A FORMAL MODEL FOR LEARNING OBJECTS

Claudia V. Pérez-Lezama, J. Alfredo Sánchez

Interactive and Cooperative Technologies Lab, Universidad de las Américas Puebla
Puebla (MÉXICO)
{clavpl03, j.alfredo.sanchez}@gmail.com

Abstract

Formal models are useful both to describe existing objects and concepts and to prescribe how they should be. This paper describes FLOD, a formal model for characterizing learning objects. FLOD details on one hand those components required for a digital resource to be considered a learning object and on the other hand states the type of interactions that should exist between the actors involved in the development process of learning objects. FLOD is divided then into a composition model and a group model. The Composition model is based on layers that formally describe the elements that make up a learning object and the way these elements must be assembled in order to be significant objects (objects with meaning). The group model identifies the main roles and activities involved in the process of generating learning objects through an organizational structure diagram, a task diagram, and a collaboration diagram. Finally, the formalization of both the composition model and the group model is stated using the Z language. Such formal specification made possible a notation where specifications are written, a universe of objects that can be specified and a set of relationships that indicates which objects correctly fulfill the specification. To guarantee the correction of the specification made, the implementation of a system based on such specification is initiated, and a system with well-defined syntax and semantics is obtained. The conclusion is that the formal model as proposed allows the definition of a notation that precisely describes the learning objects on both component and interaction levels.

Keywords: Learning objects, formal model, composition model, group model.

1 INTRODUCTION

Current research on learning objects (LO) is focused on different aspects such as: taxonomy and life cycles, granularity, standards of specifying metadata, storing, tools for development etc. However, one topic rarely taken on in literature is the specification of a formal model that allows on one hand a univocal description of the components required to enable a digital resource to be considered a LO, and on the other hand, states the conditions and properties of the object, which are required to achieve interaction with the object itself. This article states the need to have a formal model that describes LOs on the basis of components and interaction.

For the purposes of this work, the following functional definition has been adopted [1]: “a learning object is an interactive digital entity developed to support significant learning.” The term “interactive” emphasizes the fact that it is the student who controls his/her learning environment, considering that the interactivity in learning is a necessary and fundamental mechanism for acquiring knowledge and for developing cognitive skills [2]. This definition is characterized by promoting active participation of the student as s/he interacts with the LO, thus achieving significant learning that is adjusted to his/her needs, interests and experiences.

Formal specifications use mathematical notation to precisely describe the properties that an information system should have, without putting into detail the way in which these properties are reached [3]. These formal specifications are of great importance because through mathematical notation, the properties that a system should have can be precisely described without excessively restricting the way in which these properties are reached [4]. Through a formal model, a system can comply with and validate functions of restartability, equivalence, visibility, consistency and decidability. Since formal methods allow verification, validation and documentation of a system, a formal model that precisely describes the LOs is thus required, beginning with the basic components that should make up such a digital resource and the interactions that should exist among the actors involved in its developmental process. Subsequently, on the basis of this formal model, it will be possible to develop a system for generating LOs. In this paper, we propose FLOD, which stands for Formal Learning Object Design, a LO characterization model which describes the components required so that a digital resource is considered a LO and states the type of interactions that must exist between the actors

ISBN:978-84-613-9386-2

involved in the development process of the LOs. The document is organized as follows: the works related to modelling LOs are presented in section 2. Section 3 introduces a general architecture for a layered description of LOs, as well as a functional specification in detail of the architecture proposed for FLOD. The formal model using Z language is stated in section 4. Finally, section 5 describes the potential of FLOD through practical validation.

2 RELATED WORK

There are several projects related to generating LOs, some of them explain the composition of OA, others describe the creation of LOs from informal development methodologies. This section mentions the most outstanding of these projects:

A methodology for generating LOs beginning with objects already in existence is presented in [5]. The technique is based on a three-level model which includes: a model of the domain, which represents the concepts covered by the LO, a learner model, which maintains the profile of the learners, and a learning object model, which describes the content of the LO related to the domain model. The proposed methodology describes in detail the adaptive process the student follows to select the LO of interest, beginning with the LOs that are already in the object repository. However, how the LOs that already exist in the repository are composed and generated is not reported.

Farmer and Hughes [6] present an algebraically-based framework for formally constructing learning object assemblies using CASE (situated task analysis model) properties. They propose describing learning objects in terms of their proposed learning function, and according to a number of a priori conditions during selection, sequencing or association. They present a formal approach based upon instance-based learning – where one seeks to establish groups of objects based on some measurable degree of similarity – to dynamically classify learning objects across a set of closed properties. They define a learning object with the CASE attribute types Cognition, Activity, Social Organisation, and Environment, Obj(C,A,S,E) however they don´t detail which are these attributes.

Frosch-Wilke [7] introduces an information model that defines the metadata structure and elements of learning objects. Based on the IEEE LTCS LOM and the IMS CP Information Model they developed an extended information model for learning objects by using object oriented software engineering methods. Therefore they define a learning object as a package of correlated objects. For the description of their model they use the class diagram of the Unified Modeling Language (UML). They establish that the subtype *LO_Atom* represents learning objects which are not decomposable in other learning objects. All other learning objects are sets of learning objects (e.g. lesson, course) with different kinds of relationships. The class *LO_Set* represents these aggregations of learning objects whereas the subtypes of *LO_Set* models different kinds of relationships between learning objects within an aggregation. Every object of *LO_Set* is a learning object itself and can be grouped as an item into larger assemblies of learning objects. The UML description details the composition of LO but not includes information related who assemble the LO.

Hernandez et. al. [8] present a model to describe Learning Objects (LO). The model includes technical description and pedagogical description and also related uses of the learning objects. The model is based on a multi-facet representation of documents by using three ontologies: ontology of theme, ontology of the tasks, ontology of the educational theories and a LOM/SCORM description. The different actors of the system (teachers, learners) and their various tasks are considered. The model includes a semantic representation and search for relevant learning objects by the use of the LOM application profiles. They use a UML notation to describe the various aspects of representation of a learning object but not use a formal model.

Knight et. al. [9] present an ontology based approach to integrate learning designs and learning object content. They define a three part conceptual model that introduces an intermediary level between learning design and learning objects called the learning object context. They then use ontologies to facilitate the representation of these concepts: LOCO is a new ontology for IMS-LD, ALOCoM is an existing ontology for learning objects, and LOCO-Cite is a new ontology for the contextual model. The ALOCom ontology defines a number of concepts for depicting different types of content chunks in terms of their granularity (Content Fragment, Content Object and Learning Object), learning/educational role (Definition, Example, Keyword, etc.), and presentation context (Table, Image, Video, etc). They report informally who develop the learning design, the learning design context and the learning objects.

Santacruz [10] describes a model for generating, assembling, and reusing the LOs. It consists of a content model, an assembly process for the LOs, and a reutilisation process of the LOs. The content model is made up of Information Units, Content Units, and Didactic Units, each component of which comprises a multilayer structure with a distinct granularity level (N0, N1 and N2). The assembly process describes the way in which the different types of components of the content model are related. This process is carried out through the use of a mechanism, based on the application of ontology, known as OntoGlue. Santacruz formally describes the LOs from a conceptual viewpoint through the assembly process of the objects.

In the model of CISCO [11] each LO known as RLO (reusable learning object) is based on a simple objective, derived from a specific task. Each RIO (reusable information object) is based on an objective that supports the RLO objective. Each RIO is defined as a concept, a fact, a process, a principle, or a procedure, suitably labeled. Various RIOs – from 5 to 9 – are combined together to create an RLO. An RLO is the sum of the RIOs necessary to complete an objective. Each RIO can include an introduction, a summary, and an evaluation and is designed for complying with a specific objective derived from a specific task.

The aggregation content model SCORM [12] is composed of the following elements: *Assets*, which are the most elemental unit of electronic representation of information; they can be text, sound, animation, a web page, etc. *SCO*, (Sharable Content Object), which is a collection of one or more assets that can be used by the SCORM RTE (Run Time Environment) to communicate with an LMS; one example of SCO is a topic. *Content Organization* is a structure of instruction units (activities) that will be linked to resources (assets and SCO) to jointly carry out a learning experience.

Learnativity Content Model [13] defines a hierarchy structure with five content levels; the smallest level of this model has “raw media” stored at the pure data level. The second level includes the information objects, formed by a set of those data elements that create a chain of reusable information, and meaning, independent of the medium. Based on a simple objective, the data are thus selected and assembled on a third level of specific application objects or learning objects. The fourth level refers to Aggregate Assemblies which deal with larger objectives. This level corresponds to lessons or chapters. The lessons or chapters can be assembled in larger collections, such as courses or complete curriculums. The fifth level refers to collections.

With the exception of the model proposed by Santacruz, the methodologies presented in this section do not provide formal specification that clearly describes what the LO is, what its components are and how the actors interact in the development process of the LO with these components. The following sections define a formal model for characterizing LOs from their basic components and on the basis of the interactions of the actors involved in the process of their development.

3 THE LO REFERENCE MODEL

This section describes in detail the minimum components necessary for considering a digital resource to be a LO; how it can be generated from an assembly or from a combination of specific digital resources, and what the general architecture is for layers as proposed for the description of LOs; in addition, it includes a functional description of the architecture proposed for FLOD.

3.1 LO structure

In accordance with the definition of a LO, as proposed in section one, a LO represents the knowledge acquired after understanding, applying, synthesizing, and evaluating a specific topic. This includes associated knowledge, requirements for understanding it, objectives or specific educational goals, exercises or practices to allow experimentation on the issue stated in the object, and mechanisms for evaluating it and providing feedback. Thus, a LO must be composed of the following elements:

A. *Learning objectives*: The educational goals that must be reached after using the LO.

B. *Competencies/ Skills:* The abilities, attitudes and values acquired after interacting with the LO.

C. *Prerequisites:* The knowledge or competencies the learner should have acquired previously in order to be able to take advantage of the LO.

D. *Content:* The digital resources that make up the LO, including their sequencing and the navigational information on such resources.

E. *Practice:* The tasks the learner must perform while interacting with the LO.

F. *Evaluation:* The mechanisms designed to measure knowledge acquired after interacting with the LO.

G. *Metadata:* Predefined identifiers that allow LO storage, organization and searching.

3.2 The layered model

A conceptual model defined in layers, describes the elements that comprise the LO and how such elements can be assembled to obtain meaningful resources. The model is composed of digital objects (DOs), information objects (IOs), learning objects (LOs) and learning collections (LCs). The LOs are created beginning with the assembly of DOs, COs, or a combination of such elements.

A. *Digital Objects (DOs).* A digital object is a simple object, denoted as DO. Each DO contains a unique resource, possibly consisting of multimedia, which the user may access individually. A DO by itself does not complete a learning objective nor does it provide the user with knowledge about a specific subject, and thus context is required to confer educational significance to it. Individually, each DO has no educational significance, nor does it have specific application context. Examples of digital objects include texts, videos, audios, images, graphs, tables, figures, animation, and illustrations.

B. *Information Objects (IOs).* An IO is composed of various DOs assembled in a template to ensure a sequence or a logical order. Since the IOs are compound entities, they may require a certain prior knowledge in order to be comprehended. Examples of IOs are: definitions, examples, exercises, concepts and summaries. By assembling various DOs to make up an IO, a value of this object will be identified as knowledge for the individual, under a certain context. Depending on the template selected, DOs can be generated in types such as: content, practical or evaluation.

C. *Learning Objects (LOs).* A LO is formed from the combination of IOs. A LO represents knowledge acquired after understanding, applying, synthesizing and evaluating a specific subject. A LO includes content or associated knowledge, has certain requirements for its comprehension, and generates specific competencies from its use. It also includes specific learning objectives, exercises or practices that ensure understanding of the subject and an evaluation or feedback mechanism that makes it possible to measure acquired knowledge after having interacted with the object. Examples of a LO are demonstrations, principles, procedures and processes.

D. *Learning Collections (LCs).* A LC is comprised of LOs. A LC includes associated knowledge, the requirements necessary for its understanding, specific objectives, exercises or practices on the subject, and partial evaluations of such subjects. Similar to LOs, after interacting with a LC, specific competencies are developed. Examples of LCs are tutorials, sections, chapters, courses, units and topics.

3.3 LO characterization model

FLOD must, on one hand, describe the components required so that a digital resource will be considered a LO (on the basis of the structure proposed in section 3.1), and on the other hand state the type of interactions that must exist between the actors involved in the development process of the LOs. FLOD is divided then into a composition model and a group model. Fig. 1 presents a scheme showing the conceptual models that enable precise description of the learning objects, on both the component and the interaction levels.

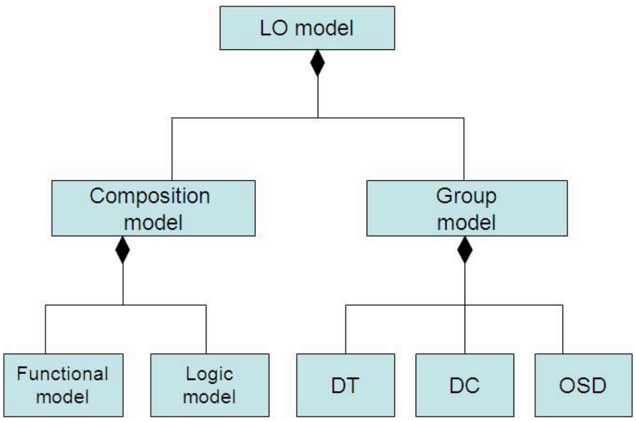

Fig. 1: FLOD scheme

The “Composition Model” presents all the aspects related to the composition of the LO and the “Group Model” presents the behaviour of the actors in the system, the latter being in charge of both representing the aspects of interaction among the users of the system and the description of roles and tasks assigned to each role.

In this scheme two conceptual models are observed: on one hand, there is the composition model based on layers (see section 3.2), which formally describes the elements making up a LO and the way in which these elements must be assembled to signify meaningful objects, and on the other hand, the group model which identifies the main roles and activities involved in the process of generating LOs.

The composition model in turn is made up of a functional model that describes in UML notation the composition process of the LO as well as a logical model that formally describes, by way of the Z language, how the LO is defined. The group model was developed from a previous identification of roles and tasks, along with the identification of the interactions among such users; in this way, a description of such was made, and three diagrams were obtained [14]: a task diagram (TD), a collaboration diagram (CD) and an organizational structure diagram (OSD).

Presented below in detail are both the functional model and the group model. The logical model will be covered in section 4.

3.4 The functional model

The functional model (Fig. 2) describes in UML notation the composition of a LO. This model is based on the layered model as stated in the section 3.2.

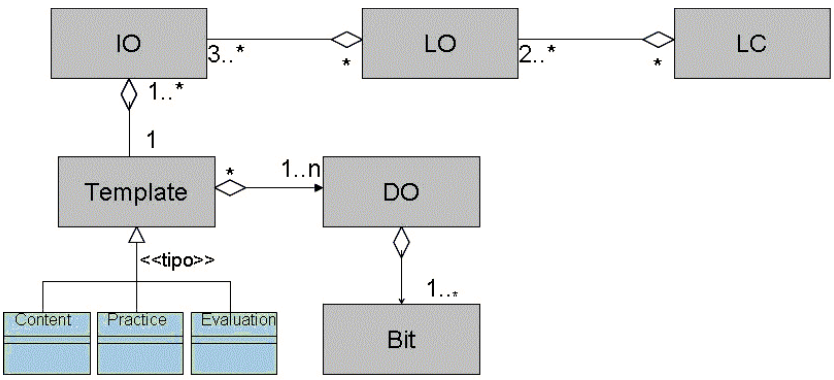

Fig. 2: Functional Model

This scheme states that a digital object (DO) is a set of bits, which, when grouped together, can comprise an image, audio, text or video. A template is a predefined format, which allows incorporation of one or more DOs in different sections within the template, achieving in such a way the composition of a particular information object (IO), the type of which can be content, practical or evaluation. Finally, various IOs must be grouped together to make up one LO.

3.5 The group model

Traditionally, this is a user model, which is used to represent the different tasks that a user may perform in a system. However, in a collaborative environment, where various users share the same scenario, the user model must extend itself to a group model in order to include social and organizational aspects of the activities performed by the users. This section will describe the details of each of the diagrams forming the proposed group model, with the basic aim being to model the behaviour of the users in the process of generating LOs by describing the organizational structure, the collaboration diagrams among the users, and finally the diagrams of role specific tasks. Firstly, the identified roles and assigned activities of each role are defined in this way:

Designer: This role is played by the facilitator, who defines the instructional design of the course, specifies which LOs must be developed for each subject and evaluates the content and usefulness of the LOs created by the developer.

Developer: This role is played by the learners, who create the LOs and their self-evaluation.

Co-evaluators: Those learners who evaluate and use the LOs produced by the developers

A. Organizational Structure Diagram

The diagram for the organizational structure (OSD) of the actors in a system represents the distribution of its different organizational elements; in other words, how the actors of the system are organized, how they relate (in structural terms) to form groups and hierarchies, and what roles they perform, etc. Fig. 3 shows the proposed OSD.

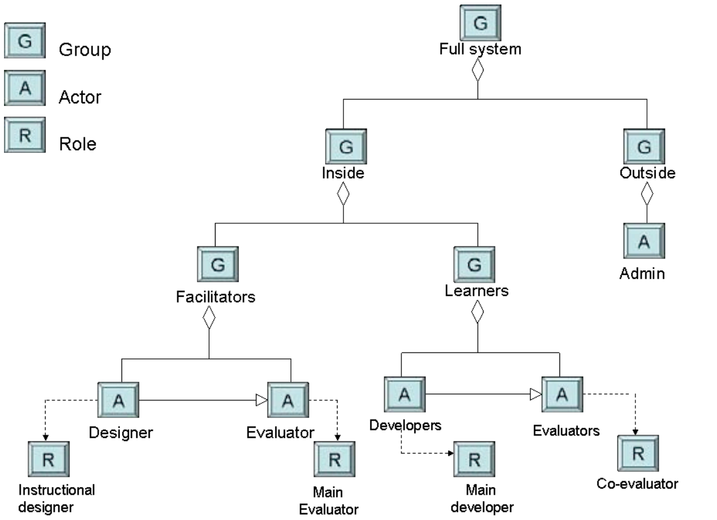

Fig. 3: Organizational System Diagram (OSD)

This diagram presents the actors involved in the development process of LOs and the roles they acquire within the community. It also shows the hierarchical structure of the main roles. As can be appreciated, for example, a facilitator who belongs to the group of facilitators may act as a designer and/or an evaluator. The designer acquires then the role of instructional designer and the evaluator acquires the role of main evaluator. As for learners, some will act as developers and others as evaluators; of the developers, one will play the role of main developer of the LO, and from among the evaluators, one of them will be co-evaluator.

B. Task diagram

The task diagram (TD) generally details the activities that must be performed in a system, and the actor who is responsible for carrying them out. Specifically, it concerns the interaction between each actor and the system. Fig. 4 describes the main tasks of the system and who is responsible for doing each of them.

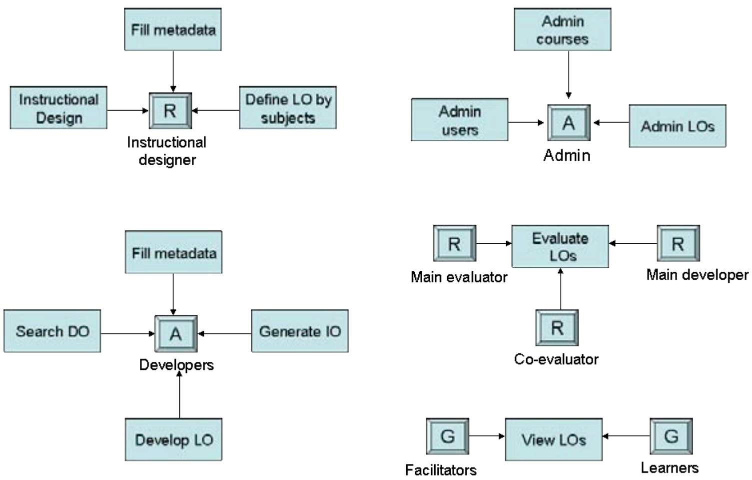

Fig. 4: Task Diagram

As can be observed in the TD, some activities are individual, meaning they are assigned to a single actor; for example, instructional design, or looking for DOs, while other tasks, such as filling metadata or evaluating the LO, are assigned to a group of actors.

C. Collaboration diagram

The collaboration diagram (CD) represents the collaborations established between the different members of an organizational structure, meaning the interactions that exist between actors of a system within the system itself. The elements employed to form a CD are the same ones as those used to represent the organizational structure by means of the OSD. A CD represents the collaboration that takes place among the participants of a system organized as shown in the OSD. As far as the possible connections between the elements, cooperative interaction relationships are stated, as described by the interaction between two participants through the system. Fig. 5 shows the CD of the system.

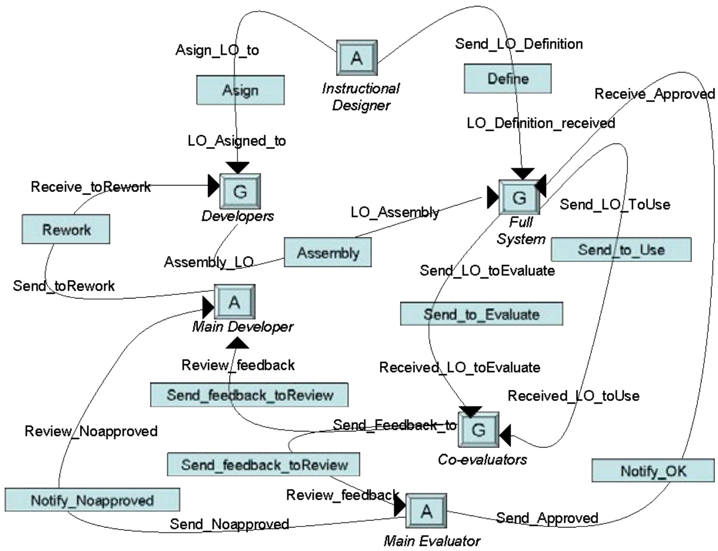

Fig. 5: Colaboration Diagram

The figure shows, for example, that the main developer receives two messages and sends one. The first message received is *Review_Noapproved*, and was sent by the main evaluator indicating that the LO sent to evaluation was not approved and must be reworked. The second message received is *Review_feedback,* and this message was sent by the group of co-evaluators to indicate that the LO was already evaluated, and that the evaluation made by each of the co-evaluators must be revised. The message the main developer sends to the developers is *Send_toRework,* and it means that the LO must be reworked.

The following section details the logical model that formalizes the LO composition proposal.

4 FORMALIZATION OF THE LOGICAL MODEL USING Z

Usually, formal methods are centred around a notation, called formal specification language [15]. Such a notation has a formal semantics, which makes it possible to unambiguously denote the meaning of a specification expressed in that language. A language of formal specification consists of a syntactic domain (the notation in which the specifications are written), a semantic domain (a universe of objects that can be specified) and a satisfaction ratio (ratio indicating which things in the semantic domain will satisfy certain specifications in the syntactic domain). There are two different techniques for formal specification of the system: the property-oriented techniques (such as CASL) are program specifications consist mainly of logical axioms; there is usually a logical system in which there is a predominant role describing the properties that the functions need in order to complete the interconnections and the model-oriented techniques (such as VDM and Z), what describe the behaviour of the system [16].

To formalize our architecture into layers of learning objects, we will use the Z language. The Z language is a formal specification language used to describe and model computing systems. It allows clear specification of computer programs and formulation of tests on the behaviour of such programs.

Z has been developed at Oxford University since the late 1970's by members of the Programming Research Group. Z includes a schema notation to aid the structuring of specifications. This provides the framework for a textual combination of sections of mathematics (known as schemas) using schema operators. Many of these match equivalents operators in the mathematical notation. The idea of an abstract Z specification is to describe what a system does rather than how it does. The Z notation is designed to be expressive and understandable (by humans) rather than executable (by computers). Z is based on a standard mathematic notation used in the Axiomatic set theory and first-order predicate logic [17]. Z contains a standardized catalogue of mathematical functions and predicates that allows the modelling of a system by representing its status, a collection of status variables and their values, and some operations that can change its status.The static aspect of a class is formalized by a Z state schema called the Class Schema. A Z schema consist of a declaration part in which variables are declared, and a predicate part that contains a predicate constraining variable values [18]. The attributes and object identifiers of a class are represented by variables in a class schema. The type name of the attribute in the class schema corresponds to the type name of the attribute in the class. A Z basic type is a set of primitive elements. If no type is associated with an attribute, then the capitalized name of the attribute is used as the type name in the Z class schema and declared as a Z basic type. The formalization of our model using Z schema is showed in the Fig. 6:

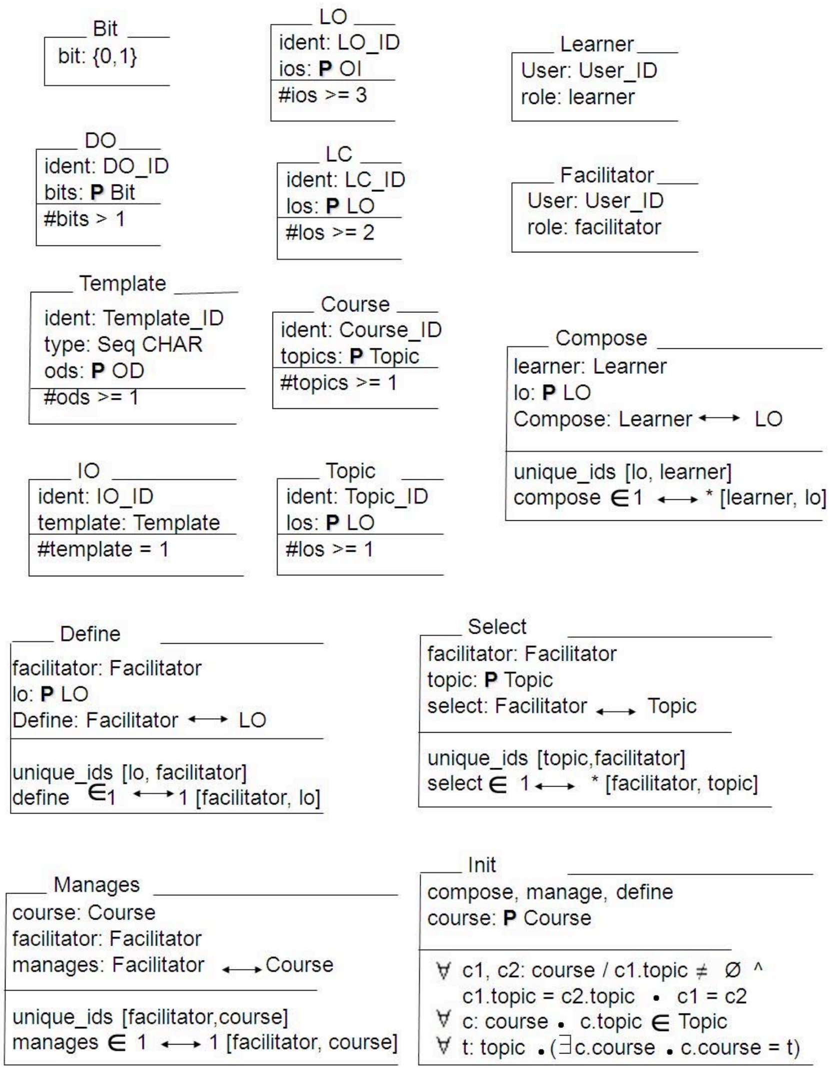

Fig. 6 Logical model in Z language

The presented schemes formalize each one of the elements of the functional model described in section 3.4 this includes the aggregation and specialization relationships. Regarding the group model are formalized the OSD (organizational structure diagram) and the DT (diagram of tasks) described in sections 3.5.1 and 3.5.2 respectively, this formalization includes the aggregation relationships and multiplicity indicators.

5 POTENTIAL OF THE MODEL

FLOD, presented and described in the above sections, allows components required for constructing a LO to be formally defined, and describes in detail the life cycle of the LO from its conception to its implementation, clearly establishing the tasks and roles involved during the development process of the object, and finally presenting the interactions that occur among such actors. In order to verify that FLOD will characterize and describe any learning object, it is necessary to validate it. Validation will be made by applying FLOD in different developments found in literature, the aim being to verify if our model allows characterization of objects developed through other methodologies and proposals. Table 1 summarizes how each model represents what a LO is.

Table 1: LO models

FLOD	DO	IO	LO	LC
Bouzeghoub	Web page, file or program	Operator-node, LO-node, query-node	ILO	
Farmer and Hughes		Obj(C,A,S,E)	A = Z Obj	
Frosch-Wilke		LO_atom	LO_Set	
Hernandez et. al.	Assets		SCO	CA
Knight et. al.		LO	LOC	
Santacruz	Information Unit	Content Unit	Didactic Unit	
Learnativity	Raw media	Information Object	Learning Object	Aggregate Assemblies
SCORM	Assets		SCO	Content Aggregation
CISCO	Content items	RIO	RLO	

As shown in the table, FLOD can characterize completely the objects developed with other methodologies in addition to FLOD formally describes the actors involved in LO development and the tasks assigned to each actor. This feature is not present in the methodologies evaluated.

FLOD can equally serve as guide for the development of LO authoring systems because to unequivocally defines the components of the LO and formally establishes who should participate during the LO generation.

6 CONCLUSIONS

The formal model as proposed allows the definition of a notation that precisely describes the learning objects on both component and interaction levels. FLOD defines the LOs and its components.

An important contribution is the proposed group model, since it defines the roles in detail, as well as the tasks assigned to each role, and the interactions among roles, through the organizational structure diagrams of tasks and collaboration, respectively. Another interesting contribution is the use of the Z language to formalize the composition model of the LO. Finally, it has been observed that the model adjusts itself to the different proposals found in literature; therefore, we may conclude that this model characterizes and describes any learning object.

Based on the proposed formal model, an object authoring tool is being implemented. As future work the system will be tested to verify that the syntax and semantics are well defined.

REFERENCES

[1] Pérez-Lezama C. V. (2007). A collaboration model for rapid generation of learning object repositories. Proceedings of the Doctoral Consortium. ENC 2007. Octubre, Morelia, México

[2] Barker, P. (1994). Designing interactive learning. In Design and Production of Multimedia and Simulation based Learning Material. T. de Jong and L. Sarti, eds. Kluwer Academic, Dordrecht.

[3] Spivey J.M. (1988), Introducing Z: A Specification Language and its Formal Semantics, Cambridge Univ. Press.

[4] Ledru, Y. (1996). Complementing semi-formal specifications with Z. In Proceedings of the 11th Knowledge-Based Software Engineering Conference (September 25 - 28, 1996). KBSE. IEEE Computer Society, Washington, DC, 52.

[5] Bouzeghoub A., Defude B., Duitama J. and Lecocq C (2006). A Knowledge-Based Approach to Describe and Adapt Learning Objects, International Journal on ELearning; ProQuest Education Journals, p. 95.

[6] Farmer R and Hughes B., (2005). A Classification-based Framework for Learning Object Assembly. Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)

[7] Frosch-Wilke D.,(2004). An Extended and Adaptable Information Model for Learning Objects. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)

[8] Hernandez N., Mothe J, Ralalason B., Ramamonjisoa B. and Stolf P (2008). A Model to Represent the Facets of Learning Objects. Interdisciplinary Journal of E-Learning and Learning Objects Volume 4, 2008. Interdisciplinary Journal of Knowledge and Learning Objects

[9] Knight, C., Gašević, D., and Richards, G. (2005). Ontologies to integrate learning design and learning content. Journal of Interactive Media in Education 2005(07). [jime.open.ac.uk/2005/07].

[10] Santacruz L. (2005). Automatización de los procesos para la generación, ensamblaje y reutilización de los objetos de aprendizaje. Tesis doctoral. Universidad Carlos III de Madrid. Escuela Politécnica Superior. Departamento de Ingeniería Telemática.

[11] Cisco Systems, Inc. (2003). Reusable Learning Objects Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches. [www.cisco.com]

[12] Advanced Distributed Learning Initiative (2001). Sharable Content Object Referenced Model. The SCORM Overview.,. Ver. 1.2. editor Philip Doods. October

[13] Hodgins W. (2004) The Future of Learning Objects. e-Technologies in Engineering Education: Learning outcomes providing future possibilities

[14] Penichet V.M.R, Lozano M.D., Gallud J.A., Tesoriero R., (2007). Análisis de un Modelo de Procesos CSCW. Organización, Roles e Interacción Persona-Ordenador-Persona. Interacción 2007 – II Congreso Español de Informática (CEDI 2007), España, Sep.

[15] Plat N., Katwijk J., Toetenel H.,(1992).Application and benefits of formal methods in software development. Software Engineering Journal September

[16] Wing, J. M. A specifier's introduction to formal methods. Computer, 23(9):8–24, September 1990.

[17] Inverno M., Ribeiro G.R., Howells P., (1996). A Formal Framework For Spefifycing Design Methods. Proceedings of the 29th Annual Hawaii International Conference on System Sciences.

[18] Shroff, M. and France, R. B. 1997. Towards a formalization of UML class structures in Z. In Proceedings of the 21st international Computer Software and Applications Conference (August 11 - 15, 1997). COMPSAC. IEEE Computer Society, Washington, DC, 646