On the Use of Educational Ontologies as Support Tools for Didactical Activities

Mihaela Oprea

University Petroleum-Gas of Ploiesti, Department of Automatics, Computers and Electronics
Bd. Bucuresti No. 39, Ploiesti, RO-100680, ROMANIA
E-mail: mihaela@upg-ploiesti.ro

Abstract
The basic characteristic of an educational process is knowledge sharing. In this context it is important to have an efficient method for knowledge representation that uses the concepts specific to the educational domain. Ontologies provide a solution for solving the problem of knowledge representation. In the last years it was highlighted the great importance of using ontologies when developing a web-based education. Educational ontologies can model the content of the course for all three phases of a didactical activity: teaching, learning and examination. The paper proposes a general framework for the development of educational ontologies as support tools for these three activities, and presents a case study of applying the framework to a course from the field of Computer Science.

Keywords: Educational ontology, Web-based education, Knowledge representation

Introduction

1. Introduction
In the last years it was highlighted the great importance of using ontologies when developing a web-based education. Educational ontologies can model a course for all three phases of an instructional activity: teaching, learning and examination. An educational ontology has general terms for any course, and specific terms for the knowledge domain of the current course. Several educational ontologies have been reported in the literature, each of them using particular development frameworks or methodologies. In this paper we propose a general framework for the development of educational ontologies of a course.

The paper is structured as follows. In section 2 it is presented an overview of some educational ontologies that were reported in the literature, and it is proposed a general framework for the development of educational ontologies as support tools for the didactical activity. A case study of applying the framework to the development of the educational ontologies of the Object Oriented Programming course, from the area of Computer Science, is described in section 3. The final section concludes the paper.

2. Educational ontologies

2.1 Generalities
Knowledge sharing is the basic characteristic of an educational process. In this context it is important to have an efficient method for knowledge representation that uses the concepts specific to the domain of education [7]. Ontologies provide a solution for solving the problem of knowledge representation and also the interoperability of the educational systems. An educational ontology is an ontology specific to the instructional process. Several models are constructed when
we develop a web-based education system or an intelligent tutoring system: the learner model, the teacher model, the domain model, the curriculum model etc. Different ways of designing the educational ontologies were proposed in the literature, so far. Also, a variety of application domains were discussed (see e.g. [11] and [17]).

2.2 Examples of educational ontologies
In this section we make a brief presentation of selected educational ontologies that were reported in the literature, and of some frameworks for educational ontologies development and application.

In [5] it is proposed an educational system based on ontologies that use metadata for finding, exchanging and managing different learning objects. The authors propose the specification of the metadata semantics by using the OWL formal ontology language.

A methodology for knowledge management that apply recommendation algorithms and is used under the framework of an e-learning system is described in [8]. Educational ontologies are used to personalize the course resources, according to the learners’ personality and preferences.

Another personalized education ontology (PEOnto) is introduced in [6]. The main characteristics of this ontology are the identification and discovery of relevant learning objects for individual needs of the learner. Five interrelated educational ontologies are included in PEOnto, that serve to address different personalized educational scenarios within a multi-agent based education system.

In [4] it is presented a method for developing educational ontologies by domain experts for use in the delivery of courseware content. The authors provided some ontological modelling guidelines that are adequate for rich domains.

The paper [12] proposes an ontology-based student model for the SoNITS educational social network developed for IT students.

In [16] it is described an ontology of educational theories and their relation to learning design. The ontological engineering was done by using the Hozo ontology editor.

The use of a computer assisted assessment system is discussed in [10]. Several learning styles are analyzed via adaptive feedback derived from the educational ontologies during the learning process. The proposed feedback deriving framework was designed to be used by intelligent tutoring systems.

In [2] it is introduced the OntoQue system, an engine for objective assessment item generation based on domain educational ontologies. The system was evaluated by using four OWL ontologies from different domains of expertise.

The use of educational ontologies in collaborative learning and knowledge generation processes is described in [1]. Several scenarios of ontology-based collaborative learning are presented. The authors highlight the knowledge creation phase when ontologies are generated and used.

An ontology-based knowledge evaluation system in higher education is presented in [9]. The authors shows a demonstration of the adaptive knowledge testing and evaluation system supported by the proposed educational ontology in fourteen higher institutions for the Business Informatics program curricula as a test environment.

An automated tool for content modeling, AIMTool, based on Java, is described in [3] for collaborative construction of the IMA-CID models (conceptual model, instructional model and didactic model). The educational ontologies are used as a supporting mechanism for modelling the course content.

The CADMOS-D method, based on UML is introduced in [14] for the educational adaptive hypermedia applications design. The authors propose a conceptual model that use RDF-based ontologies.
A semiautomatic framework, TEXCOMON, is introduced in [17], that produces domain concept maps from text, and then, derive educational ontologies from these concept maps.

An educational ontology, Univ_Edu_Onto, is presented in [13]. The ontology was developed in Protégé [15] and has general terms for a university course and specific terms for the Artificial Intelligence course that is taught to undergraduate students.

2.3 A general framework for educational ontologies development
The development of educational ontologies can be done by using some general guidelines grouped under a framework that follow a full didactical activity cycle. Figure 1 shows the block schema of a didactical activity cycle with the three phases: teaching, learning and examination, and their corresponding feedback. In our framework, the educational resources are used in the three phases of the instructional process, and they are based on the educational ontologies, specific to the domain of study and some prerequisite courses. For example, in the case of the Object Oriented Programming course some prerequisite courses are the Computer Programming Languages course, and the Data Structures and Algorithms course.

The educational ontologies include generic and specific terms for all three stages of the didactical activity. Some of the terms are domain independent, and are basic notions from education (e.g. curriculum, syllabus, educational resource, lesson structure, pedagogical roles, learner competences, teacher competences, student evaluation method, exams, test, items, assessment).

![Figure 1. The block schema of a didactical activity cycle](image-url)

The general framework for educational ontologies development is given as follows, under the form of a generic algorithm.

```plaintext
/************************************************************************/
ALGORITHM General Framework for Educational Ontologies Development
Input: course, prerequisite courses, student / learner, teacher
Output: Educational Ontologies for the course and specific student / learner competences
************************************************************************/
```
Begin

1. do Teaching Activity Ontologies Generation // for the teaching process
   * extract all the basic and advanced notions from the course and generate the Course Basic Subject Ontology and the Course Advanced Subject Ontology;
   * extract all the notions from the prerequisite courses and generate the Course Prerequisite Subject Ontology;
   * generate or use the Basic Teaching Ontology; // include teaching models

2. do Learning Activity Ontologies Generation // for the learning process
   2.1 * extract from the course all the practical activities with the needed resources and the main competences achieved by the student / learner and generate Course Practical Activities Ontology;
   2.2 * generate or use the Basic Learning Ontology; // include learning models

3. do Examination Activity Ontologies Generation // for the examination process
   3.1 * extract from the course the tests, questions, exercises, problems, assessment items and generate the Course Examination Ontology;
   3.2 * generate or use the Basic Examination Ontology; // include examination models

End.

In our general framework eight educational ontologies are generated. Each phase of the instructional process has its specific ontologies. Moreover, at each phase the ontologies of the previous phase or phases are also used. Between the eight educational ontologies there are specific links and relationships (e.g. between the Course Prerequisite Subject Ontologies of the courses that are members of a program of study curriculum). Apart from these ontologies, other ontologies can be used or generated depending on the specific course that is taught. We have simplified as much as possible the algorithm (including the generic specification for the input and output of the algorithm) in order to have a general framework. For example, the course has a detailed description with information regarding the curriculum, the syllabus, the target audience, the content, the teaching material, the resources (software and hardware), and so on.

3. Case study of applying the proposed framework to Computer Science field
In this case study we are considering a course from the Computer Science field taught to undergraduate students: Object Oriented Programming (in C++ programming language), and we present the particularization of the proposed general framework to develop the corresponding educational ontologies.

Course title: Object Oriented Programming Course (in C++ language)
The prerequisite courses for the Object Oriented Programming course are Computer Programming Languages (including the standard C programming language), Data Structures and Algorithms, and an introductory course in informatics, e.g. The Bases of Informatics. The concepts of these three courses are used as known concepts when defining the concepts specific to the Object Oriented Programming course.
The educational ontologies generated by the proposed framework are as follows:
1) Course Basic Subject Ontology
This ontology contains basic notions of the object oriented programming with reference to the C++ programming language. The content of the course introductory chapters (modules) are
represented by using this ontology. Examples of such chapters are: Introduction in object oriented
programming, Classes and methods, Inheritance, Fundamentals of object oriented modelling.
Examples of basic concepts: class, object, object variable, abstraction, abstract data type, message,
data member, function member, method, constructor, destructor, base class, derivative class,
inheritance, singular inheritance, object oriented modelling, OMT methodology etc.

2) Course Advanced Subject Ontology

This ontology contains advanced notions of the object oriented programming with reference to
the C++ programming language. The content of the course advanced chapters are represented by
using the Course Basic Subject Ontology and this ontology. Examples of such chapters are:
Polymorphism, Multiple inheritance, Object Oriented Modelling. Examples of advanced concepts
are: polymorphism, multiple inheritance, function overridden, function overloading, object models,
object oriented modelling language, UML etc.

3) Course Prerequisite Subject Ontology

This ontology contains all terms from the prerequisite courses that are necessary for defining
the concepts specific to the Object Oriented Programming course. Examples of prerequisite
concepts are: statement, sequence, decision, selection, iteration, compound statement, expression,
program structure, data type, variable, function, procedure, function call, procedure call, standard
library, programming technique, operating system (platform), algorithm, data structure, list, stack,
queue, tree, graph, search algorithm, sort algorithm etc.

4) Basic Teaching Ontology

This ontology contains terms specific to any teaching activity. Examples of such terms are:
teaching model, interactive teaching, course title, course duration, course structure, curriculum,
syllabus, target audience, teaching goals, teaching tools, course content, course outline, course
resource, educational unit, prerequisite knowledge, software, C++ language, hardware, computer
configuration, course chapter, sub-chapter, module, sub-module, section, sub-section, example,
application, problem, course presentation, course tutorial, lecture notes and readings, textbook,
course document file (ASCII text, doc, html, audio, video, slide, pdf, ps etc), bibliography,
reference, PowerPoint file etc.

5) Course Practical Activities Ontology

This ontology contains terms specific to the learning activity structured in practical activities
that corresponds to the chapters of the course. Example of such terms from the practical activities
of the Classes and methods chapter are as follows: object oriented problems solving, C++
language, defining classes in C++ (class syntax), defining the Stack class, defining the constructors
of a class (default constructor, copy constructor, constructors with list of parameters, type
conversion constructor), Borland C++, Visual C++ etc.

6) Basic Learning Ontology

This ontology contains terms specific to any learning activity: learner model, learning styles,
active reflective, sensing intuitive, visual verbal, interactive learning, learner feedback, learning
goals, practical activity, student / learner competences, learning object, resource, FAQ, lessons
learned etc.

7) Course Examination Ontology

This ontology contains terms specific to the examination activities that corresponds to the
teaching and practical activities of the Object Oriented Programming course. The key concepts
from the previous defined course ontologies are used. Additional terms are related to problems and
exercises proposed to be solved by students. An example of problem is the following: “Write a
C++ program that define the Robot base class and the Mobile Robot derivative class with
structure and behaviour at your choice, define two objects from the two classes in the main
function, and call their corresponding methods”.

"Write a C++ program that define the Robot base class and the Mobile Robot derivative class with
structure and behaviour at your choice, define two objects from the two classes in the main
function, and call their corresponding methods".
8) Basic Examination Ontology

This ontology contains terms specific to any examination activity of an instructional process. Examples of terms are: examination, assessment, self-assessment, assessment items, exercises, individualized exercises, questions, items, corrections, tests, problems, evaluation method, computer-assisted examination etc.

In this paper, we have focused on the vocabulary of the educational ontologies. However, when defining an ontology apart from the vocabulary, the relationships between the concepts and the axioms of the ontology are also specified. Examples of relationships used between the concepts and terms of an ontology are as follows: is_a, has, part_of, order, required_by etc.

The educational ontologies can be implemented in an ontology editor such as Protégé, Ontolingua etc. They are used either directly by the main actors of the instructional process (i.e. teacher, student / learner, tutor) or indirectly, by the e-learning platform or other intelligent instructional support tools (e.g. multi-agent systems, knowledge based systems) during the teaching, learning and examination phases of the didactical process.

In Figure 2 it is shown a screenshot with some classes (basic and advanced concepts) of the Object Oriented Programming course ontology, OOP_Ontology, implemented in Protégé 3.0.

![Figure 2. A screenshot with some classes of the Object Oriented Programming course ontology (in Protégé 3.0)](image)

4. Conclusion

The paper proposed a general framework for the development of educational ontologies as support tools for didactical activities: teaching, learning and examination. A case study of applying the proposed framework to a course from the Computer Science field was presented.

5. References


