Ontology-based Question Answering for Digital Libraries

Stephan Bloehdorn[1], Philipp Cimiano[1], Alistair Duke[2], Peter Haase[1], Jörg Heizmann[3], Ian Thurlow[2], and Johanna Völker[1]

[1] Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
{bloehdorn,cimiano,haase,heizmann,voelker}@aifb.uni-karlsruhe.de
[2] British Telecom, Adastral Park, Ipswich IP5 3RE, UK
{alistair.duke,ian.thurlow}@bt.com
[3] ontoprise GmbH, Amalienbadstr. 36, D-76227 Karlsruhe, Germany
heizmann@ontoprise.de

Abstract. In this paper we present an approach to question answering over heterogeneous knowledge sources that makes use of different ontology management components within the scenario of a digital library application. We present a principled framework for integrating structured metadata and unstructured resource content in a seamless manner which can then be flexibly queried using structured queries expressed in natural language. The novelty of the approach lies in the combination of different semantic technologies providing a clear benefit for the application scenario considered. The resulting system is implemented as part of the digital library of British Telecommunications (BT). The original contribution of our paper lies in the architecture we present allowing for the non-straightforward integration of the different components we consider.

1 Introduction

In the last decade, *Digital Libraries* [1] have emerged as a standard means for accessing published resources maintained electronically by libraries. Many digital libraries have evolved from traditional libraries and concentrated on making their information sources available to a wider audience. Today, many companies maintain their own digital libraries, and research and development for digital libraries now includes processing, dissemination, storage, search and analysis of all types of digital information. In contrast to physical libraries, digital libraries enable concurrent access at any time without physical boundaries. As such, digital libraries can be regarded as indispensable tools for today's knowledge workers. Digital libraries have always been an appealing playground for innovative computer science solutions. To name just a few examples, cross-referencing functionalities, document digitalization and optic character recognition (OCR), improved information retrieval techniques and recommender functionalities have changed the way we interact with digital libraries far beyond their basic functionalities. So far, functionalities of digital libraries are typically implemented in

Acknowledgements: This work was partially supported by the European Commission under contracts IST-2003-506826 SEKT, IST-2006-027595 NeOn, and IST-FP6-026978 X-Media.

a specific context and are tuned to meeting the requirements of a specific audience. Currently missing are *generic methods* for the smooth integration of content stemming from different sources and for a flexible framework for implementing new functionalities. Further, fine-grained access to resources (resource retrieval) and facts (fact retrieval) in the form of structured queries combining fine-granular metadata and fulltext content is typically not provided. In this paper we show that with the help of semantic technologies we can bridge this gap. In particular, ontologies offer a generic solution to the problem of integrating various sources. The documents in the knowledge sources are annotated and classified according to the ontology. Hereby, an ontology is essentially a logical theory conceptualizing relevant aspects of an underlying domain, in our case the domain of publications. Our ontology model consists of concepts organized hierarchically in terms of subsumption as well as of (binary) relations together with appropriate domain/range restrictions. The ontological metadata can then be exploited for advanced knowledge access, including navigation, browsing, and semantic search. Advanced semantics-based mining technology can extract fine-grained metadata from articles contained in the digital library. Finally, current reasoning techniques allow to answer structured queries to access full-text content as well as fine-grained metadata from articles from different sources in a uniform way.

In this paper, we present a principled framework for integrating digital library knowledge sources as well as facts extracted from the content under consideration by means of an ontology-based digital library system that can be used for several adaptions of the standard digital library scenario. As an example, we then present a system that allows the posing of structured natural language queries against the digital library with a well-defined semantics provided by the underlying ontology. The resulting system has been implemented as part of a case study with the Digital Library of British Telecommunications plc (BT) within the EU IST integrated project Semantic Knowledge Technologies (SEKT)). It combines a variety of tools for natural language question interpretation, information extraction, query answering and reasoning which are glued together by ontology-based knowledge representation formalisms and a corresponding ontology and metadata management system.

This paper is organized as follows: In section 2 we introduce a scenario which will serve as a running example throughout this paper. From this scenario, we derive a number of requirements for an ontology-based digital library system. In section 3, we present the architecture and the components of our system. In section 4 we refer back to the earlier requirements and describe how they are addressed and handled in our implementation. Section 5 discusses related work while Section 6 summarizes the main contribution of the paper.

2 The Scenario

In this section, we present a short scenario which will be used as a running example throughout this paper:

Scenario (BT Digital Library). *Bob works as technology analyst for British Telecom. His daily work includes research on new technological trends, market developments as well as the analysis of competitors. Bob's company maintains a digital library that* *gives access to a repository of internal surveys and analysis documents. The company also has a license with an academic research database which is accessed via a separate interface. Depending on his work context, Bob uses the topic hierarchies, the full-text search functionalities or metadata search facilities provided by the two libraries to get access to the relevant data. However, Bob is often annoyed by the differing topic hierarchies and metadata schemes used by the two libraries as well as by a cumbersome syntax for metadata queries. Questions which Bob might be interested in could include:*

1. *What articles were published by William Arms in "Communications of the ACM"?*
2. *Who wrote the book "Digital Libraries"?*
3. *What article deals with Grid Computing?*
4. *What are the topics of "The future of web services"?*
5. *Which conference papers were classified as "emerging technology trends 2007"?*
6. *Which articles are about "Intellectual Capital"?*

The user Bob in this scenario can be seen as a typical example of a knowledge worker who is using digital libraries for everyday research tasks. The scenario also points to a number of deficiencies of many of the current interfaces to digital libraries. From the initial scenario and the above questions, we derive the following requirements:

Support for Structured Queries Against Metadata and Documents: With current interfaces to Digital Libraries, users either pose keyword-based queries on document fulltexts or abstracts or they use metadata search facilities to perform document retrieval. However, question 1 requires as answers articles which fulfill the condition of having been published by William Arms in a certain journal, i.e. Communications of the ACM, and thus a structured query against the metadata of the articles. Question 2 requires the author of a certain publication, i.e. the book with title "Digital Libraries". This shows that in general we are interested in receiving as answers facts from the knowledge base rather than only relevant documents. We thus require the capability to pose structured queries to the underlying digital library which can be evaluated against the articles' metadata as contained in the knowledge base. For metadata queries, current interfaces either offer preselected attribute fields (which are interpreted as conjunctive queries over a attribute-specific fulltext search) or they require some kind of formal query language.

Integration of Heterogeneous Knowledge Sources: In general, answering questions as the above might however require uniform access to different sources or repositories. However, a common limitation is that different providers of digital libraries use different taxonomies and different metadata schemes to describe the stored content, which requires a user to switch between different representations depending on the backend system searched. A particular challenge is the integration of structured knowledge sources (e.g. document metadata) and unstructured sources (e.g. document fulltexts). We would like to access heterogenous knowledge sources via a single, unified interface that integrates different metadata schemes as well as different topic hierarchies.

Automatic content extraction and classification: Questions 3 and 4 might require fine-grained access to the topics of articles. Hereby, with topics we mean items of some classification scheme to which documents are related. Though in many cases articles are classified, we can not expect that all relevant categories are actually explicitly stated. Thus, some support for automatically capturing the content of new documents added to the library seems necessary. The content of the Digital Library is not static, but changes over time: New documents come in, but also documents may be removed from the document base. We here require means to automatically extract relevant knowledge and to classify the new content.

Natural Language Interface Finally, in order to provide intuitive access and not to impose the burden on the user of learning a formal query language, we would like to allow the user to use an intuitive query language, ideally arbitrary natural language queries. In our running scenario, Bob should thus be able to ask the questions directly in the way they are stated in the scenario above.

All of the above requirements can, of course, be tackled by ad-hoc modifications of the interfaces. The approach we take is, however, generic and can be flexibly extended. We have implemented an approach that enables the user to perform structured natural language queries against the information contained in the Digital Library. The semantics of the information and the user queries is defined by an underlying ontology. As a result, users are able to ask queries such as ”What article deals with Grid Computing?”, i.e. a query that allows for relating different knowledge sources and that does not only allow to return documents, but structured answers to the query.

3 Architecture and Components

In this section we discuss the overall architecture of the proposed digital library system. The architecture – as shown in Figure 1 – consists of the following main components:

- The *Knowledge Portal* is the user interface to the digital library. The user interacts with the portal by asking queries in natural language. The underlying transformation and query answering processes are completely transparent to the user. Figure
- The *Query Translation* component translates the natural language queries into structured logical queries against the ontology. This translation relies on a deep parsing of the questions using a lexicon that describes the possible lexical realizations of the elements in the ontology. The resulting logical queries are expressed in SPARQL [2], a query language standardized by the W3C for the Semantic Web.
- The *Query Answering* component manages the integrated ontologies and performs the answering of SPARQL queries against the knowledge sources. Extensions can be integrated that enable additional search functionalities at query time, e.g. for online-*text classification* of content. The query results are tuples of variable bindings that satisfy the query.
- The *Knowledge Base* of the digital library consists of a number of heterogeneous knowledge sources, partially structured in the form of metadata and topic hierarchies, but largely unstructured in the form of fulltext documents. All these data sources are integrated using an *ontology*. The *Ontology Learning* component is used to automatically extract structured ontologies from the unstructured text documents in the library. This allows the integration of the text documents with the other data sources such that they can be queried in a uniform way.

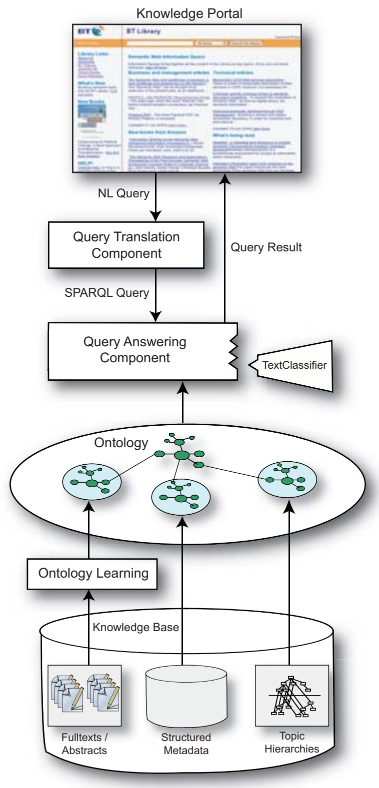

Fig. 1. Conceptual Architecture of the Application

In the following, we will discuss the individual components in more detail. While the architecture is generic in principle, we illustrate it using the components used for the implementation of the BT Digital Library.

3.1 Knowledge Base and Ontology

As shown in the bottom of Figure 1, the knowledge base of the digital library comprises a number of different knowledge sources. In the BT Digital Library, it consists of databases with bibliographic metadata, topic hierarchies, such as INSPEC[4], but also unstructured sources such as full text documents with different formats. All these heterogeneous knowledge sources are integrated using a common ontology, which is based on a designed general ontology, which we call PROTON (PROTo ONtology)[5]. Within PROTON, we adapt a layered approach: The classes in this ontology are a mixture of very general classes, e.g. `Person`, `Role`, `Topic`, `TimeInterval`, and classes

[4] `http://www.iee.org/publish/inspec/about/`
[5] `http://proton.semanticweb.org`

which are more specific to the world of business, e.g. `Company`, `PublicCompany`, `MediaCompany`. Finally, the ontology contains domain specific aspects, including classes relating to the specifics of the library, e.g. to the particular information sources available. Within PROTON there is a class `Topic` such that each individual topic is an instance of this class. However, frequently a topic will be a subtopic of another topic, e.g. in the sense that a document 'about' the former should also be regarded as being about the latter. The hierarchy of topics is modeled using a special relation subTopic. This relationship is defined to be transitive, in the sense that if A is a subtopic of B and B is a subtopic of C, then A is also a subtopic of C.

The structured information sources are integrated using a mapping of the underlying structures to the ontology. As a simple example, a database table describing persons would be mapped to the class `Person` in the ontology along with its properties, such as name affiliation, etc. The mapping formalism [3] also supports more complex mappings that establish correspondences between conjunctive queries over the sources and the ontology. For the unstructured sources – such as fulltext documents – the mapping is not as direct. Instead, we obtain structured ontologies from the unstructured sources with the help of the ontology learning tool Text2Onto [4], as explained in the following.

3.2 Ontology Learning

Ontology Learning aims at learning and extending ontologies on the basis of textual data. In our system, ontology learning is used to dynamically extract new topics, concepts and relations in the underlying document collection. For this purpose, we exploit Text2Onto, a framework for ontology learning and data-driven ontology evolution [4]. It relies on a combination of natural language processing and machine learning techniques for extracting ontologies from unstructured textual resources. In particular, it implements algorithms for learning the following ontological primitives: **concepts** and **instances** as well as **subconcept**, **subtopic**, **instance-of** and arbitrary **binary relations** between concepts. The algorithms implemented in Text2Onto build on a variety of techniques from information retrieval (e.g. statistical measures such as tf.idf for term extraction), natural language processing (e.g. lexico-syntactic patterns for extracting subconcept and instance-of relations) as well as machine learning (e.g. clustering for learning concepts hierarchies, association rules for extracting binary relations), etc.

To illustrate the ontology learning process consider the following excerpt from a digital library document about collaborative development environments: *To support remote authoring of Web pages and file contents, as well as remote source code access, GForge uses several network protocols, including SSH, SFTP, CVS pserver, and FTP.* Given a lexico-syntactic pattern matching a sequence of noun phrases such as $\mathrm{NP}_{concept}$, *including* $\mathrm{NP}^1_{\mathrm{instance}} \ldots \mathrm{NP}^n_{\mathrm{instance}}$ Text2Onto would conclude from this sentence that SSH, SFTP, CVS pserver and FTP are instances of network protocol. If the user later asks for documents about network protocols, the learned concept instantiation will enable a reasoner to infer that a document dealing with SSH might be relevant for the user – even if the term *network protocol* is not explicitly mentioned in the text.

We apply the algorithms to extract the above mentioned primitives from each of the relevant information spaces in the digital library, and merge the resulting ontologies.

Most importantly, Text2Onto keeps a document pointer to the document where a certain ontological primitive was extracted from. This allows fine-grained queries to be posed to the document collections by asking for the topics, concepts, instances, as well as different types of taxonomic and non-taxonomic relations occurring in a document. By the integration of Text2Onto we could for example answer questions like: *What network protocols are talked about in the article "The Future of Web Services"?* or *What articles are about "Intellectual Capital"?*.

3.3 Query Answering

The integrated ontology is managed by the KAON2 ontology management system [6], which acts as the query answering component. As mentioned before, the queries are logical conjunctive queries against the ontology. Query answering amounts to a reasoning process over the knowledge sources according to the semantics of the underlying ontology language OWL. To represent the conjunctive queries, we here rely on SPARQL as the query language, an ontology query language standardized by the W3C [2]. The query answering is not a mere retrieval of explicitly stated facts (as in a conventional database), but involves a deduction of answers over the knowledge base [5]. As an example, consider the following SPARQL query which asks for articles that are associated with the topic “Intellectual Capital”?:

```
SELECT  ?x WHERE {
   ?x rdf:type <http://proton.semanticweb.org/2005/04/protonu#Article> .
   ?x <http://proton.semanticweb.org/2005/04/protont#hasSubject> ?y  .
   ?y rdfs:label ?z .
   match(?z,"Intellectual Capital")
```

For the answering of queries, we follow a virtual integration approach that does not require a full integration of all knowledge sources: The actual data still resides in the individual knowledge sources, and the mapping between the ontology and the knowledge sources is only used at runtime to retrieve the facts that are actually relevant for answering a query. In a query, predicates can be used that require access to different knowledge sources. The evaluation of these predicates is automatically *pushed down* to the respective knowledge sources, e.g. as a relational query in the case of a relational database, or to a fulltext index in the case of the fulltext match predicate `match` over the topic hierarchy contained in the above query. In Section 3.5, we discuss the evaluation of a special predicate for the classification of text documents. At last the result set is processed and sent back to be displayed by the BT knowledge portal.

3.4 Query Translation from Natural Language

ORAKEL [6] is a natural language interface which translates natural language queries to structured queries formulated with respect to a given ontology. This translation relies essentially on a compositional semantic interpretation of the question guided by two lexica: a domain-specific and a domain-independent lexicon. As the name suggests, the domain-independent lexicon is a priori encoded into the system and captures

[6] `http://kaon2.semanticweb.org`

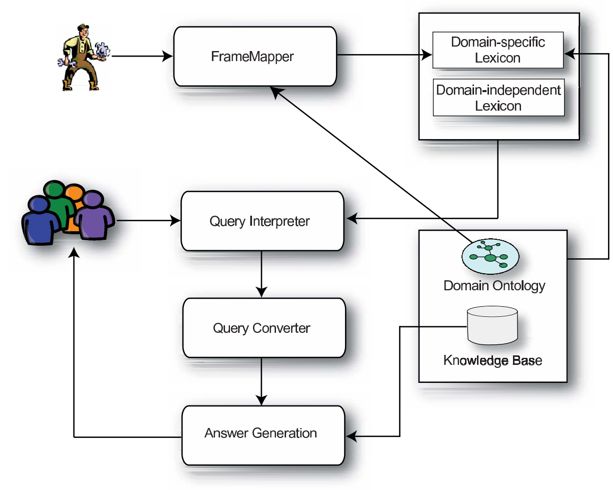

Fig. 2. Overview of the ORAKEL system

the domain-independent meaning of prepositions, determiners, question pronouns etc., which typically remain constant across domains. The domain-specific lexicon needs to be created by a *lexicon engineer* who is assumed to create the lexicon in various cycles. In fact, ORAKEL builds on an iterative lexicon development cycle in which the lexicon is constantly updated by the lexicon engineer on the basis of the questions the system failed to answer so far. The end users then are able to directly interact with the BT digital library portal by accessing the library data via natural language questions, which are translated into SPARQL queries by the ORAKEL system. The underlying mechanism however is hidden from the users - the only thing users need to do is to input the query just as their normal questions and then get the result from the portal. The obvious advantage of using such a natural language interface is that users do not have to learn nor struggle with a formal query language, while they can still benefit from the possibility to pose structured queries.

Further, ORAKEL was also modified to process quoted text by matching it against a text index of the metadata in the database using a special purpose predicate *match*. The question *"What articles are about "Intellectual Capital"?"* is translated into the SPARQL query presented above.

Figure 2 gives an overview of the ORAKEL system, which has been designed in a flexible manner allowing the system's target query language to be easily replaced. As the architecture described here relies on the KAON2 system as inference engine and knowledge repository, ORAKEL was adapted to generate SPARQL queries. Further, a lexicon was generated for the Proton ontology, which specifies the possible lexical representations of the ontology elements in the user queries.

3.5 Text Classification

Often query answering requires the evaluation of predicates whose extensions need to be determined using special purpose algorithms at query time. This is for example the case if the text documents need to be classified against parameters that are provided at query time by the user. Automated text classification [7] is a standard tool for adaptive categorization of textual data. In essence, given a collection of positive and negative examples, machine learning techniques are used to discover patterns in the text that will help in the categorization of unseen documents in the future. For topics for which no explicit library classification exists, e.g. personalized interests or emerging topics, one can train and use such a classification module. Of course, the classification accuracy for text mining tasks will depend on many factors such as representation, training algorithm, number of training documents and parameter setting. We have used a simple approach that has been shown to perform well in practice, namely the bag-of-words representation together with Support Vector Machine (SVM) classification. While training is performed offline, the resulting models can be simply integrated into the system by means of the Built-In mechanism. As a result, the special-purpose predicate "classified-as" is contained in the ontology but its extension is evaluated at query time against the document fulltext and the stored model(s)[7].

4 Scenario Revisited

The architecture described in the previous section has been implemented and deployed at BT. In contrast to the previously existing system, the resulting question answering prototype does not only allow for asking questions about existing metadata in the BT library, but also about topics found by Text2Onto as well as to invoke the automatic text classification system at runtime. The net result is that a variety of queries about authors, topics, about documents etc. could be answered successfully against the BT ontology and database. The questions introduced in section 2 are examples of supported natural language queries.

We will revisit two of these questions. Consider the question *What articles were published by "William Arms" in "Communications of the ACM"?*. The corresponding SPARQL query would look as follows:

```
SELECT ?x WHERE {
 ?x rdf:type <http://proton.semanticweb.org/2005/04/protonu#Article> .
 ?x <http://proton.semanticweb.org/2005/04/protont#documentAuthor> ?y .
 ?x <http://proton.semanticweb.org/2005/04/protonu#publishedWithin> ?z .
 ?y rdfs:label ?ys .
 match(?ys,‘‘William Arms’’)
 ?z rdfs:label ?zs .
 match(?ys,‘‘Communications of the ACM’’) }
```

As another example, consider the question *"Which conference papers are classified as "emerging technology trends 2007?"*. By means of ORAKEL and a small set of rules, the question would result in the following SPARQL query, which would use the SVM classification on document abstracts of documents of type article and the (previously trained) SVM model "emerging technology trends 2007":

[7] The underlying module for text classification and corresponding training is part of the TextGarden library. See http://www.textmining.net/ for further information.

Iteration	Recall (avg.)	Precision (avg.)
1	42%	52%
2	49%	71%
3	61%	73%

Table 1. Results for the different iterations

```
SELECT ?x WHERE {
 ?x rdf:type <http://proton.semanticweb.org/2005/04/protonu#Article> .
 ?x <http://proton.semanticweb.org/2005/04/protont#documentAbstract> ?y .
 EVALUATE ?margin:=tgclassify(?y,'emerging technology trends 2007') .
 FILTER ?margin>0 }
```

Furthermore, an evaluation of the system was carried out with BT and other users. A primary goal was to evaluate the performance of the natural language question answering over several iterations of the ontology lexicon. The lexicon was constructed in three iterations: one initial iteration of 6 hours and 2 follow-up iterations of each 30 minutes in which the questions not answered by the system were examined. The end users received written instructions describing the conceptual range of the knowledge base, asking them to ask at least 10 questions to the system. In each of three iterations, 4 end users asked questions to the system and a graduate student updated the lexicon on the basis of the failed questions after the first and second round for about 30 min., respectively. The end users were also asked to indicate whether the answer was correct or not, which allowed for the evaluation of the system's performance in terms of precision and recall. The results of the evaluation are presented in Table 1, which clearly shows that the second iteration performed much better than the first one, both in terms of precision and recall. In the third iteration, there was a further gain in recall with respect to the second iteration. Concluding, we can on the one hand indeed say that the precision of the system is fairly high. On the other hand, the lower recall is definitely compensated by a fallback strategy. In case the semantic understanding and answering of the query fails, our system resorts to standard information retrieval techniques to provide a number of relevant documents to the user's query. This move thus makes the implemented system very robust. Overall, the application of our prototype showed that we could indeed improve the access to BT's digital library by (i) integrating of data and sources using an inference engine (ii) precise question answering, (iii) exploitation of new topics detected by Text2Onto, as well as (iv) on-the-fly text classification functionality.

5 Related Work

While traditional digital library systems such as e.g. DSpace [8] or e Prints[9] have mostly focused on providing a generic infrastructure for storing digital content and metadata, the relation to Semantic Web technologies has been mostly restricted to metadata import/export functionalities. The actual combination of semantic technologies and Digital Libraries has only received increased attention in recent years. In the following, we discuss some related projects that apply Semantic Web technologies for digital libraries.

[8] http://dspace.org/

[9] http://www.eprints.org/

An initiative that is in many aspects similar to our work, is the JeromeDL project[10] [8], which employs Semantic Web technologies mainly for user management and personalized search within a digital library. In JeromeDL, full text content, bibliographic entries etc. are described with respect to the Jerome ontology. JeromeDL is distinguished by the extensive conceptualization and advanced search and personalization algorithms. However, in contrast to our approach, JeromeDL lacks any kind of knowledge extraction or natural language query functionalities. The SIMILE Project [11] ([9]), aims at enhancing interoperability among digital assets, vocabularies, metadata and services. SIMILE tackles the challenge that collections can be distributed but should be queried in a uniform way. Semantic Web technologies, in particular the Resource Description Framework (RDF) [10] are used to tackle the challenge that collections can be highly distributed but need to be queried in a uniform way. While SMILE shares the interoperability-related aspects with our work, it does not go beyond this. Recently, some initiatives have emerged with the aim of moving from a content-centered organization of digital libraries towards more service-oriented architectures. The main idea of projects such as FEDORA project[12] ([11]) is to support the whole digital content value chain from data creation, sharing, search and dynamic provision of appropriate services.

6 Conclusions

We have presented an approach that combines a number of semantic technologies, including ontology management, ontology learning and reasoning, keyword-type search and text classification in order to allow the flexible and versatile answering of natural language questions on top of a digital library. The users are able to perform structured natural language queries against a variety of knowledge sources in an integrated manner with a well-defined semantics provided by the underlying ontology. The novelty of our system lies in the combination of different tools for natural language question interpretation, ontology learning, query answering as well as reasoning. Our experience showed that with reasonable effort it is possible to apply semantic technologies to enhance a semantic library so that: (i) natural language questions can be answered precisely relying on standard (logical) querying techniques, (ii) topics can be automatically spotted over time and integrated into the system, and (iii) text documents can be classified on-the-fly given a user query. Though the integration of semantic technologies which have been developed recently at our institute (ORAKEL, Text2Onto, Text classification modules, KAON2) was far from straightforward mainly due to technical and infrastructure problems, the case study proved that semantic technologies are indeed mature enough to be integrated with reasonable effort and deliver a clear added value. In this sense we can only encourage other people to venture out and experiment with semantic technologies in order to foster the exchange of experiences between the Semantic Web and Digital Library communities.

[10] `http://www.jeromedl.org/`
[11] `http://simile.mit.edu/`
[12] `http://www.fedora.info/`

References

1. Arms, W.Y.: Digital Libraries. The MIT Press (2001)
2. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF. W3C Working Draft (2007) available at `http://www.w3.org/TR/rdf-sparql-query/`.
3. Haase, P., Motik, B.: A mapping system for the integration of OWL-DL ontologies. In Hahn, A., Abels, S., Haak, L., eds.: Proceedings of the first international ACM workshop on Interoperability of Heterogeneous Information Systems (IHIS'05), CIKM Conference, Bremen, Germany, November 4, 2005, ACM (2005) 9–16
4. Cimiano, P., Völker, J.: Text2Onto - a framework for ontology learning and data-driven change discovery. In Montoyo, A., Munoz, R., Metais, E., eds.: Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB). Volume 3513 of Lecture Notes in Computer Science., Alicante, Spain, Springer (2005) 227–238
5. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. In McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: International Semantic Web Conference. Volume 3298 of Lecture Notes in Computer Science., Springer (2004) 549–563
6. Cimiano, P., Haase, P., Heizmann, J.: Porting natural language interfaces between domains – a case study with the ORAKEL system –. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI). (2007) 180–189
7. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys **34** (2002) 1–47
8. Kruk, S.R., Decker, S., Zieborak, L.: Jeromedl - adding semantic web technologies to digital libraries. In Andersen, K.V., Debenham, J.K., Wagner, R., eds.: Proceedings of the 16th International Conference on Database and Expert Systems Applications, DEXA 2005, Copenhagen, Denmark, August 22-26, 2005. Volume 3588 of Lecture Notes in Computer Science., Springer (2005) 716–725
9. Mazzocchi, S., Garland, S., Lee, R.: Simile: Practical metadata for the semantic web. XML.com (2006)
10. Manola, F., Miller, E.: Resource description framework (RDF) model and syntax specification (2004) `http://www.w3.org/TR/rdf-primer/`.
11. Payette, S., Staples, T.: The mellon fedora project: Digital library architecture meets xml and web services. In: Proceedings of the Sixth European Conference on Research and Advanced Technology for Digital Libraries. Volume 2459 of LNCS., Springer (2002) 406–421