An Assistant for Loading Learning Object Metadata: An Ontology Based Approach

Ana Casali
Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, and Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS), Rosario, Argentina

acasali@fceia.unr.edu.ar

Claudia Deco, Agustín Romano, and Guillermo Tomé
Facultad de Ciencias Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Rosario, Argentina

deco@fceia.unr.edu.ar romano@fceia.unr.edu.ar
guillermo@fceia.unr.edu.ar

Abstract

In the last years, the development of different Repositories of Learning Objects has been increased. Users can retrieve these resources for reuse and personalization through searches in web repositories. The importance of high quality metadata is key for a successful retrieval. Learning Objects are described with metadata usually in the standard IEEE LOM. We have designed and implemented a Learning Object Metadata ontology (LOM ontology) that establishes an intermediate layer offering a shared vocabulary that allows specifying restrictions and gives a common semantics for any application which uses Learning Objects metadata. Thus, every change in the LOM ontology will be reflected in the different applications that use this ontology with no need to modify their code. In this work, as a proof of concept, we present an assistant prototype to help users to load these Objects in repositories. This prototype automatically extracts, restricts and validates the Learning Objects metadata using the LOM ontology.

Keywords: Learning Object, Learning Object Metadata, Ontologies, Metadata Extraction.

Introduction

Nowadays the web is one of the most important sources of educational material where students and teachers have a large amount of information at their disposal. For this information retrieval process, people use search engines which, unfortunately in many cases, do not return the desired information or return too many web pages. *Learning objects* are intended to help with the storage, classification, and reuse of educational resources. A Learning Object (LO) is any digital resource that can be reused to support learning

Editor: Janice Whatley

(Wiley, 2002). LOs can be used by a student who wants to learn a subject or may be used by a teacher who wants to prepare materials for his/her class. LOs are described with metadata usually in the standard IEEE LOM (Learning Object Metadata: http://ltsc.ieee.org/wg12) and they are stored in different repositories. Examples of such repositories are FLOR (www.laclo.org), Ariadne (www.ariadne-eu.org) and OER Commons (www.oercommons.org). There other standards such as the Dublin Core Metadata Initiative, or DCMI (dublincore.org), which is an open organization supporting innovation in metadata design and best practices across the metadata ecology. The Dublin Core Metadata is not focused on educational metadata but it maintains a number of formal and informal relationships with different standards bodies and it is widely used. Other metadata standards related with educational metadata are: IMS Learning Object Metadata IMS LOM (http://www.imsglobal.org/metadata), Canadian Core Learning Resource Metadata Protocol CanCor (http://www.cancore.ca) and UK Learning Object Metadata Core UK LOM (http://www.cetis.ac.uk/profiles/uklomcore/uklomcore_v0p2_may04.doc). We decided to use IEEE LOM because it is one of the most used in the community of learning objects.

Users can retrieve LOs through searches in web repositories. Thus, the importance of high quality metadata is key for a successful retrieval. Recommender systems, based on metadata information and user profiles, arise to help people to retrieve the resources that are most appropriate to user's needs and preferences (Casali, Deco, Bender, & Gerling, 2012). Nevertheless, preparing learning resources with suitable metadata is labor-intensive and, consequently, there is a lack of quality information in these metadata. Michael Sonntag (2004) analyzes the importance of metadata for learning objects, as these resources may be reused often and possibly in different contexts. Some problems that he points are the lack of metadata, the diversity of standards, and the search engine support. Thus, the development of automatic/semi-automatic extraction systems seems to be a very important step towards solving this problem. Up to now, there are not many works on automatic metadata extraction. Each of the existent tools for metadata extraction has its own objectives, architecture and uses different techniques. For instance, some extractors systems can be seen in Alfano, Lenzitti, and Visalli (2007), Li, Dorai, and Farrell (2005), Motz et al. (2009), Pire, Espinase, Casali, and Deco (2011), and Tang (2007).

In addition, there exist some metadata editors. For example, the Eureka project (http://eureka.ntic.org) is an initiative that provides a collective catalog of teaching and learning resources gathered by various organizations involved in the production of ITC educational resources. Eureka's shell is based on open source code. The data can be federated with other repositories built on a LOM application profile. Another project is Advanced Learning Object Hub Application ALOHA 1.3 (http://aloha.netera.ca/). The Learning Commons at the University of Calgary has done extensive research and development of tools and techniques to deal with workflow productivity issues related to content repurposing and metadata indexing. ALOHA's flexible interface is friendly for amateur users and customizable for the professional indexer. One of the principal goals is to work with any metadata standard including DublinCore, IMS, SCORM, CanCore, any emerging standard or any custom format. All that is required is a valid XML-schema and the output of ALOHA will be an XML file that conforms 100% to the applied standard.

Also, there exists previous works with metadata ontologies. In Sánchez-Alonso, Sicilia, and Pareja (2007), the authors introduce a mapping of the standard IEEE LOM in the ontology language Web Service Modelling Language (WSML). The objective is both to provide a basis for translating existing IEEE LOM metadata records to WSML and to serve as a basic learning object ontology from which richer ontological representations can be devised. This paper describes such mapping, formally engineered as an ontology called LOM2WSML. Another related work is Gluz and Vicari (2012), which introduces the OBAA metadata ontology which is an OWL ontology created to represent all the metadata from IEEE LOM standard and OBAA metadata proposal.

The OBAA metadata is an extension of IEEE LOM including support for adaptability and interoperability of LO on digital platforms, compatibility with international standards, accessibility and independence, and flexibility of the technology.

On the other side, the software for building open digital repositories like DSpace (http://www.dspace.org) usually specifies the metadata and their restrictions for a document collection in an implicit way, during the loading process. Thus, defining an intermediate layer for this specification seems to be a good solution to separate the metadata specification from the different processes that may use it and particularly, from the loading process.

The goal of this work is to propose an application supported by an ontology, which models LOM standard, in order to help users to load learning objects metadata. The proposed LOM ontology establishes an intermediate layer offering a shared vocabulary that specifies restrictions and gives a common semantics for any application which uses learning objects metadata. So, every change in the LOM ontology will be reflected in the different applications that use this ontology with no need to modify their code. In this work, we present an assistant prototype to help users to load LOs in repositories. This prototype automatically extracts, restricts and validates LO metadata using the LOM ontology.

This paper is structured as follows: in the next we present some basic concepts, in the third the design and implementation of the LOM ontology is shown, and in the fourth section an assistant to load metadata of learning object based on this ontology is proposed. Finally, we discuss conclusions and future work.

Preliminaries

The standard IEEE LOM is a data model, usually encoded in XML, used to describe resources used to support learning. The purpose of learning objects metadata is to support the reusability of these objects, to aid discoverability, and to facilitate their interoperability. IEEE LOM comprises a hierarchy of elements. At the first level, there are nine categories: General, Lifecycle, Meta-metadata, Technical, Educational, Rights, Relation, Annotation, and Classification. In order to specify this standard we use ontologies. The General category groups the general information that describes the learning object as a whole. The Lifecycle category groups the features related to the history and current state of this learning object and those who have affected this learning object during its evolution. The Meta-Metadata category groups information about the metadata instance itself (rather than the learning object that the metadata instance describes). The Technical category groups the technical requirements and technical characteristics of the learning object. The Educational category groups the educational and pedagogic characteristics of the LO. The Rights category groups the intellectual property rights and conditions of use for the LO. The Relation category groups features that define the relationship between the learning object and other related learning objects. The Annotation category provides comments on the educational use of the learning object and provides information on when and by whom the comments were created. The Classification category describes this learning object in relation to a particular classification system. Figure 1 shows these categories.

Learning Objects Metadata are described by *Data elements* and are grouped into these nine *categories*. The LOM data model is a hierarchy of data elements, including aggregate data elements and simple data elements. In LOM, only leaf nodes have individual values defined through their associated value space and datatype. For each data element, LOM defines the following: *name* (the name by which the data element is referenced); *explanation* (the definition of the data element); *size* (the number of values allowed); *order* (whether the order of the values is significant), and an *example* (an illustrative example). For simple data elements, LOM also defines value space (the set of allowed values for the data element) and datatype (indicates whether the values are LangString, DateTime, Duration, Vocabulary, CharacterString or Undefined). Particularly, *LangString* is a datatype that represents one or more character strings. A LangString value may include multiple semantically equivalent character strings, such as translations or alternative descriptions, and they are represented by the pair (Language, String). Besides, *Vocabularies* are defined for some data elements. A *vocabulary* is a recommended list of appropriate values.

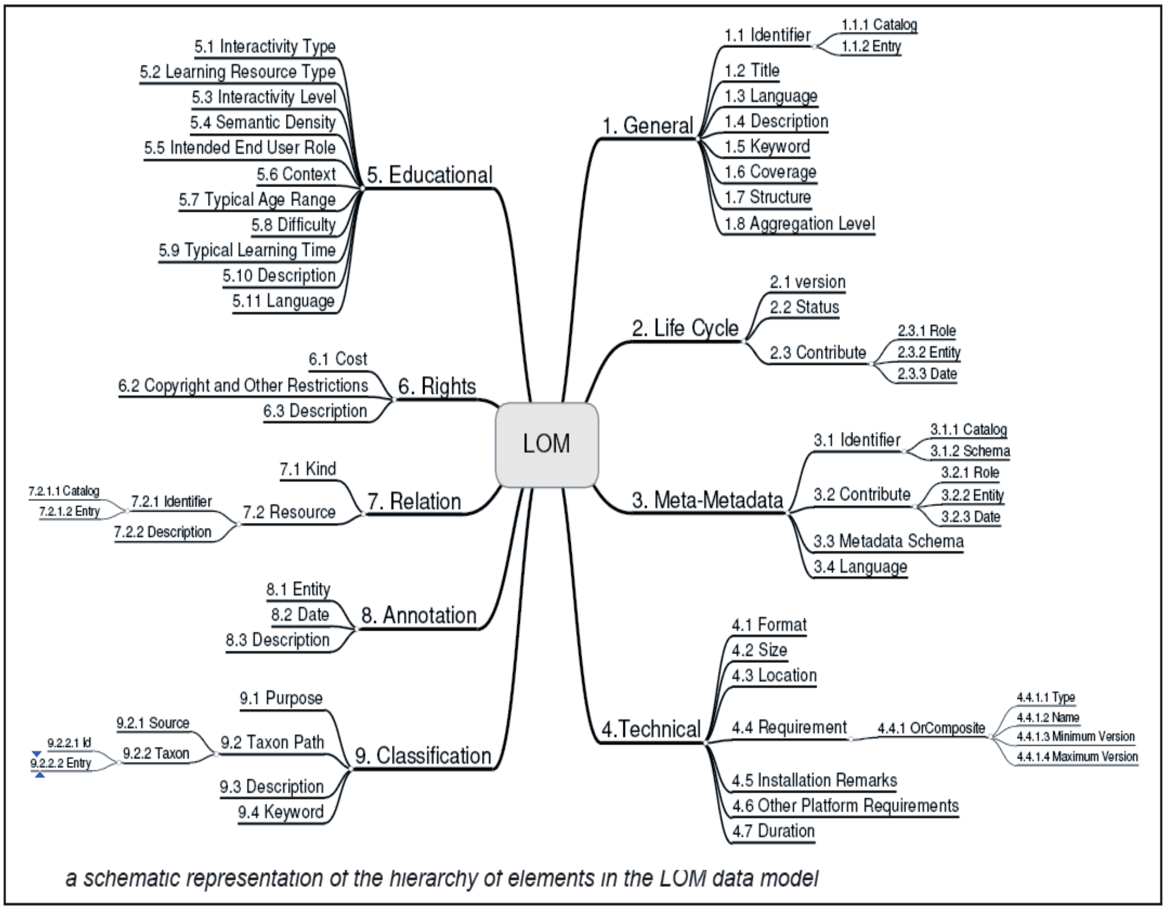


Figure 1: A schematic representation of the hierarchy of elements in the LOM data model (based on http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html)

In some instances, a data element contains a list of values, rather than a single value. This list is of one of the following kinds: ordered (the order of the values in the list is significant, e.g., in a list of authors of a publication, the first author is often considered the more important one) or unordered (the order of the values in the list bears no meaning). In LOM, smallest permitted maximum values are defined for *aggregate data elements* (an application may impose a maximum on the number of entries it processes for the value of that data element, but that maximum shall not be lower than the smallest permitted maximum value) and *data elements with datatype CharacterString or LangString* (all applications that process LOM instances shall process at least that length for the CharacterString value of that data element. In other words, an application may impose a maximum on the number of characters it processes for the CharacterString value of that data element, but that maximum shall not be lower than the smallest permitted maximum value for the datatype of the data element). In this work we propose an ontology to specify LOM standard representing all these elements and restrictions.

An ontology (Gruber, 1995) is an explicit specification of a conceptualization. Common components of ontologies include: Classes (sets, collections, concepts, types of objects, or kinds of things), Individuals (instances or objects), Attributes (aspects, properties, features, characteristics, or parameters that objects and classes can have), Relations (ways in which classes and individuals can be related to one another), Restrictions (formally stated descriptions of what must be true in order for some assertion to be accepted as input), and Axioms (assertions, including rules, in a logical form that together comprise the overall theory that the ontology describes in its domain of application). Ontologies are commonly encoded using ontology languages. The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between them. The ontology proposed in this work is modeled using OWL (http://www.w3.org/TR/owl-overview).

In this paper we propose to give semantic to LO metadata by using an ontology modeling the LOM structure. Moreover, as a proof of concept of the benefits of applications based on ontologies, we present an assistant for automatic/semi-automatic loading of metadata fields.

LOM Ontology

The Ontology to represent the standard IEEE LOM was designed following some basic steps. First, we map each item of this standard into ontology concepts. To design the proposed ontology, we defined the Classes, Categories properties and Metadata properties, and also, we established the necessary restrictions. Table 1 shows how we represent each main concept of the standard with ontology concepts and below we explain the more relevant ontology elements.

Table 1: Mapping between some LOM elements and ontology concepts

Standard IEEE LOM	LOM Ontology
Category/subcategory	Class
Metadata	Property
Size	Restriction
Datatype:Vocabulary	Data Property with enumerated range
Datatype: LangString	Class with properties language and string

Classes

- *LOM:* we defined a class that represents the learning object itself. This class has a superclass for each category. We call this class LOM.

- *Categories*: each class corresponds to a category of IEEE LOM metadata. Classes of this type have a superclass for each metadata that exists in the corresponding category in the standard. Some classes of this type are: General, Lifecycle, Meta-metadata, among others.

- *Subcategories:* correspond to subcategories of a given category.

- *Metadata:* each class corresponds to complex types of metadata that cannot be represented as a datatype. The clearest example of this type of class is the class LangString.

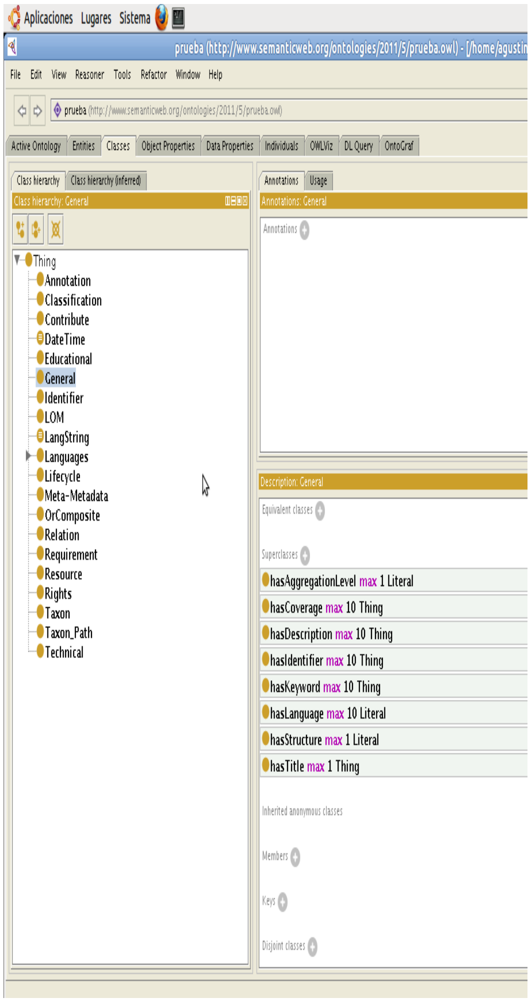
Properties

- *Categories Properties:* this type of properties is always Object Properties. They have as domain the class LOM and their range is any standard category. An example of such properties is *hasGeneral* which has as domain the LOM class and as range the General class. All superclasses of the LOM class are determined by properties of this type.

- *Metadata Properties*: can be Data Properties or Object Properties. In the case of the Data Properties, the domain is a Class Category or Subcategory, following the previous classification, and as range they have basic types (e.g., *int*, *string*, etc.). Such properties will be used to link to a category (or subcategory) with a single metadata. An example of this type of property is *hasEntry* which domain is the General class and as range the *string* type and represents the metadata Entry of the General category of the Standard.
- *Object Properties:* the domain is a Class Category and its range is a Subcategory Class or a Metadata Class. Such properties will be used to link to a category with a subcategory, or category with a complex type metadata (as LangString). An example of this type of property is *hasKeyword* which is a property with domain in the General class and range in the LangString class. This represents the Keyword metadata for the General category.

Restrictions

We model some restrictions for metadata fields using Properties. In particular, for each metadata, restrictions in the size of the field were defined. For example, for the *size* field, the standard specifies that the metadata *Keyword* of the General category can have at most 10 keywords. To reflect this in our ontology we use the restriction *max* on the *hasKeyword* property of the General class. Because of this, the property "hasKeyword max 10" is among the superclasses of the General class.



Thing
Annotation
Classification
Contribute
DateTime
Educational
General
Identifier
LOM
LangString
Languages
Lifecycle
Meta-Metadata
OrComposite
Relation
Requirement
Resource
Rights
Taxon
Taxon_Path
Technical

hasAggregationLevel max 1 Literal
hasCoverage max 10 Thing
hasDescription max 10 Thing
hasIdentifier max 10 Thing
hasKeyword max 10 Thing
hasLanguage max 10 Literal
hasStructure max 1 Literal
hasTitle max 1 Thing

Figure 2: LOM Ontology in Protégé

This ontology was developed in OWL using Protégé Editor (http://protege.stanford.edu) and the principal elements in the ontology can be seen in Figure 2.

An Assistant to upload Learning Objects

We present an assistant as a proof of concept of how an application supported by an ontology, which models IEEE LOM standard, establishes an intermediate layer offering a shared vocabulary that specifies restrictions and gives a common semantics for any application which uses learning objects metadata following this standard.

To facilitate the upload of LO in repositories, a prototype was developed in order to assist users in the process of loading the metadata fields. This prototype operates with LOs in text files and when a user submits such a file, some metadata are extracted automatically. Then, the user can confirm or modify these values and can complete other fields. Metadata values are validated using the ontology as a reference model, before uploading each LO in the repository. One of the advantages of this proposal is that, if the metadata specification changes in the ontology (e.g., a restriction is modified), the prototype will behave in a different way (e.g., restricting scenarios that were allowed previously) with no need to change its code.

We have used AlchemyAPI (www.alchemyapi.com/) for information extraction. Alchemy provides a robust, scalable platform for analyzing web pages, documents and tweets along with flexible APIs for easy integration. It utilizes statistical natural language processing technology and machine learning algorithms to analyze submitted content, extracting semantic metadata information. In particular, we choose Alchemy for extraction of keywords and language of LOs. We decided to use this extractor because in some previous experimentation on a selected set of 40 documents from our university repository we have obtained better precision on the keywords respect to other extractors.

Figure 3 shows how the different components in the system interact. The Assistant using the extractor Alchemy, obtains some LO metadata and then, applying the LOM ontology some metadata values are restricted and validated. Finally, the LO enriched with metadata can be uploaded to the repository.

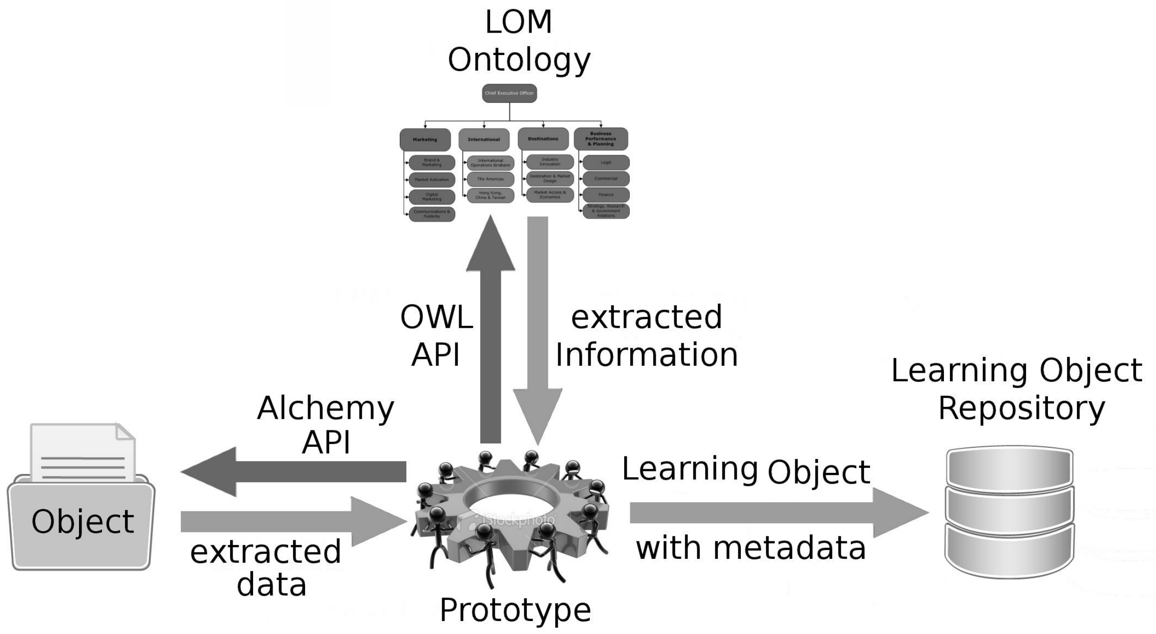


Figure 3: The Assistant prototype: Interactions and functionalities.

This prototype was developed in JAVA and it uses OWL API (http://owlapi.sourceforge.net) for communicating with the ontology and Alchemy API (http://www.alchemyapi.com) for information extraction. The user interface is in Spanish as this is the official language in our university.

From the analysis of different learning objects repositories available on the web we have selected a subset of metadata for the prototype implementation. These metadata are General::Title, Lifecycle::Contribute, General::Language, General::Keyword, Lifecycle::Status, General::Description and Educational::Difficulty.

The Assistant works as follows: when the program starts, a connection with the ontology is established, and then the application uses it as a reference for metadata loading. Particularly, the ontology is used in the following actions: restriction, extraction, and validation.

- *Restriction:* for metadata with type Vocabulary, the application retrieves from the ontology the enumerated values of metadata.

- *Extraction:* the metadata automatic extractor is restricted by the ontology specifications. For instance, if the maximum number of keywords is ten, the prototype will take a maximum of ten keywords, even if the number of extracted ones is greater.

- *Validation:* before starting the uploading process of the LO in the repository, all the fields values are checked against the ontology. If any field value does not verify its specification, an error message is shown. For instance, if the standard specifies that the metadata *Title* is mandatory and unique, and this is modeled in the ontology by a type "*constraint exactly 1"*, if the user does not complete the title, an error message will be shown and this LO will not be loaded.

A Use Case Example

We complete the description above by introducing an example to show how the assistant works during the load of metadata of a text document (in pdf format). We focus on the relationship between its behavior and the LOM Ontology used as a reference model.

Figure 4 shows the Assistant interface for loading a document on the left side, and the LOM Ontology on right side.

The principal steps in the Assistant metadata uploading are the following:

Step 1- In the initial screen we load a LO by clicking on *Archivo* (File) and then *Abrir* (Open). In this example we choose to load a PDF file containing a Computer Science classic: The Emperor's Old Clothes by C.A.R Hoare.

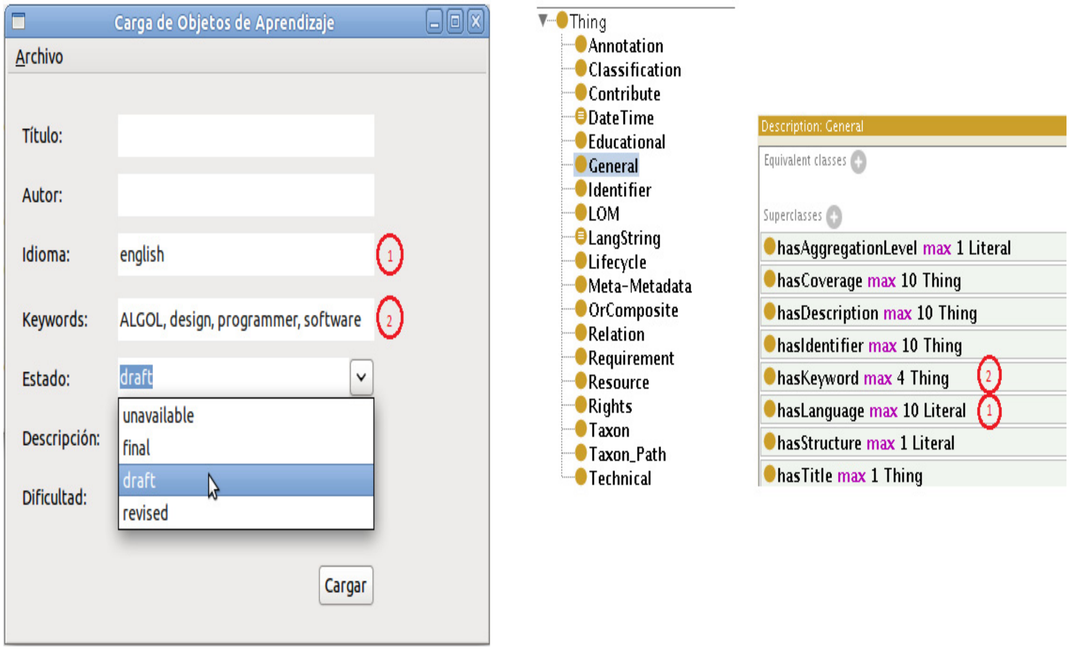
Step 2- After we load the LO the tools provided by the Alchemy API extract information from it, in this case, the language and some keywords. Although the API probably found several keywords in the input text, only the first four of them are shown. This is because the LOM Ontology specifies that there can be at most four keywords (see 2 in Figure 4 right side) and, therefore, it restricts the results obtained by the API to that number.

Note that in the real ontology the number of allowed keywords is 10; this was changed in this example for the sake of readability.

Additionally, the possible values for the field *Estado* (State) are also restricted by the LOM Ontology. Its permitted values are: unable, final, draft, and revised.

Step 3- Without adding further information we click on the button *Cargar* (Load) in order to upload the LO and the metadata we have just entered. The application proceeds to validate each field with the LOM Ontology. Finally, the load is successfully made.

This validation process checks that all mandatory fields will be present, for instance as the metadata *Title* is mandatory and unique, if this field is not loaded, the validation against the LOM Ontology will fail because the title is missing. Therefore, the LO and its metadata will not be uploaded until all the conditions imposed by the current version of the LOM Ontology are satisfied.



Título:

Autor:

Idioma: english

Keywords: ALGOL, design, programmer, software

Estado: draft

Descripción:

Dificultad:

hasAggregationLevel max 1 Literal
hasCoverage max 10 Thing
hasDescription max 10 Thing
hasIdentifier max 10 Thing
hasKeyword max 4 Thing
hasLanguage max 10 Literal
hasStructure max 1 Literal
hasTitle max 1 Thing

Figure 4: The Assistant interface for loading a document (left side) and the LOM Ontology (right side)

Conclusions

In this work, we show the importance of modeling metadata using an ontology and we have designed a LOM ontology. This specification gives a common vocabulary and precise semantics to IEEE LOM metadata. This ontology can be used and re-used by different applications in order to solve problems related to restrictions and validation. An advantage of this proposal is that if the metadata model changes in the ontology then, all applications that use this ontology as a reference model will change their behavior without changing it code. This proposal may be extended to other metadata standards by using an appropriate ontology that represents its characteristics and restrictions.

In this paper we have presented an Assistant to upload learning objects metadata giving a proof of concept of this proposal. This Assistant prototype helps users in the time-consuming task of metadata information loading. As future work, we want to improve the Assistant to extract other metadata fields by using different extractors and resources. We consider that this approach will enrich metadata information to improve LOs retrieval, decreasing the user's work in the uploading process. Furthermore, we plan to adapt and extend this prototype to assist the loading of LOs in the Repository of the National University of Rosario.

Acknowledgments

The authors acknowledge for the partial support by the project: "Hacia el desarrollo y utilización de Repositorios de Acceso Abierto para Objetos Digitales Educativos en el contexto de las universidades públicas de la región centro-este de Argentina" (PICTO-CIN N° 0143-ANPCyT) and the project Alfa III: DCI-ALA/19.09.01/11/21526/279-155/ALFA III(2011)-52.

References

Alfano, M., Lenzitti, B., & Visalli, N. (2007). SAXEF: A System for automatic eX-traction of learning object features. *Journal of e-Learning and Knowledge Society*, *3*(2), 83-92.

Casali, A., Deco, C., Bender, C., & Gerling, V. (2012). Recommender system for personalized retrieval of learning objects. In O. C. Santos & J. G. Boticario (Eds.), *Educational recommender systems and technologies: Practices and challenges* (pp. 182-210). Spain: *ERSAT*. aDeNu Research Group. UNED. (ISBN13: 9781613504895). DOI:10.4018/978-1-61350-489-5

Gluz, J., & Vicari, R. (2012). An OWL ontology for IEEE-LOM and OBAA metadata. *Proceedings of Intelligent Tutoring Systems - 11th International Conference, ITS 2012*, Chania, Crete, Greece, June 14-18, 2012. http://dx.doi.org/10.1007/978-3-642-30950-2_124

Gruber, T. (1995). Toward principles for the design of ontologies used for knowledge sharing. Originally in N. Guarino & R. Poli, (Eds*.), International Journal of Human-Computer Studies: Special issue on the role of formal ontology in the information technology, 43* (5-6 Nov./Dec.), 907-928.

Li, Y., Dorai, C., & Farrell, R. (2005). Creating MAGIC: System for generating learning object metadata for instructional content. *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 367-370, New York, NY, USA, 2005.

Motz, R., Badell, C., Barrosa, M., Sum, R., Díaz, G., & Castro, M. (2009). LooKIng4LO: Sistema Informático para la Extracción Automática de Objetos de Aprendizaje: Caso de Estudio. *IEEE-RITA* (2009) 223-229.

Pire, T., Espinase, B., Casali, A., & Deco, C. (2011). Automatic extraction of learning objects metadata for recommendation: A comparative study. *XIV Congreso Internacional de Informática en la Educación* InforEdu 2011. La Habana, Cuba, febrero 2011.

Sánchez-Alonso, S., Sicilia, M. A., & Pareja, M. (2007) Mapping IEEE LOM to WSML: An ontology of learning objects. *Proceedings of the* ITA'07 *- 2nd International on Internet Technologies and Applications*, pp. 92 - 101.

Sonntag, M. (2004) Metadata in e-learning applications: Automatic extraction and reuse. In C. Hofer & G. Chroust (Eds.), *IDIMT-2004. 12th Interdisciplinary Information Management Talks*, pp. 219-231, Universitätsverlag Rudolf Trauner, Linz, Austria, 2004.

Tang, W. Y. (2007). Automatic extraction of learning object metadata (LOM) from HTML web pages. (Master's thesis), City University of Hong Kong, May 2007.

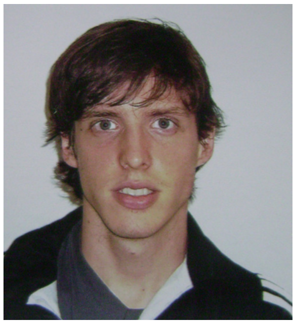
Wiley, D. (2002). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A. Wiley (Ed.), *Instructional use of learning objects*. Editorial Association for Instructional Technology.

Biographies

Ana Casali received her PhD and her Master degree in Information Technologies at the University of Girona, Spain and she obtained her degree on Mathematics at the Universidad Nacional de Rosario (UNR), Argentina. She is Head of the Computer Science Department since 2007 and Professor at the Facultad de Ciencias Exactas, Ingeniería y Agrimensura, UNR since 1991. Also, she is researcher in the Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas (CIFASIS). She has worked in several international cooperation research projects and her research interest includes agent architectures, knowledge representation, approximate reasoning, recommender systems and its applications to education. She has participated as a member of the conference program committee in many scientific events and she has been reviewer of different publications and journals. She also is author of many research articles mainly in the field of Artificial Intelligence and its applications.

Claudia Deco is Doctor in Engineering, Universidad Nacional de Rosario and she has a Master Degree in Computer Science, Universidad de la República, Montevideo, Uruguay. She obtained her degree on Mathematics at the Universidad Nacional de Rosario. She is also professor and researcher at the Research Department, Facultad de Química e Ingeniería, Rosario, Universidad Católica Argentina. Furthermore, she coordinates the Research Group of Databases at the Facultad de Ingeniería, Universidad Nacional de Rosario. Her research interest includes Databases Technologies and Information Retrieval. She participates in scientific events as a member of the conference program committee and as a reviewer of some papers and journals. She also is author of many research articles mainly in this area.

Guillermo Tomé is an advanced student of the Computer Science Degree at Universidad Nacional de Rosario, Argentina. He is currently working in the development group of the Ministery of Health of Santa Fe Province, Argentina. He is also a co-founder with fellow students, of a company dedicated to developing custom software solutions.



Agustín Romano is an advanced student of the Computer Science Degree at Universidad Nacional de Rosario, Argentina. He has worked as research assistant in research projects at IMDEA Software, Spain. He is currently working as Undergraduate Teaching Assistant at Universidad Nacional de Rosario and he is still collaborating with some projects started during his work at IMDEA Software.