AN AUTHORING ARCHITECTURE FOR ANNOTATING EDUCATIONAL CONTENTS

Domain, Sequencing and Content-Repository Ontologies

José M. Gascueña, Antonio Fernández-Caballero and Pascual Gónzalez
Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos &
Instituto de Investigación en Informática de Albacete, Campus Universitario s/n, 02071-Albacete, Spain

Keywords: E-learning, Ontology, Learning objects.

Abstract: E-learning platforms available nowadays are mainly centred in supporting management tasks, but they do not include or even consider in a too satisfactory way the adaptation to student's profile, the reusability of educational materials, or the efficient search into educational materials. By combining the paradigms of ontologies and learning objects in authoring tools it is possible to annotate educational contents for generating personalized material. The characteristics introduced in this paper are the learning style best suited to the student, the device used to access the contents and the skill to be developed when using the material. The general architecture of the proposed tool is fundamentally composed of three different and interrelated ontologies: *domain, sequencing* and *content-repository ontologies*, where all knowledge about which educative content is taught, how it is taught and how it is organized is respectively stored.

1 INTRODUCTION

Nowadays a great number of e-learning platforms are available (e.g. WebCT, Moodle, ATutor). They are mainly centred in supporting management tasks such as to administrate users and groups, or to store educational materials (Shimic, Gasevic & Devedzic, 2006). Nevertheless, these platforms generally do not include or even consider in a too satisfactory way some important issues. For instance, they offer the same materials and activities to all students; thus, the content shown is not always adapted to their knowledge, preferences and objectives, that is, to the students profiles. Moreover, there are few possibilities of reusing the educational materials due to their low granularity. Indeed, frequently it is not possible to make a search directly into files fragments (e.g. an image, a table, a graph, a schema, or a summary) without having to open and review the documents offered.

In this paper, an ontology-based authoring architecture for annotating educational contents is proposed to solve some of the problems commented above. The architecture generates adaptable material according to characteristics such as the most appropriate learning style for a given student, the device used to access to contents, and the skill that is sought to be developed when using it.

An ontology is an explicit specification of a conceptualization (Gruber, 1993); it allows defining explicitly and formally the concepts and relations that appear in the application domain. The use of ontologies in the e-learning community is useful for several reasons. (1) It helps authors in building consistent and well-elaborated material, (2) it provides facilities to construct enhanced search engines for finding more relevant learning material, (3) it enables individual adaptive delivery of learning services as well as dynamic navigational support, and, (4) it provides the means for constructing a reference model for learning resources (Leidig, 2001).

In our research only resources that may be transmitted through the Internet are considered. Thus, among the different definitions found of the concept of learning object (McGreal, 2004), the more appropriate definition to our objectives is the one provided by Wiley (2001): "Any digital resource that can be reused to support learning". The reusability characteristic inherent to learning objects is one of the reasons for the increasing interest in adopting this paradigm. It decreases time and effort required to produce learning objects in a significant

M. Gascueña J., Fernández-Caballero A. and Gónzalez P. (2007).
AN AUTHORING ARCHITECTURE FOR ANNOTATING EDUCATIONAL CONTENTS - Domain, Sequencing and Content-Repository Ontologies.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - HCI*, pages 53-60
DOI: 10.5220/0002353900530060

manner. We consider all the resources defined at the level of paragraph, image, table, diagram, audio, and so on, as learning objects to obtain a high grade of reusability.

In our architecture, learning objects are labelled according to learning styles theory in order to allow delivering didactic materials suitable to all students' learning styles. Due to the current proliferation of new and different devices, it seemed also useful to present different content formats, based on hardware, software and network connection features of the devices used by the students.

The World Wide Web Consortium (W3C) has developed OWL (Web Ontology Language) (Dean et al., 2004) to increase an expressive power that the earlier existing ontology markup languages – XML, RDF and RDF-Schema, among others – did not have. For instance, OWL incorporates Boolean combinations of classes (union, intersection, and complement), disjointness of classes, cardinality restrictions, special characteristics of properties (transitive, unique, inverse), and local scope of the properties (Antoniou & van Harmelen, 2004). All OWL ontologies introduced in this paper are encoded in OWL language and the Protégé framework has been selected to edit/construct them.

The article is structured as indicated next. In section 2 the works carried out in the last years related to the learning objects paradigm and based on ontologies are described. In section 3 the proposed architecture for annotating educational contents is introduced in extensive. Finally, some conclusions are offered.

2 RELATED WORK

In scientific literature several proposals related to the use of ontologies in educational environments may easily be found. An ontology-based metadata to achieve personalization and reuse of content in AdaptWeb project has been described (Silva & Palazzo, 2004). In mentioned project, DAML+OIL language is used to represent the ontology; but unfortunately, an inexpert user in ontologies is not provided with any user friendly interface tool to populate it. The architecture of the adaptive learning system proposed in Duitama (2005) is based on the AHAM reference model for adaptive hypermedia applications, and RDF Schema is used to represent knowledge models that compose it, which again entails the previously commented limitations. Another approach (Santacruz-Valencia et al., 2005) provides ontology-based mechanisms to carry out the leaning objects assembly, where the knowledge (requirements and competencies) associated to learning objects is kept in mind to determine if it is possible to assemble them. In this work features such as learning style, software and hardware are not considered for the purpose of the assembly. TANGRAM is a learning environment in the domain of Intelligent Information Systems (IIS) useful to authors and students interested in this domain. In this case, Semantic Web technologies and ontologies in particular are used. This environment is centred in a concrete domain, so it is not enough generic (Jovanovic, Gasevic & Devedzic, 2006). In another work (Baloian et al., 2004) a mechanism for information retrieval not only taking into account the student profiles but the equipment issues is presented. Lastly, in (Ronchetti & Saini, 2004) an architecture to help students to find materials that present different points of view or different ways to explain concepts is proposed, but again it does not make use of Semantic Web technologies.

3 FUNCTIONAL DESCRIPTION

The proposed architecture is generic, that is to say, it is not associated to a course in particular. But rather it defines the characteristics that appear in any course. The aim is to create an "instance" for each particular course from the general ontologies (mainly, *sequencing ontology* and *domain ontology*, as described next) included in the tool.

Basically, the authoring tool architecture proposed consists of several distinct ontologies (see Figure 1). The *domain ontology* describes concepts, and relations between them, that appear in the domain of a course. The *sequencing ontology* defines the possible learning paths that can be given through the concepts defined in the *domain ontology*. The *contents-repository ontology* includes metadata to describe learning objects, as well as the relationships between the different kinds of objects, used to teach the concepts belonging to the courses inserted with the tool. In this ontology, besides the definition of classes and relations among them, there will be individuals when the ontology is populated with the authoring tool. The *mapping ontology* allows establishing relations between the concepts (instances) included in the *domain ontologies* of different courses. Lastly, the *administration ontology* includes information about the authors of the contents (personal information and courses that they are authorized to access) and where the ontologies included in the tool are located.

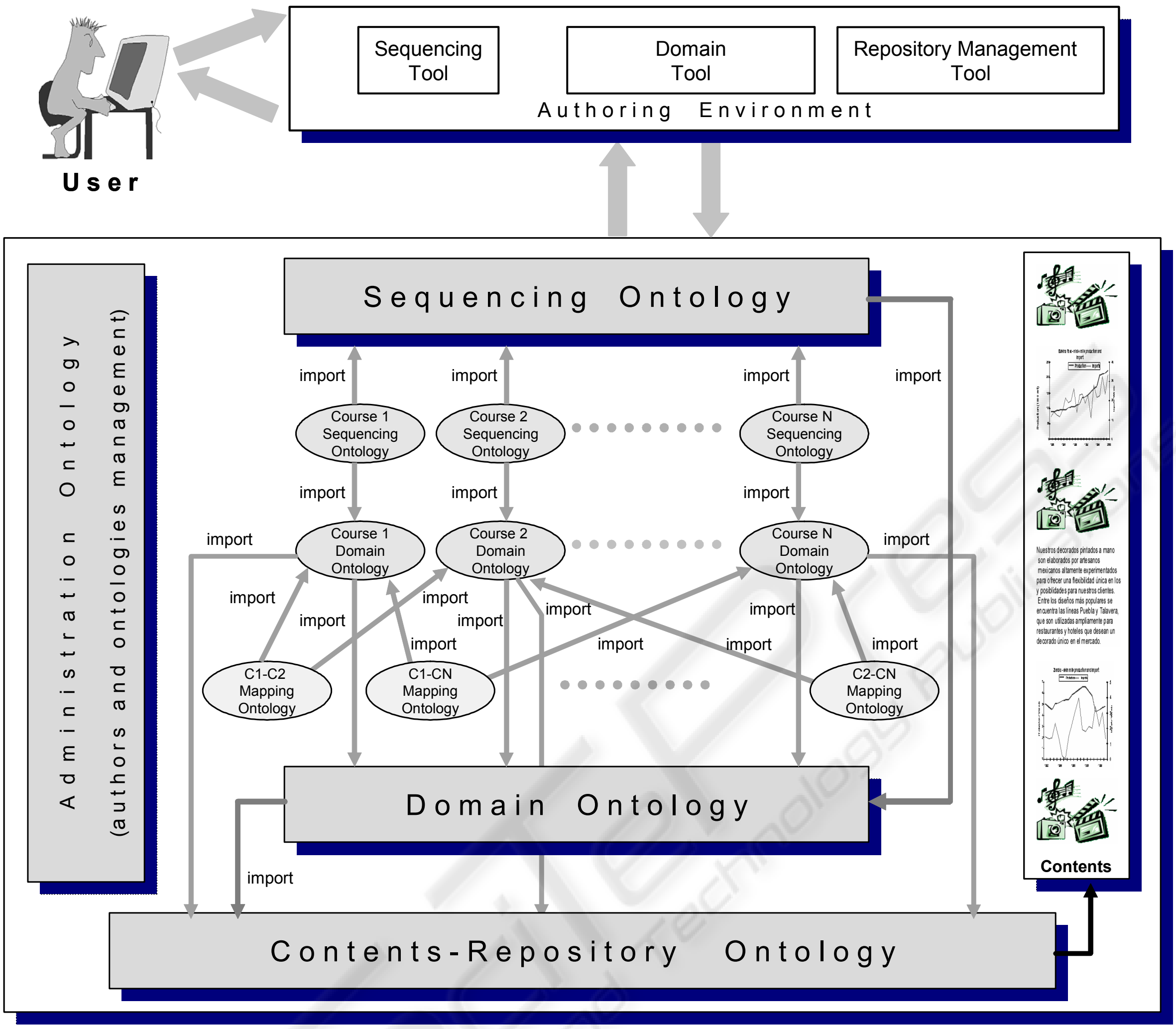

Figure 1: General layout of the authoring tool.

The courses managed with the tool are produced by creating particular *sequencing* and *domain ontologies* for each included course, besides associating the corresponding learning objects (annotated in the *contents-repository ontology*) for teaching the course matter. This separation allows a better reusability of the knowledge (e.g. in those cases in which we are in front of a course offered in different degrees whose contents differ very little). The *domain ontology* for a given course is imported from the "generic" *domain ontology* (it only contains class definition and relations, but it does not contain individuals) and the *contents-repository ontology*, both shown in Figure 1. This ontology will include the concepts of the course (individual of the class Concept) and the relations among them, and also it will point to the learning objects that can be used to learn them.

Now the *sequencing ontology* for the course imports the "generic" *sequencing ontology* (it only contains class definitions and relations, but again it does not contain individuals) and the *domain ontology* particular for this course in order to define the possible learning trajectories that the students can follow through the concepts belonging to the specific course (individuals of class *Concept*).

The author of the contents is provided with some useful utilities – a domain tool, a sequencing tool and a repository management tool – that allow him to create, delete and modify ontologies associated to the courses included with the tool. He will also be able to populate the ontologies and to manage them in a user friendly way, with no need to having any knowledge on the format of the storage chosen.

3.1 Domain Ontology

The *domain ontology* is graphically shown in Figure 2. Class *Course* represents the subjects created within the tool. For instance, Multi-agent Systems, Operating Systems and Software Engineering are some individuals of this class. The class contains two data type properties, namely *courseName* and *courseDescription,* the title and a brief description of the course, respectively, and one object property *cHasObjective* that points to the objectives to reach (class *Objective*) through the course.

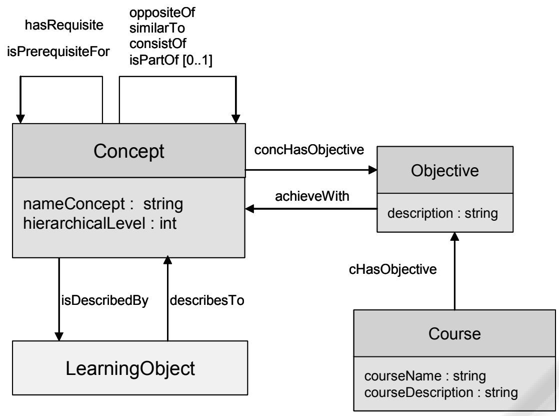

Figure 2: Domain ontology.

The concepts constitute the knowledge of the domain under study and they are collected in class *Concept*. This class contains two data type properties, *nameConcept* to identify the concept and *hierarchicalLevel* to represent the level where the concept is found in the concepts hierarchy. Notice that level 0 is assigned to the root concept. There are more object properties introduced later on to describe the possible relationships among the domain concepts.

Property *consistOf* is aimed to define a concept hierarchy, and therefore, to establish a relationship among a concept and its sub-concepts (e.g. we are able to define chapters, sections, subsections and terms which are under subsections), until reaching an atomic concept which – from the point of view of the teacher – does not need to be decomposed any more. A concept must have at least a parent concept (property *isPartOf* with maximal cardinality restriction). Properties *similarTo* and *oppositeOf* enable mapping a concept to other concepts that have the same or different semantic meaning, respectively.

In order to indicate concept restrictions through which it is possible to advance/go back to/from a given concept other properties are needed. *hasRequisite* and *isPrerequisiteFor* (its inverse) allow to point to concepts that must be known before starting to study a determined concept, and the concepts for which it is a prerequisite, respectively – some conditions should be fulfilled to access the study of the concepts.

Object property *isDescribedBy* (class *Concept*) points to digital resources that explain a concept or assess the knowledge stored about it. On the other hand, object property *describesTo* is included in *LearningObject* class to indicate which concepts a learning object is related to.

3.2 Sequencing Ontology

The *sequencing ontology* defines the possible learning paths that can be followed to learn the concepts defined in the *domain ontology*. The *sequencing ontology* imports the *domain ontology* to add to class *Concept* (defined in the domain ontology as well) the object properties shown in Figure 3 to express how the teaching of the concepts is sequenced. Object property *sequence* (functional) points to the next concept to be followed; *alternate* allows defining that it is possible to continue with any one of the pointed concepts, whereas *xorAlternate* only permits advancing through one of the related concepts. For instance, Figure 3 shows that a student starts learning concept C1, then C2 and later C3 (sequence). After this he could choose to advance independently to any of the concepts related to C3 through the alternative property (C4, C5 and C13). If he arrives at concept C8 he will have to continue to only one concept (C9 or C10).

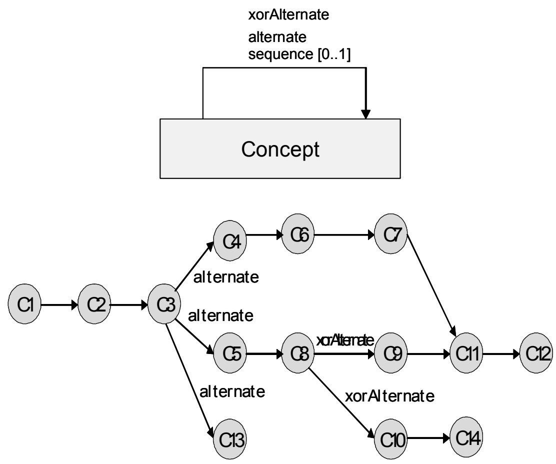

Figure 3: Sequencing ontology and example.

3.3 Contents-repository Ontology

The *contents-repository ontology* includes metadata to describe learning objects and their relationships. To determine the grade of granularity of a learning object is a fundamental decision in any project. The degree of reusability of a learning object is largely a function of its granularity, which is related with the size of an object (the lower the size of an object, the more reusable it will become) (South & Monson, 2000). Remember that in our approach the resources that have a very low granularity are precisely the learning objects (at the level of paragraph, image, table, diagram, and so on). Thus, if there are learning objects available at these levels, in every moment any e-learning system is able to add/remove contents at this level and to produce tailor-made learning materials according to the preferences of a student. Also, this facilitates showing didactical materials in those devices that have a screen of limited dimensions (e.g. a PDA) in form of a sequence of pages. In this approach, learning objects that have a greater granularity are built from smaller granularity ones. For instance, the course chapter's section will be created by mixing several little chunks.

3.3.1 Learning Objects Description

Metadata is used to describe the learning objects. In this work, some elements of the IEEE LOM (LOM, 2002) we have chosen, as this is a standard recognized internationally and many metadata schemas are based in it (e.g. IMS and SCORM metadata). On the other hand, the inspiration of our approach to describe the device features to correctly display learning objects comes from the elements of the LOM technical category and from the FIPA Device Ontology (FIPA, 2002).

Class *LearningObject* includes the metadata collection that describes a digital resource. As you may observe on Figure 4, (a) a learning object is created by one or several authors (*createdBy*), (b) it has a set of keywords that describe it (*hasKeyword*), (c) it is located in a certain direction (*location*), (d) it is written in a given language (*language*), (e) it manages a brief description (*description*), (f) it incorporates a type of interactivity - taking values active, exhibition and mixed - (*interactivityType*), and, (g) it possesses a level of difficulty - very easy, easy, average, difficult, and very difficult - (*difficultyLevel*). The type of interactivity denominated as active applies for documents where a student interacts and/or performs operations (for example, simulations, exercises, test questions), whereas exhibition is applied to documents whose objective is that the student just gets the content (for example, text, images, sound). Lastly, object properties *hasLearningStyle*, *requiresDevice* and *developSkill* are introduced in order to point to the learning styles that are better adjusted to a learning object, the device where it is more correctly visualized, and the intellectual skill developed when it is used, respectively.

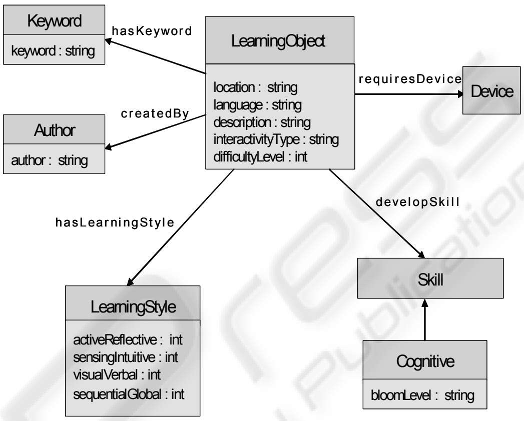

Figure 4: Description of a learning object.

So far, several learning styles theory have already been used in adaptive educational systems (Felder and Silverman, Dunn and Dunn, Honey and Mumford models, and so on). For a deeper reading, please consult (Stash, Cristea & De Bra, 2004). In the work proposed in this paper, the scheme to distinguish the student's learning style is the one proposed by the Felder-Silverman Learning Style Model (FSLSM) (Felder & Silverman, 1988). The FSLSM model distinguishes four dichotomous dimensions for learning styles: active/reflective, sensing/intuitive, visual/verbal, and sequential/global, which gives place to sixteen learning styles combinations. This decision has been taken for two reasons. First of all, this model provides a questionnaire to establish the dominant learning style of each student and its results may easily be linked to e-learning systems. Secondly, this model has been sufficiently validated in many other adaptive environments (Shi et al., 2003; Hong & Kinshuk, 2004; Capuano et al, 2005; Peña, Marzo & De la Rosa, 2005). Therefore, it seems reasonable to annotate learning objects according to Felder and Silverman model for the purpose of choosing the best learning objects adapted to the student's learning style. Object property *hasLearningStyle* points to suitable learning styles for a learning object. The class *LearningStyle* offers four properties of type integer that correspond to the four dimensions of the FSLSM.

Object property *requiresDevice* in class *LearningObject* of the *contents-repository ontology* points to the device necessary to display a learning object in a most satisfactory way. The class *Device* describes the device technology that should be used for displaying the contents correctly and in a suitable time. A device satisfies certain hardware (class *Hardware*) and software (class *Software*) requirements. With regard to hardware, we consider the following features to achieve a good user satisfaction: the computer CPU type (*cpu*), the network connection required (*networkConnection*), the necessary memory (object property *hasMemory*), the user interface characteristics (object property *hasUI*), the capability to receive audio input (*audio-input*) or to produce audio output (*audio-output*), as well as information on the video card (*videoCard*). Property *hasMemory* allows pointing to the description of the features related to memory – the amount of memory that the user device should incorporate to show the learning object (*amount*) and in which unit this amount is expressed (*unitMemory*). Now, property *hasUI* allows pointing to the information that describes the user interface - the width of the screen (*width*), the height of the screen (*height*), the unit for the width and height parameters (*unit*) and if a colour screen (*color*) is necessary. Regarding the software, features such as the minimum and the maximum version capable of using the resource (*minimumVersion* and *maximumVersion,* respectively) and the name that identifies it (*nameSoftware*) are included. We distinguish the browser (*Browser*), the operating system (*OperatingSystem*) and the pluggins (*Pluggin*) as kinds of software.

Moreover, the taxonomy of learning objectives proposed by Benjamin S. Bloom (Bloom, 1956) is selected to annotate the learning objects according to the instructional pedagogic role that they play from a cognitive perspective. This supposition has been adopted because this taxonomy is easy to understand and it has also been widely applied (Soldatova & Mizoguchi, 2003; Buckley & Exton, 2003; Ullrich, 2004). Class *Cognitive* has object property *bloomLevel* to represent the intellectual skill that the student develops when using it: knowledge, comprehension, application, analysis, synthesis and evaluation, just as identified in the cognitive domain of the Bloom Taxonomy.

To conclude, class *LearningObject* also includes object properties *isEquivalentTo* and *complementsTo*. The fist one is for knowing the resources that have the same semantic meaning, whereas the second allows reusing resources that the author of the contents thinks that are necessary to be grouped together in a course. For instance, a diagram should be grouped together with the paragraph and/or audio that describes it; a simulation should be reused together with the paragraphs that explain a concept, etc.

3.3.2 Types of Learning Objects

The individuals of class *LearningObject* may be theoretical explanations, practical explanations, or evaluation questionaires belonging to classes *TheoreticalContentObject*, *PracticalExplanation* and *IndividualEvaluation,* respectively. Notice that these classes are subclasses of class *LearningObject*. They are related to each others, as represented schematically in Figure 5 through rTCO_IE, rTCO_PE and rIE_PE. Let us highlight that in the ontology, instead of rTCO_PE there is an object property to indicate that a theoretical content object can be related with several practical explanations. Another object property expresses that a practical explanation can be related with several theoretical content objects. Exactly the same idea is applicable to rTCO_IE and rIE_PE.

Classes *TheoreticalExplanation* and *PracticalExplanation* represent the theoretical and practical explanations, respectively, that are shown to the students (see Figure 5). In order to compose the theoretical explanations several types of formats are proposed (classes *Text*, *Audio*, *Video*, and *Image*). This way, the theoretical explanations that appear to the verbal students are formed by text and/or audio, whereas videos and images are shown to the more visual students. Classes *Text*, *Audio*, *Video* and *Image* are subclasses of *Theory*, which includes the data type property *order* to describe how to link the theoretical contents. Class *Text* has the property *role* that allows expressing whether it is a text chunk associated to a definition, summary, law, theorem, or proof, among others; whereas class *Image* has the property *type* to describe that it is a graph, figure, graphical scheme, etc. In order to realize practical explanations, examples, simulations and animations (classes *Example, Simulation, Animation*) can be used.

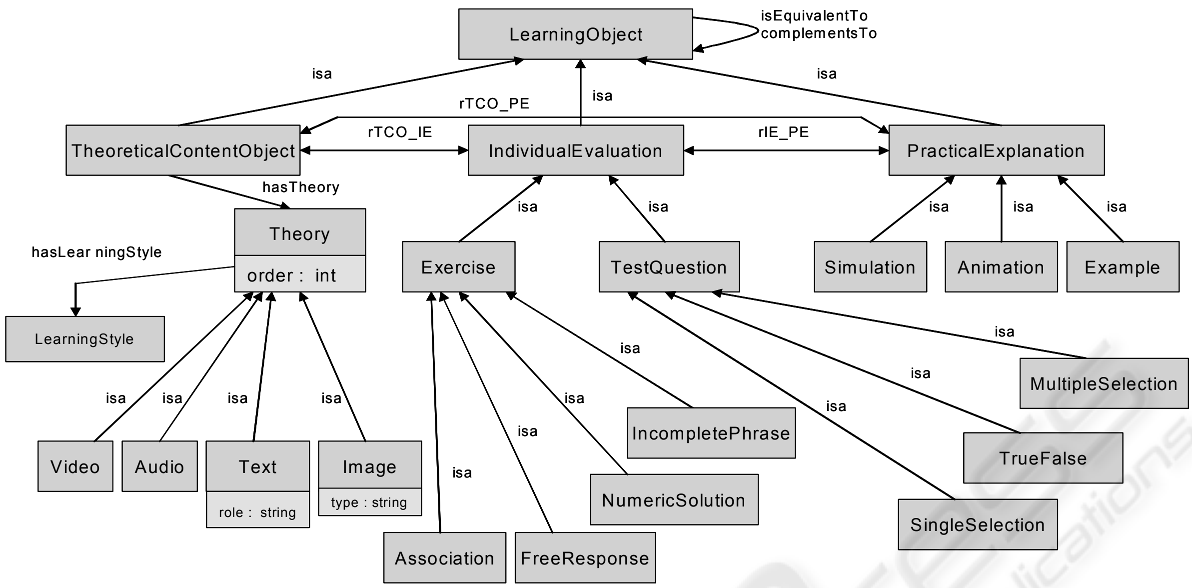

Figure 5: Types of learning objects.

The individuals of class *IndividualEvaluation* are used to evaluate the knowledge acquired by the student. Class *IndividualEvaluation* has two subclasses (*Exercise* and *TestQuestion*) that contain the exercises and the test questionnaires, respectively, that a student has to solve. Class *Exercise* contains four subclasses associated to different kinds of statements in which (1) a question is posed where the student has to answer with a numerical solution (*NumericSolution*), (2) it is necessary to complete in one o more points with a phrase, a word or a cipher (*IncompletePhrase*), (3) the student has to answer with one or several paragraphs (*FreeResponse*), and, (4) he must establish the relations between the elements of two columns (*Association*). On the other hand, class *TestQuestion* has several subclasses to highlight the different types of test questionnaires that are shown to the student. For instance, a student has to choose between one of two alternatives (*TrueFalse*), one of three or more alternatives (*SingleSelection*), or all the correct ones from a series of alternatives (*MultipleSelection*).

4 CONCLUSIONS

In this paper a proposal for an authoring architecture based in several ontologies, namely *domain ontology, sequencing ontology* and *contents-repository ontology*, has been introduced. To annotate a learning object, characteristics such as the most appropriate learning style to a student, the device used to access the contents and the skill that is expected to be developed when the student approaches the contents are considered in the *contents-repository ontology*. In the *domain ontology* the concepts and the relations between them we describe, just as appearing in the course domain. In the *sequencing ontology* the possible learning paths that can appear through the concepts defined in the *domain ontology* we define.

As future work, it would be important to provide a more sophisticated learning objects assembly mechanism allowing to generate them according to formats generally used at present (IMS and SCORM) and fitted to the profile of each student. Moreover, we are engaged in using these objects in an agent-based intelligent tutoring system proposed very recently (Fernández-Caballero et al., 2006) in order to improve the proposal's adaptivity capacities.

ACKNOWLEDGEMENTS

This work is supported in part by the Spanish CICYT TIN2004-08000-C03-01 and the Junta de Comunidades de Castilla-La Mancha PAI06-0093 grants. José M. Gascueña is the recipient of a Predoctoral Scholarship awarded by Junta de Comunidades de Castilla-La Mancha under grant PBI06-0099.

REFERENCES

Antoniou, G. & van Harmelen, F. (2004). A Semantic Web Primer. The MIT Press, Massachussets Institute of Technology. Cambridge, Massachussets.

Baloian, N.A., Galdames, P., Collazos, C.A. & Guerrero, L.A. (2004). A model for a collaborative recommender system for multimedia learning material. Lecture Notes in Computer Science, 3198, pp. 281-288

Bloom, B.S. (Ed.). (1956). Taxonomy of educational objectives: The classification of educational goals: Handbook I, Cognitive Domain. New York, Toronto: Longmans, Green, 1956.

Buckley, J. & Exton Ch. (2003). Blooms' taxonomy: A framework for assessing programmers' knowledge of software systems. Proceedings of the 11th IEEE International Workshop on Program Comprehension, IWPC'03, pp. 165-174.

Capuano, N., Gaeta, M., Micarelli, A., & Sangineto, E. (2005). Automatic student personalization in preferred learning categories. In 11th Conference International on Human Computer Interaction, HCI International.

Dean, M. & Schreiber, G. eds. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. & Stein, L.A. (2004). OWL Web Ontology Language Reference. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-ref/

Duitama Muñoz, J.F. (2005). Un modèle de composants pédagogiques pour des cours adaptatifs sur le Web. Ph. D. Thesis, Institut National des Télécommunications, France. http://etna.int-evry.fr/~duitama/thesis-home/PhD-Thesis-05-04-2005pdf.zip.

Felder, R.M, Silverman, L.K. (1988.) Learning and teaching styles in engineering education. Journal of Engineering Education, 78(7), pp. 674-681. Retrieved October 25, 2006, from http://www.ncsu.edu/felder-public/Papers/LS-1988.pdf

Fernández-Caballero, A., Gascueña, J.M., Botella, F., Lazcorreta, E., 2006. Distance learning by intelligent tutoring system. Agent based architecture for user-centred adaptivity. In Enterprise Information Systems VII.

FIPA (2002). FIPA Device Ontology Specification. 6-12-2002. Retrieved October 25, 2006, from http://www.fipa.org/specs/fipa00091/SI00091E.pdf

Gruber, T. R. (1993). A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2), pp. 199-220.

Hong H., Kinshuk. (2004). Adaptation to student learning styles in web based educational systems. Proceedings of ED-MEDIA 2004, World Conference on Educational Multimedia, Hypermedia & Telecommunications, pp. 491-496.

Jovanovic, J., Gasevic, D. & Devedzic, V. (2006). Ontology-based automatic annotation of learning content. International Journal on Semantic Web and Information Systems, 2(2), pp. 91-119.

Leidig, T. (2001). L3—towards an open learning environment. ACM Journal on Educational Resources in Computing, 1(1), article #5, 11 pages.

LOM (2002). IEEE 1484.12.1-2002, Draft Standard for Learning Object Metadata, (2002).

McGreal R. (2004). Learning objects: A practical definition. International Journal of Instructional Technology and Distance Learning, 1(9), pp. 21-32.

Peña, C.I., Marzo, J.L., & De la Rosa, J.L. (2005). Intelligent agents to improve adaptivity in a web-based learning environment. In C. Ghaoui, M. Jain, V. Bannore & L.C. Jain (Eds.), Knowledge-Based Virtual Education, Springer-Verlag, pp. 141-170

Ronchetti, M. Saini, P. (2004). Knowledge management in an e-learning system. Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies, ICALT 2004, pp. 365-369.

Santacruz-Valencia, L., Navarro, A., Aedo, I. & Kloos C. (2005). An ontology-based mechanism for assembling learning objects. Proceedings of E-Learning on Telecommunications, ELETE 2005, pp. 472-477.

Shi, H., Rodriguez, O., Shang, Y. & Chen, S. (2003). Integrating adaptive and intelligent techniques into a web-based environment for active learning. In Ed: C. T. Leondes (Ed.), Intelligent Systems: Technology and Applications. CRC Press/Lewis Publishers.

Shimic, G., Gasevic, D., & Devedzic, V. (2006). Classroom for the Semantic Web. In M.D. Lytras, A. Naeve (Eds.), Intelligent Learning Infrastructure for Knowledge Intensive Organizations. Information Science Publishing, pp. 251-283.

Silva Muñoz, L. & Palazzo M. de Oliveira, J. (2004). Adaptive web-based courseware development using metadata standards and ontologies. Proceedings of the 16th International Conference on Advanced Information Systems Engineering, CAISE 2004. Lecture Notes in Computer Science, 3084, pp. 414-428.

Soldatova, L., & Mizoguchi, R. (2003). Ontology of tests. Proceedings of Computers and Advanced Technology in Education, pp.175-180.

South, J.B. & Monson, D.W. (2000). A university-wide system for creating, capturing, and delivering learning objects. Section 4.2 of the book The Instructional Use of Learning Objects: Online version http://www.reusability.org/read/chapters/south.doc

Stash, N., Cristea, A., & De Bra, P. (2004). Authoring of learning styles in adaptive hypermedia: Problems and solutions. Proceedings of the 13th International World Wide Web Conference, pp. 114-123.

Ullrich, C. (2004). Description of an instructional ontology and its application in web services for education. Proceedings of the Workshop on Applications of Semantic Web Technologies for E-learning, SW-EL'04, pp. 17-23.

Wiley, D.A. (2001). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D.A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version.