

Towards an Ontology to Describe the Taxonomy of Common Modules in Learning Management Systems.

Carlos E. Montenegro-Marin¹, Juan Manuel Cueva-Lovelle², Oscar Sanjuan², Edward Rolando Nuñez-Valdez²

¹Engineering Faculty, Distrital University, Bogotá, Colombia.

²Computer Science Department, Oviedo University, Oviedo, Spain

Abstract — This article has the objective to create an ontology for "common modules in Learning Management Systems", the steps for building the ontology were: Determine the domain and scope of the ontology, Consider reusing existing ontologies, Enumerate important terms in the ontology, Define the classes and the class hierarchy, Define the properties of classes—slot and Define the facets of the slot, finally, explain how the ontology is composed.

Keywords— Ontology, class, Learning Management Systems, common modules, Protégé.

I. INTRODUCTION

A very big problem that is found in the integration or migration of platforms e-learning or Learning Management Systems (LMS) is the incompatibility between these, this incompatibility is due to lack of unification regarding the appointment and composition of modules and submodules that integrate the different LMSs, for example the module file management in the Atutor platform is named "File Administrator" while that in other platforms like Claroline or Moodle is named "Documents" and "Resources" respectively.

Before working with the LMS platforms it is necessary to select LMS platforms. In this sense we divided the LMS platforms into two types; the private platforms like BlackBoard and the Open Source platforms like Moodle.

Due to the huge amount of LMS platforms that exist in the market, it is necessary to select the most commonly used. For the selection of LMS, we chose Moodle, Sakai, and DotLRN because works such as [1] show that Moodle is the most used open source LMS in Spanish Universities with over 45%. Furthermore, Sakai has 5%, and DotLRN has 4%. Claroline and ATutor were selected because they are also used worldwide. For instance, according to [2] each of these platforms is used by 3% of the Italian Universities.

This article presents an ontology of modules common in LMS platforms, in the first stage had been realized the "MODELING AND COMPARISON STUDY OF MODULES

IN OPEN SOURCE LMS PLATFORMS WITH CMAPSTOOL" [3], in this moment we tested five LMS (Moodle, Sakai, DotLRN, Claroline and Atutor), for each of them we constructed your knowledge map with CmapTools and we obtained a comparative table between its modules. This result is the information source for building the ontology. For the construction of the ontology, the methodology used was "Ontology Development 101: A Guide to Creating Your First Ontology"[4] and this step is developed in section 2, the section 3 explained the ontology composition and finally in section 4 are located the conclusions.

II. METHODOLOGY FOR CREATING ONTOLOGY.

For creating an ontology we used the guide official for the ontology development that Stanford University recommends, Stanford University is the creator of Protege [4] and the name of the guide is "Ontology Development 101: A Guide to Creating Your First Ontology" [5]. Here we define the concept of the ontology how "is a formal explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties)), and restrictions on slots (facets (sometimes called role restrictions)). An ontology together with a set of individual instances of classes constitutes a knowledge base" [5] and proposed a methodology for the construction of ontologies, comprising the following steps:

1. Determine the domain and scope of the ontology.
2. Consider reusing existing ontologies.
3. Enumerate important terms in the ontology.
4. Define the classes and the class hierarchy.
5. Define the properties of classes—slots.
6. Define the facets of the slots.
7. Create instances.

The following chapters develop the proposed methodology adapted to our special needs. Respect to coding standards, uses the same Java recommended[6].

A. Determine the domain and scope of the ontology.

The problem domain this bounded, to the management of an LMS Components and especially of the LMS appointed in the section one, with a focus at the creation of courses. The Ontology is employed for the get respect necessary knowledge at the modules that make up a LMS and the create courses within him, Similar modules or homologous modules between LMS, the ontology may answer questions as ¿A forum is part of the tools of the LMS? Or ¿a teacher is a type of user of a LMS? it's very important clarify that ontology is created to a changing and evolution context, therefore it is necessary maintain, upgrade and expand according to the context need for the ontology.

B. Consider reusing existing ontologies.

To consider progress in the area of ontologies for LMS, we made a search of available ontologies, in order to work and enrich more, the search was conducted in the following browsers ontological, Recommended in [5]:

- <http://swoogle.umbc.edu/>: Swoogle is a Ontology search engine for the Semantic Web on the Web [7].
- http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library: Page of Ontology's, is organized into the following groupings [4]
- <http://www.ksl.stanford.edu/software/ontolingua/>: Ontolingua provides a distributed collaborative environment to browse, create, edit, modify, and use ontologies. The server supports over 150 active users, some of whom have provided us with descriptions of their projects [8].
- <http://www.daml.org/ontologies/>: The DARPA Agent Markup Language (DAML) Program officially began in August 2000. The goal of the DAML effort is to develop a language and tools to facilitate the concept of the Semantic Web. Michael Pagels is the DARPA Program Manager for DAML. The DAML program will end in early 2006 [9]
- <http://www.unspsc.org/>: The United Nations Standard Products and Services Code (UNSPSC) provides an open, global multi-sector standard for efficient, accurate classification of products and services. Search the code on this website to locate commodity codes that can be used by your company. The UNSPSC offers a single global classification system that can be used for: Company-wide visibility of spend analysis, Cost-effective procurement optimization, Full exploitation of electronic commerce capabilities. You may browse and download the current version of the code at no cost [10].
- <http://www.dmoz.org/>: The Open Directory Project is the largest, most comprehensive human-edited directory of the Web. It is constructed and maintained by a vast, global community of volunteer editors [11].

The search was unsuccessful in these containers and browsers ontological, since all ontology were offered concept of Learning Management System, but do not model the

ontology of the LMS, so it is not feasible to use one of these ontologies.

The next step was to find an ontology is search in the academic databases such as ISI Web of Knowledge, Springer, Ebsco, Dialnet, Proquest, ACM, IEEE, Google Scholar and other similar, there are some jobs such as:

- *A Learner Oriented Ontology of Metadata to Improve Effectiveness of Learning Management Systems*: This paper presents ontology for an e-Learning Management System (LMS), which arranges metadata, and defines the relationships of metadata, which are about learning objects; belong to academic courses and user profiles. This ontology has been incorporated as a critical part of the proposed architecture. By this ontology, effective retrieval of learning content, customizing LMS is expected. Metadata used in this paper are based on current metadata standards. This ontology specified in human and machine-readable formats. In implementing it, several APIs were defined to manage the ontology. They were introduced into a typical open sourced LMS. Proposed ontology maps user preferences with learning content to satisfy learner requirements. These learning objects are presented to the learner based on ontological relationships. Hence it increases the usability and customizes the LMS [12].
- *Towards an ontology about LMS*: This article is a review of the LMS concept and proposes ontology based on the latest definitions. This article is part of an investigation in progress that aims to clarify the systemic quality in the process of implementing an LMS in an organization [13].
- *Justification and description of the domain of knowledge of an Ontology for the formalization and automatization of education scenarios*: This article justifies the need to build an ontology with order to provide technical support for a specification of learning scenarios and a tool for development and validation of new scenarios. Specifically, it justifies the need for ontology and described in natural language the first approach to domain knowledge of it [14].
- *Knowledge Representation of LMS using Ontology*: Though there are no distinct classifications of the approaches while implementing, the union of all the viewpoints is not dealt / applied completely by any of the author. This paper focus on integrating the above said principles on semantic educational servers with the power of eLearning standards. The knowledge items (learning objects) are linked to commonly agreed ontology [15].
- *Towards an ontology of lms a conceptual framework*: This article presents a research in progress whose final objective is to develop a method to select, implement and integrate an LMS into an organization with a systemic quality approach. As a first step, in this article is presented an ontology to conceptualize

the terms associated to LMS, unifying them through their relations [16].

All these papers raise the creation of an ontology in their respective contexts and approaches made to the ontology, but not analyzed common modules between LMS platforms, All these ontologies are used as input for the a ontology building common between some LMS platforms.

C. Enumerate important terms in the ontology

For the enumeration important terms we used a list of terms available in [3], in this list is presented a approximation of modules compatible between LMSs and a generic name for them to define the LMS ontology, this is shown in the table 1.

TABLE I

COMPARISON OF MODULES BETWEEN SOME LMS PLATFORMS AND A GENERIC NAME					
Generic	Atutor	Claroline	Moodle	.RLN	Sakai
File manager	File Administrator	Documents	Add resources	Documents	Resources
Announcement	Announcement s	Announcements	Home	∅	Announcement
Help	Help	Help	Help	Help	Help
Chat	Chat	∅	Chat	Chat	∅
Management Curricula	Contents	∅	Add resource	Class material	Resources
			Add activity	Learning contents	∅
Educational Design	Administration	Learning path	Site Administration	Control panel	Portfolios Preferences
Course	Course	Course claroline	Module	Course	Portfolios
Authentication	Directory	Users	Users	Teachers	Accounts
Survey	Quiz	∅	Quiz	∅	∅
Evaluation System	Tests and Quiz	Evaluation System	Questions	Evaluations Questions	Evaluation System
Forums	Forums	Forums	Forums	Forums	Messages and Forums
Glossary	Glossary	∅	Glossary	∅	∅
Groups	Groups	Groups	∅	Groups	∅
Work Group	Networking	∅	∅	Communities	Profile2
FAQ	FAQ	∅	∅	FAQ	∅
Activity calendar	∅	Calendar	Activities	My calendar	Calendar
News	∅	∅	∅	News	News
Wiki	∅	Wiki	Wiki	∅	∅
Rating system	Test and task	Exercises	Module	Evaluates	Event
			Advanced File	Test	
		Task	Uploading	Projects	
			Upload File	Task	
Administration	Preference	Yes	Site Administration	Control Panel	Membership (Site administration)

D. Define the classes and the class hierarchy

For the definition of the classes, we chose a combined process mix between top-down process in which one begins by defining the general concepts and lay the composition of these, for example a course has tools, and process bottom-up which begins with the definition of more specific classes and then generalized, for example a forum is part of communications.

The end result of this step, we can see in Figure 1.

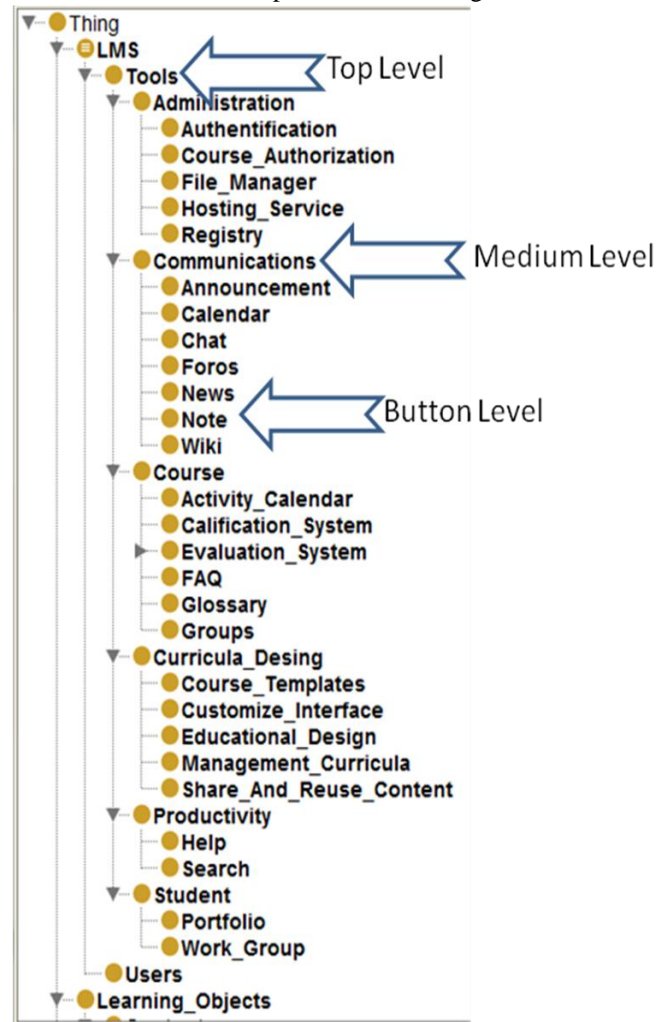


Figure 1. View of the class in Protege with classification levels

E. Define the properties of classes—slots

The properties of the classes are defined basing on types of components that they have, this classification of the components had already been done on the job [3](to appear), using the leaf nodes used knowledge maps for each class used in the ontology, is searched common components in knowledge maps for example, in the conceptual maps of moodle and claroline for the Forum module, we can determine that the common elements for the forums are: forumName, message, subject and attachment is optionally, this means that these are the properties of our classes and are of type: textBox, textBox, textBox and file respectively why this is shown in the leaf nodes, the Figure 2 show the members of the Foro Class and Figure 3 show the type of forumName.

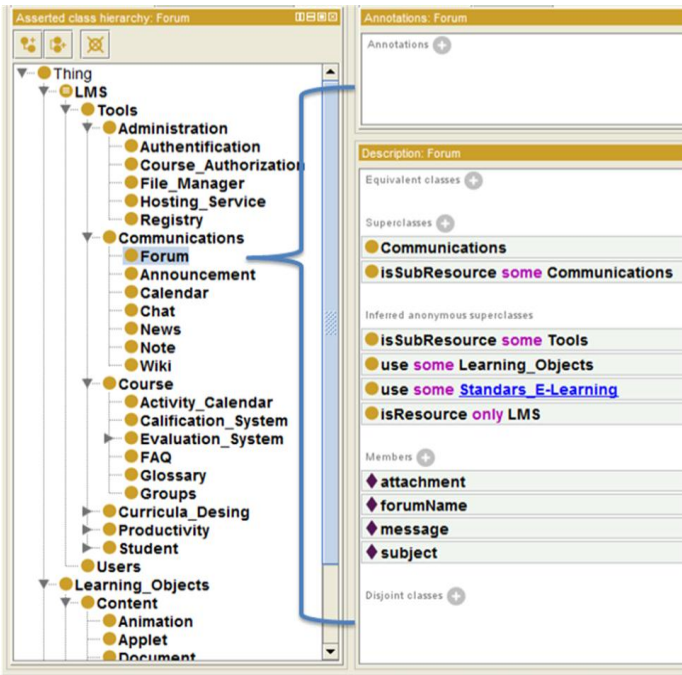


Figure 2. Members of the Forum Class

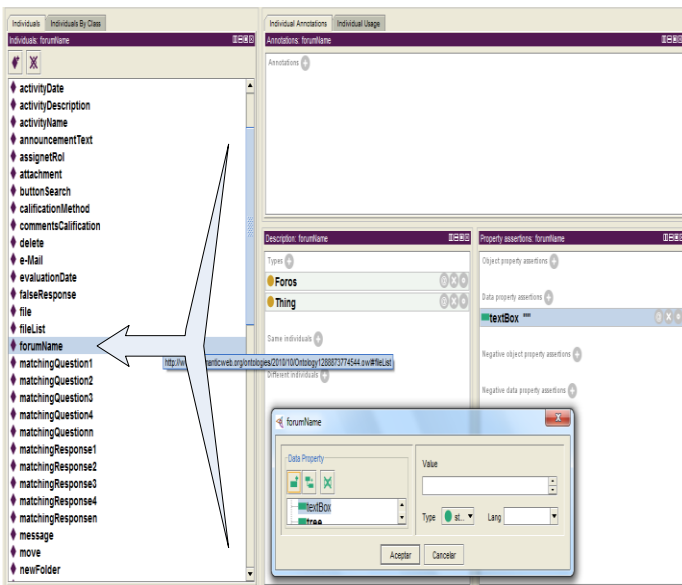


Figure 3. Type of the forumName

F. Define the facets of the slots

Here we define the facets that describe the type of value, valid values, the number of values (cardinality) and other characteristics that the slots can take, for example the item evaluationDate is a type calendarDate and the values it receives only Activity_Calendar, are as shown in Figure 4 and the same types of securities may be as common as strings or numbers.

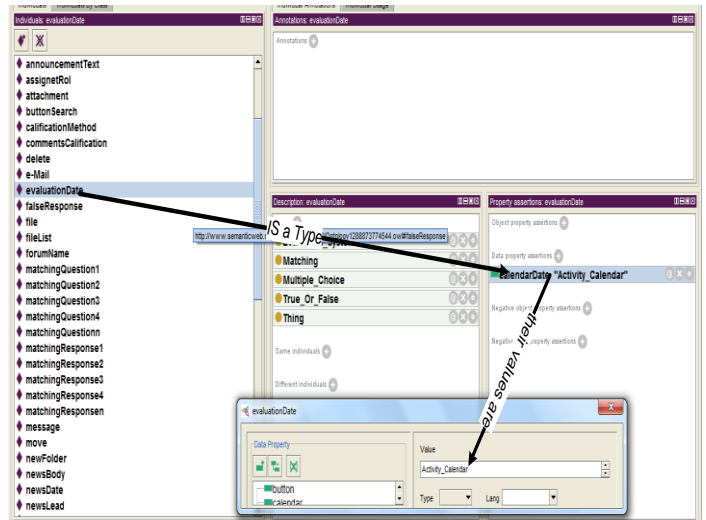


Figure 4. Element evaluationDate, that is of the type calendarDate and their values are Activity_Calendar

G. Create instances

This last step is not proved, because not is provided in the scope of the this article and our case will be the responsibility of the domain specific language for the generation of learning management systems modules [17], this project was developed in the Informatics department of the University of Oviedo.

III. ONTOLOGY GENERATION

The generated ontology is composed of 50 classes including the Thing Class, 4 Object Properties, 16 Data Properties, 69 Individuals, each of these components with their respective descriptions the more general classes are LMS, Learning_Objects and Standars_E-Learning, this classes show in the Figure 5, their classes visualization is done in Asserted class hierarchy of Protege ant the view offered by OWLViz Plugin.

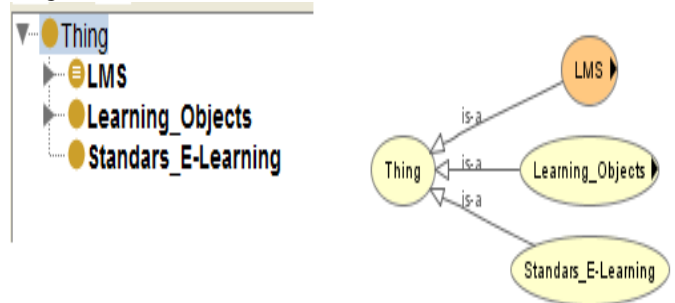


Figure 5. View of more general class in Protege Asserted class hierarchy and OWLViz Plugin

The LMS Class has Users and Tools, the Users has assignerRol, e-Mail, password and UserName how members of him, and the Tools class has the subclasss: Tools Administrations, Communications, Course, Curricula_Desing, Productivity, and Student, Figure 6 shows these classes.

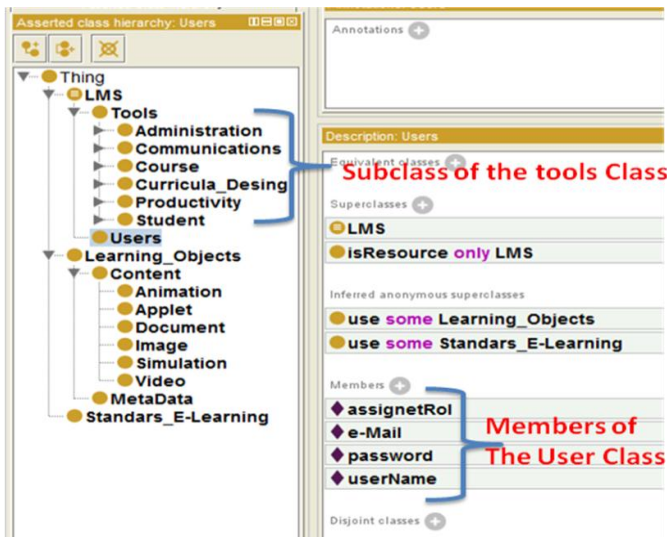


Figure 6. Subclass of Tools and Members of the Users

The Administration class is son of Tools class and have five subclasses; Authentication, Course_Authorization, File_Manager, Hosting_Service, Registry. The Authentication class, have a password of textBox type and userName of textBox type too. The Course_Authorization class, have a assigneRol of comboBox type and userList of list type. The File_Manager class, have a delete of event type, fileList of list type, move of event type, newFolder of button type, rename of event type and uploadfile of explorator type. The Registry class, have a registerOfUsers of link type. And the Hosting_Service class. All this is show in figure 7.

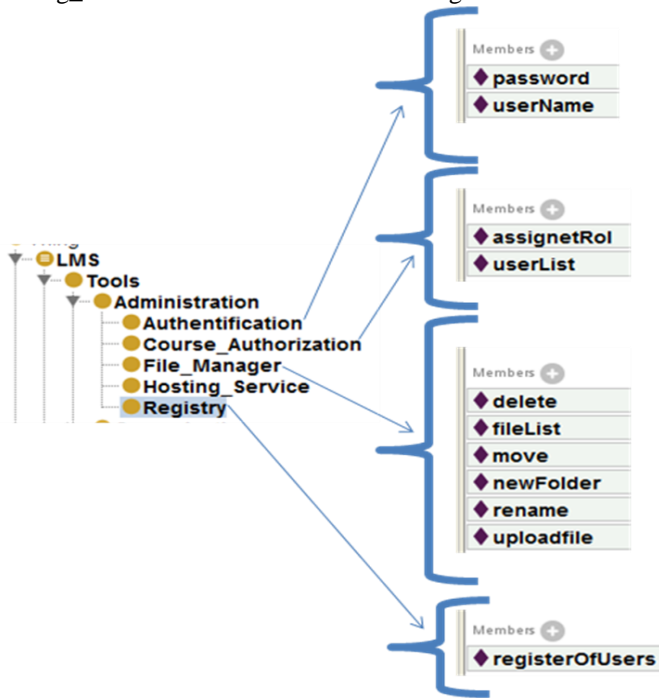


Figure 7a. Sons of the Administration class and your members

The Communications class is son of Tools class and have seven subclasses; Announcements, Calendar, Chat, Forums, News, Note and Wiki. The Announcements class, have a announcementText of textArea type. The Calendar class, have

a showCalendar of calendar type. The Chat class, have a roomName of textBox type, send of button type, sendText of textBox type, textWrittenByUsers of nonEditableTextArea type and userList of list type. The Forum class, have an attachment of file type, forumName of textBox type, message of textBox type and subject of textBox type. The News class, have a newsBody of textArea type, newsDate of textBox type, newsLead of textBox type and newsTitle of textBox type. The Note class, have a message of textBox type, send of button type and userName of textBox type. And the Wiki class, have a wikiEdit of textArea type and wikiView of view type. All This is show in figure 8.

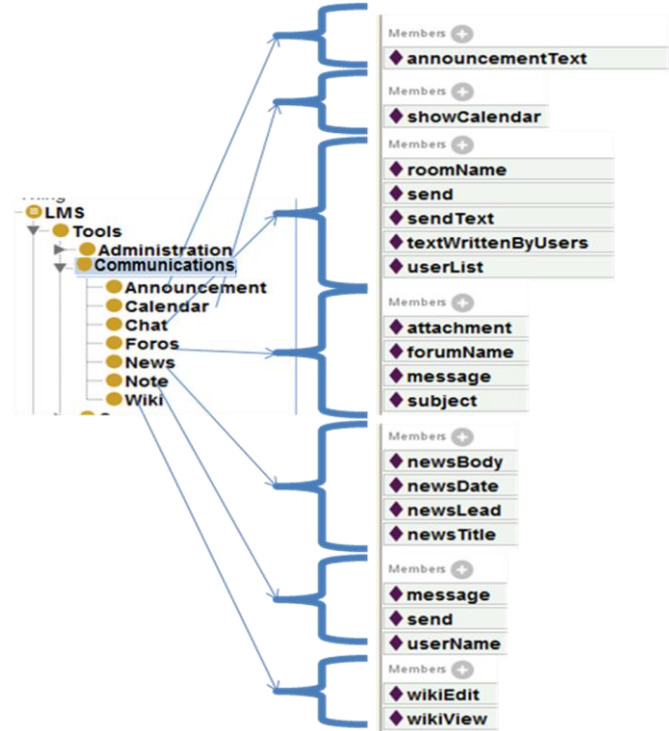


Figure 7b. Sons of the Communications class and your members

The Course class is son of Tools class and have six subclasses; Activity_Calendar, Calification_System, Evaluation_System, FAQ, Glossary and Groups. The Activity_Calendar class, have a activityDay of calendarDate type, activityDescription of textArea type, activityName of textBox type and attachment of file type. The calification_System class, have a activityName of textBox type, commentsCalification of textArea type and valueCalification of textBox type. The Evaluations_System class, have a calificationMethod of textBox type, evaluationDate of calendarDate type, statementQuestion of textArea type, and has three subclasses, Matching class that additionally has, matchingQuestion1, 2, 3, 4 to n of textBox type and matchingResponse1, 2, 3, 4 to n of textBox type. Multiple_Choise class that additionally has, MultipleResponse1, 2, 3, 4 to n of textBox type and percentageHitResponse1, 2, 3, 4 to n of textBox type and True_Or_False class that additionally has, falseResponse of textArea type and trueResponse of textArea type. Other sons

of Course class, are FAQ class, have a viewFAQ of textArea type, Glossary class, have a viewGlossary of textArea type and Groups class, have a userList of list type. All This is show in figure 8.

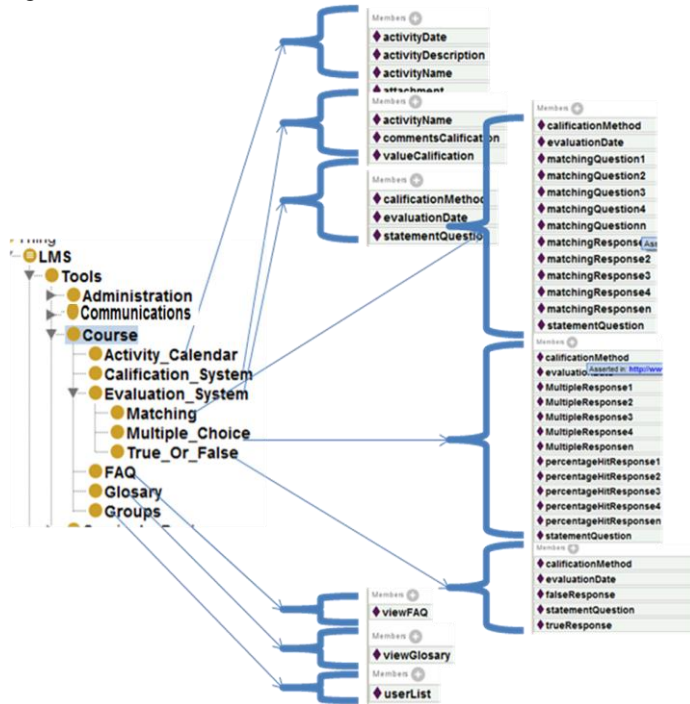


Figure 8. Sons of the Course class and your subclass and members

The Curricula_Design class is son of Tools class and have five subclasses; Course_Templates, Customize_Interface, Educational_Design, Management_Curricula, Share_and_Reuse_Content. The Course_Templates class, have a viewCourseTemplate of view type. The Customize_Interface class, have a viewcustomizeInterface of view type. The Educational_Design class, have a viewEducationalDesign of view type. The Management_Curricula, have a viewmanagementCurricula of view type and Share_And_Reuse_Content class, have a fileList of checkBox type. All This is show in figure 9.

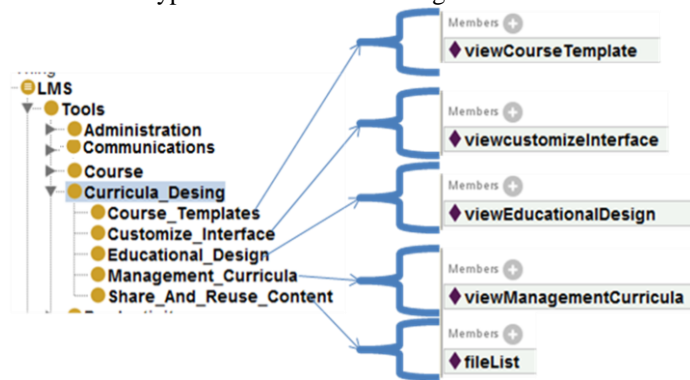


Figure 9. Sons of the Curricula_Design class and members

The Productivity class is son of Tools class and has two subclasses; Help and Search. The Help class, have a viewHelp of view type. The Search class, have a buttonSearch of button type and have textSearch of textBox. All This is show in figure

10.

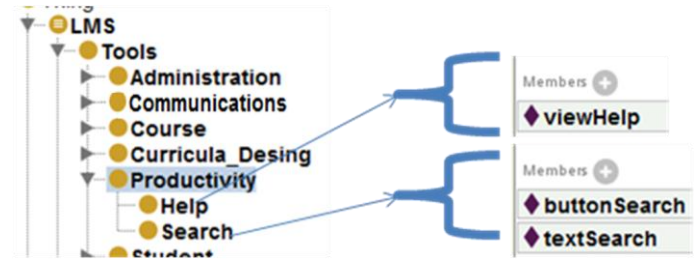


Figure 10. Sons of the Productivity class and your members

The Student class is son of Tools class and has two subclasses; Portfolio and Work_Group. The Portfolio class, have a viewPorfolio of view type and the Work_Group class, have a viewworkGroup of view type. All This is show in figure

11.

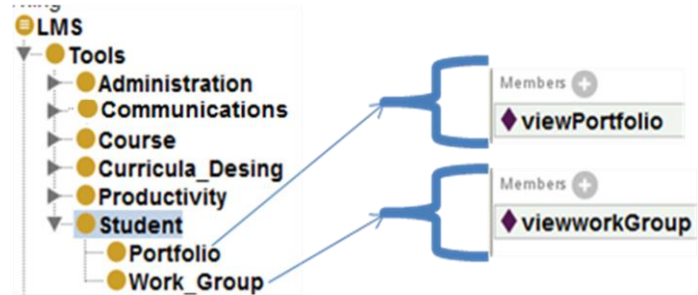


Figure 11. Sons of the Student class and your members

And finally, Standars_E-Learning class and Learning_Objects class, whose subclass are Metadata and Content. The Content class has Animation class, Applet class, Document class, Image class, Simulation class and Video class. This is show in the figure 12.

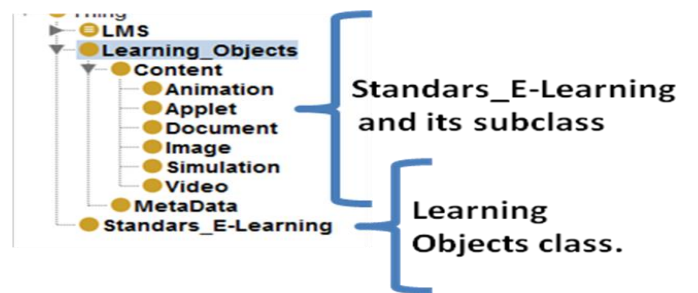


Figure 12. Standars_E-Learning and its subclass. and Learning_Objects

IV. CONCLUSIONS

The complete Model of the Ontology is show in the figure 13, and shows similar elements between the LMS analyzed, and the modules common among LMS, these are : File_Manager, Registry, Announcements, Calendar, Chat, Foros, News, Note, Wiki, Activity_Calendar, Calification_System, Evaluation_System, FAQ, Glossary, Groups, Help, Search, portfolio, Work_Group, Users, and Learning_Objects.

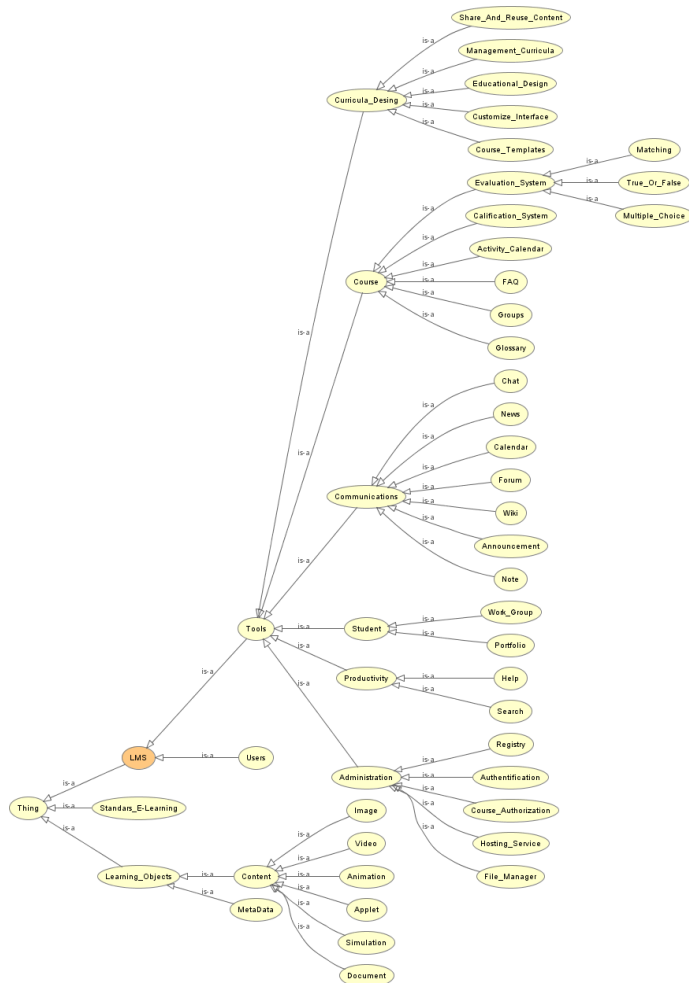


Figure 13. LMS Ontology

The LMSs use a different Standards, but the principal modules its similar in all standards, but the problem is that technology implementation in each LMS is different and this is cause of incompatibility, is necessary create a method for the create modules independent of the platform, one solution could be applied Model Driver Engineering or web services, or other technologies.

The previous step for the perform the ontology, is know the domain of the context, And a good way for this, is interact and navigate for each of the platforms, for exploring the modules that comprise it.

V. REFERENCES

[1] V. Álvarez, "Voice Interactive Classroom, a service-oriented software architecture to enable cross-platform multi-channel access to Internet-based learning," Computer science, University of Oviedo, Oviedo, 2010.

[2] S. Campanella, *et al.*, "E-learning platforms in the Italian Universities: the technological solutions at the University of Bari," *WSEAS TRANSACTIONS on ADVANCES in ENGINEERING EDUCATION*, vol. 5, January 2008 2008.

[3] C. Montenegro, *et al.*, "MODELING AND COMPARISON STUDY OF MODULES IN OPEN SOURCE LMS PLATFORMS WITH CMAPSTOOL," *International Journal of Interactive Multimedia and Artificial Intelligence newsletter*, 2010.

[4] Stanford Center for Biomedical Informatics Research. (2010, Nov). *Protege Ontology Library*. Available: http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library

[5] F. N. Natalya and L. M. Deborah, "Ontology Development 101: A Guide to Creating Your First Ontology," ed. Stanford,, 2005.

[6] S. Microsystems, "Java Code Conventions," in *Java Code Conventions*, ed. Sun Microsystems: Sun Microsystems, 1997, p. 25.

[7] UMBC. (2007, Nov). *Swoogle*. Available: <http://swoogle.umbc.edu/>

[8] S. University. (2005, Nov). *Ontolingua*. Available: <http://ksl.stanford.edu/software/ontolingua/>

[9] D. s. I. E. Office. (2000, Nov). *The DARPA Agent Markup Language Homepage*. Available: <http://www.ai.sri.com/daml/ontologies/>

[10] T. U. N. S. P. a. S. Code. (2010, Nov). *The United Nations Standard Products and Services Code UNSPSC*. Available: <http://www.unspsc.org/>

[11] Netscape. (2002, Nov). *DMOZ Open Directory Project* Available: <http://www.dmoz.org/>

[12] S. R. Heiyanthuduwage and D. D. Karunaratne, "A Learner Oriented Ontology of Metadata to Improve Effectiveness of Learning Management Systems," *International Journal of the Computer, the internet and management*, vol. 14, 2006.

[13] G. Díaz-Antón and M. Pérez, "Hacia una ontología sobre LMS," presented at the Hacia una ontología sobre LMS, Colima, México, 2005.

[14] À. Rius, *et al.*, "Justificación y Descripción del Dominio de Conocimiento de una Ontología para la Formalización y Automatización de Escenarios Educativos.," presented at the El IV Simposio Pluridisciplinar sobre Diseño, Evaluación y Desarrollo de Contenidos Educativos Reutilizables (SPDECE07), Bilbao, Spain, 2007.

[15] H. Srimathi, "Knowledge Representation of LMS using Ontology," *International Journal of Computer Applications*, vol. Volume 6–No.3, 2010.

[16] G. Díaz-Antón and M. A. Pérez, "TOWARDS AN ONTOLOGY OF LMS A Conceptual Framework," presented at the 8th International Conference on Enterprise Information Systems, Paphos - Cyprus, 2006.

[17] C. E. Montenegro Marin, *et al.*, "APPLICATION OF MODEL-DRIVEN ENGINEERING (MDA) FOR THE CONSTRUCTION OF A TOOL FOR DOMAIN-SPECIFIC MODELING (DSM) AND THE CREATION OF MODULES IN LEARNING MANAGEMENT SYSTEMS (LMS) PLATFORM INDEPENDENT," *DYNA*, vol. 78, p. 258, October 2011 2010.