Ontological semantic search of learning objects: experiments and results

João Carlos Gluz[1], **Luiz Rodrigo Jardim. da Silva**[1]

[1] Programa Interdisciplinar de Pós Graduação em Computação Aplicada (PIPCA) - Universidade do Vale do Rio dos Sinos (UNISINOS) - Caixa Postal 275 - 93.022-000 - São Leopoldo - RS - Brasil

jcgluz@unisinos.br, rodjle@gmail.com

Abstract. *Generating relevant results in semantic searches typically involves the context of the search term, and may include, besides elements of the search context, other elements like the location, intention, variation of words and synonyms. Ontologies and metadata are important computational tools to bring the semantics for learning object queries. The semantic search system presented in this paper aims to retrieve learning objects organized by OBAA metadata. Its search engine supports the integration of multiple educational ontologies with the OBAA metadata ontology. However, when distinct ontologies need to interoperate, semantic compatibility problems can lead to unpredictable, ambiguous or incomplete results. The semantic search system used in this paper addresses these issues through a combination of ontology aligning and mapping mechanisms. This paper presents the mapping and alignment algorithms used in the search engine. Experiments were conducted to evaluate the quality of the information returned by the search from the point of view of users. These experiments showed positive results.*

1. Introduction and Methodology

The generation of significant results in semantic searches can involve, for instance, the understanding of the intention of the user and the context of the search term, either on the Web or within a closed system [Gunter, 2009; Sujatha et al, 2011]. Thus semantic search engines usually should consider several points, including the search context, its location, the intention of the user, the variation of the words, and treatment of synonyms, possible concept correspondences and even natural language structure of the query, to provide relevant search results.

Ontologies and metadata are important computational tools to bring semantics when querying for web information. Metadata [NISO, 2003] are used to represent information about a particular object as: name, location, description, technical characteristics, relationship with other objects, etc. An important feature of the metadata is that its data elements and types can be considered symbolic structures that can be efficiently handled by current techniques of knowledge representation and natural language processing.

The properties that characterize a search of objects as semantics, for example, the ability to understand the user's intention or understand the context of the search term [Gunter, 2009; Sujatha et al, 2011], require an epistemic basis, which will define

what is knowledge, and how it is can be "understood" by the system. This epistemology must be supported by a effective technology to make the semantic search feasible. Computational ontologies offers a technological solution to this issue. A computational ontology is a formal and explicit definition of the conceptual categories existing in some knowledge domain [Berners Lee et al., 2001]. An ontology incorporates an axiomatic structure, based on the description of these concepts, which defines the semantic relationships between them, plus of its attributes, properties and relations.

However, given the heterogeneity of existing ontologies [Ehrig, 2007] one of the main challenges for current semantic search engines, particularly in the case of the semantic search for Learning Objects (LO), is how to enable the integration of heterogeneous ontologies, not only from distinct domains of knowledge , but even within the same domain. Ontologies belonging to same domain can often be written using different vocabularies, hindering interoperability between them.

Nowadays, LO technology is a critical element in the design, and implementation of any digital educational system. Repositories form a central piece of the LO technology, providing storage spaces where learning objects can be cataloged, located, and retrieved. In general, LO repositories contains only the metadata, which is used to catalog, and locate the objects, allowing that the corresponding content to be stored in other web servers. Currently the most prominent LO repositories are based on DSpace technology (*http://www.dspace.org*), and use relational databases for storing metadata.

This kind of technology usually allows only syntactical searches in the repository. In this kind of search the semantic of terms contained in LO metadata is not considered, only the syntactical and morphological aspects of these terms are taken into account in the search process. A semantic search is another form of search were the meaning of the words used in the search (or even the meaning of entire sentences) is considered in the search process. In a semantic search, the generation of relevant results could involve, for instance, understanding the intention of the user and the context of the search terms [Gunter, 2009; Sujatha et al, 2011]. Techniques based on ontologies could help in this situation. These techniques already are suitable for the representation of learning domains, educational applications, and, thus, learning objects [Bittencourt, 2009; Mizoguchi, 2007].

The MSSearch system presented in this work uses advanced ontology alignment techniques to create a semantic search engine, and a native OWL LO repository. The OBAA metadata ontology [Gluz & Vicari, 2012] was chosen to represent, and store LO metadata, because this OWL ontology not only fully represents IEEE-LOM metadata, but also provides new metadata for accessibility issues, multimedia, and multi-platform contents, and could represent all non-qualified DublinCore metadata. This system relies on the use of ontology alignment techniques, software agents and inference mechanisms for retrieving information semantically annotated.

2. Related Work

There are some works that try to explore the use of ontologies, and ontology alignment techniques, to provide semantic search services. For instance, the D-OSWS [Ochs et al., 2011] system uses alignment mechanisms to build an intermediate ontology to conduce

searches of famous people in DBpedia, and the BROAD [Teixeira et al., 2012] tool currently provides RESTful services able to search LOs using SPARQL queries, and is incorporating inference engine support to provide semantic search of LO. However, when compared with MSSearch semantic search engine presented in this work, they have some important limitations. D-OSWS system explores, and advances alignment mechanisms, but cannot be directly applied to standard learning objects. The BROAD tool is more similar to MSSearch, but it stops on the semantic search based only on the metadata ontology, ignoring the problem that metadata ontologies offer a shallow semantics for metadata. Different than MSSearch, this tool does not considers the alignment of independent ontologies for learning domains, teaching strategies, or other educational topics.

Surely, there are some problems in the use of ontology technology. One important problem addressed in the present work is how to correlate LO metadata stored in the repository to educational ontologies, which represent, for instance, the learning domains, teaching strategies, and other educational topics of these LO. The process to establish the relation among metadata and educational ontologies, or among distinct, but generally heterogeneous educational ontologies could be very complex, and, if done manually, very tiring. Fortunately, there are some techniques that can make this process easier, allowing the automatic, or semi-automatic establishment of the relations among the ontologies. Ontology alignment [Ehrig, 2007; Euzenat, 2007] is currently regarded as an important mechanism for the integration of semantically heterogeneous databases, and as an enabling technology to provide semantic searches on these databases. However, sometimes the complexity involved in this process can require a lot of computing power [Shvaiko & Euzenat, 2011]. Thus, it is necessary be careful with the implementation of these techniques, always verifying the resulting performance of the system.

3. The Multiagent Architecture

The architecture of MSSearch system was divided into three main layers: (a) the ontology layer, which specifies the knowledge that will be shared among agents, (b) the agents layer, and (c) the interface layer, which allows agents to interact with users, LO repositories, databases, and other external applications.

The overall architecture of MSSearch is shown in Fig. 1. The ontology layer is formed by a set of educational ontologies aligned to the metadata ontology. The interface layer contains the web interface with common users (*WebQueryInterface*) and administrators (*WebAdminInterface*), the web services interface (*RESTfulInterface*), and the interface with learning object repositories through the OAI-PMH[1] harvesting protocol (*OAIPMHInterface*).

The agents layer is composed by the following agents:

– *MetaQuery*: agent responsible for executing the queries in semantic repository;

– *MetaUpdate*: agent that updates metadata stored in the repository;

– *MetaLoad*: agent, which is charged with the task of to populate the database with learning object metadata;

1 http://www.openarchives.org/pmh

– *OntoAlign*: agent that perform the alignment of ontologies;

– *SemanticSearch*: agent that implements the semantic search mechanism. This agent also implements the relevancy-based ordering of query results;

– *RDFBaseManager* : this agent encapsulates the storage facility of native RDF triples storage, which currently is the graph storage system provided by JENA TDB;

– *OWLReasoner* : agent that encapsulates the OWL inference engine used in MSSearch. Currently this agent is integrated with the Pellet reasoner.

All agents were developed using the JADE framework to allow an easy integration with web interfaces. The reasoning and knowledge representation processes of agents were implemented with the help of ontology-based tools, including JENA, Pellet and OWL-API.

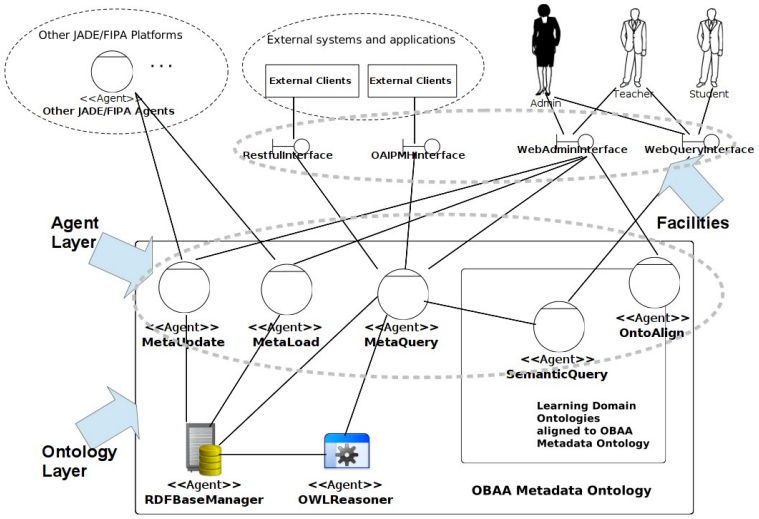

Fig. 1. MSSearch Architecture

3.1. Semantic Repository

Agents *MetaQuery*, *MetaUpdate*, *MetaLoad*, *OWLReasoner*, and *RDFBaseManager* are the core subsystem of MSSearch. This subsystem combines the JENA TDB RDF database, with the Pellet reasoner, to provide a native OWL repository able to store, locate, and retrieve LO metadata. The remaining agents implement the semantic search, and alignment functionality. Using query in the format SPARQL/TERP [Sirin et al., 2010] it is possible to locate LOs which are currently available on the database. An very simple initial search to find these LO, could be implemented by the following query:

```
SELECT ?lobj
       WHERE {?lobj a LearningObject}
```

This query will identify all objects belonging to class LearningObject currently contained in the test base. This query assumes that only test objects will be contained on the database. However, if it is necessary to locate only the objects belonging to the catalog "OBAA Test Objects", then the following query should be used:

```
SELECT ?lobj
WHERE   {?lobj a
           (LearningObject and
```

```
            (hasMetadata some
                    (hasIdentifier some
(itsCatalogIs value "OBAA Test Objects")))).}
```

The variable ?lobj will be replaced only by individuals LearningObject class, which have some identification metadata (restriction defined by the clause "hasMetaData some (hasIdentifier ... ") , whose catalog information (restriction indicated by the itsCatalogIs attribute) is "OBAA Test Objects ". Currently the answer to this query is the same as the previous query, but if other catalogs of objects may be stored in the database, then results could differ in future. Additional information can also be obtained. The following SPARQL/TERP query, gets the title and location of all LO that are supported by the UNIX operating system:

```
SELECT ?title ?loc
WHERE  {?lobj a (hasMetadata some
               (hasRequirement some
                 (hasOrComposite some(itsTechNameIs value
"unix")))).
       ?lobj hasMetadata ?mdtit, ?mdloc .
       ?mdtit itsTitleIs ?title .
       ?mdloc itsLocationIs ?loc. }
```

4. Ontology Alignment and Mapping

The alignment of ontologies may require a large computing power [David et al, 2011; Shvaiko & Euzenat, 2011]. To minimize this problem is possible to employ heuristics that reduce the number of steps in the comparing process. This heuristics are typically derived from the structure of the application domain. Besides the use of text-based techniques to align ontologies, the MSSearch semantic search engine also incorporates alignment heuristics in the form of annotations on the ontology.

The "Alias" annotation is used to include heuristic information on educational ontologies. For instance, to circumvent the issue of not having a good public domain lexical vocabulary of Portuguese language, it is possible to use values of "Alias" annotation initiated by the term "Align" to explicitly indicate synonyms for some class or relationship. These annotation values are understood by the alignment algorithm and used to find equivalent classes or relationships in other ontologies. A simple example could illustrate the use of this annotation: in Brazil, the secondary education period is currently called "Ensino Médio", thus a class "EnsinoMédio" could be used to represent this period of education. However, some time ago this period was called "Ensino de Segundo Grau" or "Ensino Secundário", thus "Alias" annotation values "Align.EnsinoSegundoGrau" and "Align.EnsinoSecundario" associated to this class name could allow the alignment of this ontology to older educational ontologies.

The alignment process, while necessary, is not sufficient to implement the mechanism of semantic search. For this mechanism to work is necessary to map the terms and concepts of educational ontologies to the LO metadata represented by metadata ontology. This process is called mapping and can not be characterized as an alignment between ontologies, because there is an important conceptual difference between educational ontologies and the metadata ontology used by MSSearch: educational ontologies typically identify concepts and terms related to educational contents, teaching methods and learning processes, while this metadata ontology represents the data types and possible values of LO metadata information.

The mapping process is helped by "Obaa" annotations. These annotations are understood by the search algorithm as a mapping between the ontology concept and some corresponding metadata value in the metadata ontology. For instance the values "Obaa.General.itsDescriptionIs" and "Obaa.General.itsKeywordIs" when associated to some class will indicate that the name of this class (or some equivalent class discovered by the alignment process) could appear in the Description or Keyword general metadata of some learning object.

Note that the process of annotation educational ontologies, even when done manually, do not need to be a tedious process: only the higher levels of the class hierarchy need to be annotated. The alignment process implements a mechanism of inheritance of annotations which ensures that the remaining levels of the hierarchy are automatically annotated.

The *AlignOnto* agent is responsible by alignment process. This agent extends the AlignApi API [David et al, 2011] to implement this process. This extension was necessary because of the increase on the complexity of the alignment process when there is a big number of elements belonging to the ontologies that will be aligned. Because of this possibility, the similarity algorithm was modified to avoid comparison with all entities, enabling only the comparison with annotated elements. The alignment process was divided in the following phases:

– *Input*: the alignment process begins with the passage of ontologies as a parameter to AlignApi functions;

– *Extraction of Concepts* (Classes): in this step is extracted a list of the labels of ontology classes;

– *Extraction of Annotations*: for each class in the previous list, is extracted a list of its Alias annotations;

– *Computation of Similarity*: in this phase the Monger-Elkan is used to compute the similarity between entities of ontology O1 and O2 , becaus of its good performance [3];

– *Threshold Checking*: this phase checks thresholds for acceptance of alignment. The degree of similarity of an alignment is in the range [0..1]. Degrees outside a pre-defined range will be automatically disregarded ;

– *Validation*: in this phase the reasoner is called to validate the structure, taxonomy and relationships of the resulting ontology;

– *Output* : after the validation, a file is generated containing the OWL axioms that establish the correspondences among the ontologies.

4.1. Semantic Search Engine

The semantic search engine aims to retrieve learning objects from a repository, which supports OWL (SPARQL) queries. Users needs only to provide the context of the search, in the form of keywords. After that the engine will correlate the context information with aligned ontologies to create an appropriate SPARQL query. The search mechanism is implemented by *SemanticSearch* agent according to the following steps:

– *Step* 1: the Orengo algorithm [Orengo & Huyck, 2001] is applied to remove suffixes and stop words from the keywords informed by the user;

– *Step* 2: the base of aligned ontologies is consulted to extract all annotations and terms semantically related to the keywords provided by the user. In this step if Obaa mapping annotations are present, they are used to relate keywords to specific learning object metadata. However, if no mapping annotation is found, the algorithm will correlate user provided keywords as target values of "itsDescriptionIs", and "itsKeywordIs" relationships;

– *Step* 3: a TERP/SPARQL query is built combining the list of terms extracted from the aligned ontologies with relationships that identify learning objects metadata, then this query is sent to *MetaQuery* agent and the resulting RDF triples are stored for posterior processing;

– *Step* 4: the information to be presented is organized, and formatted in HTML. First, it is sorted, in decreasing order, according to its relevancy. The evaluation of the relevance takes into account the number of occurrences found in the text of the metadata(higher is this number, the greater is the relevance. After that a filter is applied to eliminate less relevant results (the number of results can be defined by the user).

5. Experiments and Results

5.1. Performance Evaluation

The goal of performance evaluation experiment was to measure the execution time of operations to load, and query learning objects in the semantic repository when an increasing quantity of LO are stored in the repository. To do so, it was selected an external LO repository to be the source of metadata information to be used in the experiment. The source repository chosen was BIOE (*http://objetoseducacionais2.mec.gov.br/?locale=en*), which at the time of the tests contained approximately 17,600 learning objects. In Table 2 it is possible to check the amount of learning objects loaded (Load operation) along with their respective load times in the repository.

Table 2. Load operation performance experiment

# of LO	Time(s)	RDF Triples	Triples per Sec.
99	13	2354	181.08
198	14	5107	364.79
412	17	9836	578.59
897	20	17928	896.40
1888	24	39883	1661.79
4196	33	72446	2195.33
11088	66	192785	2920.98

Based on these results, it is possible to infer that the load time remained linearly proportional to the number of objects, indicating a possible maximum complexity of order O(n) for this process. The Load operation even showed a better performance with loads above 2000.

In another test a complex SPARQL query was performed (see Fig. 2), aiming to recover all learning objects stored in the semantic repository, ordered by title.

```
SELECT ?lobj ?key ?desc ?loc ?plat ?title
WHERE {
    ?lobj a obaa:LearningObject  .
    ?lobj obaa:hasMetadata ?mdata.
    ?mdata obaa:itsKeywordIs ?key.
    ?lobj obaa:hasMetadata ?mdtit. ?mdtit obaa:itsTitleIs ?title .
    ?lobj obaa:hasMetadata ?mddesc. ?mddesc obaa:itsDescriptionIs ?desc
.   ?lobj obaa:hasMetadata ?mdloc. ?mdloc obaa:itsLocationIs ?loc .
}
order by ?title
```

Fig. 2. SPARQL query used in query performance experiment

The time spent for query execution can be seen in Table 3. According to the data presented in this Table, the performance of query operation appears to be logarithmically proportional when the number of LO stored in the repository ranges from 99 to 4200, passing to a more linear performance after 4200.

Table 3. Query operation performance experiment

# of LO	Time(s)	LO per Sec.
99	1.78	55.6
198	2.11	93.8
412	2.81	146.6
897	3.27	274.3
1888	5.07	372.3
4196	6.30	635.7
11088	14.28	776.4

Despite the need for further testing, these data are indicative of a possible optimal performance of order O(log(n)) for the search, with a possible maximum of order O(n), both very good results for queries. The results of the performance experiments suggest that semantic repository technology may already support a relatively big amount of metadata without compromising the performance.

5.2. User Perception Evaluation

The goal of this experiment was to evaluate the quality of query results returned by MSSearch when compared to the results returned by the search engine of BIOE for similar queries. Fig. 3 show the user interface of MSSearch and typical results of a query.

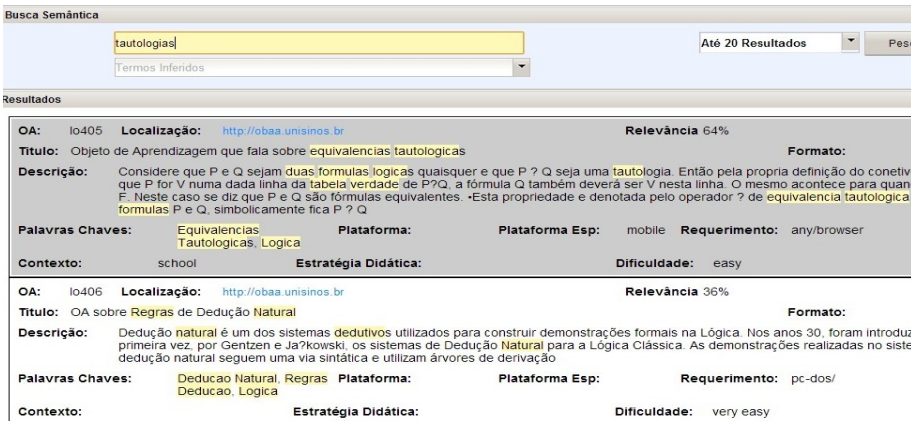

Fig. 3. MSSearch user interface with some query results

The learning domain chosen for the experiment was high school mathematics. To execute the experiment the semantic repository of MSSearch was populated with more than 11.000 learning objects from BIOE. This included all BIOE objects that

contained educational materials related to mathematics. The experiment was conduced by four teachers. All teachers had post-graduation titles in mathematics, with approximately 10 years of experience teaching mathematics. In the experiment teachers could create terms for the search (eg "polynomials") of your free choice, and then submit them to MSSearch and BIOE. Based on the query results returned, the teachers completed an assessment questionnaire. The evaluation experiment required that each teacher make two distinct searches, with two different keywords of their choice. Table 4 summarizes the results of the assessments made by teachers.

The results from Table 4 show that, from the point of view of its users, MSSearch consistently returned best query results than BIOE. Two items are important to note: the relevancy of results returned by MSSearch, and the ordering of these results. Both items were considered by users highly satisfactory for MSSearch, at least when compared with BIOE: 62.5 % of users considered MSSearcg results relevant and well ordered against only 12.5 % of BIOE.

Table 4. Results from User Perception Experiment

The system returned some result?	**Yes**	**No**	**Partial**
MSSearch	75.0%	0.0%	25.0%
BIOE	75.0%	25.0%	0.0%
The results were as expected?	**Yes**	**No**	**Partial**
MSSearch	62.5%	25.0%	12.5%
BIOE	37.5%	50.0%	12.5%
The results were relevant?	**Yes**	**No**	**Partial**
MSSearch	62.5%	0.0%	37.5%
BIOE	12.5%	50.0%	37.5%
The results were in the context of the search?	**Yes**	**No**	**Partial**
MSSearch	50.0%	25.0%	25.0%
BIOE	37.5%	37.5%	25.0%
The results were well-ordered by their relevancy?	**Yes**	**No**	**Partial**
MSSearch	62.5%	12.5%	25.0%
BIOE	12.5%	75.0%	12.5%
The number of results were limited as asked?	**Yes**	**No**	**Partial**
MSSearch	75.0%	12.5%	12.5%
BIOE	25.0%	25.0%	50.0%
The quantity of information was satisfactory?	**Yes**	**No**	**Partial**
MSSearch	50.0%	25.0%	25.0%
BIOE	37.5%	50.0%	12.5%
The answering time was OK?	**Yes**	**No**	**Partial**
MSSearch	50.5%	50.0%	0.0%
BIOE	50.0%	12.5%	37.5%

6. Conclusions

The main goal of this work was to present a system that combines state of the art agent and ontology technologies, with advanced alignment techniques to built a semantic search engine for learning objects. The paper address, from software engineering perspective, how to integrate ontology and agent engineering to built a successful application.

The MSSearch system got positive reviews by users who used it. Among these reviews, we highlight its performance in search and presentation of results, and also the quality of information retrieved. This is due mainly because the use of inference

mechanisms and ontology alignments techniques, in the context of an native OWL repository. The good results of the performance experiments also indicate that this system has good possibilities to become a fully production system, being able to provide the core, and search facility of LO repository. It can, indeed, to become a fully operational semantic repository of learning objects.

References

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) The Semantic Web, Scientific American. May.

Bittencourt, I. I., Costa, E., Silva, M., Soares, E. (2009) A computational model for developing semantic web–based educational systems. Knowledge-Based Systems, v.22, n.4, p. 302 - 315.

Cohen, W., et al. (2003) A comparison of string distance metrics for name-matching tasks. Procs. of IJCAI-03 Workshop on Inf. Integration on the Web.

David, J., Euzenat, J., Scharffe, F., Santos, C. (2011) The Alignment API 4.0. Semantic Web, v.2, p.3-10.

Ehrig, M. (2007) Ontology alignment: bridging the semantic gap. Springer.

Euzenat,J., Shvaiko, P. (2007) Ontology matching. Springer.

Gluz, J.C., Vicari, R. (2012) An OWL Ontology for IEEE-LOM and OBAA Metadata. Procs. of ITS 2012, Crete. LNCS. New York: Springer, 2012. v.7315. p.696 - 698.

Gunter, D. (2009) Semantic search. Bull. of the Amer. Soc. for Inf. Sci. and Tech., Oct-Nov, v.36, p.36.

Mizoguchi, R., Hayashi, Y., Bourdeau, J. (2007) Inside Theory-Aware & Standards-Compliant Authoring System, Procs. of SWEL'07, p.1-18.

NISO. (2004) National Information Standards Organization. Understanding Metadata. Bethesda, MD, USA: NISO Press.

Ochs, C., et al. (2011) Google Knows Who Is Famous Today - Building an Ontology From Search Engine Knowledge and DBpedia. Procs. of Fifth IEEE Int. Conf. on Semantic Computing.

Orengo ,V., Huyck, C. (2001) Stemming Algorithm for Portuguese Language. Procs. of the Symposium on String Processing and Information Retrieval.

Shvaiko, P.; Euzenat, J. (2011) Ontology matching:state of the art and future challenges. IEEE Transactions on Knowledge and Data Engineering, vol.PP, no.99.

Sirin, E., Bulka, B., Smith, M. (2010), Terp: Syntax for OWL-friendly SPARQL Queries. In: 7th OWL Experiences and Directions Workshop. San Francisco.

Sujatha, R. et al. (2011) Semantic Search Engine: A Survey. International Journal of Computer Technology and Applications, v.2. n.6, p.1806.

Teixeira, T.N., Campos, F. Braga, R., Santos,N. Mattos, E. (2012) BROAD Project: Semantic Search and Application of Learning Objects. IEEE Technology and Engineering Education (ITEE), v.7, n. 3, p.23-32.