technology building up the semantics from the low level, combining “atoms” of knowledge of each user. Using Web 2.0 tools in the learning corresponds with the Connectivism learning theory (Pitner, 2007).

Semantic E-learning Project

The goal of SELE project is a framework of tools combining Web 2.0 approach, more elaborate backbone technology, and higher usability than semantic-web prototype tools. Requirements evolved as a result of semantic tools survey:

- The user experience is more important than level of contemporary semantic technology usage, i.e. ontologies used in tools are lighter, tools shall use a set of corresponding collaborating specialized ontologies rather than one heavy-weight;
- Standardized technology (OWL for semantics, Fresnel for visualization etc.), open-source and open interfaces to other processes (learning etc.) are vital. Framework should have modular architecture.
- Tool, as a part of the learning process, should be easy to understand, correct, concise. That eases the shared consensus creation of the concepts of the learned domain. It is very important requirement, because students do not dispose any base knowledge in the beginning.

Any Web 2.0 service is centered on repository of some interesting data – maps, wiki pages, documents etc. In SELE it is the (e-learning) document repository. There are plenty of legacy learning documents on web or specialized data not initially intended to be used as learning resource. The work with general formats allows users to use corresponding processors with better abilities and user experience than web CMSs have.

As our team teaches a couple of well-attended courses (Java Programming, Mark-up languages), another requirement is to use tools not even FOR the learning processes, but also to use their development in the learning (as Master, Bachelor theses or interim projects) – as a part of the Inclusive
universal access methodology (Pitner et al., 2007)); and to integrate learning tools already used – like project management system (Gregar et al., 2009).

Document Structure

We view the document as a general multimedia container, containing the (structured) text, set of binary data, and semantic metadata (like EXIF, ID3 etc.) in SELE:

\[ D = T + S + B \]  \hspace{1cm} (1)

Where \( T \) is the textual content, set of words, \( S \) is the set \((S_1 \ldots S_m)\) of semantic information about the document or some of its parts; \( B \) is the set of binary data. I.e. metadata could describe text in wiki page, pictures, or document per se.

The goal is to identify, to detect and to reveal this logical structure in every type of document the framework could be engaged. Maximizing the dimension of \( T \) (and \( S \)) allows better computerization and utilization – i.e. to use same processes on the documents originally from different processors. We designed the XML format (DocBook derivate) for the formatted text description. It allows profiting from semantic web and Web 2.0 technologies with preservation of easy document authoring in known user interface. The main issue is importing and exporting various document types. The easier process is indeed for the XML-based documents like ODF (first implemented module), Office Open XML or DocBook, where XSLT can be incorporated. But every document type \( t_{\text{import}} \) identifies unambiguously the function \( f_{\text{trans}} \), which for the input document can return its sub-elements:

\[ D = f_{\text{trans}}(t_{\text{import}}, t_{\text{DocBook}}, D_{\text{import}}) \]  \hspace{1cm} (2)

That transformation is not always symmetrical (DocBook can have lower descriptive power). API of the format can be incomplete, i.e. reverse transformation of the \( D \) into \( t_{\text{import}} \) do not end with \( D_{\text{import}} \). Hence we store also copy of initial document state: \( B = \{B, D_{\text{import}}\} \).

Semantic description of the document has two subsets. There are general level metadata (subset \( S_G \), major part is gained within the import, using standards like Dublin Core), and results of voluntary annotation of terms \( T_j \), binary elements \( B_k \), or its segments (subset \( S_A \)). Annotation function \( SF_A \) declares the projection of term to its semantic representation.

\[ S_A = \{S_j | S_j = SF_A(T_j), T_j \subseteq T\} \cup \{S_k | S_k = SF_A(B_k), B_k \subseteq B | | B_k \subseteq B_l \subseteq B\} \]  \hspace{1cm} (3)

Binary structures are bound to the document by the DocBook vocabulary. Semantic data are connected by standard RDF structures and Linked Data rules. Export from the internal structure can be performed to the initial format, or any other known by the system. The storage of semantics gives us the opportunity to show formerly hidden knowledge of the documents (i.e. \( S \)), sometimes as the only information to export (Gregar, 2009).
Content Management System

The module responsible for the processing of the documents, their importing and exporting, is called ASCMS (Apache Servicemix Content Management System) (Mudrák, 2009). The internal structure of the system is based on Service oriented architecture (SOA). ASCMS use Apache ServiceMix ESB framework and Apache ODE for service orchestration and BPEL language for service composition. Because of ServiceMix bug, system is temporarily exposed in Apache Tomcat via Codehouse XFire.

Framework works with Entrypoint interface for describing services, which can import (Capture process in ECM terminology) documents (see Figure 1). Such services implement transforming function $f_{trans}$. Reference Entrypoint implementation (called DocTransformer) works with OpenOffice.org (OOo) server side instance. It use library ASCMS2OO to access it and transform content of OOo processable formats into internal structure. But such interface could be implemented also in dedicated modules (DOCX, DocBook, XHTML is under development). The Entrypoint services can be used also for export of the document.

Repository

Document repository is composed of the three databases:

- XML database (eXist and dbXMLTL library of the ASCMS) bound via JBI (Java Business Integration). XML structure allows text mining, indexing, simple fulltext search (implemented by ASCMS module XMLDocStorage).
- Relation database (Apache Derby, persistency is based on JPA+ implementation EclipseLink). It stores initial version of the document and binary parts of it. Binary data can be searched via the similarity search.
- Semantic database (OpenRDF Sesame 2 and eXist). Smaller part stored in XML database is the directory service for managing the documents in CMS (managed by ASCMS module Metadata). Data store for the document metadata and stored semantics is OWL repository Sesame 2. Module SemanticMetadata use it in CMS-related processes for storing document-metadata in Dublin Core vocabulary. Sesame 2 was selected as a result of our internal survey.
We selected the lightweight approach of semantic storage via instantiating the set of domain ontologies (Al-Khalifa, 2008) – for example project development description ontology should annotate only bits of document which have some value (project author, collaborators etc.). Such instantiating of independent ontologies creates the set of independent semantic views. And that independence restricts future utilization. But it can be withdrawn with Linked Data (http://linkeddata.org/) principles or use of upper ontologies. The semantic utilization also needs different views on the declared semantics. We use Fresnel Lens (Pietriga, 2005) for defining the intention-ruled sub-graphs.

Visualization

For better knowledge transition we need to visualize the semantics. There are more attempts to visualize the RDF data. It became obvious that RDF browsers are almost always solving the same issues: selection of relevant subset of RDF data and adding visual information to it. Hence W3C started an effort to provide a declarative and universal way to handle these common issues, to standardize visualizing approaches and avoid reimplementation. As a result Fresnel standard was developed in June 2005. It is a display vocabulary for RDF (Pietriga, 2005), capable to select the data (Fresnel Lens) and visual style and data binding (Fresnel Format).

Some applications already used parts of Fresnel standard, for example RDF Browser IsaViz (http://www.w3.org/2001/11/IsaViz/). There is Java library JFresnel implementing the basic vocabulary of the standard. The nonexistence of authoring tool inhibited the standard. We developed the modular application Fresnel Editor (FE) for semi-WYSIWYG editation and further visualization. JFresnel was used as a technological base (it was extended during development). FE uses framework Spring on the application level and component Lobo for the web-visualization.

FE creates an environment for the accessing the semantic data (stored in Sesame 2 repository). It also use the repository of Fresnel definitions (also in RDF, hence these two data connections can point to one repository). The initial version of the system implements the modules for definition of Lens and Formats in semi-WYSIWYG way. It allows binding them in Groups and activating the visualization. The alternative Lens and Formats modules for full WYSIWYG definition (for non-IT specialists) are under development, as well as alternative visualizer capable to export the output in SVG, or the set of specialized Fresnel Stylesheets. More information about this can be found in (Gregar et al., 2010).

![Fresnel Editor structure](image)

**Figure 2: Fresnel Editor, structure.**

Project Management System

We analyzed the requirements for different project categories managed at MU (Gregar et al., 2009) and described the processes identified and used to cope with them.

We used the framework of *Subversion* for storage of the research projects, *Wiki* for output presentation (with light usage of Wiki as a blended learning system), project describability and project
status examining, and Maven 2 for quality measurements of the project. Later we created a fork of the Open Source system Trac called Deep Thought (DT). DT can manage multiple projects and integrate the former distinct tools in one mash (Pitner et al. 2007). The second-level integration comes with the utilization of the semantic repository in the wiki module. The module utilizes library RDFLib and RDFAlchemy to connect to semantics repository of the framework (Sesame 2).

The system gained the set of wiki-macros to annotate the content or visualize automatically selected semantic information (like automatically generated list of software projects or project calendar). The macros are ready for a set of widely used, Linked Data aware ontologies, like DOAP (description of the project) or FOAF (Friend of the friend), O3 (ontology metadata), GEO (geographical localization), DC (Dublin Core metadata vocabulary) etc. But the content can be annotated with any RDF-based ontology.

Recently we started to develop the GUI for easy annotation of the content. The wiki-module, as well as the system itself is under testing now. It is used for the management of the interim student projects.

**Iris, Image Annotation**

Framework subsystem for picture annotation is called Iris. It is the simple graphic editor capable of segment the images automatically, describe their content via MPEG-7 descriptors, store that information and also suggest concepts for picture segments depending on the selected semantic domain. User can follow that suggestion, or assign semantics independently. The semantic information about the graphic segmentation, its properties and also the resulting annotation is stored in general multimedia ontology COMM (Core Ontology for Multimedia, http://comm.semanticweb.org/) (Arndt et al., 2007). The tool has two distinct prototypes – one developed in .NET framework (Java API of COMM was translated in C#), another is under development in Java. The MPEG-7 descriptors are extracted via library Caliph and Emir.

Contribution of Iris is its more mature user interface than existing image-annotation tools. It also works as a browser of already defined annotations. The semiautomatic proposition of the concept is based on the computer vision techniques, uses MPEG-7 standard descriptors, COMM ontologies for describing the picture structure and semantics, and processes of machine learning (Gregar and Pospíšilová, 2006).

**Conclusion**

The paper gave you the brief overview of the framework structure and their modules in particular. Its development continues. This semester we started to use the semantically enriched project management system in the learning, so some interesting usage data will follow. The mayor part of the tools were developed as Master or Bachelor theses, the visualization editor came public and it is available on sourceforge.org. There are also other modules, like ontology editing interface into editor Protégé, or newly designed web interface for browsing stored RDF definitions LinkedData-friendly, i.e. in the human readable way.

**Acknowledgement**

Semantic e-learning project development is partly supported by Rector’s programme to support MU student’s creative work, ID: MUNI/G/0121/2009.

**References**


Mudrák, J, 2009: Document processing in distributed environment, Diploma Theses, MU (in Czech)