Journal of Theoretical and Applied Information Technology
31st May 2016. Vol.87. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BUILDING AN E-LEARNING SYSTEM'S OWL ONTOLOGY BY EXPLORING THE UML MODEL

[1]BOUCHRA BOUIHI, [2]MOHAMED BAHAJ
[1,2] LITEN Laboratory, Faculty of Science and Technology, University Hassan 1st, Morocco
E-mail: [1]bouchrabouihi@gmail.com, [2]mohamedbahaj@gmail.com

ABSTRACT

The relevance of information in the field of e-learning is strongly requested. The semantic web is the discipline that guarantees this relevance with its ontology system. Since this trend is still evolving, we propose in this article to bridge the gap between the semantic web and the software engineering by making the ontology development closer to the widespread software engineering community. We propose how to transform an e-learning UML class diagram into an OWL ontology. This transformation is highlighted by the conversion of some UML concepts like: class, instance, attributes, class identifier, inheritance, association relationship and composition relationship. The added value of our work is to propose to convert in a better way the UML class diagram features, the inheritance and the composition relationships by exploring respectively OWL-DL advanced features: the Value Partitions Design Pattern and the Collection Ontology. The resulting ontology is designed by the open-source ontology editor and framework for building intelligent systems Protégé 4.3.0.
Keywords: *UML Model, OWL Ontology, E-learning, Value Partitions Design Pattern, The Collection Ontology.*

1. INTRODUCTION

The growing need for technologies that make the web more intelligent gives birth to a promising technology that is the Semantic Web. According to the W3C, "The semantic web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". The semantic web is also called "web of data", it will allow to make the semantic content of the web interpretable not only by human but by the machine. The machines will be able to interpret and process information in order to better meet the human's need of intelligent web that could be transformed into an intelligent guide, able to provide full and immediate answers to natural language queries, and to encourage the development of new forms of collective intelligence. Ontology and the different languages designed for publishing data: Resource Description Framework (RDF), Web Ontology Language (OWL), and Extensible Markup Language (XML) are the pillars of this technology.

Given its increasing impact in the education field, E-learning remains one of the areas in which we can explore the intelligent features of semantic web. Because semantic web technologies offers to develop e-learning platforms content that are more personalized and more adapted to the users skills and needs. The development of an e-learning platform in the semantic web universe starts with designing the ontology which is the heart of the linked data world. It's the knowledge data domain that will be shared and explored. The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation [1]. To bridge the gap between software engineering communally conceived in UML and the semantic web to make the ontology development, this paper is intended to build an e-learning system ontology in OWL by exploring the UML class diagram. As a matter of fact, we propose a new UML-to-OWL conversion that better convert the inheritance and the composition relationships using OWL-DL advanced features: The Value Partitions Design Pattern and The Collection Ontology.

The paper is structured as follows: In the following section, we will compare our approach with related works. Section 3 describes the class diagram of an e-learning system. Section 4 discusses our approach to build our e-learning ontology by converting the UML class diagram into OWL and a conclusion ends the paper.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. RELATED WORKS

In recent years, many researches have been made to build OWL ontologies from UML class diagram. [2] Proposes an automatic method to convert a UML class diagram into an ontology using OWL / XML language with keeping the features meaning of the diagram. The main contribution of [2] is to preserve the semantic features of UML concepts like:
• Encapsulation: this concept is converted in OWL by using a super datatype property called “Attribute”, it has three subproperties that are Public, Private and Protected.
• Composition and Aggregation: they are converted thanks to the superclass “Association” which has 2 subclasses that are Composition and Aggregation.
However, the limits of the conversion process adopted in [2] are that there is no difference between inheritance and composition relationships in OWL unless they are extremely different in UML. That’s why in our approach we propose to better convert these two concepts by making the difference between them.

In conjunction with [2],[3] proposes the conversion of the aggregation and composition relationship from UML to OWL FULL using respectively the antisymmetric object property and the concept of container materialized by “Bag” (not sorted resources) or “Seq” (sorted resources). But, this conversion process is only doable in OWL-FULL and not in OWL-DL, because Containers and Collections are missing in OWL-DL. That’s why in our approach we propose to import and integrate The Collection Ontology which handles the lack of Collections in OWL-DL to better convert the composition relationship.

In other scientific approach, [4] Proposes a transformation between UML class diagrams and OWL 2 ontologies in the M2 level using the QVT transformation language and the meta-models of UML and OWL 2. More than that, [5] Investigates UML and OWL 2 similarities and differences by specifying and implementing the transformation on the meta-model level using the QVT transformation language.

[6], [7], [8] establish a precise conceptual correspondence between UML and OWL through a semantics-preserving schema translation algorithm.

Coming to e-learning within these transformation approaches, the work presented in [9] outlines the knowledge engineering approach for constructing semantics for e-learning domain entities using Web Ontology Language OWL.

Aiming at improving and extending what was proposed in these works, we propose in this paper a new approach of UML-to-OWL conversion by exploring in depth the nature of inheritance and composition relationship In UML. To propose a new conversion of these UML concepts we have recourse to the Value Partitions Design Pattern to convert the inheritance and the Collection Ontology to convert the composition relationship. This process of conversion is implemented in Protégé 4.3.0 by building an e-learning ontology from its UML class diagram.

3. E-LEARNING SYSTEM’S UML MODEL: CLASS DIAGRAM

In Figure 1, we present the class diagram of the e-learning system that we propose as a basic prototype. It illustrates the following scenario: in an e-learning system, the instructor and the student are the main users who interact with the system after connecting. In order to achieve the intended objectives, the student adopts a learning strategy involving a certain area of knowledge. This strategy has three major components: courses, assignments, and assessments. These elements are supervised and managed by the instructor. To support the advancement of the student, the instructor may issue from time to time some announcements or feedbacks.

4. FROM UML CLASS DIAGRAM TO OWL ONTOLOGY

In order to build the ontology of our e-learning system, we opted for the UML to OWL conversion process adopted in [2]. Significant improvements have been made to this process, especially in relationships conversion. Table 1 shows an overview of our conversion process:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 1: From UML to OWL conversion

UML		OWL
Class		Class
Instance		Individual
Attribute		Functional Datatype Property
Identifier		Inverse-Functional Property
Relationship	Association	ObjectProperty with a specification of Domaine and Range
	Inheritance	-A relationship hierarchy (is_a) between two concepts in ontologies. -Exploring the Value Partitions Design Pattern for more than one subclass.
	Composition	Its conversion depends on the sublanguage used: -OWL Full: Using rdf: Bag or rdf: Seq -OWL DL: Exploring the CO (the Collection Ontology)

4.1 Class

The concept of class in UML is the same in OWL. Each ontology is composed of several classes also called concepts. UML classes inherit from the Object class, but in OWL they inherit from the Thing class.
OWL code:

```
<owl:Classrdf:about="&e-learning-
ontology;Course"/>
```

4.2 Instance

UML class instances can be interpreted as OWL class individuals.
OWL Code:

```
<owl: NamedIndividual rdf:about="&e-
learning-ontology;Lina">
<rdf:type rdf:resource="&e-learning-
ontology;Student"/>
</owl:NamedIndividual>
```

4.3 Attribute

UML class attributes are converted into OWL datatype property. An OWL Datatype Property may have more than one value for one individual, so to maintain the UML atomicity we should declare it functional as follows:
OWL Code:

```
<owl:DatatypeProperty rdf:about="&e-
learning-ontology;has_login">
<rdf:type
rdf:resource="&owl;FunctionalProperty"/>
<rdfs:domain rdf:resource="&e-learning-
ontology;Student"/>
<rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
```

4.4 Identifier

The class identifier may be converted into a functional datatype property as we described before, but it's not sufficient because this identifier makes every instance unique in a similar way as a table's primary key. In this case the Datatype Property should be an Inverse-Functional Property. This kind of properties is only supported by the sublanguage OWL Full.
OWL Code:

```
<owl:InverseFunctionalProperty
rdf:about="#id_student"/>
```

4.5 Relationships

In our e-learning UML model, there are three kinds of UML relationships that are linking the classes: Association, Inheritance (Specialization / Generalization [4]) and composition. In this part, we explain how we converted each relationship from UML to OWL.

4.5.1 Association

This UML relationship is interpreted as an OWL Object Property. An Object OWL Property is a relationship that links an individual to another one [10]. It's considered as a role specifying the domain and the range.

Journal of Theoretical and Applied Information Technology
31st May 2016. Vol.87. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OWL Code:

```
<owl:ObjectProperty rdf:about="&e-learning-ontology;adopts">
    <rdfs:range rdf:resource="&e-learning-ontology;LearningStrategy"/>
<rdfs:domain rdf:resource="&e-learning-ontology;Student"/>
<owl:inverseOf rdf:resource="&e-learning-ontology;adoptedBy"/>
    </owl:ObjectProperty>
```

Analogically to the UML cardinality, in OWL we can describe the class of Individuals that have at least, at most or exactly a specified number of relationships with other Individuals or datatype values. The restrictions that describe these classes are known as Cardinality Restrictions [11].
OWL code:

```
    <owl:Class rdf:about="&e-learning-ontology;Student">
<owl:equivalentClass>
<owl:Restriction>
<owl:onProperty rdf:resource="&e-learning-ontology;adopts"/>
<owl:onClass rdf:resource="&e-learning-ontology;LearningStrategy"/>
<owl:minQualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minQualifiedCardinality>
</owl:Restriction>
</owl:equivalentClass>
<rdfs:subClassOf rdf:resource="&e-learning-ontology;User"/>
    </owl:Class>
```

4.5.2 Inheritance

Inheritance is a common concept between UML and OWL. Each class may be a superclass for other classes, as it can also be a subclass of another class. UML Object class is the superclass of all classes as in OWL; the Thing class is the superclass of all classes. To make better use of this hierarchical relationship and to refine our description of various classes having more than one subclass, we propose to explore the Value Partitions Design Pattern.

Value Partitions as defined in [11] and [12] are not part of OWL, or any other ontology language, they are a Design Pattern. Design patterns in ontology design are analogous to design patterns in object oriented programming; they are solutions to modelling problems that have occurred over and over again. These design patterns have been developed by experts and are now recognised as proven solutions for solving common modelling problems. Value Partitions restrict the range of possible values to an exhaustive list, for example, in our e-learning system, the User class will restrict the range into 'Student' and 'Instructor', It will be covered only by these two types of users [11] as depicted in Figure 4:

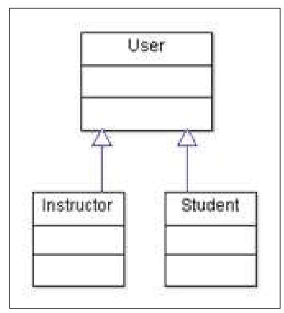

Figure 2: Inheritance in UML

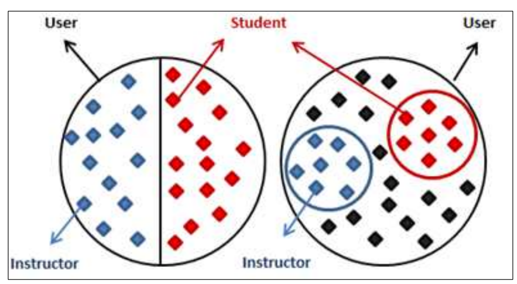

Figure 3: With Value Partitions Figure 4: Without Value Partitions

Creating a Value Partitions in OWL under Protégé 4.3.0 consists of several steps [11]:

1. Creating a class to represent the Value Partitions. For example, the User class.
2. Creating subclasses of the Value Partition to represent the possible options for the Value Partitions. For example, we might create the classes Student and Instructor as subclasses of the User class.

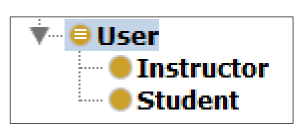

Figure 5: User Class subclasses

Journal of Theoretical and Applied Information Technology
31st May 2016. Vol.87. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. Make the subclasses of the Value Partitions class disjoint.
4. Provide a covering axiom to make the list of value types exhaustive.

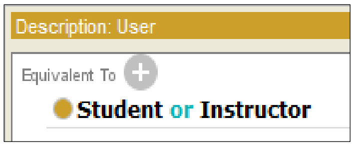

Figure 6: The covering axiom

5. Creating an object property for the Value Partitions. For example, for our User Value Partition, we might create the property hasRole.
6. Making the property functional.
7. Set the range of the property as the Value Partition class. For example for the hasRole property the range would be set to User.

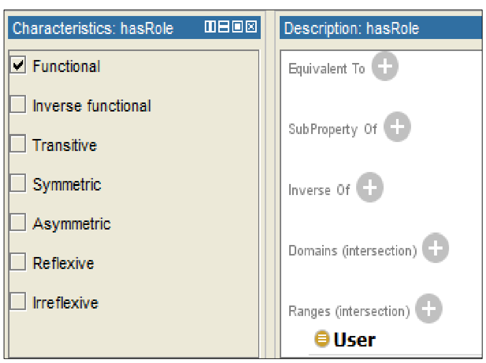

Figure 7: The functional object property "hasRole"

In this way, we explored the Value Partitions Design Pattern to better express the inheritance relationship.

OWL Code:

```
<owl:Class rdf:about="&e-learning-ontology;User">
<owl:equivalentClass>
<owl:Class>
<owl:unionOf rdf:parseType="Collection">
<rdf:Descriptionrdf:about="&e-learning ontology;Instructor"/>
<rdf:Description rdf:about="&e-learning-ontology;Student"/>
</owl:unionOf>
</owl:Class>
</owl:equivalentClass>
</owl:Class>
```

4.5.3. Composition

In UML, the composition relationship is a strong aggregation. The life cycle of components depends on that of the composed, e.g. if the composed is destroyed, the components will be automatically destroyed. Seeking to model this relationship in depth pushed us to improve what was proposed in [2]. The way of converting of this relationship depends on the sublanguage used. In the case of:

4.5.3.1. OWL-FULL

The best way to express the composition in OWL-FULL is to use the containers "Bag" and "Seq". They are Collections that are making possible to mean that the disappearance of the composed involves the disappearance of the components. It's perfectly explained in [3].

4.5.3.2. OWL-DL

To better express the composition relationship in OWL DL, we propose in this paper to explore the "Collection Ontology". The "Collection Ontology" is the result of the work of Dr. Paolo Ciccarese and Dr Silvio Peroni, who proposed it as a remedy for non-existent standard and accepted way for defining collections within OWL DL frameworks unlike in OWL FULL. It's conceived for creating sets, bags and lists of resources, and for inferring collection properties [13].

In our e-learning system we have: a student adopts a Learning Strategy that is composed of several Courses, Assessments and Assignments. In UML, it's modeled with composition relationship

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

as depicted above. To better view it in OWL DL, we consider that the Learning Strategy has the same behavior as the Bag Collection of Collection Ontology. Courses, Assessments and Assignments are their elements. The choice of Bag Collection is justified by the fact that a bag is a collection having non-ordered items referring to repeatable elements [13]. The Figures 8, 9 and 10 illustrate the Collection Ontology implementation in our e-learning ontology:

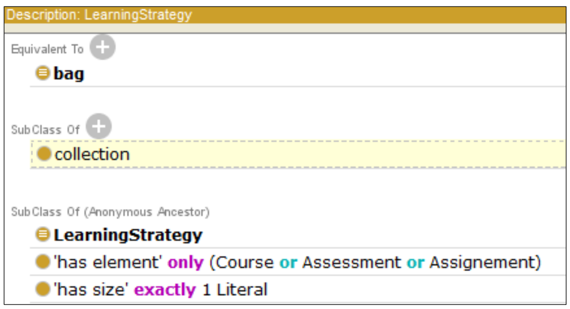

Figure 8: Description of LearningStartegy Class

5. CONCLUSION

Data modeling is one of computer science's pillars, it's crucial to represent, and organize data in a common model. To ensure the interoperability between software engineering commonly modeled in UML and semantic web, we proposed in this work a conversion process to convert a class diagram into an OWL ontology. Our work joins several works have been devoted to proposing different processes, but what is new in our one, is that we proposed to deeply interpret the inheritance and the composition relation by exploring the Value Partitions Design Pattern and The Collection Ontology. E-learning is a trendy education area to take advantage of the semantic web. That's why we proposed to practically implement our conversion process by building an E-learning ontology from its class diagram. The consistency of the built ontology is tested by Pellet Reasoner under Protégé 4.3.0.

REFRENCES:

[1] T. Berners-Lee, J. Hendler, O. Lassila. The Semantic Web. *Scientific American: The Semantic Web*. May 17, 2001.

[2] M. Bahaj, J. Bakkas . "Automatic Conversion Method of Class Diagrams to Ontologies Maintaining Their Semantic Features». *International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-2, Issue-6, January 2013. pp.65-69

[3] M. Ahlonsou, E. Blanchard ,H. Briand, F. Guillet . "Protégé at the heart of the transformation from uml 2 class diagram concepts to full owl". *EGC 2005, workshop Modelling of Knowledge*. Paris, January 2005.

[4] J. Zedlitz , J. Jorke *, N. Luttenberger." From UML to OWL 2". Knowledge Technology 2012*. Springer Berlin Heidelberg, pp.154-163

[5] J. Zedlitz , N. Luttenberger. "Transforming Between UML Conceptual Models And OWL 2 Ontologies". *Terra Cognita 2012 Workshop*

[6] Z. Xu, Y. Nu, W. He, L. Lin, Q. Yan." Automatic extraction of OWL ontologies from UML class diagrams: a semantics- preserving approach". *World Wide Web*. Springer. September 2012, Volume 15, Issue 5, pp 517-545.

[7] P. Pabitha, A. S. Priya and M. Rajaram, "An Approach for Detecting and Resolving Inconsistency using DL Rules for OWL Generation from UML Models", *European Journal of Scientific Research*, vol. 72, no. 3, (2012), pp. 404-413.

[8] D.Park, S.Park." Ontology Mapping for Enhanced Interoperability of S-100 Geographic Information Registers". *International Journal of Software Engineering and Its Applications* Vol.8, No.1 (2014), pp.225-234.

[9] M.F. Begamin, G. Ganapathy." Knowledge Engineering Approach for constructing Ontology for e-Learning Services". *Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on. IEEE*. Jakarta, 17-18 Dec. 2011, pp.125 – 132.

[10] S. Bechhofer, F .V Harmelen, J. Hendler , I. Horrocks , D. L McGuinness, P.F Patel-Schneider, L.A Stein. http://www.w3.org/TR/owl-ref/. Accessed January 2015.

[11] M. Horridge ." A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools". Edition 1.3. The University Of Manchester. March 24, 2011. pp.68-73.

[12] M. E Aranguren, E. Antezana , M. Kuiper , R. Stevens . "Ontology Design Patterns for bio-ontologies: a case study on the Cell Cycle Ontology". *BMC Bioinformatics 2008*, 9 (Suppl 5):S1.

[13] P. Ciccarese, S. Peroni. "The Collections Ontology: creating and handling collections in OWL 2 DL frameworks". *Semantic Web Journal*, Volume 5, Number 6 / 2014, pp. 515-529.

Journal of Theoretical and Applied Information Technology
31st May 2016. Vol.87. No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

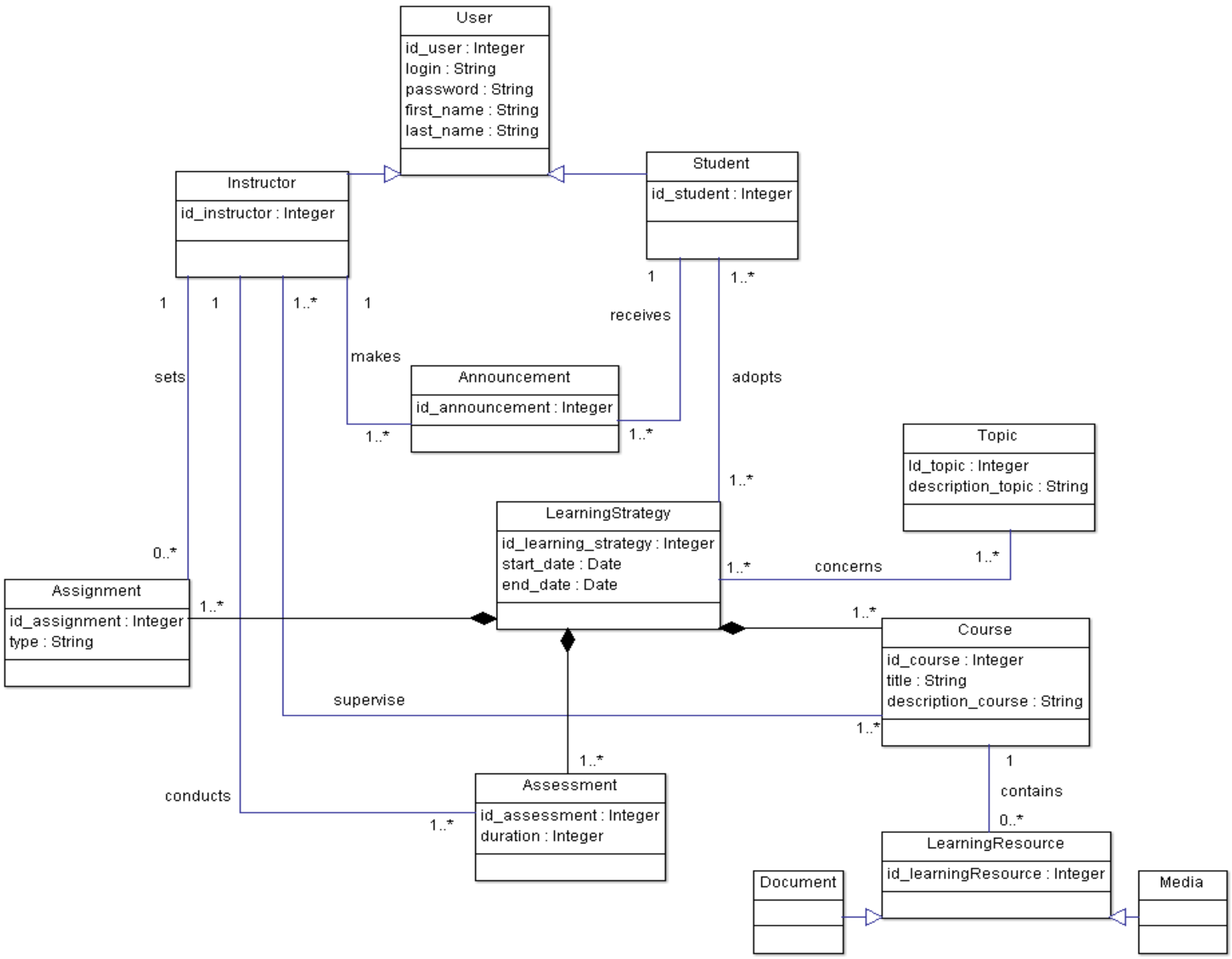

Figure 1: The class diagram of the E-learning system

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

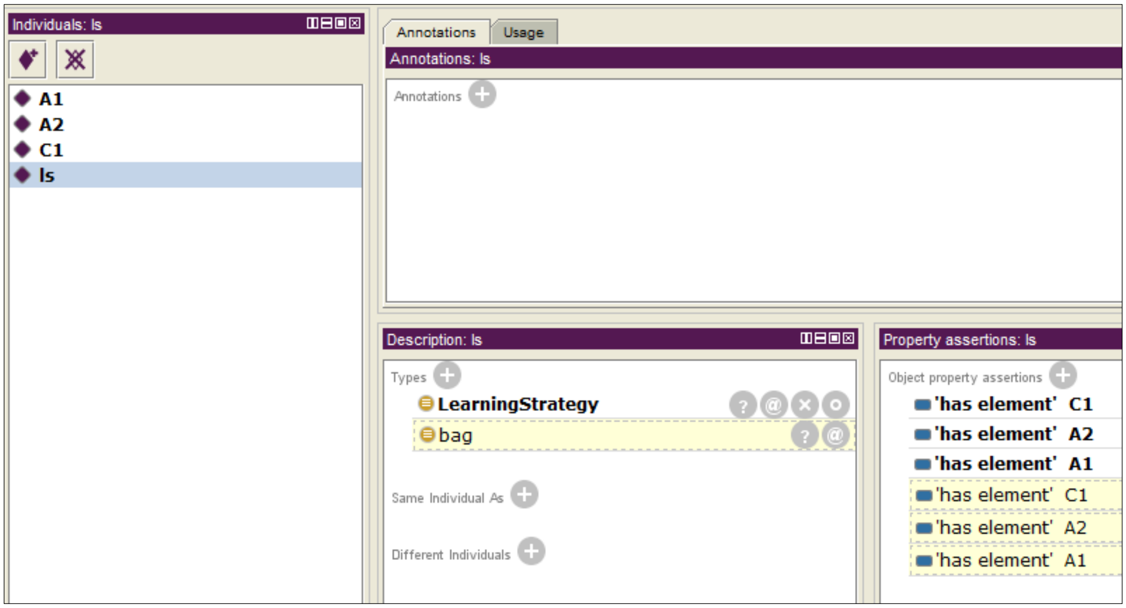

Figure 9: Individuals example. A1: Instance of Assessment class A2: Instance of Assignment class, C1: Instance of Course Class

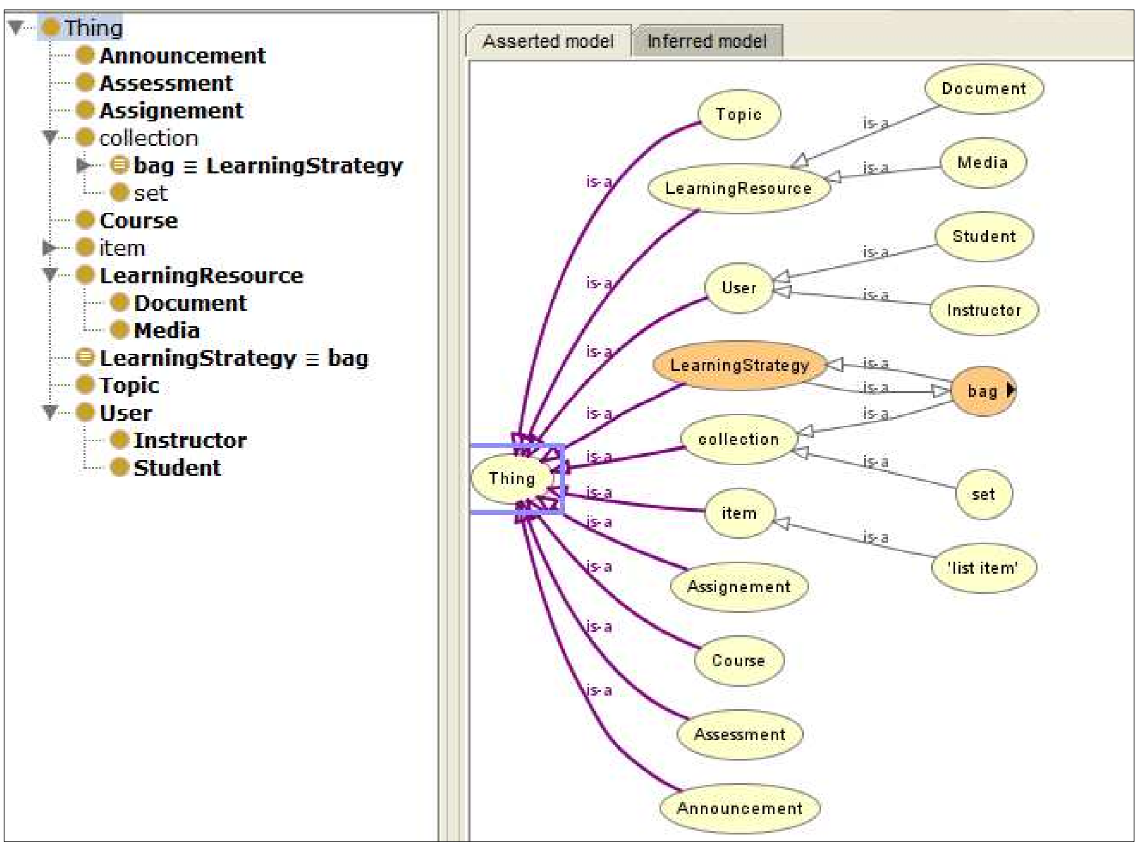

Figure 10: The Collection Ontology within the e-learning Ontology