

# OBAA-LEME: A Learning Object Metadata Content Editor supported by Application Profiles and Educational Metadata Ontologies

André Behr<sup>1</sup>, Tiago Thompsen Primo<sup>2</sup>, Rosa Vicari<sup>1</sup>

<sup>1</sup>Informatics Institute – Federal University of Rio Grande do Sul  
Postal Code 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

<sup>2</sup>SAMSUNG Research Institute – Campinas, SP – Brazil

arbehrr@inf.ufrgs.br, tiago.t@samsung.com, rosa@inf.ufrgs.br

**Abstract.** *This paper proposes an editor for Learning Object Metadata compatible with the Semantic Web. This tool uses two base ontologies, OBAA and VideoAula to provide the semantics to describe learning objects. These ontologies are the core for a tool to verify the consistency of the user-input data and realize inferences about the described information. Furthermore, this tool is mainly based on application profiles (derived from the base ontologies) that copes with the reasoning engine that verifies the description of the learning objects. The tool also serializes a learning object metadata ontology in an open server that can be reused in third-party applications.*

## 1. Introduction

The online learning faces new challenges with the growth of ubiquitous computing. We refer to this relationship as ubiquitous learning. A research realized by The Engineering Education Report<sup>1</sup> points out three major alternatives to be applied in the near-future world education. In descending order of relevance: E-learning Platforms and Architectures, E-books and digital libraries, and Massive Open Online Courses.

Those alternatives usually encompass the usage of learning objects (LOs), that are defined by the [LTSC 2002] as “an entity, digital or non-digital, that may be used for learning, education, or training”.

Nowadays, metadata are largely used to describe LOs on the Web, helping in its search and reuse. According to [Bargmeyer et al. 2000], “metadata is data that is used to describe other data” and the metadata standards “...help promote interoperability between organizations, systems, and people”.

The usage of such technologies are growing. As an example, the International Bank of Educational Objects (BIOE)<sup>2</sup> has 19,802 objects published. The total amount of objects published in BIOE is increasing during the years<sup>3</sup>. In the period 2009-2010 it was 38%, 2010-2011: 36%, 2011-2012: 11%, and 2012-2013: 12%. The learning objects are described with the Dublin Core metadata standard [Weibel et al. 1998].

---

<sup>1</sup>[http://ohm.ieec.uned.es/eer/consulta\\_years.php](http://ohm.ieec.uned.es/eer/consulta_years.php)

<sup>2</sup><http://objetoseducacionais2.mec.gov.br/>

<sup>3</sup>This data was collect by the Internet Archive with use of the wayback machine as the BIOE just inform the amount of objects in the current day.

There is no metadata standard that can be used for all applications [Kraan 2003]. As an alternative, the combination and extension of existing metadata standards can be a solution. This concept is called application profile [Duval et al. 2002].

Nevertheless, the correct usage of metadata leads to a time-consuming situation. The diversity of metadata standards and its combinations also increases the complexity of understanding and its usage.

The purpose of this work is to propose a LO metadata content editor based on ontological application profiles. This tool helps the users along with the metadata editing process and eases the burden to cope with complex metadata standards.

To cope with that, we choose to use ontologies as the base technology and knowledge representation. Ontologies are compatible with the Semantic Web [Shadbolt et al. 2006] and were described with the OWL [Antoniou and Van Harmelen 2004] language and with a defined ontology engineering methodology [Behr et al. 2012].

The proposed research will present a tool that is able to import an application profile ontology (APO), and present to the final user an interface that will (i) allow the description of the meta-information described in APO; (ii) it will use a reasoner to verify any inconsistencies at the fulfillment process, and (iii) provide an explanation interface that is presented to support the user, in case of inconsistency.

The paper follows presenting a theoretical background in Section 2. The related works to the work reported here is described in Section 3. The main characteristics of the metadata content editor proposed (OBAA-LEME) in this paper are showed in Section 4 with reusing of Videoaula [Primo 2013] as a study case. In the end, the conclusion and future work are presented on Section 5.

## **2. Theoretical Background**

In this section will be presented some associated topics to the well understanding of this paper main topics. Brief definitions of ontologies and the Videoaula metadata standard will be described. In the end, the methodology of the information flow will be explained.

### **2.1. Ontologies**

The ontology definition was improved during the years. [Studer et al. 1998] merged Gruber and Borst definitions stating that “An ontology is a formal, explicit specification of a shared conceptualization”.

In order to aggregate theories and technologies for the ontology construction, the ontology engineering plays an important role in the ontology research. According to [Mizoguchi and Ikeda 1996], the ultimate purpose of ontology engineering is to “provide a basis for building models and all things in which information science is interested in the world”.

The Semantic Web gives semantic for the web data, i. e. a “meaning” and logical connections to terms. The Semantic Web that is expected makes considerable reuse of existing ontologies and data. Ontologies will furnish the semantics for the Semantic Web must be developed, managed, and confirmed. It’s a linked information space to data be added and improved [Shadbolt et al. 2006].

The interoperability of learning objects in different learning systems is supported by the use of ontologies. The ontologies that will describe the structure of learning objects [Mohan and Brooks 2003].

In the work reported here, each LO is described as an ontology. Classes, individuals, and properties are going to represent the metadata structure. This representation will be made through the OWL.

## 2.2. Videoaula: Application Profile Domain Ontology

The combining and adapting of the various metadata standards or specifications can be made to meet the community context-specific needs [Kraan 2003]. This conceptualization is called application profile. Modularity and extensibility are principles utilized through the application profiles. An application profile aims to combine or adapt existing schemas and add local metadata elements [Duval et al. 2002].

The OBAA Metadata Standard has an extensive set of metadata. This is necessary because the standard aims interoperability and adequacy to the Brazilian educational context. In [Primo 2013], a Videoaula application profile based on the OBAA Metadata Standard was proposed to represent learning objects composed by multiple synchronized medias.

The Videoaula application profile is composed by some elements of the groups 1. General, 2. Lifecycle, 3. MetaMetadata, 4. Technical, 5. Educational, 6. Rights, and 7. Relation, with addition of some new metadata items. An ontology<sup>4</sup> that represents this application profile was created. Furthermore, an equivalent class to represent and classify a LO metadata with minimum items of the Videoaula application profile is showed in Figure 1.

|   |  |
|---|--|
| <pre>CopyrightAndOtherRestriction and Duration and Format and General.Description and General.Keyword and General.Language and General.Title and Location and MetadataSchema and Rights.Description and Size and Status and SupportedPlatforms and (hasEducational some Context) and (hasEducational some Educational.Description) and (hasEducational some IntendedEndUserRole) and (hasEducational some LearningResourceType) and (hasEducational some TypicalLearningTime) and (hasEducational some Program) and (hasGeneral.Identifier some General.Identifier.Catalog) and (hasGeneral.Identifier some General.Identifier.Entry) and (hasLifecycle.Contribute some Lifecycle.Contribute.Date) and (hasLifecycle.Contribute some Lifecycle.Contribute.Entity) and (hasLifecycle.Contribute some Lifecycle.Contribute.Role) and (hasMeta-Metadata.Contribute some Meta-Metadata.Contribute.Date)</pre> | <pre>and (hasMeta-Metadata.Contribute some Meta-Metadata.Contribute.Entity) and (hasMeta-Metadata.Contribute some Meta-Metadata.Contribute.Role) and (hasMeta-Metadata.Identifier some Meta-Metadata.Identifier.Catalog) and (hasMeta-Metadata.Identifier some Meta-Metadata.Identifier.Entry) and (hasRelation some Kind) and (hasRelation some Resource.Description) and (hasEducationalChain some Code) and (hasEducationalChain some Course.Title) and (hasRelationChain some Resource.Identifier.Catalog) and (hasRelationChain some Resource.Identifier.Entry) and (hasRequirementChain some OrComposite.Name) and (hasRequirementChain some OrComposite.Type) and (hasServiceChain some ServiceLocation) and (hasPlatformSpecificFeatures some SpecificAspectRatio) and (hasPlatformSpecificFeatures some SpecificStandardResolution) and (hasService some Essential) and (hasService some Protocol) and (hasService some Provides) and (hasService some Service.Name) and (hasService some Service.Type) and (hasRelatedMedia some RelatedMedia.Catalog) and (hasRelatedMedia some RelatedMedia.Entry) and (hasThumbnail some Thumbnail.Catalog) and (hasThumbnail some Thumbnail.Entry)</pre> |
|---|--|

**Figure 1. OBAA-VideoAula equivalent class to represent the application profile minimum metadata items.**

In this paper, a metadata which has sub-metadata items in its hierarchy is called

<sup>4</sup><http://gia.inf.ufrgs.br/ontologies/VideoAula.owl>

container. Considering the Meta-Metadata group of LOM, the Meta-Metadata itself, Identifier, and Contribute are considered containers.

### **3. Related Work**

There are a several tools that describe learning objects according some existent metadata standard. Those tools have standard functionalities as exportation, help, and visualization. However, the tools that are compatible with the Semantic Web are not so established.

Most of the metadata content editors are based in the IEEE-LOM metadata standard. For example, the authoring tool integrated in a Intelligent Tutoring System [Garofalakis et al. 2007]. Although, the tool is not Semantic Web compatible, it uses a database to manage the metadata relations. An XML can be viewed and exported.

There are metadata tools that are very functional, having superior abilities for managing the IEEE-LOM. The LOM Editor [LOMEditor 2014] and LOMPAD [LOMPAD 2014] are well known examples of managing and exporting the metadata.

The SCAM [Palmér et al. 2004] framework has the main objective to store and load the metadata in a easy way for softwares that use it as base. The framework is independent of application profiles. It uses RDF bindings of Dublin Core and IEEE-LOM.

The eXe Learning [Bulegon and Mussoi 2010] is an authoring tool that describes learning objects according the Dublin Core metadata standard. The LO exportation to the IMS Content Packaging [IMS 2000] and SCORM [ADL 2001] can be made as well. The tool is not Semantic Web compatible. Each field has a help option to access.

In [Casali et al. 2013], it is proposed an assistant for loading learning object metadata. The prototype stores the LO in a repository and produces an ontology metadata description that is validated. Some metadata content for LO text files is recommended through Alchemy API. It only works with a restricted small subset metadata of LOM and do not present the basic functionalities as exportation, help, and visualization.

Another tool compatible with the Semantic Web is LOIT [Ghebghouba et al. 2009], an indexing tool based on a LOM ontology. It is possible to export the metadata and access helping information. LOIT does not use application profiles.

This work proposes a prototype tool that will describe the learning object metadata information to be used in the Semantic Web. Besides using the common editors functionalities, our proposal differ of the related works in the fact that it loads the metadata through the application profiles, aiming flexibility in its definition and providing just the minimum quantity of metadata to be filled. Also, it is possible to verify the metadata content that the related works are not able to.

#### **3.1. Methodology**

The OBAA-LEME proposal tries to behave like the traditional metadata editors. Moreover, it intends to add important functionalities through ontologies as content verification, documentation embedded, and Semantic Web compatibility.

In this way, from the LO metadata content information, (1) the user has to input

this data in the corresponding fields in OBAA-LEME. Afterward having all the information filled and consistent, (2) the OBAA-LEME will publish the ontology metadata information in a web server. This information flow is represented in Figure 2.



Figure 2. OBAA-LEME information flow.

## 4. OBAA-LEME

The tool was developed using Java programming language with the OWLAPI<sup>5</sup>. It imports a VideoAula, OBAA, IEEE-LOM, and IMS AccessForAll ontologies to get the metadata's semantic.

In a first moment, the user has to provide the physical location of the learning object. It can be the location in the user computer or an URI of the file. Whether it is a local file, then it will load this file on server, creating a new URI for it. Later, the tool already filled the Location metadata with this information because this metadata item is responsible to store the string used to access the learning object.

### 4.1. Application Profile Choose

In a second moment, the user can view and choose the existent application profiles to represent LOs with minimum metadata items. Each metadata that composes the application profile is showed in an alphabetical order as illustrated in Figure 3.

Each application profile is also represented by ontologies, which are imported and grouped in the Profiles ontology<sup>6</sup>, further loaded by the OBAA-LEME content editor. When a new application profile is defined, importing this new ontology in the Profiles ontology will automatically allow the tool to load it.

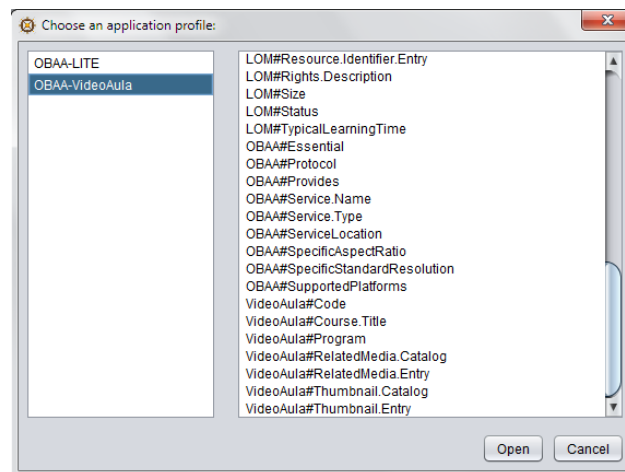
### 4.2. Filling Metadata Information

After the user choose the application profile, all its metadata are loaded, and divided into tabs. Containers with cardinality greater than one will create a specific tab. Metadata that belongs to a container with a cardinality equals to one will be associated to the previous container in the hierarchy (with cardinality greater than one). Whether all the previous containers have cardinality one, the metadata will be associated to a tab with the application profile name.

The Figure 5 presents the disposal of the tabs for the OBAA-VideoAula application profile. Additional metadata containers that belong to the application profile also can

<sup>5</sup><http://owlapi.sourceforge.net/>

<sup>6</sup><http://gia.inf.ufrgs.br/ontologies/Profiles.owl>



**Figure 3. Application profile choose and listing of metadata names.**

be added. As an example, another Educational tab can be added to the learning object metadata description.

A help button (“?”) was added on the side of each metadata field. It is possible to access some documentation information about the metadata, as its explanation, data type, example, cardinality, and value space.

Combo boxes are shown instead of text fields when a metadata has a value space defined. When the metadata cardinality is one, just a combo box is shown. Otherwise, a text field is added before and the user can add values to it by pressing the “+” button or just text it by comma separation.

Therefore, the tool barely uses a comma separation for multiple values to metadata with cardinality greater than one. For example: “diagram, exercise” filled in the *Learning Resource Type* metadata are considered distinct contents. They will result in two data properties values in the ontology.

### **4.3. LO Metadata Exportation**

As even the application profiles can have a large metadata information, to have all the metadata contents done can be time-consuming. The user can just fill a portion of the metadata information and then export this information to a local file.

The exported file will be already in an OWL ontology format. Thus, it is possible to open it in the tool and load the metadata information previous filled. It is also possible to fill all the learning object metadata and export to a file for further publication.

### **4.4. LO Metadata Visualization**

In the learning object visualization option is possible to see the entire tabs relationships represented by ellipses. This visualization is made through the Graphviz<sup>7</sup>. Each ellipsis will result in an individual in the learning object metadata ontology and represent the metadata containers with cardinality greater than one.

<sup>7</sup><http://www.graphviz.org/>

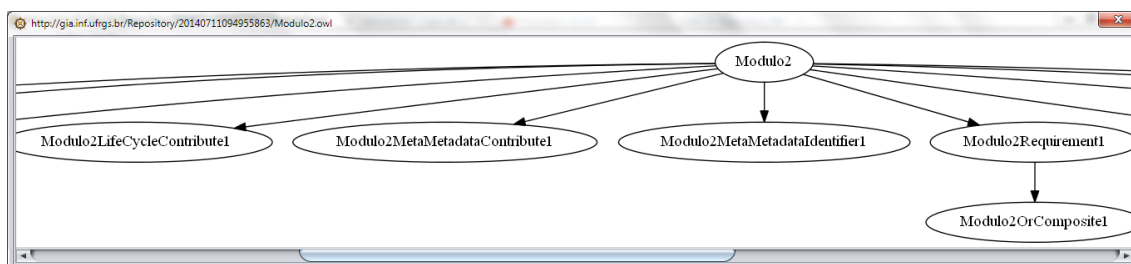


Figure 4. Visualization of Modulo 2's tabs relationships.

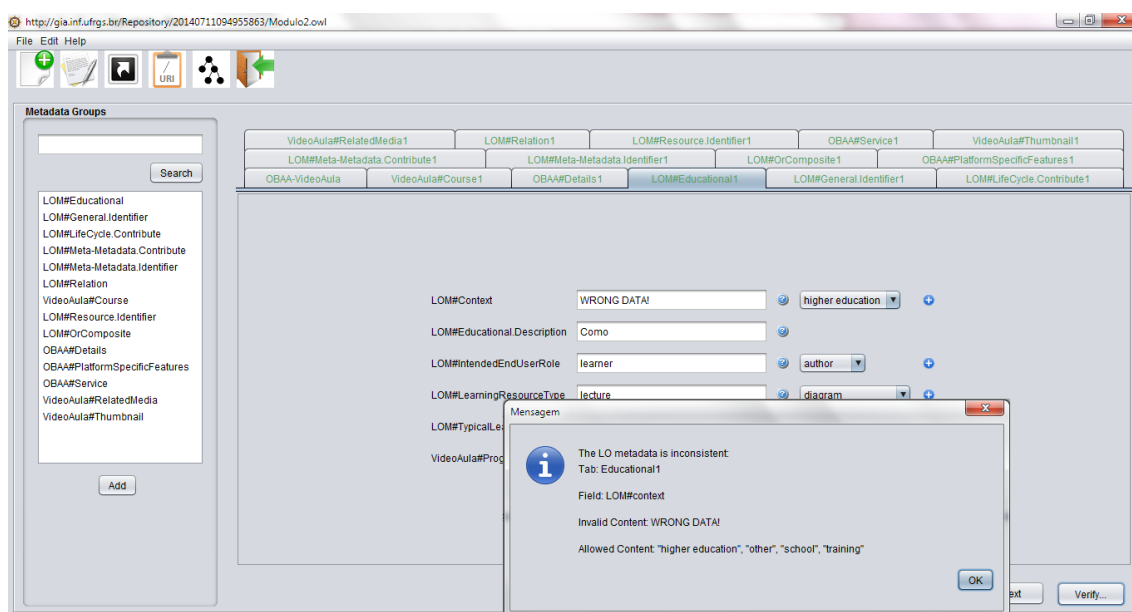


Figure 5. Modulo 2 inconsistency data example.

The partial general metadata representation of Modulo 2<sup>8</sup> is showed in Figure 4. The learning object was designed according the OBAA-VideoAula application profile.

#### 4.5. Reasoning

At the time that the user filled all the metadata items, the LO consistency must be verified before be published on the server. After the processing time, the tool will show a message saying if the LO is well described or if there is some inconsistency.

The metadata value space can be an example of reasoning, it checks if a value is or not one of the value space defined. The Figure 5 shows the inconsistency captured by the reasoner, where the value filled (“WRONG DATA!”) was not in the value space acceptable (“higher education, other, school, or training”) to the IEEE-LOM Context field. The tab, the field with the wrong value, and also the acceptable space value are showed.

Another examples of inconsistency can be check the metadata maximum cardinality or conditional values. In addition to check if all the application profile metadata items are filled.

<sup>8</sup>[http://va05-cps.rnp.br/riotransfer/rnp/treinamentos/videoaulas/modulo\\_2/modulo\\_2.xml](http://va05-cps.rnp.br/riotransfer/rnp/treinamentos/videoaulas/modulo_2/modulo_2.xml)

#### 4.6. LO Metadata Publication

After the Reasoner verifies and accepts the new ontology, the user can publish the LO. When accepted, the LO ontology will be put in the server. If the user does not want to publish the LO in that moment than the ontology is removed.

All the LOs are in an open Repository<sup>9</sup>. Each LO will be stored in a folder named with a timestamp, working like a key, where the LO local file previously mentioned and its ontology will be stored. Besides this together storage, the timestamp also allows that a LO with the same name can be published, as example for further versioning.

As a case of study, a complete resulting ontology description of the Modulo 2 LO metadata<sup>8</sup> can be observed. This ontology is available at <http://gia.inf.ufrgs.br/Repository/20140403211232415/Modulo2.owl>

#### 5. Conclusion and Future Work

In this work OBAMA-LEME was presented. This tool makes substantial use of Semantic Web technologies and intends to minimize the burden of filling metadata and the content verification based in application profiles.

Along with this tool, we intended to contribute to the future of education by exploring the usage of Semantic Web technologies. In this way, ontologies were used to describe the semantics that are underlying to metadata standards of learning objects.

Through this adoption, we achieved the reusability of learning object in Semantic Web domains, and further, the semantic educational repositories.

Another contribution of this work, relates to the fact that the users are free to choose an authoring tool to build a LO. The presented research and tool only deals with the metadata information of the learning objects.

That information is mapped to ontologies which are openly stored in a internet server. This outcome copes with the free and open educational resources initiative, that is an international effort to publish educational contents, that will are open e freely available for students and educators.

The current research was supported by an ontology engineering methodology. On the one hand, it can increase the complexity of the software. On the other hand, there is the possibility of external reasoners to process the metadata without the necessity of LOs repositories installation and maintenance. Besides that, it has a greater knowledge extraction potential and can be use with intelligent algorithms.

In order to continue this work, it is planned new versions of the tool and experiments, including an improved usability, architecture, and recommendation aspects. For instance, we are working to add a recommendation feature to support the relationships between Learning Objects.

#### References

ADL (2001). Sharable content object reference model (scorm). *Advanced Distributed Learning*, <http://www.adlnet.org>.

---

<sup>9</sup><http://gia.inf.ufrgs.br/ontologies/Repository>



- Antoniou, G. and Van Harmelen, F. (2004). Web ontology language: Owl. In *Handbook on ontologies*, pages 67–92. Springer.
- Bargmeyer, B. E., Agency, E. P., Gillman, D. W., and Statistics, B. O. L. (2000). Metadata standards and metadata registries: An overview.
- Behr, A., Primo, T., and Viccari, R. (2012). An ontology for the obaa metadata standard. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 23.
- Bulegon, A. M. and Mussoi, E. M. (2010). exe learning: uma ferramenta de autoria para o ensino e aprendizagem.
- Casali, A., Deco, C., Romano, A., and Tomé, G. (2013). An assistant for loading learning object metadata: An ontology based approach.
- Duval, E., Hodgins, W., Sutton, S., and Weibel, S. (2002). Metadata principles and practicalities. *D-Lib Magazine*.
- Garofalakis, J., Hatzilygeroudis, I., and Kyriakou, P. (2007). A tool for managing domain metadata in a web-based intelligent tutoring system.
- Ghebghouba, O., Abelb, M.-H., Moulinb, C., and Benmmohamedc, M. (2009). Loit: An indexing tool based on lom ontology.
- IMS (2000). Version 1.1. 2 final specification. *IMS Global Learning Consortium*, <http://www.imsproject.org/metadata/index.html>.
- Kraan, W. (2003). No one standard will suit all. *The Centre for Educational Technology Interoperability Standards*, Presentation available online.
- LOMEditor (2014). Retrieved on 19.03.2014 from <http://dbis.rwth-aachen.de/cms/projects/lomeditor>.
- LOMPAD (2014). Retrieved on 19.03.2014 from <http://helios.liceef.ca:8080/lompad/en/index.htm>.
- LTSC (2002). Ieee standard for learning object metadata (draft). ieee standard 1484.12.1.
- Mizoguchi, R. and Ikeda, M. (1996). Towards Ontology Engineering. Technical Report AI-TR-96-1, The Institute of Scientific and Industrial Research, Osaka University.
- Mohan, P. and Brooks, C. (2003). Learning objects on the semantic web. In *Advanced Learning Technologies, 2003. Proceedings. The 3rd IEEE International Conference on*, pages 195–199. IEEE.
- Palmér, M., Naeve, A., and Paulsson, F. (2004). The scam framework: helping semantic web applications to store and access metadata. In *The Semantic Web: Research and Applications*, pages 167–181. Springer.
- Primo, T. T. (2013). *Método de representação de conhecimento baseado em ontologias para apoiar sistemas de recomendação educacionais*. PhD thesis, Universidade Federal do Rio Grande do Sul (UFRGS).
- Shadbolt, N., Berners-Lee, T., and Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.
- Studer, R., Benjamins, R., and Fensel, D. (1998). Knowledge engineering: principles and methods. *Data and knowledge engineering*, 25:161–197.

Weibel, S., Kunze, J., Lagoze, C., and Wolf, M. (1998). Dublin core metadata for resource discovery. *Internet Engineering Task Force RFC*, 2413(222):132.