

ONTOSEARCH2: SEARCHING AND QUERYING WEB ONTOLOGIES

Jeff Z. Pan, Edward Thomas and Derek Sleeman
University of Aberdeen
Aberdeen, UK
{jpan, ethomas, dsleeman}@csd.abdn.ac.uk

ABSTRACT

Ontologies are important components of web-based applications. While the Web makes an increasing number of ontologies widely available for applications, how to discover ontologies in the Web becomes a more challenging issue. Existing approaches are mainly based on keywords and metadata information of ontologies, rather than semantic entailments of ontologies. In this paper, we present a Semantic Web engine, called ONTOSEARCH2, which searches and queries Web ontologies by creating and storing a copy of ontologies in a tractable description logic. ONTOSEARCH2 allows formal querying of its repository, including both the structures and instances of ontologies, using the SPARQL query language. Furthermore, this paper reports on preliminary, but encouraging, benchmark results which compare ONTOSEARCH2's response times on a number of queries with those of existing knowledge base management systems.

KEYWORDS

Semantic Web, Ontology, Description Logics, Search, Query

1. INTRODUCTION

Ontologies play a key role in the Semantic Web [BHL01], where the W3C standard ontology language OWL [SWM03] has become the defacto standard for ontologies and semantic data. A growing library of these ontologies is available online, covering the definitions of a very wide range of subjects. While the Web makes an increasing number of ontologies widely available for applications, how to discover relevant ontological material on the Web becomes a more challenging issue.

Currently it is very difficult to find ontologies suitable for a particular purpose. Semantic web engines like Swoogle [D04] and OntoKhoj [P03] allow ontologies to be searched using keywords, but further refinement of the search criteria based on semantic entailments is not possible. These unstructured searches are not enough to perform searches on the highly structured data of the Semantic Web. In this paper, we present a Semantic Web engine, called ONTOSEARCH2, to search and query Web ontologies and their associated data-sets. The core of ONTOSEARCH2 is an inference engine for the DL-Lite ontology language [C04], which is a sub-language of the OWL DL (a language in the OWL family, see Section 2.1). DL-Lite is chosen because it provides a good balance between expressive power (e.g. it is able to express most features in UML class diagrams) and efficiency (query answering is polynomial).

The applications of ontology searching and querying have been identified in several areas.

- **Information Retrieval:** Information retrieval has been an important problem for the Web and Internet-based information systems – ontologies in the Semantic Web enable information to be retrieved by making use of annotations of Web resources based on ontologies. For example, when using the FOAF (Friend of a Friend¹ ontology, currently the most popular use of RDF [D05]), it is very easy to answer a query such as “list all the people that have been listed as a friend by person X”

¹ <http://www.foaf-project.org>

by looking at the properties of the RDF resource at X. However, answering a similar query “list people who list the person X as their friend requires searching and querying multiple ontologies based on FOAF. The above example also indicates that ontology searching and querying services should enable richer navigations. For example, once we retrieve the URIs of these friends, it is possible to further navigate more information about them.

- **Information Integration:** Information integration has been another important problem for the Web and Internet-based information systems – it is hard to combine new information with any other piece of related information we already possess, and to make them both available for application queries. With ontologies being shared understanding of certain application domains; ontology-based integration is a promising direction. In this task, while the role of ontology querying is obvious, ontology searching can also help identify potential ontologies that we use for integration. For example, by indexing the WordNet² lexical database, which is available in RDF form, searches can be combined with the WordNet “SimilarTo” information to allow matching to extend to synonyms of the original keyword in the query. These queries would then include any match which used a synonym of a query term, without the need to add any additional reasoning capabilities to the search engine.
- **Ontology Management:** As progressively more applications use ontologies to represent semantic information, how to support ontology reuse and, in general, ontology management are becoming more important. Searching and querying ontologies are among the first steps of ontology reuse – users want to specify some queries, e.g., to specify a particular class hierarchy or the associations of some datatype properties for a certain class. Searching and querying are also very useful in other ontology management tasks. For example, ontology querying can be used to test if ontologies match the intended meaning of users, which would be very useful in ontology construction. When an ontology becomes very large and contains many modules, ontology search can help locate the modules that satisfy a user’s criteria.

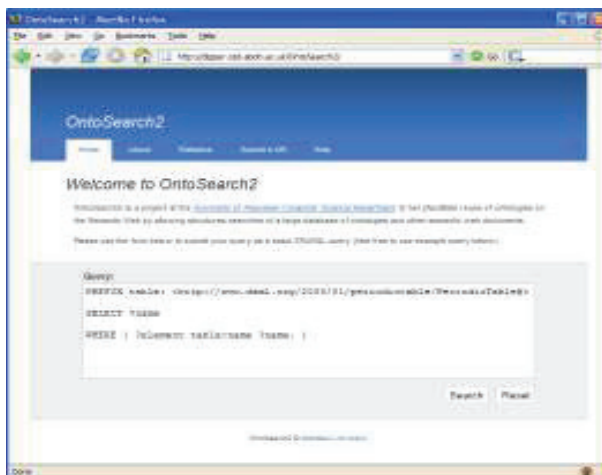


Figure 1: ONTOSEARCH2 query form

ONTOSEARCH2 is implemented as a set of Java Servlets and JSP pages, which allow ontologies (such as those found by some Web crawlers) to be added to the repository and queries to be made using the Semantic Web standard RDF query language SPARQL [PS05]. A screenshot of the query form of ONTOSEARCH2 is shown in Figure 1. The rest of the paper is structured as follows. After briefly discussing

² <http://wordnet.princeton.edu/>

ontologies and DL-Lite (Section 2), we present the architecture of ONTOSEARCH2 (Section 3), and evaluate it using the well known Lehigh University Benchmark (Section 4). We will also discuss the related work (Section 5), before concluding the paper (Section 6).

2. PRELIMINARY

2.1 Ontology

An ontology [UG96] typically consists of a set of classes, properties, and constraints about these classes and properties. An ontology language provides constructors to construct class and property descriptions based on named classes and properties, as well as some forms of axioms about classes, properties and individuals. For example, RDFS [15] provides some axioms (such as domain and range axioms), but no class or property constructors. OWL DL [16] provides class constructors (e.g. conjunction $C \sqcap D$ and number restriction $\leq nR$), property constructors (e.g. inverse properties R^{-}) and more kinds of axioms (such as individual equality axioms $a \approx b$) than RDFS. Usually, we call the set of class and property axioms TBox, while the set of individual axioms ABox. OWL DL is an expressive Description Logic [B03], in which the complexity of logical entailment is NEXPTIME.

2.2 DL-Lite

Looking for a balance between expressive power and complexity has been one of the main themes in Description Logics. Recently, Calvanese et al proposed DL-Lite [C04] which can express most features in UML class diagrams but still has a low reasoning overhead [C05] (worst case polynomial time, compared to worst case exponential time in the case of most widely used Description Logics). DL-Lite supports the following axioms: (1) class inclusion axioms: $B \sqsubseteq C$, where B is a basic class $B := A \mid \exists R \mid \exists R^{-}$ and C is a general class $C := B \mid \neg B \mid C_1 \sqcup C_2$ (where A denotes an atomic class and R denotes an atomic property); (2) functional property axiom: $\text{Func}(R)$, $\text{Func}(R^{-})$; individual axioms: $B(a)$, $R(a,b)$, where a and b are individual. Querying over DL-Lite ontologies can be carried out by an SQL engine (query rewriting is needed), thus taking advantage of the query optimisation strategies provided by current DBMSs [C05].

3. ONTOSEARCH2

ONTOSEARCH2 is the framework which we have developed to support ontology searching and querying. It provides functionality to query the ontology repository using a restricted subset of SPARQL or to add additional ontologies to the index by providing the URI of the ontology or RDF data.

The core of ONTOSEARCH2 is an inference engine for the DL-Lite ontology language. It implements the algorithms of TBox normalisation, ABox storage, ontology consistency and query reformulation (PerfectRef) presented in [C05]. For ABox query answering, a query in SPARQL is parsed and converted to the DL-Lite conjunctive query format [ibid], which is reformulated based on the normalised axioms (of the forms: $B_1 \sqcap B_2$, $B_1 \sqcup \neg B_2$, $\text{Func}(R)$, $\text{Func}(R^{-})$) into a set of SQL queries, which can be carried out by an SQL engine. This is executed on a JDBC database used as the repository and the results are returned to the user formatted as HTML or as XML depending on the parameters used while searching. An example of ABox query is shown in Figure 1.

In order to enable ontology search, we also need to support TBox queries. In the ONTOSEARCH2 framework, we maintain a meta-ontology³ for DL Lite, using the Fixed layered metamodeling Architecture [PH02,PH03], so that TBox queries can be handled in a similar way of ABox queries. For example, we can

³ <http://www.csd.abdn.ac.uk/~ethomas/dl-lite.rdf>

query classes (instances of owl:Class) which are in the domain of a certain property (instance of owl:ObjectProperty) which has a particular class A as its range. This is shown in SPARQL in listing 1. Note that the following query is against the whole repository, rather than a single ontology.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>

SELECT ?x WHERE {
  ?x rdf:type owl:Class .
  A rdf:type owl:Class .
  ?r rdf:type owl:ObjectProperty .
  ?r rdf:domain ?x .
  ?r rdf:range A .
}
```

Listing 1: A TBox query in SPARQL

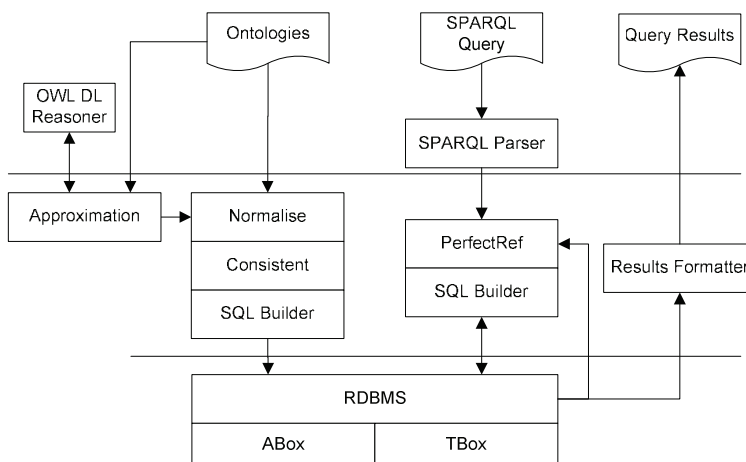


Figure 2: ONTOSEARCH2 system architecture

Although DL-Lite is rich enough to express most features of UML class diagrams, it is only a sub-language of OWL DL. According to a recent survey of Web ontologies [WPH06], there exist many ontologies which exceed the expressiveness of DL-Lite. In order to support these ontologies in ONTOSEARCH2, we approximate an OWL DL ontology O with its DL-Lite counterpart O' . That is, given the set of names of classes, properties and individuals used in a target ontology O , we use an OWL DL reasoner to calculate all valid normalised and entailed axioms of the following forms: $B_1 \vee B_2$, $B_1 \sqsupset B_2$, $\text{Func}(R)$, $\text{Func}(R)$, $B(a)$, $R(a,b)$. Note that O' contain all the explicit and implicit axioms which are entailed by O and are expressible in DL-Lite. In the current implementation of ONTOSEARCH2, we use PELLET⁴ as the OWL DL reasoner, since it is also implemented in Java.

Figure 2 shows the system architecture used for ONTOSEARCH2. The elements shown in the centre of the diagram are the rulesets which mediate between the standards used on the Semantic Web for ontologies and querying (OWL and SPARQL respectively) and the DL-Lite derived methods used by ONTOSEARCH2 to store and query the ontologies. These convert ontologies from a Jena OWL model into the internal DL-Lite representation. SPARQL queries are converted into DL-Lite conjunctive queries which can be expanded and converted to SQL queries to run on the database.

⁴ <http://www.mindswap.org/2003/pellet/>

As we show in the next section, ONTOSEARCH2 is scalable, which makes it possible for Semantic Web applications to query millions of instances to find information. It also gives knowledge engineers access to a huge variety of pre-built ontologies and classes which they can reuse in other applications.

4. EVALUATION

We have evaluated the ONTOSEARCH2 system using the Lehigh University Benchmark (LUBM) [GPH05] to measure its performance on large data sets. We have run benchmarks using generated data sets representing 1, 5, 10, 20 and 50 universities, these are generated using the same seed and index values as used in [GPH04] so we can directly compare the performance of the ONTOSEARCH2 system against the systems tested in [ibid]. The smallest dataset (0,1) contains approximately 136,000 triples in 14 RDF files, and the largest dataset (0,50) contains approximately 6,800,000 triples in 999 RDF files.

The test machine specifications are listed below in Table 1. The Java platform used was JDK 1.5.0, we used PELLET 1.3 as the OWL DL reasoner and PostgreSQL 8.1 as the RDBMS. These were setup with default installations, no additional configuration was performed.

Model	Dell Precision 370
CPU	Intel Pentium 4 3.0GHz
RAM	1024Mb
Hard Disc	80Gb (ATA)

Table 1: Test machine specification

The data set was loaded separately for each set of queries, and the database was cleared between each data set. After loading the data, the command “VACUUM FULL ANALYSE” was executed on the database to remove redundant data and update the system catalogue with accurate statistics about each table to allow the queries to be executed in as efficient a manner as possible. The timings recorded were the total time taken for the query to be parsed from SPARQL, expanded, converted to SQL, for this query to be sent to the database, and the results retrieved. The time taken for the results to be sent back to the web browser of the test machine is not included.

The results obtained are shown in table 2. The columns represent the time taken to execute the query for each data set (T), the precision of the results (P, where 1.00 is every result returned being a valid result for the query) and the recall of the results (R, where 1.00 is every valid result for the query is returned). The precision and recall figures are calculated for the results of the query on the first dataset only, as reference result sets for the other datasets are not currently available.

Query	T[0,1] (ms)	T[0,5] (ms)	T[0,10] (ms)	T[0,20] (ms)	T[0,50] (ms)	P[0,1]	R[0,1]
Q1	156	185	435	597	921	1.00	1.00
Q2	220	301	503	2013	9250	1.00	1.00
Q3	172	224	375	609	1468	1.00	1.00
Q4	83	102	231	309	399	1.00	1.00
Q5	96	142	373	504	1313	1.00	1.00
Q6	204	263	1364	2120	9875	1.00	1.00
Q7	108	165	353	879	1781	1.00	1.00
Q8	166	264	738	1432	2201	1.00	1.00
Q9	820	1224	2193	8720	201351	1.00	1.00
Q10	741	1405	2932	6339	15745	1.00	1.00
Q11	232	294	823	1568	2567	0.00	0.00
Q12	189	216	302	499	781	1.00	1.00
Q13	67	65	72	76	78	1.00	1.00
Q14	236	302	398	460	822	1.00	1.00

Table 2: Results of the Lehigh University Benchmark queries against different data sets

We see from these results that with one exception, the results obtained are returned to the user within 20 seconds. The more promising results are given for dataset (0,50) where, excluding Q9, the lowest time is 0.078 seconds, and the highest is 15.745 seconds (average 3.630 seconds). Query 9 requires three of the largest datasets to be joined, which caused a considerable amount of disk access on the database. We will try to mitigate this in future revisions of the ONTOSEARCH2 software by tuning the database and making more RAM available for index caching.

The precision and recall figures show perfect precision and recall for all queries except for Q11. This is described in the benchmark as:

In this query, property subOrganizationOf is defined as transitive. Since in the benchmark data, instances of ResearchGroup are stated as a sub-organization of a Department individual and the later suborganization of a University individual, inference about the subOrganizationOf relationship between instances of ResearchGroup and University is required to answer this query. Additionally, its input is small.

<http://swat.cse.lehigh.edu/projects/lubm/query.htm>

ONTOSEARCH2 fails Q11 as it does not deal with transitive axioms in its current approximation (transitive axioms are clearly beyond DL-Lite), which will be solved in our next version of ONTOSEARCH2.

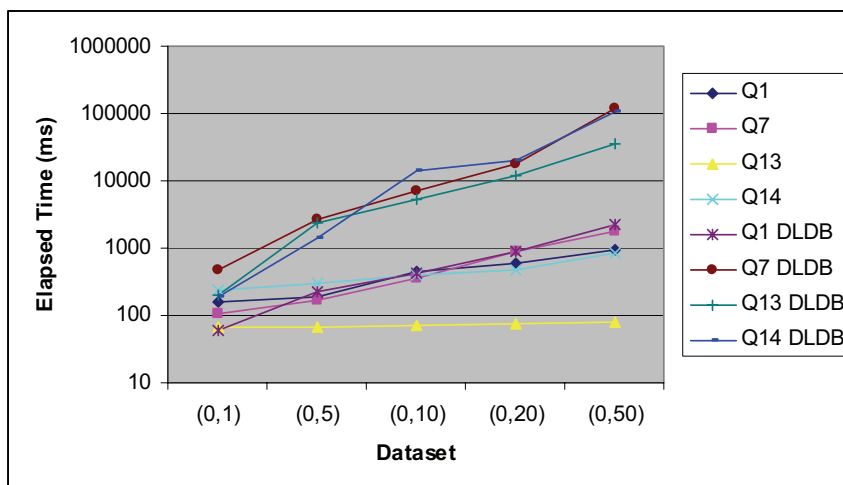


Figure 3: Performance of ONTOSEARCH2 vs. DLDB as data size increases

Figure 3 shows how the performance of ONTOSEARCH2 changes with 4 different queries from the LUBM query set. As the size of the data set grows, the time required to complete each query grows, but this is a linear growth when compared to the increase in size of the data set. This is partially due to the query overhead to parse the query and expand the query using the PerfectRef algorithm which remains constant for queries no matter the size of the DL-Lite ABox being queried, and partially due to the way that PostgreSQL uses indexes which are increasingly efficient for larger data sets. Also included in this chart are the results of the DLDB-OWL system, taken from [GPH04]. This is the only knowledge management system from the comparison performed by Guo et al which was able to complete the queries with the largest data set. While the two benchmarks were performed at different times and on different machines, we can draw certain conclusions from the results obtained. Although in some cases, DLDB was able to return results for smaller datasets quicker than the ONTOSEARCH2 system, the time taken to return results for the larger datasets grows far quicker with DLDB than with ONTOSEARCH2.

5. RELATED WORK

5.1 Ontology Searching

ONTOSEARCH [ZVS04] is the predecessor to the ONTOSEARCH2 system. It was developed to allow simple keyword based querying of ontologies by passing the keywords to Google; further these were packaged in such a way as to only return ontological data in RDF. This was supported by functions to show the context of matching terms in the ontology, and a simple visualization of the structure of the ontologies found.

Swoogle [D04] is a Semantic Web Search Engine developed at the University of Maryland. Swoogle crawls and indexes all types of Semantic Web Documents (SWDs), these documents are indexed and stored in a triple store database. Swoogle allows this database to be queried using a simple keyword based query interface; all ontologies which match the given keywords are returned to the user, with additional contextual descriptions given using information from linked SWDs. The interface to Swoogle limits its flexibility as a query tool. Only simple keyword based searches are possible, and additional work must be performed if one requires ontologies which contain certain structures or specific data.

OntoKhoj [P03] is a system developed by the University of Missouri. It crawls the Web searching for ontologies which it aggregates and classifies. Searching its index is performed using a keyword based search interface, which is similar to Swoogle. OntoKhoj differs from ONTOSEARCH and Swoogle in that it only allows searching of ontologies, not of other Semantic Web documents which reference ontologies.

5.2 Ontology Metamodelling

RDFS(FA) [PH02][PH03] and OWL(FA) [PHS05] are fixed layered meta-modelling architectures for RDFS and OWL. These try to remove the confusion caused by the dual roles given to certain structures in both OWL and RDFS by creating new modelling primitives to describe classes and properties at the object, language, and ontology layers. In this model, a class at the ontology layer is an instance of a class at the language layer, which in turn is an instance of a class at the metalanguage layer. We have adopted a similar model to this to provide the metamodelling capabilities used in ONTOSEARCH2. This allows us to describe TBox information in terms that can be queried using DL-Lite ABox query algorithms. The TBox used for these queries is a meta-TBox which describes the top level structure of DL-Lite.

6. CONCLUSION AND OUTLOOK

We have shown that ONTOSEARCH2 is able to reliably query large data sets faster than comparable database driven knowledge management systems. The recall and precision figures from the tests performed are encouraging but there are situations in which incomplete results can be returned, further work on the approximation component of ONTOSEARCH2 will try to fix this. Additional experimental work is also required to maximize the performance of the database subsystem.

Future development work will position ONTOSEARCH2 as a general purpose framework for building Semantic Web applications. An online service will provide general querying of Semantic Web data, the repository being expanded by using a search engine style spider to find new semantic web documents and ontologies. The current web interface will be supplemented with a Web Service interface to allow agents and other Semantic Web applications to directly query the ONTOSEARCH2 repository. A stand alone version of the software will be made available for local ontology repositories so that applications which require high performance querying of very large ontologies (such as the Gene Ontology⁵) will be able to make use of the ONTOSEARCH2 technology. This stand alone application will have support for more complex queries than the web service.

⁵ <http://www.geneontology.org>

We are investigating related projects which could provide ranking for the results of queries based on an analysis of the structures of an ontology [AB05], and are investigating ways of linking ONTOSEARCH2 to a natural language query engine which will map natural language queries such as “Which papers are written by students at the University of Aberdeen”, to improve its usability for a wider body of users.

ACKNOWLEDGEMENT

This research was partially supported by (1) the Advanced Knowledge Technologies (AKT⁶) project which is sponsored by the UK Engineering and Physical Sciences Council under grant number GR/N15764/01 and (2) the FP6 Network of Excellence EU project Knowledge Web (IST-2004-507842).

REFERENCES

- [AB05] Alani H, Brewster C, 2005. Ontology ranking based on the analysis of concept structures. *In proceedings of Proceedings of the 3rd International Conference on Knowledge Capture*, Banff, Alberta, Canada, pp. 51-58.
- [B03] Baader F et al, 2003, The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press.
- [BHL01] Berners-Lee T, Hendler J, and Lassila O, 2001. The Semantic Web. *Scientific American*, 284(5), pp34-43.
- [C04] Calvanese D et al, 2004. DL-Lite: Practical Reasoning for Rich DLs. *In Proceedings of the 2004 Description Logic Workshop*, Whistler, British Columbia, Canada.
- [C05] Calvanese D et al, 2005. DL-Lite: Tractable Description Logics for Ontologies. *In proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, USA, pp. 602-607.
- [D04] Ding L et al, 2004, "Swoogle: A Search and Metadata Engine for the Semantic Web", *In the Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*.
- [D05] Ding L et al, 2005, How the Semantic Web is Being Used: An Analysis of FOAF Documents. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, Track 4.
- [GPH04] Guo Y, Pan Z, Heflin J, 2004. An Evaluation of Knowledge Base Systems for Large OWL Datasets. *Third International Semantic Web Conference*, Hiroshima, Japan, LNCS 3298, 2004, pp. 274-288.
- [GPH05] Guo Y, Pan Z, and Heflin J, 2005. LUBM: A Benchmark for OWL Knowledge Base Systems. *Journal of Web Semantics*, 3(2), pp. 158-182.
- [P03] Patel C et al, 2003. Ontokhoj: A semantic web portal for ontology searching, ranking, and classification. *In Proc. 5th ACM Int. Workshop on Web Information and Data Management*, New Orleans, Louisiana, USA, pp. 58–61.
- [PH02] Pan J, and Horrocks I. Metamodeling, 2002. Architecture of web ontology languages. *In the emerging semantic web, Frontiers in artificial intelligence and applications*. IOS press, Amsterdam (NL), 2002.
- [PH03] Pan J, and Horrocks I. RDFS(FA) and RDF, 2003. MT: Two semantics for RDFS. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *Proc. of the 2003 International Semantic Web Conference (ISWC 2003)*, number 2870 in Lecture Notes in Computer Science, pages 30-46. Springer.
- [PHS05] Pan J, Horrocks I, and Schreiber G, 2005. OWL FA: A Metamodeling Extension of OWL DL. *In Proc. of the International workshop on OWL: Experience and Directions (OWL-ED2005)*.
- [PS05] Prud'hommeaux E, and Seaborne A, 2005, SPARQL query language for RDF. Technical report, World Wide Web Consortium. <http://www.w3.org/TR/rdf-sparql-query/>.
- [SWM03] Smith M, Welty C, and McGuinness D, 2003, "Owl web ontology language guide". <http://www.w3.org/TR/owl-guide>.
- [UG96] Uschold, M., Gruninger, M.: Ontologies: Principles, Methods and Applications. *The Knowledge Engineering Review*, 1996.
- [WPH06] Wang T, Parsia B, Hendler J, 2006. A Survey of the Web Ontology Landscape. *In proceedings of International Semantic Web Conference*, Athens, Georgia, USA.
- [ZVS04] Zhang Y, Vasconcelos W, and Sleeman D, 2004, OntoSearch: An Ontology Search Engine. *Twenty-fourth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Cambridge, UK.

⁶ <http://www.aktors.org>