Journal of Universal Computer Science, vol. 22, no. 1 (2016), 16-36
submitted: 30/6/15, accepted: 20/12/15, appeared: 1/1/16

Towards a Semantic Repository for Learning Objects: Design and Evaluation of Core Services

João Carlos Gluz
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
jcgluz@unisinos.br)

Ederson Luis Silveira
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
esilveira1@gmail.com)

Luiz Rodrigo Jardim da Silva
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
rodjle@gmail.com)

Jorge Luis Victória Barbosa
(University of Vale do Rio dos Sinos (UNISINOS), São Leopoldo, Brazil
jbarbosa@unisinos.br)

Abstract: Repositories form a central piece of the learning objects technology, providing storage spaces where the objects can be catalogued, located, and retrieved. However, repositories usually support only syntactical and morphological aspects of learning objects metadata for cataloguing and searching purposes. This article proposes two integrated systems, which provide the core services of a semantic repository of learning objects. MSSearch system uses ontology alignment techniques to create a semantic search engine and a semantic database for learning objects metadata. MSSearch supports the integration of multiple educational ontologies through a combination of ontology aligning and mapping mechanisms. In turn, Linnaeus system offers intelligent support for the creation and editing of learning object metadata. This tool employs the technologies of intelligent agents and educational ontologies to provide an intelligent semi-automatic metadata filling service. This article presents the main architectural components, ontological models and interface facilities of these systems. The text finishes with the presentation of the experiments conducted to validate Linnaeus and MSSearch.

Keywords: Semantic Repository, Semantic Search, Cataloguing, Learning Objects, Ontologies, Intelligent Agents.
Categories: L.3.0, L.3.2

1 Introduction

Nowadays, Learning Objects (LO) technology is a critical element in the design, and implementation of any digital educational system [Polsani 2003]. LO technology is based on the assumption that it is possible to create educational components, and arrange them in order to promote reuse, saving production cost, and time [Wiley 2002]. To enable reuse, discovery and to facilitate interoperability, several LO metadata standards, such as IEEE LOM [IEEE 2002], Dublin Core [Kunze and Baker

2007], SCORM [ADL 2004] and OBAA [Vicari et al. 2010] were created. In these standards, learning objects are analysed on two abstraction levels, the metadata level, related to information about the object and the content level, related to the encapsulation, and organization of the learning material itself.

Agents, ontologies and metadata technologies are useful to bring semantics for educational contents when designing digital educational environments [Bittencourt et al. 2009; Bittencourt et al. 2008]. Software agents derived from Distributed Artificial Intelligence research [Weiss 2013]. They differ from the usual notions of software processes or software objects because their requirements of autonomy, reactivity, pro-activity and sociability [Wooldridge 2009]. Intelligent agents are a refinement of software agents, applying knowledge representation and reasoning techniques to interpret perceptions, to plan and decide what actions should be execute to meet their goals, and to evaluate the results of these actions [Wooldridge 2009]. Computational ontologies are formal and explicit definition of the conceptual categories existing in some knowledge domain [Gruber 1993]. Ontologies incorporate an axiomatic structure, based on the description of these concepts, which defines the attributes, properties and relations between them. Ontologies are also commonly associated with the Semantic Web (SW) initiative, which offers a future vision for the Web where information is given an explicit meaning, making it easier for machines to automatically process and integrate information available on the Web [Berners-Lee et al. 2001]. Developments in Description Logics (DL) [Baader et al. 2007] resulted in OWL [W3C 2012], a logical language used to specify SW ontologies. Metadata [NISO 2004] provide the data elements and types used to represent information about contents: title, author, location, description, technical characteristics, relationship with other objects, etc.

In terms of digital educational environment, ontologies can specify the properties of educational contents and applications, but they are not tools to implement the active components of a system. On the other hand, software agents can implement the active components of the system, relying on ontologies as the source of the semantics of shared knowledge related to educational content. In this context, metadata elements and types define the basic linguistic terms used in educational ontologies.

There are several cases in the literature of independent use of agent, ontology or learning objects technologies in the design of educational systems. But the combined and integrated use of these three technologies in the design, and development of educational applications is not very common and is a key differential of the present work.

Repositories form a central piece of the LO technology, providing storage spaces where learning objects can be catalogued, located, and retrieved. In general, LO repositories contain only the metadata, which is used to catalogue, and locate the objects, allowing that the corresponding content is stored in other web servers. Currently the most prominent LO repositories are based on *DSpace* technology [Smith et al. 2003], and use relational databases for storing metadata. This kind of technology usually allows only the use of syntactical and morphological aspects of LO metadata for metadata editing and for searching purposes.

The main hypothesis of the present work is that the technologies of agents, ontologies and LO metadata can be successfully integrated to provide the semantics for two of the most critical services of a LO repository: the search service, used to find LO appropriate for some educational purpose and the LO metadata editing service, essential for the correct cataloguing, indexing, searching and retrieving of this kind of objects.

In a semantic search the meaning of the words used in the query is considered in the search process. This involves, 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; Ramachandran and Sujatha 2011]. However, the properties that characterize a search of objects as semantics, like the ability to understand the user's intention or understand the context of the search term, require an epistemic basis, which will define meanings, and how they can be understood by the system. This epistemology must be supported by an effective technology to make the semantic search feasible. Ontology technology could help in this, already being suitable for the representation of learning domains, educational applications, and learning objects [Bittencourt et al. 2009; Mizoguchi et al. 2007].

The *MSSearch* system proposed in this article implements a semantic search engine together with a LO repository that represents LO metadata in OWL. Besides, being an example of how to combine agent, ontology and LO metadata technologies to build a semantic search engine, *MSSearch* contributes with a new approach to help in scalability issues: it uses ontology alignment techniques [Ehrig 2007; Euzenat and Shvaiko 2013] to integrate new ontologies on the search engine.

The main problem related to LO metadata creation and editing, is the complexity and extension reached by current LO metadata standards, like IEEE-LOM [IEEE 2002]. Asking for content designers (or teachers) to understand the meaning and applicability of all data elements of these standards to correctly catalogue their material is simply ineffective, increasing substantially the time and effort spent to correctly fill the metadata. Metadata creation and editing are crucial activities of LO repositories, LO correctly catalogued and indexed is a necessary condition for obtaining relevant material from search and retrieving processes. Thus, the creation of intelligent tools able to assist these activities becomes a key point for LO repository technology.

This is the main goal of the *Linnaeus* tool proposed in this article. This tool assists the activities of creating and editing LO metadata for educational domains. *Linnaeus* is also based in a successful combination of agents, ontologies and LO metadata technologies, but its main contribution is the automatic metadata filling mechanisms based on inferences over the aligned educational ontologies produced by *MSSearch*.

These two integrated systems *Linnaeus* and *MSSearch* provide the core services of the semantic repository of learning objects proposed in this work. The OBAA metadata ontology [Gluz and Vicari 2012] represents, and stores the LO metadata of the semantic repository. This OWL ontology was selected, because it fully represents IEEE-LOM metadata, providing additional new metadata able to represent accessibility, multimedia, and multi-platform information associated with contents. Moreover, OBAA ontology can represent all non-qualified Dublin Core metadata [Kunze and Baker 2007].

2 Related works

As independent technologies, agents, ontologies and learning objects are being applied to the design and development of educational systems for a while. There are some cases of integrated use of two of these technologies, but the combined and integrated use of these three technologies in the design, and development of educational applications is not very common.

Agent-based technology has been instrumental for the design of educational systems such as Intelligent Tutoring Systems (ITS) for a long time [Gürer 1998; Giraffa and Vicari 1998]. Intelligent agents able to take educational, or pedagogical roles to facilitate, or assist teaching processes are called pedagogical agents [Gürer 1998; Giraffa and Vicari 1998]. Recent examples of this literature are the works [Bittencourt et al. 2008; Frasson et al. 2005; Sklar and Richards 2006]. Agent technology has also been applied in the design of other types of educational systems, with some emphasis on the use of agents in Virtual Learning Environments (VLE) [Arias et al. 2009] and digital contents [Dietze et al. 2007], as exemplified by the Intelligent Learning Object (ILO) approach proposed in [Silveira et al. 2004].

Most of the works on agents still are very focused on the ITS [Bittencourt et al. 2008; Frasson et al. 2005; Sklar and Richards 2006] and VLE [Arias et al. 2009] perspectives, not intersecting with the LO semantic repository perspective considered in this work. The application of agent technology for LO is more promising, however the works observed in the literature [Stoilescu 2008; Silveira et al. 2004] also do not take into account the requirements and questions related to LO repositories.

The use of ontology technology already has proven useful in designing various types of educational environments including, among others, web environments [Bittencourt et al. 2009], and formal educational models [Hayashi et al. 2009]. In this context, ontologies are used to define the properties of elements and entities related to the educational system. There is a tendency to follow the structure of the ITS, and divide educational ontologies into three types: a) ontologies for the learning domain, b) ontologies about teaching methods, and c) ontologies about the student model. However, there are also proposals that integrate partial aspects of these three types [Mizoguchi et al. 2007; Hayashi 2009].

Generally, works on the application of ontologies in education [Bittencourt et al. 2009; Mizoguchi et al. 2007; Hayashi et al. 2009] were more interested on the impact and benefits of this kind of technology on learning environments, exploring how ontologies could represent learning domains, educational applications, students profile, curricular structure and similar issues. The relation with our work is irregular, being more predominant when issues related to learning contents are considered. When this occurs, the most important differential in our proposal is the dynamic alignment mechanism that allow the semantic repository to keep for searching purposes an updated an integrated ontology, composed not only by the metadata ontology, but by educational ontologies representing distinct learning domains, curricular structure and pedagogical knowledge.

The application of ontology technologies derived from Semantic Web initiative [Berners-Lee et al. 2001], on the design of educational environments and systems is a relatively recent trend [Isotani et al. 2009; Bittencourt et al. 2008]. The works [Dietze et al. 2009; Dietze et al. 2007] are directly related to the approach employed in the present work, focusing on the creation of semantic web-services able to represent and manipulate LO metadata through the ontologies. The work [Dietze et al. 2007] defines an OCML ontology, which partially represents the SCORM 2004 [ADL 2004], IMS LD [IMS, 2014] and IEEE-LOM [IEEE 2002] metadata standards. A semantic web-service architecture is defined to represent and handle LO metadata. The work [Dietze et al. 2009] proposes a content adaptation model based on goals associated to a learning process and the LO metadata of the educational resources used in this process. It uses the same OCML ontologies presented in [Dietze et al. 2007].

Besides the use of agents instead of web-services, the most important differential that *MSSearch* and *Linnaeus* have in respect to [Dietze et al. 2009; Dietze et al. 2007] is the metadata ontology employed to represent LO metadata. This ontology is based on OWL standard and fully represents all IEEE-LOM [IEEE 2002] and non-qualified *DublicCore* metadata [Kunze and Baker 2007]. *MSSearch* also suffers from the same scalability issues of [Dietze et al. 2009] being dependent on a unique aligned ontology for searching purposes. However, *Linnaeus* does not have this problem, handling sets of several ontologies to help users to fill metadata.

More recent works have started to consider the Linked Data (LD) approach for the design of educational applications [Yu et al. 2012; D'Aquin et al. 2013]. The LD approach begins by admitting that concrete real-world (i.e. web) data is essential for any Semantic Web (SW) application. To be useful for SW purposes, LD should be in the form of RDF triples that link web documents (or elements of documents) to properties and concepts (classes) eventually available in the form of ontologies. Due to the non-constrained form of LD (they do no need to be related to any OWL ontology), there is a real necessity to use data mining, knowledge discovery or other analytics technique to understand the meaning of these data. Because of this, even assuming that LD offers a very interesting research possibility, there is a very minimal intersection on this kind of research with the work presented in this paper.

Much of the work related to LO has focused on creating tools to assist in the creation of learning objects. FreeLoms [Freeloms 2014], eXe Learning [G.-Barbone and A.-Rifon, 2010] and Xerte [Ball and Tenney 2008] tools are the outstanding examples of this kind of work. All of them provide a graphical interface to fill the metadata. Another relevant area of study is dedicated to create repositories to store, edit and retrieve LO metadata. These studies resulted in systems like *DSpace* [Smith et al. 2003)] that uses the DCMI metadata and FEB [Schreiner et al. 2012] based on OBAA metadata [Vicari et al. 2010].

LO authoring tools [Freeloms 2014; G.-Barbone and A.-Rifon 2010; [Ball and Tenney 2008] provide mechanisms for information entry supported by vocabulary and explanations about the semantics of metadata, but the information still has to be filled manually by the user. They suffer from the lack of intelligent mechanisms to help in the filling of LO metadata automatically. Current technologies for LO repositories, like the *DSpace* system [Smith et al. 2003] can edit and store LO metadata through a web interface, and retrieve these metadata through an OAI protocol interface. However, the semantics of the information stored in LO metadata is not considered for these operations.

There are some works that provide semantic search services based on ontologies and ontology alignment techniques. For instance, the D-OSWS [Ochs et al. 2011] system uses alignment mechanisms to build an intermediate ontology to search famous people in DBpedia. The BROAD [Teixeira et al. 2012] tool provides RESTful services to search LO using SPARQL queries. This tool incorporates an inference engine to provide semantic search of LO. However these works have some gaps specifically addressed by *MSSearch*. D-OSWS [Ochs et al. 2011] explores the alignment mechanisms, but cannot be directly applied to standard learning objects. The BROAD [Teixeira et al. 2012] uses the semantic search based on the metadata ontology and focused on LO, but ignores that the metadata usually define data-types of items. Therefore, this tool does not explore the concepts and relations that could exist among these items. Moreover, this tool does not consider the alignment of independent ontologies for learning domains, teaching strategies, or other educational topics.

3 Software Architecture

The software architecture of the semantic repository was designed according to the guidelines of the MILOS [Gluz et al. 2012]. MILOS is an agent-based infrastructure that provides intelligent support for all activities involved in the life cycle of OBAA LO. The architecture is divided into three abstraction layers: (a) *Ontology layer*: it is the conceptual layer of the infrastructure that specifies the knowledge that will be shared among agents; (b) *Multiagent systems layer*: it is the action layer that implements the requirements foreseen in OBAA proposal; (c) *Interface facilities layer*: it provides the communication facilities of MILOS agents, allowing the agents to interact with users, LO repositories, databases and other external applications. This layered structure was used to design the semantic repository proposed in this article, resulting in the software architecture presented in Figure 1.

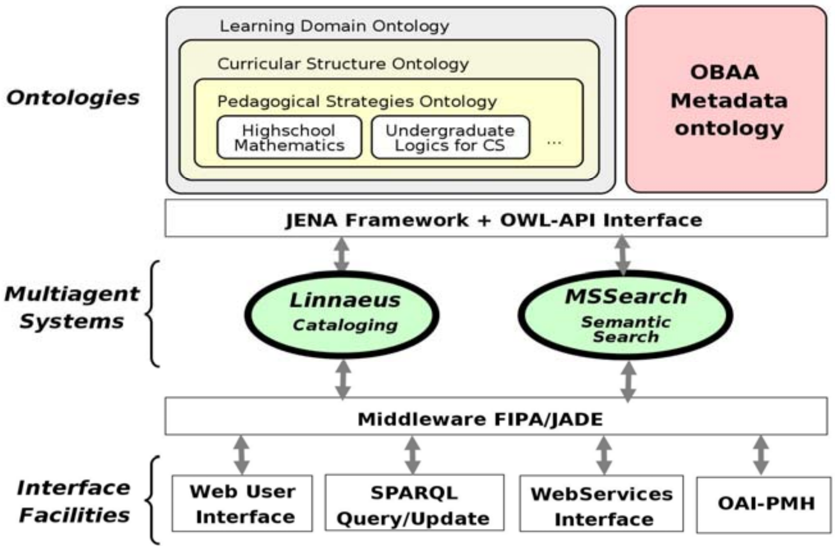

Figure 1: Overview of Semantic Repository Architecture.

In this architecture the ontology layer is divided in the OBAA metadata ontology, which represents the basic knowledge about LO stored in the repository, and a set of educational ontologies that contains the knowledge about the learning domains of these objects, the knowledge about the curriculum where these objects are inserted and the knowledge about the pedagogical strategies and teaching methods to be used with these objects. The intelligent services provided by the semantic repository are divided in two multiagent systems: the *MSSearch* semantic search engine and the *Linnaeus* system, which helps users to create and edit LO metadata.

The overall architecture of *MSSearch* is shown in Figure 2. The UML 2.0 communication diagram shown in Figure 2 is used in all MILOS projects. It extends the UML notation in some aspects that is worth to explain. Agents' stereotypes can be used as lifelines of the diagram. Ontologies are represented by rounded rectangles. If some ontology encloses an agent, this implies that the agent uses the ontology to give meaning to its interactions with other agents. Thus, the communication links connected to this agent that pass over the ontology rectangle must contain messages related to the ontology. If some ontology is enclosed inside other ontology, then the enclosing ontology provides terms and relations that are used in the enclosed ontology.

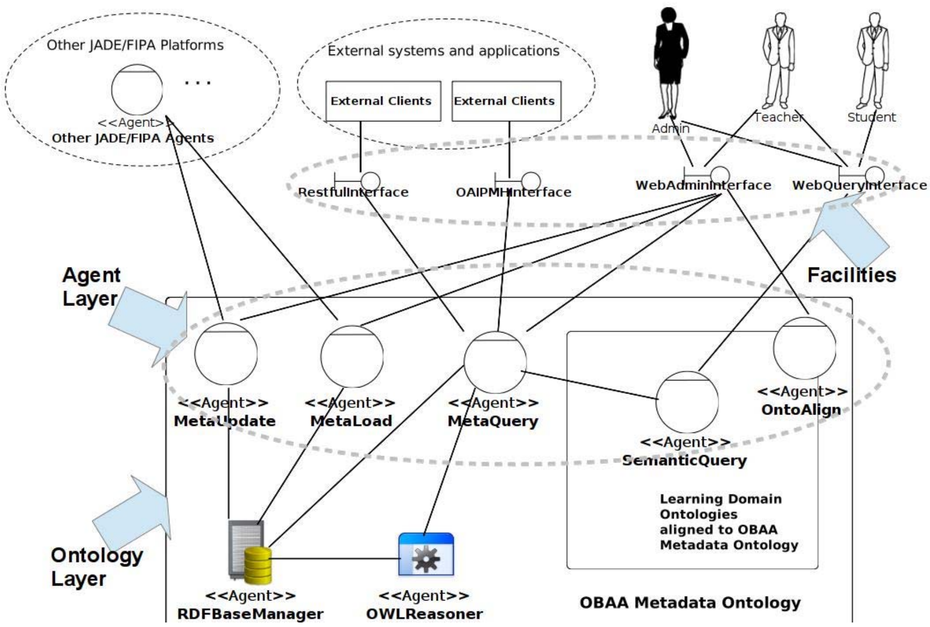

Figure 2: MSSearch system architecture.

The *WebQueryInterface* facility provides the web interface with end users, while the *WebAdminInterface* facility provides the web interface for administrators. The web services interface is implemented by the *RESTfulInterface* facility, and the

OAIPMHInterface facility implements the access gateway to LO repositories through the OAI-PMH harvesting protocol .

The *MetaQuery* agent is responsible for executing the queries in the semantic repository. *RDFBaseManager* agent encapsulates a RDF triples storage, which is the database used by the semantic repository. *MetaLoad* agent makes bulk loads of learning objects metadata to initialize the database, while *MetaUpdate* agent updates metadata stored in the database. The *OntoAlign* agent performs the alignment of ontologies. The *SemanticSearch* agent implements the semantic search mechanism and the relevancy-based ordering of query results. This agent works with the *OWLReasoner* agent that encapsulates the OWL inference engine used in *MSSearch*.

The *MetaQuery*, *MetaUpdate*, *MetaLoad*, *OWLReasoner* and *RDFBaseManager* agents form the core subsystem of the semantic repository. It combines the JENA TDB RDF database, with the *Pellet* reasoner [Clarksia 2014] to provide a semantic database able to store, locate, and retrieve LO metadata in RDF format.

The *Linnaeus* system helps the users in the activities of creating and editing LO metadata information. The software architecture of *Linnaeus* is aligned with MILOS infrastructure [Gluz et al. 2012] and integrated with the *MSSearch* system, as shown in the Figure 3. The main function of *Linnaeus* is to provide a set of wizards to support the filling of LO metadata. The *Linnaeus* provides an intelligent and proactive service that helps users without technical knowledge about standards of LO metadata.

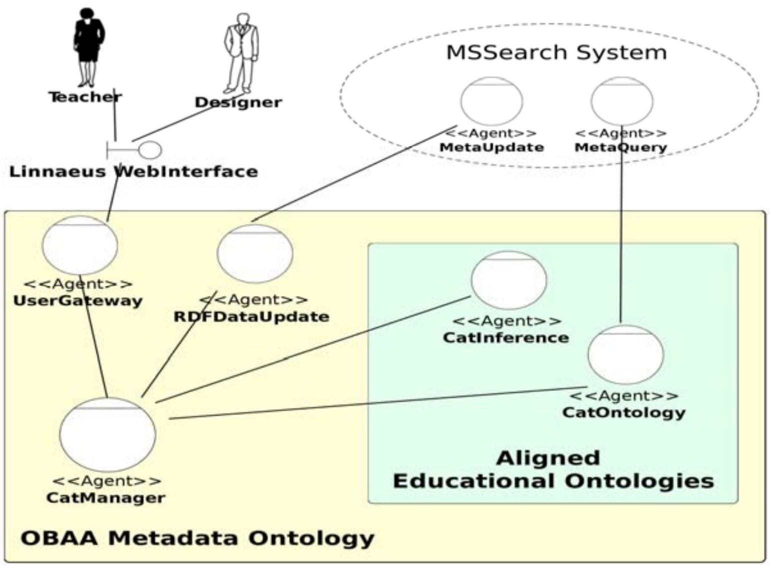

Figure 3: Linnaeus system architecture.

The intelligence of *Linnaeus* user interface was based on wizards. The creation of the LO metadata occurs semi-automatically with little user involvement, guided by inference mechanisms. The inference process can generate and validate LO metadata considering the previous experience of user with learning domains. The *UserGateway* agent is responsible for all user interaction. Requests or responses from users are formatted and sent to the *CatManager* agent that in turn will activate other agents.

The *CatManager* agent coordinates the metadata generation process. In the first step of its basic cycle of operation, *CatManager* activates the *CatOntology* agent to get information about educational and metadata ontologies currently available in the semantic store provided by *MSSearch* system. After, the *CatManager* requests *UserGateway* agent to obtain more information from the user. Having the ontology and user information, *CatManager* requests the *CatInference* agent to infer new metadata values based on the educational and metadata ontologies. This basic cycle repeats if the set of metadata is not complete. When the set of metadata for the LO being catalogued is completely generated, the *RDFDataUpdate* agent sends this set to the semantic store provided by *MSSearch* system. The *RDFDataUpdate* also is responsible for the update of metadata information already present in this store.

4 Ontological Model

Ontologies and metadata can be used in conjunction to support semantics either to help users to fill the LO metadata or to query for information in the repository. Figure 4 shows the overall structure of the ontological model adopted by the semantic repository. The OBAA metadata ontology [Gluz and Vicari, 2012] provides the basic vocabulary of linguistic terms that can be applied to learning objects. This ontology defines the RDF syntax, and the OWL data type to be assigned to all OBAA [Vicari et al. 2010] and IEEE-LOM [IEEE 2002] metadata.

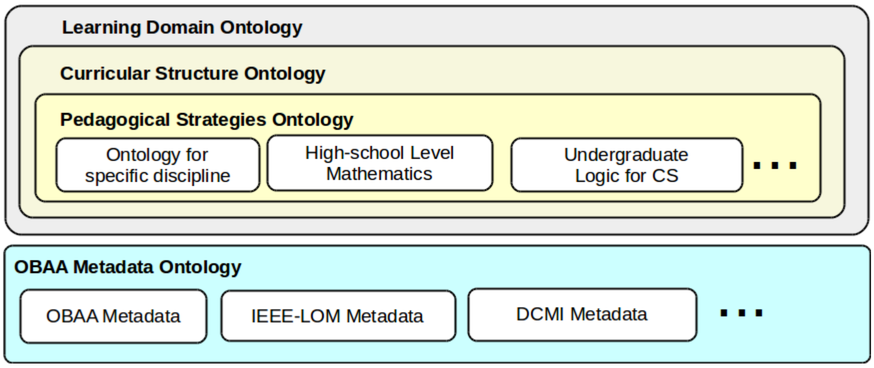

Figure 4: Ontological model of the semantic repository.

The knowledge represented in these ontologies allow semantic search engines to take into account the educational context where the LO is being used. In addition, authoring tools can use the knowledge available in educational ontologies, like the conceptual hierarchy and the relationships of learning domains, curricular structures and teaching strategies, to help users in their authoring activities.

These educational ontologies are structured in levels, starting from the most general level formed by the Learner Domain ontology, which defines high-level classes of learning domains, passing by the Curricular Structure ontology that define general concepts related to curriculum of courses and disciplines, and reaching to the Pedagogical Strategies ontology that defines overall concepts related to strategies and methods for teaching subjects. Ontologies for specific disciplines should include general educational ontologies (the Learner Domain, Curricular Structure and Pedagogical Strategies ontologies), and specialize the higher-level concepts, classes and relationships defined in these ontologies.

Educational ontologies are fundamentally distinct from the metadata ontology in the context of the MILOS reference architecture [Gluz et al. 2012]: educational ontologies identify concepts and terms related to educational contents, teaching methods and learning processes, while the metadata ontology represents the data types and possible values of LO metadata information. Concepts from educational ontologies could appear in the values of metadata elements from metadata ontology, but this relationship is not defined in any one of these ontologies. This relationship is established by a *mapping* of the terms and concepts from educational ontologies to the LO metadata elements and values of metadata ontology. The mapping process is helped by "*Obaa*" annotations, which are interpreted by the search algorithm as a mapping between the conceptual ontology and some corresponding metadata value in the metadata ontology. Note that the annotation of educational ontologies, even when done manually, does not need to be a tedious process. Only the higher levels of the class hierarchy need to be annotated. The search engine supports the inheritance of annotations, which ensures that the remaining levels of the hierarchy are automatically annotated.

Several processes are used to dynamically add new ontologies to the ontological model. If some educational ontology already follows the general classes and relationships currently used in the ontological model, then the ontology can be simply loaded in the semantic database by the *RDFDataManager* agent and recognized by *OWLReasoner* agent (see Figure 2). If this is not the case, then the ontology alignment operation provided by the *OntoAlign* agent (see Figure 2) can be used to align new educational ontologies to the ontological model. The alignment of ontologies may require a large computing power [Shvaiko and Euzenat 2011; David et al. 2010]. To minimize this problem, our approach was to employ heuristics derived from the structure of the application domain to reduce the number of steps in the comparing process. Thus, besides the use of standard text-based techniques supported by lexical vocabularies to align ontologies [Ehrig 2007; Euzenat and Shvaiko 2013], the *OntoAlign* agent also incorporates alignment heuristics in the form of annotations on the ontology. This agent extends the *AlignApi* API [David et al. 2010] to implement the alignment process. The similarity algorithm was modified to avoid comparison with all entities, enabling only the comparison with annotated elements.

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.

An example could illustrate the use of this annotation. In Brazil, the secondary education period is currently called "*Ensino Médio*", thus a class "*EnsinoMedio*" 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 "*Align*" annotation values "*Align.EnsinoSegundoGrau*" and "*Align.EnsinoSecundario*" associated to this class name could allow the alignment of this ontology to older educational ontologies.

5 Inferences

The semantic search engine used in *MSSearch* aims to retrieve learning objects from a repository that supports OWL (SPARQL) queries. Users need only to provide keywords, and then the search engine deduces the appropriate semantic context for the search. To this, the *MSSearch* engine correlates context information with aligned ontologies to create an appropriate SPARQL query. The search mechanism is implemented by *SemanticQuery* agent (see Figure 2). The Orengo algorithm [Orengo and Huyck 2001] is applied first to remove suffixes and stop words from the keywords. Then, the base of aligned ontologies is consulted to obtain all LO semantically related to the keywords. In this step, if "*Obaa*" mapping annotations are present, they are used to relate keywords to specific LO metadata. However, if no mapping annotation is found, the algorithm will correlate the keywords provided by user to target values of *descriptionIs*, and *keywordIs* metadata relationships.

A SPARQL query is then built combining the list of terms extracted from the aligned ontologies with relationships that identify learning objects metadata. This query is sent to *MetaQuery* agent (see Figure 2) and the resulting RDF triples are formatted in HTML and sorted in decreasing order, according to the relevancy. The evaluation of the relevance takes into account the number of occurrences found in the text of the metadata. A filter is applied to eliminate less relevant results.

The *Linnaeus* system uses a rule-based inference engine to generate the contents (values) of metadata. Examples of inference rules used by *Linnaeus* in natural language format are presented in Figure 5.

Examples of rules used to fill the general group of metadata:

```
IF Language is informed by user
THEN fill languageIs metadata with Language;
IF LearningDomain is informed by user
THEN search LearningDomain.Ontology;
IF LearningDomain.Ontology is defined AND
     LearningDomain.Topic is informed by user
THEN search LearningDomain.Topic.Description in
     LearningDomain.Ontology AND
THEN fill descriptionIs metadata with
     LearningDomain.Topic.Description;
```

Technical group metadata information can be filled based on the content files associated to the learning object. Some rules for this group are presented below:

```
IF ContentFile is informed by user
THEN fill specificSizeIs metadata with ContentFile.FileSize;
IF ContentFile is informed by user AND ContentFile.Type is video
THEN fill durationIs metadata with ContentFile.VideoTime;
```

Figure 5: Examples of Linnaeus inference rules.

This engine extracts the knowledge represented in the hierarchy of concepts and axioms of the educational ontologies to infer information that will be stored as metadata values. The inference engine produce new metadata values, combining this ontological knowledge with information provided by users.

6 Interfaces

MetaQuery agent accepts SPARQL queries and *MetaUpdate* agent accepts update commands with SPARQL Update syntax. In turn, *MetaLoad* agent implements bulk transfers of metadata using RDF triples. The semantic database can be accessed by and agent-based FIPA/JADE interface through the *fipa-request* protocol. Two additional interfaces are also implemented: an OAI-PMH interface and a RESTfull web services interface that supports queries and updates using SPARQL commands.

The *Linnaeus* provides distinct modes of user interfaces that adapt to the knowledge that the designer has in relation to the technical aspects of LO. The easy mode is for users without technical knowledge about LO or metadata. The expert mode interface is similar to *DSpace* interface, allowing users with good knowledge about LO metadata to have access to all metadata available in OBAA [Vicari et al. 2010]. In the intermediate mode interface the user interacts with wizards, but can complement the remaining of metadata in any point of the interaction. The easy mode interface is fully guided by wizards that interact with the user to extract the information necessary to infer the remaining metadata. The metadata generated by wizards is presented to the user in the end of the interactions for a revision.

The end user web interface of *MSSearch* is implemented by the *WebQueryInterface* facility (see Figure 2). To use this interface, end users do not need advanced knowledge about LO; they only need to enter the keywords to search. The search returns a list of LO, with information about their metadata structured by OBAA standard. The *MSSearch* system also allows the users to select the most relevant results and also to limit the amount of results to be presented.

7 Experiments

We have conducted several experiments to evaluate *Linnaeus* and *MSSearch* systems. The test platform was formed by a cluster composed by eight computers isolated from the Internet. Each computer of the cluster has the same configuration and runs Linux as the operating system, the JADE platform as communication middleware for agents and Glassfish/Apache Tomcat as web application providers. To provide external access to the systems running on the cluster, the cluster is connected to a dual port gateway computer. This gateway also runs the Apache web server that provide the interface to the cluster's systems. The experimental systems can be accessed by the project's portal page *http://obaa.unisinos.br/*. The next subsections present these experiments and their results.

7.1 Usability of the easy mode interface

The goal of this experiment was to validate usability of the easy mode interface of *Linnaeus*. An objective metric, which considers the number of user interactions necessary to generate the set of metadata for a given LO, was used for validation purposes. In this metric, a smaller number of iterations corresponds to a best index of usability. The number of iterations of the *Linnaeus* system was compared with the number of interactions needed to create the same metadata via a metadata editing interface without wizard support, indicating how much work *Linnaeus* has saved for the user, if the user have decided, instead, to use a tool like *DSpace*.

For this experiment were selected 10 learning objects from BIOE [MEC 2014], which is a well-established repository used by various educational institutions. The learning domain chosen was the high school Mathematics due to the available support of a teacher with more than five years of experience in this domain. This teacher does not interfere with the usability experiment, but was instrumental in the assessment of the quality of the metadata generated by *Linnaeus*. The content of these 10 objects were presented to a user with some experience in the use of tools to generate LO metadata. However, the test subject was not a teacher and had no previous experience with this learning domain. Table 1 presents the results of the usability experiment, showing that the number of iterations to request information from the developer was low compared to the quantity of metadata generated. These results suggest that the *Linnaeus* needs a low number of interactions to fill a significant amount of metadata. The average of 27% of interactions obtained by the easy mode interface is quite significant, resulting in a low number of iterations when compared with the equivalent number of interactions necessary with expert mode interface or with *DSpace*.

Learning Object	# of interactions in easy mode	# of metadata generated	% of interactions from expert mode
The_height_of_the_tree	11	47	23%
Creation_of_Logaritms_Part_I	13	51	25%
Creation_of_Logaritms_Part_II	13	51	25%
Bets_on_the_clock	11	43	25%
Building_curves_hyperbola	15	48	31%
Affine_function	14	45	31%
Geometry	15	53	28%
3x+1_part1	14	54	25%
3x+1_part2	14	54	25%
Cylinder model	17	52	32%
TOTAL	**137**	**498**	**---**
AVERAGE	**13.7**	**49.8**	**27%**

Table 1: Results of Usability Experiment.

7.2 Quality of generated metadata

This experiment used qualitative and quantitative analysis to compare the metadata generated in the usability experiment against the metadata originally stored in the BIOE repository. A teacher of High School Mathematics with five years of experience made the qualitative analysis. Analysis results show that the quality of metadata generated by the system depends on: (1) the quality of learning domain ontology; (2) the quality of the metadata inference rules; (3) the quality of answers produced by the user. The results indicated that, in the case of technical and educational metadata, *Linnaeus* was able to correctly deduce the metadata related to compatibility with operating systems, type of interactions and modes of use in classroom, as well as some aspects of accessibility.

Metadata with textual descriptions were obtained by inferences over the learning domain ontology combined with input data provided by the user. The text generated by *Linnaeus* usually introduced the general context of the teaching topic associated with the LO, but do not include details about its use or application. This can be highlighted comparing the text "*Objeto para Matematica Ensino Medio, com Conteudo Matematica Ensino Medio em Sistemas Lineares na área de Equacoes Lineares*" (freely translated as "*Object for High School Mathematics with High School Mathematics Contents about Linear Systems in Linear Equations area*"), generated by *Linnaeus*, with the text of the equivalent metadata in BIOE "*O programa apresenta a conjectura do problema 3x+1 e discute algumas curiosidades em torno dela para mostrar que, mesmo parecendo verdade, os matemáticos só consideram verdadeiro aquilo que é provado lógica e matematicamente*" (freely translated as "*The program presents the conjecture of 3x+1 problem and discusses some curiosities about this conjecture to show that, even seeming to be true, mathematicians only consider true something, which is mathematically and logically proved*"). The text produced by *Linnaeus* correctly defines the general context of this LO, but does not show any specific detail about the intended use of the LO. The text stored in BIOE is almost the opposite. The general context is limited, but there are several details that the teacher wants to associate with this object. Considering that the designer can change the generated text, this does not seems to be a problem.

The quantitative analysis compared the quantity of metadata elements generated by *Linnaeus* with the quantity of metadata elements stored on BIOE. Table 2 shows the results of this comparison.

Learning Object	**# metadata gen. by Linnaeus**	**Linnaeus X BIOE(%)**
The_height_of_the_tree	18	86%
Creation_of_Logaritms_Part_I	17	81%
Creation_of_Logaritms_Part_II	17	81%
Bets_on_the_clock	19	90%
Building_curves_hyperbola	16	76%
Affine_function	19	90%
Geometry	18	86%
3x+1_part1	20	95%
3x+1_part2	20	95%
Cylinder model	16	76%
TOTAL	**180**	**---**
AVERAGE	**18.0**	**85.7%**

Table 2: Quantitative analysis of metadata generated by Linnaeus.

Note that BIOE only uses DCMI metadata. The 10 selected LO used 21 distinct DCMI metadata elements. The results in Table 2 show that *Linnaeus* can generate from 76% to 95% of the metadata elements stored in BIOE. Results from Table 1 show that the easy mode interface requires an average of 13.7 interactions to generate LO metadata. Results from Table 2 show that easy mode interface is able to generate an average of 18 metadata from BIOE's set of metadata for the LO used in the experiment. Thus, on average 3 metadata must be added to complete the set of BIOE's metadata, requiring 3 more interactions outside of easy mode interface, besides the average 13.7 interactions already made. This is a 20% average reduction in the number of interactions necessary in the expert mode interface (or *DSpace* interface) to generate the same set of 21 metadata. Note that this is not the only gain from easy mode interface. This interface does not generate only the set of 21 metadata originally stored in BIOE for these LO, but can generate several other information (see Table 1).

7.3 Performance Evaluation

The goal of performance evaluation was to measure the execution time of operations to load and query LO in the semantic database, when an increasing quantity of LO is stored in the database. An external LO repository was selected to be the source of metadata information. The repository chosen was BIOE [MEC 2014] that at the time of the tests contained approximately 17,600 learning objects. Then the semantic database of *MSSearch* was populated with more than 11.000 learning objects from this repository. This included all BIOE objects that contained educational materials related to Mathematics.

Table 3 shows the amount of learning objects loaded (# of LO) along with their respective load times. Moreover, the table shows the number of RDF triples needed to store LO metadata as well as the average number of triples loaded by second. The load time remained linearly proportional to the number of objects, indicating a possible maximum complexity of order $O(n)$ for this process.

# of LO	Time (sec.)	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

Table 3: Load operation performance experiment.

Another experiment executed a complex SPARQL query, in order to recover the identifier, keywords, description, location, hardware platform and title of all learning objects stored in the semantic repository, sorted by title. The time spent for this query execution can be seen in Table 4. According to the results, 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. 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 good results for queries. The results of the performance experiments suggest that semantic repository technology supports a relatively large amount of metadata without compromising performance.

# 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

Table 4: Query operation performance experiment

7.4 User Perception Evaluation

The goal of this experiment was to evaluate the quality of query results returned by *MSSearch* when compared to results returned by BIOE [MEC 2014] search engine for similar queries. The learning domain was High School Mathematics. The experiment was conducted by four teachers with post-graduation in Mathematics and 10 years of experience teaching Mathematics. Teachers could create terms for the search of their choice (eg "polynomials") and then submit to *MSSearch* and BIOE. Each teacher made two separate searches, with two different keywords. Based on query results, teachers filled an assessment form. Table 5 summarizes the results of the experiment.

MSSearch consistently returned best query results than BIOE. In particular, the relevancy of results returned by *MSSearch* (third item) and the ordering of these results (fifth item) were considered by users highly satisfactory for *MSSearch*, when compared with BIOE: 62.5% of answers produced by *MSSearch* were considered relevant and well-ordered against only 12.5% of the answers produced by BIOE for the same searches.

8 Conclusions

This article shows how to combine the state of the art in agent, ontology and LO metadata technologies to build the basic services of a LO semantic repository. The article addresses how to integrate ontology and agent engineering to build this application.

The *Linnaeus* system provides the metadata creation and editing services of the semantic repository. This system was designed to reduce significantly the amount of work necessary to fill the metadata for a particular LO. Usability experiments showed that a significant reduction in the work is possible. Quality experiments highlighted the need for good quality learning domain ontologies to generate metadata of quality, but also showed evidences that *Linnaeus* can succeed in attain this goal if this condition is satisfied.

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 was 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%

Table 5: Results from User Perception Experiment.

The use of ontology alignment techniques in *MSSearch* system allows integrating ontologies from different sources, resulting in greater coverage of knowledge and information sharing in the search engine. The direct use of alignment techniques sometimes does not present satisfactory results, mainly when there are big differences in the structure and vocabularies of the ontologies. To solve this issue, *MSSearch* implemented and tested a new ontology alignment mechanism based on OWL annotations that enabled the system to locate learning objects based on the semantic context. The initial experiment on the quality of query results produced by *MSSearch* was positively evaluated by users. However, more tests should be executed for a better assessment of this quality. On the other way, the good results of the performance experiment and the volume of data processes in this experiment, indicate that *MSSearch* has good possibilities to become a fully production system, being able to provide the core to search facilities in a semantic LO repository. These results provide evidences of the feasibility of integrated use of *Linnaeus* and *MSSearch* to provide the core services of a semantic repository of learning objects. They also show that this kind of solution for a LO repository can bring good results for the users.

In this work we applied alignment techniques only to educational domain ontologies. However, we intend to apply these techniques with ontologies from other areas. This is a relevant new line of research based on the results presented in this article. Another new possibility of research is to explore automatic ontology annotation mechanisms. *MSSearch* results show that the use of annotations can be an efficient tool to aid the aligning of OWL ontologies. However, the annotation process is done manually, requiring some prior knowledge of the structure of the ontologies to be aligned. So, a relevant research approach is to discover mechanisms and techniques that allow these annotations to be made with minimum interference from users, preferably automatically.

References

[ADL 2004] Advanced Distributed Learning (ADL) - SCORM 4th Edition Specification, 2004, http://adlnet.gov/adl-research/scorm/scorm-2004-4th-edition.

[Arias et al. 2009] Arias, F., Moreno, J. Ovalle, D.: Integration Model of E-Learning based on Pedagogical Software Agents and Collaborative Learning Environments; In Proc IFIP World Conference on Computers in Educaton (WCCE), 2009, 1–10.

[Baader et al. 2007] Baader, F., Horrocks, I., Sattler, U.: Chapter 3: Description Logics; in Harmelen, F., Lifschitz, V., Porter, B. (eds), Handbook of Knowledge Representation, Elsevier, 2007.

[Ball and Tenney 2008] Ball, S., Tenney, J.: Xerte - A User-Friendly Tool for Creating Accessible Learning Objects; in Proc. 11th Int. Conference on Computers Helping People with Special Needs, Lecture Notes in Computer Science, vol. 5105, Springer, 2008, 291–294.

[Berners-Lee et al. 2001] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web; Scientific American Magazine, 29-37, May 2001.

[Bittencourt et al. 2009] Bittencourt, I., Costa, E., Silva, M., Soares, E.: A computational model for developing semantic web-based educational systems; Knowledge-Based Systems, 22, 4, 302–315, 2009.

[Bittencourt et al. 2008] Bittencourt, I., Costa, E.B., Neto, B.F.S., Menezes, J.G.M., Melo, J.S.S., Fernada, E., Silva, A.P, Brasil, L.M.: Constructing Intelligent Tutoring Systems Based on a Multiagent Architecture; in Vicari, R.M., Jaques, P.A., Verdin, R. (eds.), Agent-Based Tutoring Systems by Cognitive and Affective Modeling, IGI Press, 28–59,2008.

[Bittencourt et al. 2008] Bittencourt, I., Isotani, S., Costa, E., Mizoguchi, R.: Research Directions on Semantic Web and Education; Scientia, 19, 1, 60–67, 2008.

[Clarksia 2014] Clarksia: Pellet System, 2014, http://clarkparsia.com/pellet.

[Cohen et al. 2003] Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks; in Proc. 18th Int. Joint Conference on Artificial Intelligence (IJICAI) - Workshop on Information Integration on the Web, 2003, 73–78.

[D'Aquin et al. 2013] D'Aquin, M., Adamou, A., Dietze, S.: Assessing the Educational Linked Data Landscape; in Proc. 5th Annual ACM Web Science Conference (WebSci), 2013, 43–46.

[David et al. 2010] David, J., Euzenat, J., Scharffe, F., Santos, C.T.: The Alignment API 4.0; Journal Semantic Web, 2, 1, 3–10, 2010.

[Dietze et al. 2007] Dietze, S., Gugliotta, A., Domingue, J.: A Semantic Web Service oriented Framework for adaptive Learning Environments; in Proc. 4th European Semantic Web Conference (ESWC), Lecture Notes in Computer Science, vol. 4519, Springer, 2007, 701–715.

[Dietze et al. 2009] Dietze, S., Gugliotta, A., Domingue, J.: Supporting Interoperability and Context-Awareness in E-Learning through Situation-driven Learning Processes; International Journal of Distance Education Technologies, 7, 2, 20–43, 2009.

[Ehrig 2007] Ehrig, M.: Ontology Alignment: Bridging the Semantic Gap, Springer, 2007.

[Euzenat and Shvaiko 2013] Euzenat, J., Shvaiko, P.: Ontology Matching, Springer, 2013.

[Frasson et al. 2005] Frasson, C., Chaffar, S., Abdel Razek, M., Ochs, M.: Emotion Recognition Agent for On-Line Learning Systems; in Proc. 18th Int. FLAIRS Conference, 2005,1-2.

[Freeloms 2014] Freeloms Systems, 2014, http://www.freeloms.org.

[Giraffa and Vicari 1998] Giraffa, L., Vicari, R.M.: The Use of Agents Techniques on Intelligent Tutoring Systems; in Proc. 18th International Conference of the Chilean Computer Science Society (SCCC), 1998, 12–14.

[Gluz and Vicari 2012] Gluz, J.C., Vicari, R.M.: An OWL Ontology for IEEE-LOM and OBAA Metadata; in Proc. 11th International Conference on Intelligent Tutoring Systems (ITS), Lecture Notes in Computer Science, vol. 7315, Springer, 2008, 691–693.

[Gluz et al. 2012] Gluz, J.C., Vicari, R.M., Passerino, L.M.: An Agent-Based Infrastructure for the Support of Learning Objects Life-Cycle; in Proc. 11th Int. Conf. on Intelligent Tutoring Systems (ITS), Lecture Notes in Computer Science, vol. 7315, Springer, 2008, 696–698.

[G.-Barbone and A.-Rifon 2010] Gonzalez-Barbone, V., Anido-Rifon, L.: From SCORM to Common Cartridge: A step forward; Computers & Education, 54, 1, 88–102, 2010.

[Gruber 1993] Gruber, T.R.: A Translation Approach to Portable Ontologies Specifications; Knowledge Acquisition, 5, 2, 199–220, 1993.

[Gunter 2009] Gunter, D.W.: Semantic search; Bulletin of the American Society for Information Science and Technology, 36, 1, 36–37, 2009.

[Gürer 1998] Gürer, D.: The Use of Distributed Agents in Intelligent Tutoring; in Proc 2nd ITS Workshop on Pedagogical Agents, 1998, 20–25.

[Hayashi et al. 2009] Hayashi, Y., Isotani, S., Bourdeau, J., Mizoguchi, R.: Toward a Learning/Instruction Process Model for Facilitating the Instructional Design Cycle; in Tatnall, A., Jones, A. (eds.), Education and Technology for a Better World, Springer, 138–147, 2009.

[IEEE 2002] IEEE Standard for Learning Object Metadata-1484.12.1-2002, 2002, http//dx.doi.org/10.1109/IEEESTD.2002.94128.

[IMS 2014] IMS Learner Information Packaging Information Model Specification, 2014, http://www.imsglobal.org/profiles/lipinfo01.html.

[Isotani et al. 2009] Isotani, S., Mizoguchi, R., Bittencourt, I., Costa, E.: State of the Art in Semantic Web and Web 2.0: Potentialities and Tendencies of the New Generation of Educational Environments on Internet; Brazilian Journal on Informatics in Education, 17, 1, 30-42, 2009.

[Kunze and Baker 2007] Kunze, J., Baker, T.: The Dublin Core Metadata Element Set, 2007, http://www.ietf.org/rfc/rfc5013.txt.

[MEC 2014] MEC: BIOE - International Database of Educational Objects - Ministério da Educação e Cultura (MEC), 2014, http://objetoseducacionais2.mec.gov.br/?locale=en.

[Mizoguchi et al. 2007] Mizoguchi, R., Hayashi, Y., Bourdeau, J.: Inside Theory-Aware and Standards-Compliant Authoring System; in Proc 5th International Workshop on Ontologies and Semantic Web for E-Learning (SWEL), 2007, 1–18.

[NISO 2004] National Information Standards Organization (NISO), Understanding Metadata, 2004, http://www.niso.org/publications/press/UnderstandingMetadata.pdf.

[Ochs et al. 2011] Ochs, C., Tian, T., Geller, J., Chun, S.A.: Google Knows Who is Famous Today - Building an Ontology from Search Engine Knowledge and DBpedia; in Proc 5th IEEE International Conference on Semantic Computing (ICSC), 2011, 320–327.

[Orengo and Huyck 2001] Orengo, V. M., Huyck, C.: Stemming Algorithm for Portuguese Language; in Proc 8th International Symposium on String Processing and Information Retrieval (SPIRE), 2001, 186–193.

[Polsani 2003] Polsani, P. R.: Use and Abuse of Reusable Learning Objects; Journal of Digital Information, 3, 4, 1–10, 2003.

[Ramachandran and Sujatha 2011] Ramachandran, A., Sujatha, R.: Semantic Search Engine: A Survey; International Journal of Computer Technology and Applications, 2, 6, 1806–1811, 2011.

[Schreiner et al. 2012] Schreiner, P., Rossi, L., Nunes, M., Vicari, R.M.: FEB + OBAA: Cataloging and retrieving Learning Objects effectively; In Proc Workshop on Applications to Provide Learning and Teaching Support (CBIE), 2012, 1–5.

[Shvaiko and Euzenat 2011] Shvaiko, P., Euzenat, J.: Ontology Matching: State of the Art and Future Challenges; IEEE Transact. on Knowledge and Data Engineering, 25, 1, 158–176, 2011.

[Silveira et al. 2004] Silveira, R., Gomes, R. E., Pinto, H.V., Vicari, R. M.: Intelligent learning objects: An agent based approach of learning objects; in Proc 7th International Conference on Intelligent Tutoring Systems (ITS), 2004, 886–888.

[Sklar and Richards 2006] Sklar, S., Richards, D.: The use of agents in human learning systems; in Proc 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2006, 767–774.

[Smith et al. 2003] Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., Walker, J.H.: DSpace: An Open Source Dynamic Digital Repository; D-LIB Magazine. 9, 1, 2003.

[Stoilescu 2008] Stoilescu, D.: Modalities of Using Learning Objects for Intelligent Agents in Learning; Interdisciplinary Journal of E-Learning and Learning Objects, 4, 2008.

[Teixeira et al. 2012] Teixeira, T.N., Campos, F., Braga, R., Santos, N., Mattos, E.: BROAD Project: Semantic Search and Application of Learning Objects; IEEE Technology and Engineering Education, 7, 3, 23–32, 2012.

[Vicari et al. 2010] Vicari, R.M., Gluz, J.C., Passerino, L., Santos, E., Primo, T., Rossi, L., Bordignon, A., Behar, P., Filho, R., Roesler, V.: The OBAA Proposal for Learning Objects Supported by Agents; in Proc 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2010, 1–4.

[W3C 2012] W3C: OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (Second Edition), 2012, https://www.w3.org/TR/owl2-syntax.

[Weiss 2013] Weiss, G.: Multiagent Systems, MIT Press, 2013.

[Wiley 2002] Wiley, D. A.: Connecting Learning Objects to Instructional Design Theory: A Definition, A Metaphor, and A Taxonomy; in Wiley, D. A. (ed.), The Instructional Use of Learning Objects, Association for Educational Communications and Technology, 3–23, 2002.

[Wooldridge 2009] Wooldridge, M.: An Introduction to MultiAgent Systems, John Wiley & Sons, 2009.

[Yu et al. 2012] Yu, H., Pedrincaci, C., Dietze, S., Domingue, J.: Using Linked Data to Annotate and Search Educational Video Resources for Supporting Distance Learning; IEEE Transactions on Learning Technology, 5, 2, 130–142, 2012.