STAFFORDSHIRE UNIVERSITY

TEI OF LARISSA

A DISSERTATION SUBMITTED TO THE STAFFORDSHIRE UNIVERSITY, TEI OF LARISSA SCHOOL IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTER'S THESIS

Student Modeling and Related Reasoning Issues

BY

ANTIGONI K. VENIADOU

DEPARTMENT OF COMPUTER SCIENCE

May, 2011

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CERTIFICATE

This is to certify that the dissertation entitled “Student Modeling and Related Reasoning Issues” submitted by Antigoni Veniadou in partial fulfillment of the requirement for the award of degree Master’s Thesis to Staffordshire University, TEI of Larissa School, department of Computer Science is a record of the candidate’s own work carried out by her under my supervision. The matter embodied in this dissertation is original and has not been submitted for the award of any other degree.

Date: 13/05/2011 (Dr. C. Hartonas)

Supervisor

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

To

My beloved Chris, who has always been there for me and I will always stand by him.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Abstract

The current thesis concentrates its goal on studying and constructing a student model component for the inner area of teaching Semantic Web, which plays a very important role on Web Based Educational Systems as it constitutes the encoding of knowledge of a system regarding its user as a means of enhancing its interaction. Specifically, on those factors that are considered important for using and updating of the student model so that the latter helps to enhance personalized teaching by customizing abilities; That's the analysis of reasoning issues.

Description of the creation of the static structure of the student model will be done with the aid of the CROP Reference Architecture. The CROP Reference Architecture based on the notions of Concepts, Resource, Order and Product and aimed to put forward a standard of internal structure that a Learning Object is to abide by, without making any commitment to particular educational/instructional theories, styles or preferences. A Learning Object is a resource, usually digital and web-based, that can be used and re-used to support learning. One of the key issues in using Learning Objects is their identification by search engines. This is usually facilitated by assigning descriptive Learning Object Metadata. As we said, the procedure of adaptation in terms of our student model constitutes a major issue for our research and we take it into account too.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Acknowledgements

Firstly, I would like to acknowledge my supervisor Professor C. Hartonas, whose comments, criticisms, inspiration and encouragement have propelled me this far. It would never have been possible to complete this study without an untiring support from him.

My family, who supports me financially, encourages me to go on and help me in every possible way they can.

Finally, I will take up on this opportunity to thank my friends for their moral support and understanding.

Thanks a lot to everyone,

Antigoni Veniadou

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Table of Contents

Abstract v

Acknowledgements vi

List of Figures ix

List of Tables xii

CHAPTER 1: Introduction 1

1.1 Preliminaries and Motivation 1

1.2 Statement of the Problem 4

1.3 The Goals of This Research 5

1.4 Organization of This Thesis 6

CHAPTER 2:Student Modeling – State of the Art 7

2.1 Student Modeling in ITSs 7

2.2 The Content of the Student Model 9

2.3 Characteristics of Students 11

2.4 Methods of Student Modeling 13

2.4.1 Overlay Model 14

2.4.2 Buggy Model 15

2.4.3 Stereotype Model 16

2.4.4 Uncertainty Model 18

2.5 General Considerations about Student Modeling in ITSs 19

2.6 The Personalization Issue 20

2.7 The IMS Learner Information Package (LIP) 21

CHAPTER 3:Learning in the Semantic Web 27

3.1 Ontologies and the Semantic Web 27

3.1.1 The Semantic Web 27

3.1.2 Ontologies 29

3.1.3 Querying an Ontology 33

3.2 Learning Objects Theory 35

3.2.1 IEEE Learning Object Metadata (LOM) 37

3.2.2 Shareable Courseware Object Reference Model (SCORM) 40

3.2.3 Learning Management Systems (LMSs) 44

3.3 The CROP View 45

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

3.3.1 CROP Learning Objects 45
3.3.2 CROP Learning Services and Learning Domains 54
3.4 Related Web Learning Projects 58
3.5 Learning Styles 59
CHAPTER 4: Learner & Learning Object Ontologies 64
4.1 An Owl Binding for Learning Object Metadata 65
4.2 An Owl Binding for Learner Information 75
4.2.1 An ontology for the IMS LIP Package 75
4.2.2 An Ontology for Learning Style 87
4.3 A Learner Model Ontology 94
4.3.1 The Learning Object Ontology 100
4.3.2 The Learning Behavior Ontology 101
CHAPTER 5: Reasoning Issues 107
5.1 Using and Managing the Learner Model 109
5.1.1 Extending the Student Model Graph 109
5.1.2 Monitoring Performance, Actions and Actions-Sequences 110
5.2 Reasoning about Learning Objects 111
5.3 Reasoning about the Learner 116
5.3.1 Inferring Style Characteristics from User Actions 125
5.4 Reasoning about the Learner – Learning Object Interaction 125
CHAPTER 6: Conclusions and Further Research 130
REFERENCES 133
CHAPTER 1 133
CHAPTER 2 134
CHAPTER 3 135
CHAPTER 4 138
CHAPTER 5 138

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

List of Figures

Figure 2-1: Architecture of Web-Based Intelligent Tutoring System 8
Figure 2-2: Overlay Student Model 14
Figure 2-3: Differential Student Model 15
Figure 2-4: The IMS Learner Information Package (LIP) core data structures 23
Figure 2-5: The principle LIP data structure 25

Figure 3-1: Architecture of Semantic Web 28
Figure 3-2: LOM elements & structure 39
Figure 3-3: Package Structure from the SCORM 1.2 CAM specification 42
Figure 3-4: Organizations and Items from the SCORM 1.2 CAM specification 43
Figure 3-5: CROP Learning Objects and their Content Ontology 46
Figure 3-6: Knowledge Objects Structure 47
Figure 3-7: Knowledge Resources Structure 48
Figure 3-8: Knowledge Products Structure 49
Figure 3-9: The Concept Graph of a Learning Object for Photography 50
Figure 3-10: KRC Graph Structure of Learning Object 52
Figure 3-11: Learning Domain 55
Figure 3-12: Roles in a Learning Domain and their correlations 57
Figure 3-13: The Taxonomy of the Learning Style Dimension Designators 63

Figure 4-1: LOM ontology ontograph diagram 66
Figure 4-2: Class hierarchy of LOM ontology 67
Figure 4-3: General class structure... 69
Figure 4-4: LifeCycle class structure 70
Figure 4-5: Meta-metadata class structure.. 71
Figure 4-6: Technical class structure... 71
Figure 4-7: Rights class structure.. 73
Figure 4-8: Relation class structure... 73
Figure 4-9: Annotation class structure 74
Figure 4-10: Classification class structure.. 74

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Figure 4-11: LIP ontology ontograph diagram 76
Figure 4-12: Class hierarchy of LIP ontology 77
Figure 4-13: Address class restriction axioms 79
Figure 4-15: Demographics class restriction axioms 80
Figure 4-16: Educational Objective class restriction axioms 81
Figure 4-17: LIP class restriction axioms 81
Figure 4-18: LIPContentType class restriction axioms 82
Figure 4-19: Accessibility LIPElement restriction axioms 83
Figure 4-20: Activity LIPElement restriction axioms 84
Figure 4-21: Affiliation LIPElement restriction axioms 84
Figure 4-22: Competency LIPElement restriction axioms 85
Figure 4-23: Goal LIPElement restriction axioms 85
Figure 4-24: Identification LIPElement restriction axioms 86
Figure 4-25: Qcl LIPElement restriction axioms 86
Figure 4-26: Name restriction axioms for Name class 87
Figure 4-27: Class hierarchy of LearningStyle ontology 87
Figure 4-28: LearningStyle ontology ontograph diagram 88
Figure 4-29: CognitiveStyle class and its subclasses 90
Figure 4-30: Abstract, Active, Concrete, Holist, Reflective and Sequential class structure 91
Figure 4-31: InformationProcessingStyle class and its subclasses 91
Figure 4-32: AudioVisual and Verbal class structure 92
Figure 4-33: InstructionalPreferencesStyle class and its subclasses 92
Figure 4-34: Expository, NavigationDependent, NavigationIndependent, Participatory, SelfEvaluation and TutorEvaluation class structure 93
Figure 4-35: LearningAttitudeStyle class and its subclasses 93
Figure 4-36: Continuity and Persistence class structure 94
Figure 4-37: Diagram of Learner Model structure 95
Figure 4-38: Learner Model ontology ontograph diagram 96
Figure 4-39: Class hierarchy of Learner Model ontology 97
Figure 4-40: The Learner Model Graph node class 98
Figure 4-41: ActivityHistory class structure 98
Figure 4-42: ActivityRecord class structure 99
Figure 4-43: LearningObject class structure 100

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Figure 4-44: LearningBehavior class structure..101
Figure 4-45: LearningBehavior ontology ontograph diagram..................................102
Figure 4-46: OwlViz diagram from NavigationAction class 104
Figure 4-47: OwlViz diagram from RequestAction class 105

Figure 5-1: Reasoning Process..108

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

List of Tables

Table 3-1: The roles and participants that implement them 57
Table 3-2: Learning styles models 61

Table 4-1: General Category Base Schema Structure from [33] 68
Table 4-2: LifeCycle Category Base Schema Structure from [33] 70
Table 4-3: Address Class Base Schema Structure 78
Table 4-4: ContactInfo Class Base Schema Structure... 80

Table 5-1: LearningBehaviour and LearningStyle ontology 118

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 1: Introduction

1.1 Preliminaries and Motivation

The current research focuses on student modeling and related reasoning issues. That is, we are interested in design aspects of a Student Model component in a web based learning system, as well as in reasoning issues that are involved in the use and update of the student model. We study this issue on a Semantic Web based approach. In the view we adopt [1], we consider Learning Services, a sub-category of Semantic Web Services, as the owners of Learning Objects.

A Learning Object is a resource usually digital and web based that can be used and re-used to support learning [2]. The current specification standard for the structure of a Learning Object is the Sharable Content Object Reference Model (SCORM) and Aviation Industry Computer Based (AICC) standards. With appropriate metadata descriptions [3], Learning Objects can be used as modular units that can be assembled together to form lessons and courses (larger Learning Objects). The current standard for Learning Object Metadata is the IEEE LOM standard [4].

In a Learning Domain as we envision it, after [1,5] the Learner is the basic entity. Other components of a Learning Domain, in the view we adopt here, are Learning Services, the Domain Broker, the Student Model, Learning Objects and the Domain Ontology.

An ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to reason about the properties of that domain and may be used to describe it.

Learners come with a knowledge level, preferences, a learning style, while they need to learn. Learners can interact with Learning Services at a physical level (Learner-Learning Service dialog). At a modeling level, we need to provide an account of activities related to the knowledge and comprehension of the student, the student's behavior as well as the (adaptable) behavior features of the service.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Student modeling [6] is the process by which Web Based Education Systems (WBES) acquire information about a user. This process is a life-cycle integrated by tasks as follows: knowledge elicitation about the student, construction, maintenance and exploitation of the knowledge repository. As a result a mental representation composed by attributes and preferences of the student is set in a module called Student Model [6]. The student modeling issue has been addressed, to some extent in the context of Intelligent Tutoring Systems. ITSs are educational software systems often using artificial intelligence techniques [7]. They can be used one-on-one and therefore they can adapt their tutoring to the needs of individual students. One key component that makes ITSs more intelligent or more adaptive to the needs of individual students is that ITSs have the ability to maintain a student model. In a web based environment, single purpose ITS's are replaced by Learning Management System (LMS), ie systems that are designed to help educators create quality online courses and manage learner outcomes. An LMS typically includes a set of tools, functions and features for learning. Typical LMSs are best understood as web platforms providing a number of different services to different categories of end users (teachers, authors of learning content, learners, and administrators). So, as we can see LMSs are global platforms in contrast with Intelligent Tutoring Systems (ITSs) [8].

In a network where reasoning can be implemented, such as the Semantic Network, adaptation (personalization) is an issue of significant importance. Adaptation needs to take into account the current state of knowledge of the student, the student's learning goals and his/her learning skills, preferences and style, as well as more mundane information such as the language of teaching, the available time for learning etc. Adaptation, sometimes also called "personalization", is an issue of significant focus of research nowadays.

It is in this direction that this current research attempts to provide answers in a significant number of issues that arise, such as:

- What is the static structure of a student model? In this point we should consider issues such as, what kind of information are contained in the student model component, which is the structure of data that are contained that information, how is this information categorized, based on the preferences of users and many other related issues. In order to determine this static structure of the student model, we adopt the CROP Reference Architecture [1], which is a novel proposal for a standard of internal structure that a Learning Object is to

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

abide by, “without making any commitment to particular educational/teaching theories, instructional strategies, styles or preferences, being flexible enough to accommodate different stances on such issues, while providing a solid foundation and laying out the way for the deployment of relevant technology to assist the development of Learning Objects” [1].

- The way the student model is used for selection of appropriate Learning Objects (LOs), given some learning objective. What methods are used in order to select the appropriate LOs for a specific user/student who wants to deal with a particular learning objective?
- How is it that an appropriate entity such as a Profile Manager modifies the student model so that it is always updated with the current knowledge the student has for a specific learning objective at any time.
- How is it that the execution monitor can be also assigned the task of issuing reports on student-Learning Object interaction dysfunctionalities, where these reports are addressed at the Learning Service that owns the Learning Object and which then must respond appropriately to modify the object so that it is better adapted to the student needs?
- Extracting information from a student model and reasoning on that information in order to offer to the student the best Learning Services that are available for the student’s needs at that time.
- Updating of the student model. Here, we should consider issues such as, what the update processes are, who is responsible to do that, how can we be sure the model remains consistent, etc.

It is necessary to obtain a global view on the structure and functioning of a Learning Domain in the Semantic Web in order to successfully deal with such issues. Also in order to achieve the above goals we must take advantage of existing reasoning mechanisms (ontological reasoning, based on systems of Description Logic (DL)) and tools (DL reasoners such as Fact++ etc). In our current research we focus on learning processes in the Semantic Web, placing the student modeling problem in that context.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

1.2 Statement of the Problem

As mentioned above, the student model component plays an important role in the intelligent operation of Web Based Education Systems. On Semantic Web approach is very crucial for the student model to has the capability to explicitly represent the properties of a particular student and also analyze the related reasoning that is involved in the usage and update of this student model.

In order to achieve this, we adopt the CROP Learning Domain approach [5], which we briefly present in the following sections.

In a CROP Learning Domain, Learners may send requests for some learning objective to the Domain Broker. At first, we should clarify issues such as: what are the necessary exchanges between the roles (played by participants, i.e. services and actors) that are involved in a Learning Domain? What reasoning issues need to be performed and who is it that does this reasoning? Our objective is to address issues relating to the adaptability of the learning provider, custom-selecting or constructing Learning Objects to suit the needs and requirements of the learner, even dynamically modifying the delivered product when shortcomings or learning difficulties are detected.

In the CROP view, Learning Objects by themselves have no adaptation capability. It is the learning providing system that must cater for this need. CROP Learning Objects are modifiable even at run time, but they do not modify themselves (as an adaptive response). It is the supporting system that has both the authorization and the capability to do that, in principle. This implies that there is a Monitoring role in the model. Some entity gets informed about the interaction of the Learner with the Learning Object. This entity can be endowed with the capability to discern shortcomings or difficulties in the student performance, to monitor persistent requests, such as for example that a Learner frequently asks for more examples, more practicing material, and then based on such information that is being passed to it by the Monitor, the Learning Service may react appropriately.

Moreover, while the learning interaction unfolds itself, the student acquires some knowledge. This must be recorded in the student model. Finally, if a learning interaction gets interrupted for some reason, suspending continuation until later, then some kind of session information needs to be kept in the model. It needs to be clarified

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

precisely what constitutes the “session information”. The identity of the Learning Object used must be also recorded in the model.

As we can infer from the above, the problem of this proposed research has to do with the recording to the student model component of all essential information that take place in a CROP Learning Domain. We need to think about which roles take part in the teaching process and what kind of information each one gives so that it is recorded in the student model. Afterwards, recorded information on the student model is used to extract useful conclusions about the emerging needs of the system.

1.3 The Goals of This Research

This research assumes a certain level of clarification, modeling and development of CROP Learning Domains. Within such domains, it further assumes a certain level of clarification about the structure and functionality of Learning Services, as well as of auxiliary participating entities such as a Domain Broker, a Monitor, perhaps a Rating Service etc.

Our central objective, in this context, is to provide a technical account of a **global student model** structure, to be used not in connection with a single Learning Object, but as a shared knowledge base carrying information about the Learner, to be used by a number of entities such as the Learning Services, the Domain Broker etc for reasoning purposes relating to more accurately capturing the features of the Learner that may influence a decision on what Learner Object to propose to him/her, or how to dynamically modify perhaps the execution model of a Learning Object, in order to suit the Learner observed preferences and behaviors, or even how to dynamically modify the content of a running Learning Object in order to cater for diagnosed Learner needs.

Since reasoning about both Learners and Learning Objects is involved and since the setting we adopt is that of a Semantic Web, all necessary modeling, as well as Knowledge Representation will be carried out by means of Ontologies.

In our research, we first tried to clarify issues like the above. The clarification of such issues needs to become carefully so that we can go ahead and design and implement a student model, carrying all these methods that may be needed to perform updates. Furthermore, there is the task of reasoning on the information contained in the student model. One reasonable approach is to let the student model do all the reasoning

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

and everybody else just submit queries about what they need to know (rather than allow them to look inside the student model, thus revealing the content of the model to the public). Another is to involve a Personal Learning Assistant who has full access to the model and performs the necessary reasoning and negotiation tasks. We shall choose an appropriate approach that will be implemented in the student model that we propose. The purpose here is to clarify the interactions among student model and everybody else. This involves clarifying the types of messages sent back- and-forth.

1.4 Organization of This Thesis

The structure of the current thesis is as follows:

In the first Chapter has presented an overview of the background and rationale for this study. In addition, the status of the problem is emphasized, as well as the special goals set for the thesis. In Chapter II, we investigate (through existing bibliography for “Student Modeling in ITSs”) the content that a student model component should have, plus the basic features of the student registered in it. In the subsequent parts of the chapter, we refer to some methods used to represent these features, and some general considerations about student modeling in ITSs are also mentioned. In the end of the chapter, there is a discussion on the “personalization issue” regarding the ITSs and their need to evolve into AEHSs, plus a short introduction to the specifications of the IMS Learner Information Package (LIP) which defines the general student profile. Chapter III discusses learning in the Semantic Web, introduces terms such as “Ontology”, Learning Objects and some of the technologies that those use. Additionally, we promptly describe the logic behind the CROP Reference Architecture based on which we will apply our own approach of the student model component. The chapter closes with a discussion about learning styles. In Chapter IV, we extensively describe the developing process of the basic ontologies of the Learner, the Learning Object’s and finally the Learner Model’s with the aid of the Protégé 4.0.2. The reasoning issues that arise from developing these ontologies are further analyzed in Chapter V. The last Chapter VI summarizes this thesis and discusses the significance of this research.

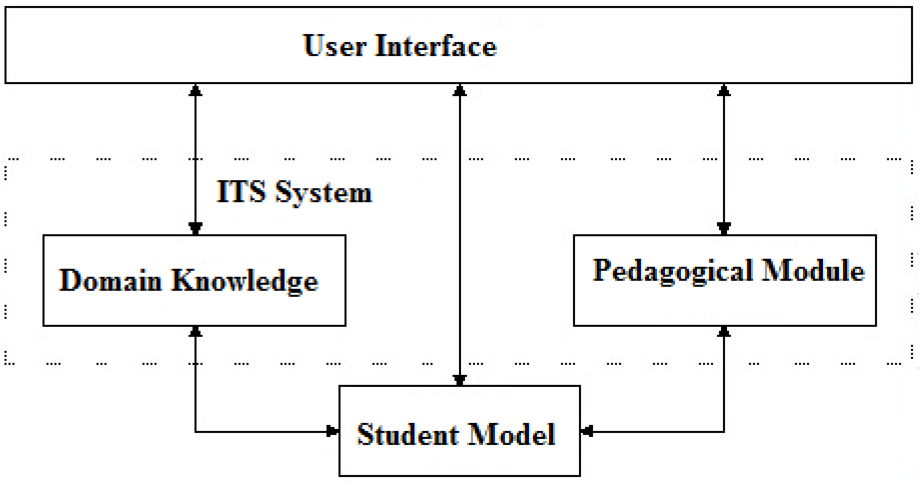
Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 2:

Student Modeling – State of the Art

2.1 Student Modeling in ITSs

The student modeling component constitute an essential part of an ITS. An Intelligent Tutoring System or ITS, is a computer program that instructs the student in an intelligent way. An important feature of ITSs is their ability to adjust the presentation of the teaching material to the needs of the users. This is achieved by using methods of Artificial Intelligence to represent teaching decisions and information relevant to each student. Intelligent Tutoring Systems were usually implemented as stand-alone systems. ITS's features are their knowledge of the field, the student and the strategies to support a flexible and customized teaching process. This means they know of the content and the connections inside it and they try to choose the best strategy depending on the user's information. Student features and their progress are stored on what we call a "student model". An ITS according to [9,10] knows what to teach (knowledge range), how to teach (teaching methodology) and to whom (student model). Intelligent Tutoring System basic architecture consists of the following basic features:



Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Figure 2-1: Architecture of Web-Based Intelligent Tutoring System [9]

The Domain Knowledge includes the teaching material presented to the users of the system. This material refers to a variety of subjects that begin with introductory education issues and climax to more advanced ones. The teaching material of each subject is organized in chapters, sub – chapters and lessons. Every teaching unit is defined by some cognitive connotations. These connotations can be either pre- required (known to the user so as to comprehend the content of the teaching unit), or acquired during the process of learning the selected unit. Connotations relate to each other by each describing those connotations that are pre-required. Examples help the user to understand the important points of the theory. The number of examples presented depends on the student model.

The Pedagogical Module represents the teaching process. It provides the knowledge needed so as to adapt the presentation of the teaching material to the data of the student model. The Pedagogical Model includes information about different teaching approaches. These approaches determine a subject's structure. Moreover, the pedagogical model includes data related to choosing different education units according to the user's features.

The User Interface is responsible for the interaction of the system with the user. The main goal is to design a user interface that can be used by users with different features, needs, requirements and preferences.

And finally the feature which is also the subject of our study, the student model that keeps information relevant to the student's knowledge level as well as other of his features. Basically, it forms a data collection and assumptions for the users (either as one, or as a group) that are necessary for adjusting the system to their needs. An intelligent education system should have that for every student in order to offer specialized services. Without it, an education system behaves the same way to all students without taking into account any of their special abilities/features. In particular, student model's role is to provide information about the goals and plans of the student, about their knowledge and its distance from the knowledge offered by the system, so as to help in the tracing- diagnosis of misleading, omissions and mistakes from the student.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Possible student features included in the model are the following:

- goals,
- existing knowledge-abilities,
- special features,
- preferences, interests,
- peculiarities-style,
- behavioral motives between user-student,
- Conclusion extraction mechanisms and others.

The last but not least process running by the student model, is to observe how well the student performs on the subject taught. A possible added process is to report false student conceptions. All gathered information collected by the student model, must be ready to use by other system models (i.e. the Pedagogical Model). Information kept by the student model reflects the system's perception regarding the student's knowledge.

2.2 The Content of the Student Model

There are many possible student features that can be stored on the student model and a problem there is choosing the most appropriate of them. The student model should not be either incomplete (the system's adjustability will fail), or too complicated (causing a big congestion on the functionality of the system). The feature values of student come up either during interaction with the system during the learning process or given directly by him. The system must be able to recognize any changes made to the student features during interacting with him and update the appropriate fields on the student model.

According to [11], the student model includes static information that doesn't change during the learning process. Information such as the student's identity and his final goal are variable information that changes dynamically through the learning process. Similar to that, is information about the student's understanding of concepts and the

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

next material that has to be taught. According to [12], the student model usually consists of two parts.

The first part includes information about the student. That's information about his profile (preferences, needs, favorite learning methods). There is also information about his initial knowledge level about his current objective/subject (beginner, intermediate, advanced). The student can see his profile either through a questionnaire or a dialog window. Additional information recorded by the system refers to his learning and concentration ability. The student's responses during interacting with the system determine how high or low these values would be.

Moreover, information about the student's preferences regarding the multimedia type of the teaching units is also recorded. All these preferences are being recorded (that's one version) when the student gains an account in the system though he can change this information during the learning process. The system assumes that the educational goal is a sum of concepts that need to be well understood by the student. Conquering this goal will be done with the help of adjustable navigation mechanisms, beginning with some basic concepts and moving sequentially to the next concepts of the current goal. The student's goals are a feature that changes quite often. These goals could be either of high (cognitive goals) or low level (exercise solving). [13].

In the second part, which is actually more interesting, there is information regarding the student's relation to concepts of the subject to be taught, or in other words, how well he knows these concepts. Some of the models used to describe this kind of student knowledge are coming from ITS and we will describe them in the following section 2.4 (Methods of Student Modeling).

Certainly the level comprehension of the students regarding the learning subject needs to be recorded. But in what way should it be represented? It would be quite exaggerated for the guide to say that "the student doesn't know this subject" or "the student knows this subject". Another exaggerated case would be for the guide to report every single action of the student. Most student models are somewhere amongst these cases and try to model the student in the same detail that the field's knowledge is represented. Additionally, by recording student comprehension level, a student model can include even general pedagogical information for the student. This kind of

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

information includes his general preferences, like for example whether he likes to study examples before answering questions or not.

2.3 Characteristics of Students

Every customized system takes into account some certain student features and that's what makes them different. Brusilovsky (1998) [14] defined seven features, that are being used by already existing systems. These features are:

- *knowledge*
- *goals*
- *background*
- *hyperspace experience*
- *preferences*
- *interests*
- *individual traits*
- *learning rate*

Knowledge

The knowledge of student on the subject represented in cyberspace is the most important customizing feature. Based on student's answers to question, exercises etc, the system indirectly attempts to define the student's current knowledge. This determination can also be done indirectly or directly by the student alone when given the option, for example, to update the system about which concepts he is familiar with and at what degree. Both approaches have advantages and disadvantages. Very often a combination of these two is chosen. The objective is for the customizing mechanism to have valid data available so it can proceed to make the right customizing actions.

Goals

Student's goals or processes are features related to the student's work content in a hyper-media, rather than the student as an isolated unit. The student's learning goal could be either a) static or b) adjustable by the student or tutor. A goal is considered to

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

be “static” when choosing it inevitably leads to a certain and pre-defined sum of cognitive concepts. On the other hand, it’s called “adjustable” when the student or teacher have the option to chose a subset of the sum of cognitive concepts of the current learning goal. According to another kind of grouping, goals are distinguished in a) high level and b) low level. For instance, in education systems the teaching goals are high level goals, whereas problem solving goals are low level. This way they can switch from a teaching problem to another numerous times in a work session.

Background and Experience

We consider background to be general knowledge, abilities, familiarity etc of the student, so it is things that are related indirectly to the subject to be learned. For example, knowledge of a programming language by the student could be a useful clue by system during teaching another language. In the same way, experience in a defined hyper-space would be the level of familiarity a student has about the structure of the hyper-space and therefore how easy it is for him to navigate in it.

Preferences

Every student can have different preferences in a series of issues such as the way of presentation examples, use of images, sound, font type and color etc. These preferences cannot come up by the system guessing itself, but only by the student’s input of such information. In some cases a system allows the student s to adjust some features directly while in others it adjusts the settings automatically after receiving their preferences. As a result, such preferences differ from other student model features because the student must inform the system directly or indirectly about them.

Interests

Student interests have been adopted from “*Information Recovery Hypermedia Systems”,* which try to model long-term student interests, so as to use them along with their searching goals. This way information filtering is enhanced. These features have been applied by many online information systems as well.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Personal Traits

By “personal traits”, we mean all those features that define the student as person like for example, personality traits, cognitive factors, learning types and others. These factors affect learning and therefore it would be useful to model them. Contrary to other features groups; these features offer great stability over time. To define them, we use special psychological tests since this data cannot be obtained by the user directly.

Learning Rate

Students with different learning rate of the teaching material should be regarded differently by it. For example, the system should decrease the help level faster on students with high learning rate. This rate is a user feature compliant to cognitive teaching theories. Knowledge atomization for a subject, for instance, is needed in the field of cognitive psychology. Thus, learning rate definition can lead to added, appropriate to his level occupation of the student with a subject that contains added learning value.

2.4 Methods of Student Modeling

The student model keeps information about the student based on which the system adjusts its functions to the student’s requirements. The most important student model features are the following:

- Knowledge of the system’s knowledge field.
- User’s goals
- Background and experiences
- Preferences

The values of student features occur either during his interaction with the system during the learning process or given directly by him. The system must recognize any changes in student features during interaction and appropriately update the student model. There are many ways to represent student data. Many researchers have tried to

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

classify them and formalize them in a unified framework. Oftenly used are the following techniques:

2.4.1 Overlay Model

One of the most important models used to represent knowledge is the Overlay Model. According to Brusilovsky (1994) [15], for every knowledge field concept we assign a value that depicts how well the student is familiar with it. This model (overlay model) is a system approximation of how well the student knows this concept. It divides the knowledge levels of students using quantitative (from 1 to 100) and qualitative (weak-average-good) values. The student's knowledge level in this model is represented as a part of experts' knowledge of the subject to be learned. He can increase his knowledge by learning but he cannot learn anything more or different than the experts' knowledge.

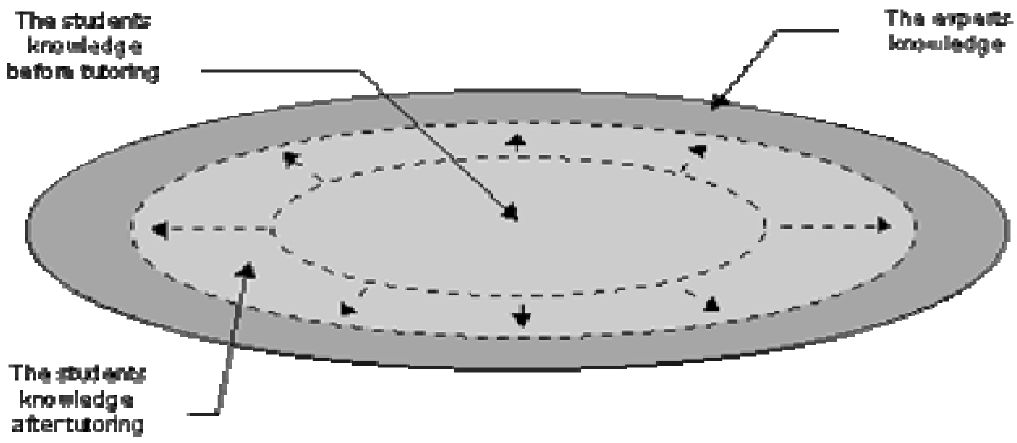

The students knowledge before tutoring
The experts knowledge
The students knowledge aftertutoring

Figure 2-2: Overlay Student Model [16]

The model shown in Figure 2-2 is dynamic, flexible and can be used by many different system sections. It can record the student's knowledge value on a subject regardless of other subjects and because of that we can use the network architecture of the knowledge field on which each concept is a node and links between nodes show the concepts relations. Value updating in this model is easy and is done according to the student's interaction with the system (answers to questions, visited pages, solution to problems).

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

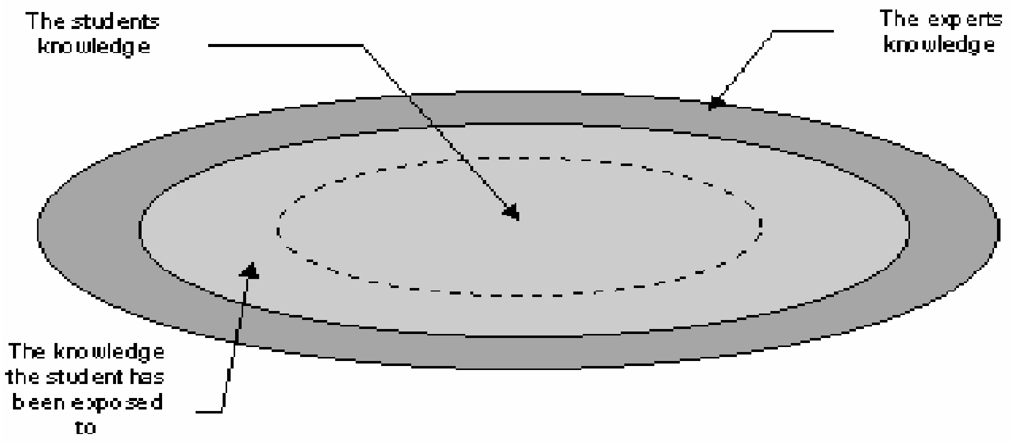


Figure 2-3: Differential Student Model [17]

The Differential Student Model is similar to the overlay model on the basis that the student cannot gain more knowledge than the expert. However, according to the differential model the object to be learned by the student is not 100% sure to be properly understood by him so that it would be equal to the knowledge of the expert. Thus, in the differential model it is not absolutely necessary to avoid differences in knowledge level between the student and the expert, whereas the overlay model makes this a requirement; the student's knowledge spreads until it reaches that of the expert's.

2.4.2 Buggy Model

A disadvantage of the overlay model is its weakness to represent possible misunderstandings of the student. For this reason there is another model invented, the Buggy Model [18], which represents student's knowledge as the union of the subsets of knowledge field and a set of its misunderstandings. The buggy model helps on enhancing user error correction since having a full view about the wrongful knowledge of the students is very useful pedagogically.

There are two variations of the buggy model: bug catalogue and bug-parts-library models. In the bug catalogue model there is a big library of pre-defined misconceptions and it's used to add relevant misconceptions to the student model. A disadvantage of this model is the difficulty in creating the misconception library. On the second

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

variation, student misconceptions are created by the tutoring process by a bug-part library. Usually the library contains symbolic rules with conditions and actions executed when true.

2.4.3 Stereotype Model

According to [19], the stereotype model has some student models stored and classifies a student to a specific category. It’s initialized at the student’s registration by his subject list priority. Usually this is done using a diagnostic test or a questionnaire. Model categories are usually classified according to their cognitive level. The system records the student’s interactions with the subject and updates the student model accordingly.

More specifically Judy Kay refers that [20], a stereotype represents a collection of attributes that often co-occurs in people. An appealing property of the stereotype is that it should enable a system to get started quickly on its customized interaction with the student. That quick start is often based upon a brief initial interaction with the user or less commonly a short period observing the user. For example, we consider the case of a system which teaches Java. If it knows nothing about the student it would logically have a default initial student model for typical person and this might reasonably set all components of the student model to indicate the student knows no Java concepts. This is the implicit stereotype of the typical beginner’s programming book. A second stereotype can be triggered by the user’s claim of no knowledge of Java. This could assign the value “unknown” for components representing the student’s knowledge of the detailed syntax and idiom of Java. Yet another stereotype inference could assign the value unknown to those Java concepts which are quite different from anything in C++. It could also set as unknown those Java concepts which clash with knowledge of C++ because there are similar elements but important differences. The trigger for this stereotype is the user’s claimed expertise in C++ combined with their claimed ignorance of Java.

Therefore stereotypes set pre-defined user classes. A stereotype model is represented as a set of “stereotype-value” pairs where the value determined whether the student belongs to that specific stereotype or not. Stereotype models are simpler

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

and hence they can be initialized and maintained more easily unlike other models. Some models of stereotypes are:

- **Hand-crafted Stereotypes:** In this stereotype, the designer of the system makes assumptions about the stereotype groups. For example, there may be stereotypes for the beginner and the advanced student. Although this approach may be often be ad-hoc its value and importance should not be underrated. Another important potential role for handcrafted stereotypes arises in local customization of systems. For example an experienced teacher can observe their own students. In addition that teacher knows the context of the learning activities. So, that teacher is ideally placed to define stereotypes of the individual knowledge, learning goals and common problems for their own students.
- **Empirically-based Stereotypes:** These approaches do not rely on elicitation of an expert teacher's knowledge of students. Instead, we collect data about students and use this to construct stereotypes. This has considerable appeal where a student works with an online tool such as a spreadsheet. In such cases, it is straightforward to monitor their actions. More broadly, there is an important role of machine learning in acquiring stereotypes as well as careful study of empirical data to identify stereotypes.
- **Stereotypes Inference:** Collection of information for triggering stereotypes comes from three main sources:
  - Directly elicit information from the student
  - Observe the user interacting with the system
  - Diagnostic tasks

The nature of stereotypes makes them especially important as targets for user access and correction. There are two levels of control associated with stereotypes:

- *The whole stereotype*: The student can decide that an active stereotype should be deactivated or vice-versa. So, for example the student can decide to deactivate the beginner stereotype and possibly choose to activate some other.
- *Individual inference level*: The student can alter the value any single inference. For example the student may be content to have the beginner stereotype active.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

A nice approach for the student model is a combination of the stereotypes and overlay models. The overlay model stores a value for every field concept. This value is the system's estimation of the student's familiarity with the concept. Using these values we classify students at stereotype model classes. For example, if a student is classified "intermediate", all "beginner" concepts are omitted.

2.4.4 Uncertainty Model

In some cases, uncertainty models are used to model knowledge and the evaluation process. Such models are probabilistic using e.g. Bayesian networks, fuzzy, neurofuzzy [18]. A Bayesian network is a graphic model that encodes probabilistic relations between variables as well as historical information of these variables.

Bayesian (or probabilistic) networks are graphs. Each variable in a Bayesian network represents statistical objects and links represent mostly causal relations between them. Each node has states or in other cases a set of possible values matched to each variable. Nodes interconnect with each other by pointing arrows (edges) that show the interrelationship of the variables and indicating influence direction. On each link, a possibility is assigned that represents the possibility of the home subject to lead to the linked subject. The possibility of each node to arrive at a state without a current running indication is described using a table of conditional probabilities. These probabilities represent probabilities based on previous information and past experience. Probabilities in some nodes are affected by other node states depending on their co-relationships. Previous information regarding node relationships indicates in some cases the probability of a node to be in a state directly dependant of another's node state.

Fuzzy expert systems constitute the most important applications of fuzzy logic. Such systems - fuzzy rules sets – are used to extract conclusions based on input data. Fuzzy rules include fuzzy variables. The process of extracting conclusions consists of three phases: 1) the process of converting input to fuzzy data (fuzzification) using a relation function, 2) fuzzy rules applying and 3) undoing the fuzzification (to receive the output).

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

2.5 General Considerations about Student Modeling in ITSs

More generally, we can conclude that in the beginning of the teaching process that's done on the internet, the tutor knows very few things about his students and more precisely, about the possible knowledge they have on the subject to be learned. However, given the fact that using the Internet allow for personalized education, at that point it is necessary to collect some specific information about the students so that the teaching environment adapts to the user's features and special abilities from the first minute of teaching.

The easiest way to update the system about a student is directly through the student himself. This is not always easy, so the system must be in the position to be able to create assumptions about the student. Acquiring explicit information can be done in several ways: questionnaires, talking with the student about his preferences, features, learning style, interests. Acquiring implicit information on the other hand is done by heuristics, goal recognition or/and statistics models.

Implemented heuristic rules observe right use, wrong use, student feedback (positive or negative), percentage of known concepts, requests for further explanations or information, and by those the system is led to assumptions about the student. The system submits questions to the student so as to understand his knowledge level. Upon finding inconsistencies, it asks for previously answered questions again, grades the importance and difficulty of concepts and at the same time has rules for acquiring implicit information based on the concepts difficulty.

By goal recognizing, the system tries to trace the student's goals. This is a difficult technique because it is not clear whether the student changed his goal or not and when, because small actions can belong to one or more goals, or the student might pause for a bit or abandon a goal and also because usually there are more than one ways to reach a goal. The techniques used in that case are goal libraries, where there is a very detailed description and for this reason it is only feasible in some restricted fields. In order to create goals, it is necessary to record and compose all possible actions, their pre-requirements and their results.

Statistic models, on the other hand, use sample values they have acquired to extract conclusions about unknown dependent variables. Such variables could be the

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

student's future behavior, his goals, and his next actions and so on. Statistic models claim that the same student will behave in a similar manner in similar situations, and therefore his behavior is predicted by his behavior in similar circumstances. This view is useful for users with idiosyncratic behavior. The disadvantage of this approach is that it requires lots of data by the student himself.

At this point we should refer that it is possible for inconsistencies in student model to exist. This can happen either because of the nature of learning and the changing student features, or due to problems-weaknesses of the information acquiring and updating mechanisms of the student model. Specifically, the student may have learnt something he didn't know, or forgot something he knew, overcame a misunderstanding, changed his preferences, goals, and may have been helped by another person and so on. At the same time, system questions may create a problem due to misunderstanding them, and thus the conclusions extracted by the system may collide with those the user has stated. Besides, implicit information acquiring mechanisms can be wrong statistically from the student's point of view. For all these reasons, control by the student is advised in some cases.

2.6 The Personalization Issue

In this chapter we saw the student model component as part of an ITS system. ITSs while at first had many strategies and used their knowledge for the field and the user to offer personalized learning, they fall short of offering student-centered teaching (Brusilovsky [14,21]) because all ITS systems guided the user directly, or they offered some kind of menu to choose from. There was a need for creating a part that gave the user the ability to follow his own course through the material. The solution was found by including a hyper-media part in the ITSs environment. On Adaptive Educational Hypermedia Systems (AEHSs) users have enough freedom of navigation choices as opposed to Intelligent Tutoring Systems that didn't have that. On the latter, the system controls what appears to the user at a great scale. Therefore, AEHSs constitute a student-centered approach contrary to ITSs that adopt a teacher-centered approach in learning. The biggest part of early customizable hypermedia education systems had a strong connection to ITSs. Hypermedia can offer the base for learning based on

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

discovery but they need a new part that can guide the user. This part can come from ITSs. ITSs and AEHSs look like two different approaches of using the computer in education. In reality though, those two overlap and fill the gaps of each other and systems that include a tutor for guiding the user and a hypermedia part can offer both teacher and student-centered teaching at the same time. The student model of the ITSs can be used by the hypermedia part.

Adaptive Educational Hypermedia Systems (AEHSs) [14,21] develop a standard of goals, preferences and knowledge for each user, and use it during interaction with him, in order to adapt to the user's needs. Therefore the student model is an important element of these systems as it represents certain student features and is being refreshed and modified for all the time of the interaction to adjust several visible elements of the system. On AEHSs users have extensive navigation freedom unlike ITSs where the systems control what appears to the user most of the time. AEHSs try to adjust the content and links of a hypertext to the user's demands. Their goal is to increase functionality of hyper media by personalizing them. This field of research is newer than ITSs. They use different types for student models to adjust the content and links in the user's pages. As stated previously, AEHSs borrow some features of ITSs. Thus, we usually come across features that can be matched to those of ITSs.

In this current research, in accordance to what is stated before (about the student model's operability in ITSs at first and AEHSs later), we interact with the student model as if it is a *global student model*. This means that its operation (a *global student model*) does not aim specifically to teaching a specific Learning Object, but any Learning Object that a student wants to learn at any time. This approach sets some different requirements which we will discuss in the following chapters.

2.7 The IMS Learner Information Package (LIP)

The standards that define the student profile have a goal to determine which information is important for each student, so that they can be shared between different systems. Nowadays the most important user profiling standard is the Learner Information Package (LIP) standard of the IMS organization. The IMS LIP [22] is a data collection for a student or learning content creator. It was designed to abide to the following requirements:

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- Distributed information: A learner information system may in fact consist of multiple distributed systems that share learner information or that store learner information in a distributed fashion. This necessitates the inclusion of adequate indexing and time stamping of learner information data as it is packaged.
- Scalability: To support large-scale systems it is necessary to exchange and reassemble chunks of arbitrary granularity as well as bulk transfer. Packaging of multiple LIPs will use the IMS Content &Packaging specification.
- Privacy and Data Protection: Learner information systems must be able to implement privacy and data protection policies and insure the integrity of data.
- Flexibility and External references: Learner information includes many constructs, such as learning objectives and learning history, which are in practice represented by different structures in different contexts. Learner information data models must be flexible enough to accommodate this need.

IMS Learner Information Package is a structured information model. The information model contains both data and meta-data about that data. The model defines fields into which the data can be placed and the type of data that may be put into these fields. Typical data might be the name of a learner, a course or training completed, a learning objective, a preference for a particular type of technology, and so on. Meta-data about each field can include:

- Time-related information,
- Identification and indexing information,
- Privacy and data protection information.

The Learner Information Model can be viewed in three different ways:

- A tree,
- An object model,
- A tabular representation.

All three ways are explained in the specification [22]. The Learner Information is separated into eleven main categories (as shown in Figure 2-4). These structures have been identified as the primary data structures that are required to support Learner

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Information. This composite approach means that only the required information needs to be packaged and stored.

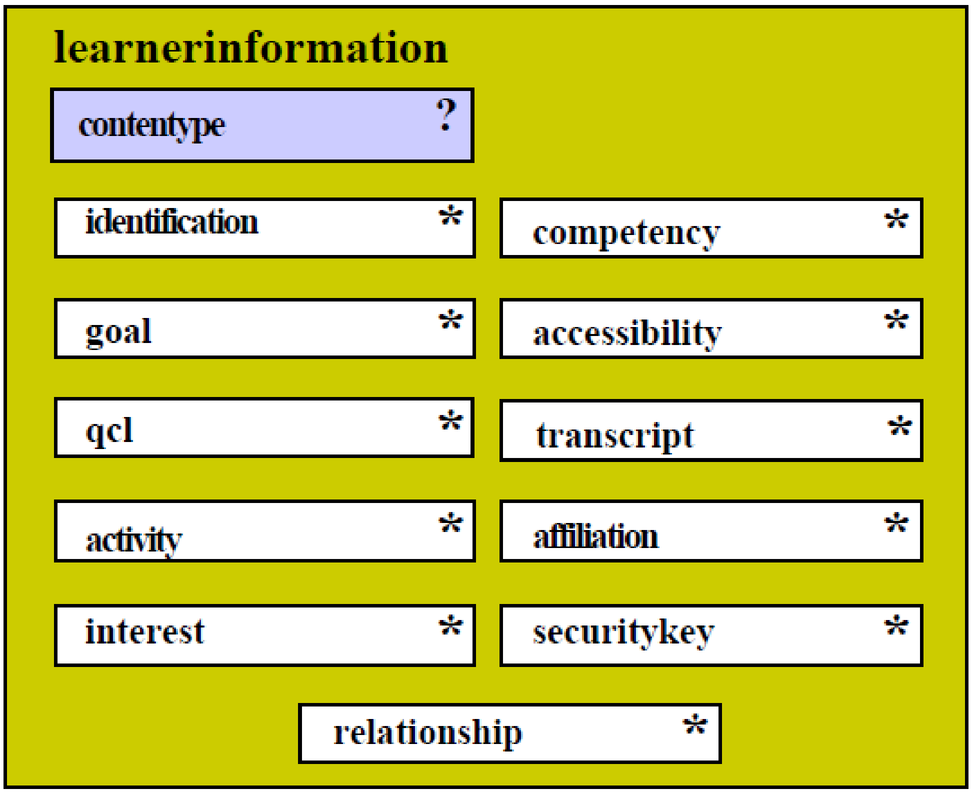


Figure 2-4: The IMS Learner Information Package (LIP) core data structures [22]

- *Identification*: The identification learner information contains all of the data for a specific individual or organization. This includes data such as: name, address, contact information, agent and demographics.
- *Goal*: The goal learner information consists of the description of the personal objectives and aspirations. These descriptions may also include information for monitoring the progress in achieving the goals. A goal can be defined in terms of sub-goals. A different 'goal' structure will be used for each entry.
- *Qualifications, Certifications and Licenses (qcl)*: The qcl learner information consists of the qualifications, certifications and licenses awarded to the learner i.e. the formally recognized products of their learning and work history. This includes information on the awarding body and may also include electronic copies of the actual documents. A different 'qcl' structure will be used for each qualification, etc.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- *Activity*: The activity learner information consists of the education/training, work and service (military, community, voluntary, etc.) record and products (excluding formal awards). This information may include the descriptions of the courses undertaken and the records of the corresponding assessment. A separate ‘activity’ structure will be used for each entry.
- *Transcript*: The transcript learner information is used to store the summary records of the academic performance at an institution. This information may contain an arbitrary level of detail and so there is no proscribed structure for a transcript.
- *Interest*: The interest learner information consists of descriptions of hobbies and other recreational activities. These interests may have formal awards. Electronic versions of the products of these interests may also be contained. Each interest will be described within its own ‘interest’ structure.
- *Competency*: The competency learner information consists of the descriptions of the skills the learner has acquired. These skills may be associated with some formal or informal training or work history (described in the ‘activity’) and formal awards (described in the ‘qcl’). A different ‘competency’ structure will be used for each competency through an external reference mechanism. The adopted competency definition follows the work of the IMS Competency Definition working-group.
- *Affiliation*: The affiliation learner information is used to store the descriptions of the organization affiliations associated with the learner. These affiliations may include education groups e.g. classes, cohorts, etc. but it is expected that these will be exchanged using the IMS Enterprise specification technique.
- *Accessibility*: The accessibility learner information consists of the cognitive, technical and physical preferences for the learner, disability, eligibility and language capabilities. These describe the learner’s capabilities to interact with the learning environment.
- *Security key*: The security key learner information is used to store the passwords and security codes that are to be used when communicating with the learner. A different ‘securitykey’ structure will be used for each key and class of key.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- *Relationship:* The relationship learner information is used to store the description of the relations between the other core data structures. All of the relationship information has been removed from the other structures to enable these to be collected at a single place. This structure may also be used to describe mapping relationships to be used by the communicating systems.

Above, we describe the eleven core information types that are considered fundamental to the learner information data structures and the content information used to store information describing the content.

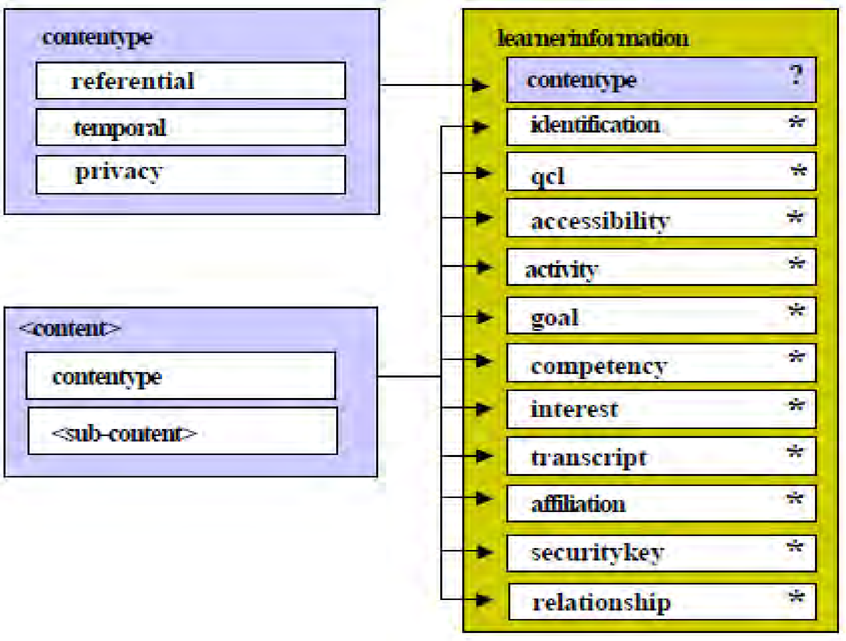


Figure 2-5: The principle LIP data structure [22]

The control information describing the learner information as a whole is contained within the ‘contentype’ class. This class includes the container for the control information that is used to describe the learner information. This information consists of referential, temporal and privacy information and is applied to each of the ‘atomic’ parts of the learner information structure.

- Referential – The referential information is used to uniquely identify the learner information record as a whole and the individual data components within that record. These enable each piece of information to be identified. The actual identification system is outside the scope of this specification.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- Temporal – This information is used to describe any time-based dependencies of the data. This includes information such as the date of creation, time-stamp and expiry date of the learner information.
- Privacy - All of the data relevant to the privacy, authenticity and integrity of the learner information is contained within this structure. The actual privacy etc. mechanism and architectures used to support the learner information are outside of the scope of the specification but they interact with the learner information through these structures.

These eleven categories were chosen to meet the requirements of a large variety of use cases and to facilitate mapping among IMS and other relevant specifications. Also, an XML binding for these categories has been defined but is not meant to exclude other bindings. So, it is possible to exchange information between different LISs.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 3:

Learning in the Semantic Web

3.1 Ontologies and the Semantic Web

The purpose of semantic web research is to allow the vast range of web-accessible information and services to be more effectively exploited by both humans and automated tools. To facilitate this process, RDF and OWL have been developed as standard formats for the sharing and integration of data and knowledge the latter in the form of rich conceptual schemas called ontologies.

3.1.1 The Semantic Web

The Semantic Web [23] is the outcome of the efforts to build a new architecture on which the function of controlling the content with computer systems is upgraded by adding meaning to it. Adding meaning to the content is the most important feature of Semantic Web. To create this feature, Semantic Web is constructed by the following levels:

- **URI/Unicode level**, which is the unique characteristic to of everything on the World Wide Web,
- **XML level**, which represents the data structure of information and its goal, is to achieve interoperability,
- **RDF/RDF-S level**, which represents the meaning of data and its goal, is to provide further interoperability amongst application,
- **OWL level**, which represents agreements on the meaning of data and therefore its goal, is to create meanings. It can be used to define concepts inside a cognitive space and their interconnections,

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- **Digital Signature level**, this level's mission is to trace document changes and along with the ontology level these levels are standardized by the W3C workgroups,
- **Logic, proof and trust levels** constitute the higher architecture levels while at the same time some simple proofing applications are made for them. The level of logic allows for rule writing, while the proof level executes the rules and with the help of the trust level evaluates the applications to decide whether to trust the given proof or not.

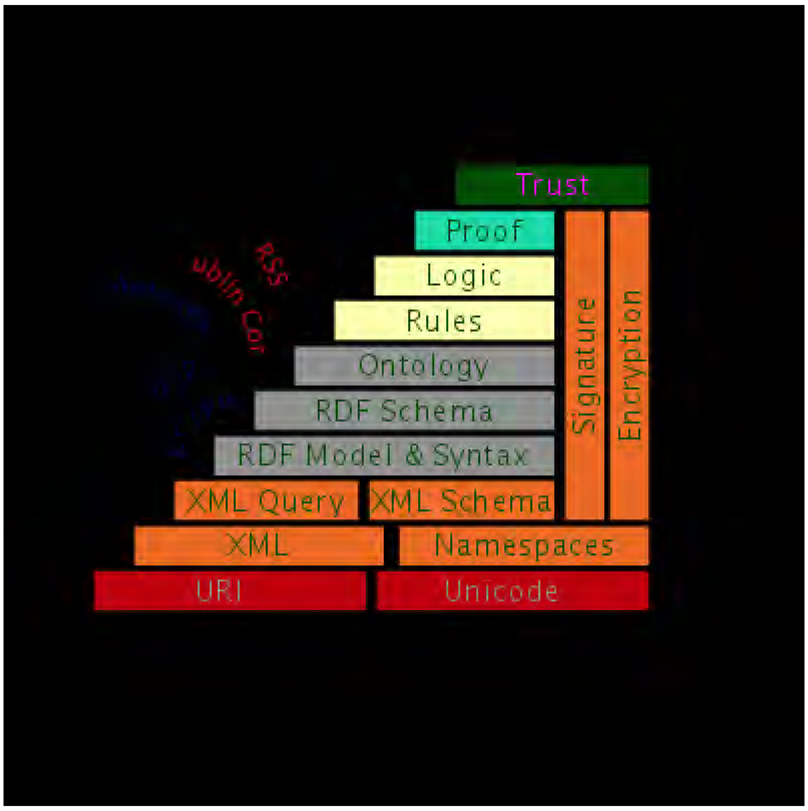


Figure 3-1: Architecture of Semantic Web [24]

The features embedded in the technologies of Semantic Web, constitute it a powerful environment, rich and appropriate to develop e-learning with user personalization abilities. Educating material, in the form of Learning Objects (SCORM), which is digital entities with robust learning value, and with semantic information included in them, can be used in the sense of creating an adaptive system. According to learning goals, preferences, records, efficiency, previous knowledge, and possibly more parameters that define a student, an adaptive system can give him with more precision

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

the suitable learning material for him or the navigation options that cover his needs better.

3.1.2 Ontologies

The Semantic Web is the basic ingredient to create a fully distributed form of artificial intelligence. As we know, artificial intelligence deals with two subjects: 1) Knowledge Representation and 2) Reasoning and searching methods. Thus, in order for the Semantic Web to work, computers must have access to structured knowledge bases and reasoning rules that can use to carry automatic reasoning/inference.

Regarding knowledge representation, the main tool for Semantic Web is Ontologies. Indeed the Semantic Web initiative's greatest contribution until today is standardization it provides to languages and development technologies and ontology usage. More precisely, the most widespread ontology creating language today is Web Ontology Language (OWL). This language's syntax is based on XML and the RDF/RDF Schema (Resource Description Framework). OWL's expressiveness and semantics is defined mostly by Description Logics. Reasoning is done with rule technologies (combined with ontologies) and mostly with Semantic Web Rule Language (SWRL).

The Description Logics term refers to a structured knowledge representation, with whose help we can represent concepts and correlations of an application domain in a formalistic way that allows conclusion extraction through reasoning. Description Logics are subsets of the first logical class and their origin is the Semantic networks and frame systems. They constitute a language family (and not just one language) for defining the vocabulary and description of concepts and correlations that compose an application domain. For this reason, Description Logics are "armed" with typically defined semantics, which is based on logic and oriented towards reasoning processes. Such typical reasoning processes are knowledge base consistency checking, satisfaction checking and instance checking. The basic elements that form such a language are concepts, roles, features and instances. A concept is a description of the common features that describe the instances that in turn belong to that concept. DL's concepts actually represent object classes, which is instance sets, while sets represent binary correlations between instances and actually describe the concepts' features.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

DL’s formalism and semantics are based in logic science. Another important feature that is based on the fact that they are based in logic is the emphasis they give to the conclusion extraction mechanism. Description Logics were the basis of creating Web Ontology Language (OWL), whose target is to represent knowledge on the Internet.

Ontologies constitute a somewhat updated method of representing knowledge, but also, a research field of artificial intelligence in general. An apt and succinct term for ontology would be: “*Ontology is a formal, explicit specification of a shared conceptualization”* [25]. Ontology is a strictly mathematical description of a knowledge domain and includes object classes, which is concepts related to objects. Correlations usually refer to hierarchical dependencies between terms. Other kinds of information that can exist in ontology can be concept properties, restrictions around them, equivalence relations, as well as semantic correlations between concepts using logic. The ontology we design for a domain, is used form a commonly accepted domain vocabulary and to be able to extract conclusions by taking advantage of the modeling elements and their semantics.

There is the case, one application uses multiple ontologies, especially when using modular design of ontologies or when we need to integrate with systems that use other ontologies. In this situation, some operations on ontologies may be needed in order to work with all of them. One of such operations is the merge of ontologies. Merge of ontologies means creation of a new ontology by linking up the existing ones. This procedure between ontologies it is known that appears various types and classification of difficulties with result to give off a number of problems that is required to be solved.

In the literature [26] it is referred that ontology merging problems could broadly be classified as technical problems and practical problems. The technical problems refer to the difficulties related to conceptual representation, modeling languages and development of merging tools while the practical problems deal with difficulties associated with the use of merging tools.

The technical problems associated with ontologies deal with two types of difficulties:

- Mismatches, which is the existence of differences between two ontologies and
- Versioning, which is related to the evolution of ontologies over time.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Ontology mismatches problem has to do with two levels of difficulties. At the first level, different ontologies use different conceptual models to represent a domain of interest while at the second level different ontologies use different modeling languages to represent their conceptual models. In our approach, the case of "different modeling languages" is not valid because all our ontologies were created by the OWL language, and only that.

In terms of the first level, conceptual mismatch, we can refer two main categories. These categories have to do with differences that exist on the concept / class or relation level. A class mismatch occurs when classes and their subclasses are differently conceptualized. For example, two ontologies could use the same name for two concepts, but have different restrictions between them, therefore a different conceptualization. Or another example could be an ontology that has an entry named A, and that same named entry to exist in another ontology named B, plus all the remaining restrictions and concepts being identical between them. In that case we are talking about the same entry appearing in both ontologies with a different name. This category includes also a categorization mismatch or aggregation-level mismatch. A categorization mismatch arises when two ontologies recognize the same class, but each decomposes this class to different subclasses while an aggregation-level mismatch exists when both ontologies model the same class, but each defines the class at a different level of abstraction. In the case of relation mismatch there is a difference in the relations between ontology concepts / classes. This includes hierarchal relations, other ontological-relations, and assignment of class attributes. For example a structure mismatch arises when two ontologies share classes, but differ in the way these classes are structured through ontology relations.

At the second level, language mismatches, according to [26] there are four types:

- *Syntax*: Clearly, different languages have different syntax. A syntax difference is considered the simplest type of mismatch. Such mismatch is easily overcome by using rewriting technique.
- *Logical representation*: Different languages may use different representations to express logical sentences. This type of mismatch is relatively simple, since logical equivalence is easy to be defined.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- *Semantics of primitives*: Same syntax may have different semantics in different languages. So, two languages may appear to use the same syntax, although in fact syntax may differ in semantics.
- *Language expressivity*: A languages may not be able to express functions or logical sentences that are expressed by another language. This constitutes the most critical type of language mismatches.

Regarding the case of the practical problems, are separated into three main problems [26]:

- *Finding mappings*: Identifying matching candidates is a difficult task. The difficulty arises from the possibly different similarities that could exist between two matching candidates: 1) both terms are exactly matching, 2) both terms are equivalent in meaning, 3) one term is more general than the other, 4) or both terms are incompatible.
- *Diagnosis*: Implications of matching are difficult to assess.
- *Repeatability*: Ontology sources used to generate matching evolve over time. Thus repeated merging and matching may be required.

As is referred in above paragraph the OWL (Web Ontology Language) is a family of knowledge representation languages for authoring ontologies. It has been designed to be used by applications whose goal is to process information [27]. OWL facilitates internet content translation ability and by appropriate engines that can’t use it. This capability of framing content is better than that offered by XML, RDF and RDFS *RDF Schema), which are considered to be the main content describing languages alongside OWL. According to its creators [27,28], it includes an additional vocabulary compared to the languages mentioned above, and for this reason it is more powerful at content describing. An important detail about OWL is that it is an ontology itself as well. This results in many ontology standards existing on the internet with namespaces ready for use.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

3.1.3 Querying an Ontology

This part mainly focuses on how knowledge can be retrieved from the ontology. This can be achieved with the help of query languages. Ontology languages and corresponding query languages play key roles for representing and processing information about the real world for the emerging Semantic Web. Ontology query languages were developed to query the information defined by these ontology languages and reasoning systems. There are many different query languages (RQL, SquishQL, RDFQL, SPARQL, VERSA, SQWRL) that have been created for searching and acquiring information. Out of those, SPARWL is the most widespread, as it is the evolution product of rdfDB, RDQL, SeRQLand it is also the choice of W3C [29]. SPARQL is a query language that provides access to RDF graphs and also methods for accessing primary information, plus the ability to export sub-graphs and create graphs based on the answer to a query. However, SPARQL has no native understanding of OWL. In opposite, SQWRL (**S**emantic Query-Enhanced Web Rule Language) is an SWRL-based language for querying OWL ontologies. It provides SQL-like operations to retrieve knowledge from OWL.

As discussed in the SQWRL documentation [30], SQWRL takes a standard SWRL rule antecedent and effectively treats it as a pattern specification for a query. This language replaces the rule consequent with a retrieval specification. SQWRL uses SWRL's built-in facility as an extension point. Using built-ins, it defines a set of operators that can be used to construct retrieval specifications. The attractiveness of this approach is that no syntactic extensions are required to SWRL. In the sequel we present some explanations about SQWRL taken from its documentation, together with some examples of our own.

Core Language Features

The core SQWRL operator is *sqwrl:select*. It takes one or more arguments, which are typically variables, used in the pattern specification of the query and builds a table using the arguments as the columns of the table. For example, the following query retrieves all students in an ontology with a known height that is less than 1.70 cm, together with their ages:

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Student (?s) ^ has Age (?s, ?b) ^ swrlb:lessThan (?s, 1,70) → sqwrl:select (?s, ?b)

This query will return pairs of individuals and ages with one row for each pair. Results can be ordered using the *orderBy* and *orderByDescending* built-ins. For example, a query to return a list of students ordered by age can be written:

Student (?s) ^ has Age (?s, ?b) → sqwrl:select (?s, ?b) ^ sqwrl:orderBy (?b)

The left hand side of a SQWRL query operates like a standard SWRL rule antecedent with its associated semantics. So, for example the atom *Student (?s)* will match not only all OWL individuals that are directly of class *Student* but will also match individuals that are entailed by the ontology to be individuals of that class. These (*select* and *orderBy*) are some of the built-ins that SQWRL language supports, of course there are more that we don't mention. An OWL reasoner that wishes to support SQWRL must obviously implement its built-in operators.

Moreover, Protégé 4 comes equipped with reasoners and a query language (DL Query). The Protégé DL query tab combined with the Pellet reasoning tool are able to retrieve all instances or related classes, given a class expression query in DL. Therefore, any question which can be correctly translated into such class expression query will be answered with the facts that exist in the knowledge base. Of course, translating normal questions into DL queries involves knowing the taxonomy (classes and properties) of the ontology. The query has to be in a standard DL form. The results are the individuals from the resulting fact classes. Any subclasses, superclasses or equivalent classes can also be retrieved easily by simply asking to display them in the results. For example let's suppose that we have an ontology like this:

- Class:
  - Student
- Data Properties:
  - hasGivenName
  - hasSurname

And suppose also that we have several hundred instances of class Student in our ontology. To find an individual named "Chris", we could enter the following query:

hasGivenName value "Chris

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

But clicking on the execute button may not return any results. We also need to check the "Individuals" option. Any individuals found will then be displayed in the query results as shown below:

Students and hasGivenName value "Chris"

We could also show all instances of person by simply providing the class in the query like this:

hasAge value "17"^^long

3.2 Learning Objects Theory

In a general definition a "Learning Object" is every ontology or "unit" that can be (re)used to support learning processes. Every digital resource can be reused to support learning. It's a small, reusable teaching bit of digital information that teachers and tutors can archive and use to build their subjects, and also to share it with others [31]. Learning Objects are based on object-oriented programming and new teaching approaches and designs.

A Learning Object can be considered to contain three elements: teaching material, metadata, and ability of communication with an administration system. While even a simple image can be considered a Learning Object, usually they are more complex. Also, although there's been much speculation about Learning Object being made automatically, they are still being created by real people. Archiving and distribution ability help avoiding re-creating already existing Learning Object, they promote cooperation in education globally, and provide important support at learning. The Learning Object concept supports the idea of teaching material moving inside the Internet and combine in bigger teaching sections [32]. These sections can appear in any form and contain text, video, sound, graphics and multimedia. At the same time, they can include university traditions, scenarios, emulations, subject plans, case studies and valuations [31]. The teaching process in distance learning is different this way and the teaching material plays the most important role.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

The Learning Objects are designed to support concept or process understanding. Their goal is not to teach a whole subject or section. Learning is offered in the form of a “package”, which the user is either obliged to study in its fullness, or choose what he‘s looking for after overviewing the package.
A notable Learning Object feature is reusability. This creates some important financial gains and also helps the scientific community a lot in evolving already created knowledge. This results in time saving and promoting interdisciplinary. In order for learning material sharing to be possible, every Learning Object is accompanied by additional which describes the type and form of its content, the concepts involved in it, its creator, difficulty level for an average student, the type and level of interaction it offers and more. All this information is in specific format and it is encoded in XML in order to be globally understandable by every learning material composition and administration system. Most researchers suggest the method of metadata, kept in a separate file which the system can access without having to open or show the Learning Object contents.

Metadata allows software agents or systems to choose Learning Objects from global repositories, according to already given search criteria. Learning Object Metadata (LOM) were created and developed to solve the problem of finding material by tutors. Through optimized metadata connected to teaching the objects, it is possible for teachers to find, collect and use parts of teaching material.

Applying metadata in Learning Objects means adding appropriate descriptions and values to digital resource elements. This process must be implemented by the right people, meaning the teachers and learning material designers because it requires precise determination and careful labeling so that the concepts are given correctly and are related to the main subject to help users make their searches easier. Libraries are usually responsible for organizing and describing metadata and in this case they need to cooperate with the right specialists to achieve correct description. Standards allow interoperability, while sorting helps ordering as they sort and group according to presumed natural relationships.

In order to achieve inter-operability, accessibility and reusability of the learning content must follow some rules. The e-learning metadata standards are typical descriptions of the terms used for the semantic commenting of the learning material. An e-learning standard can refer either to the structure of a Learning Object (using

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

metadata), or to content sharing by using content structure models, though there are standards that refer to both. The constantly raising need for standard definition led many known Organizations and Foundations to work on this subject and publicize their ideas:

- **ADL/SCORM** – Advanced Distributed Learning/Sharable Content Object Reference Model
- **AICC** – Aviation Industry Computer Based Training Committee
- **ARIADNE** – Alliance of Remote Instructional Authoring & Distribution Network for Europe
- **DC** – Dublin Core
- **IEEE/LOM** – Institute of Electrical and Electronic Engineers/Learning Object Metadata
- **IMS** – Instructional Management System

The majority of existing standards today use XML (IEEE LOM, IMS, and SCORM) as the coding language for specifications. Lately, an effort is made to define RDF bindings for some of these standards (IEEE LOM, IMS). This move is led by the fact that RDF has some important advantages over XML. Firstly, the use of RDF enhances inter-operability between standards. This is achieved thanks to the unique storing model used for different data and shape types. Secondly, a common problem is that there are not standardized ways of vocabulary encoding and distribution. RDF solves problems related to vocabularies, as these constitute a fundamental part of the RDF schema description. It is also important that metadata reusability is made easier. Defining metadata of any level is as easy as defining simple metadata.

3.2.1 IEEE Learning Object Metadata (LOM)

As defined by the Learning Object Metadata Standard [33], its goal is to “facilitate searching, evaluating, obtaining and using of learning content by students, tutors or automatic software processes”. Additionally, this multi-protocol facilitates sharing, diffusion and Learning Object exchanging, making general and special catalog

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

development easier while taking the cultural and language environment variety on which their metadata will be reused into account.
The standard defines nine optional categories for over seventy metadata elements related to Learning Objects, in respect to the following granularity hierarchy [33]:

- **1st level:** Curriculum
- **2nd level:** Course
- **3rd level:** Unit
- **4th level:** Topic
- **5th level:** Lesson
- **6th level:** Fragment

a) Category **General** groups the general information that describes the learning content as a whole.
b) Category **Lifecycle** groups all the features related to the history and current status of the learning content and of anything that has affected it during its evolution.
c) Category **Meta-metadata** groups information about the metadata instance itself (not for the Learning Object, which describes the metadata instance).
d) Category **Technical** groups the necessary technical requirements and features of the learning object.
e) Category **Educational** groups the educational and pedagogical features of the Learning Object.
f) Category **Rights** groups copyrights and usage conditions of the Learning Objects.
g) Category **Relation** groups features that define the relation between the Learning Object and other relevant Learning Objects.
h) Category **Annotation** provides comments about the education usage of the Learning Object and information about when and from whom these comments were made.
i) Category **Classification** describes the Learning Object in relation to a certain classification system.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

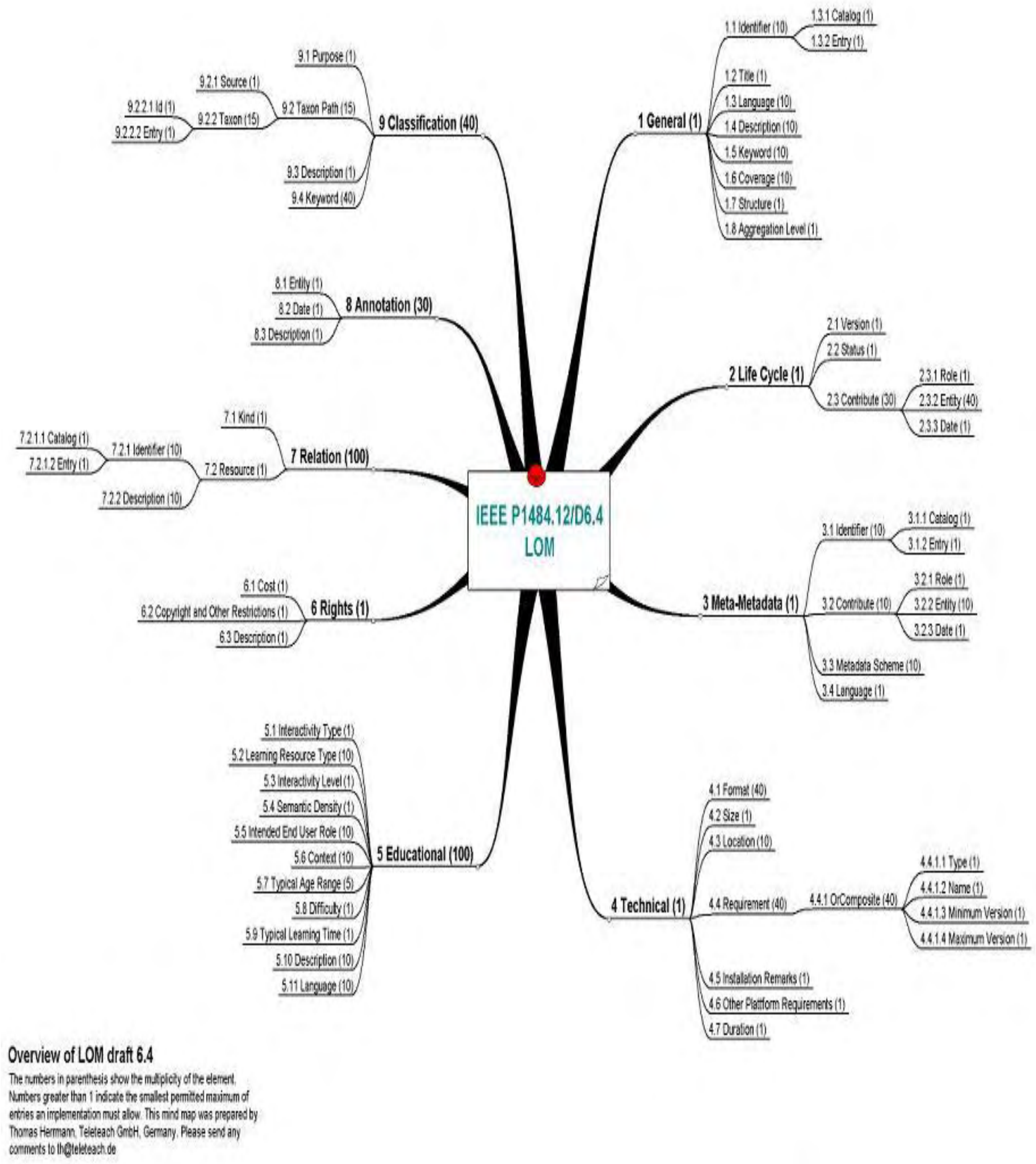


Figure 3-2: LOM elements & structure [34]

Generally we could say that the LOM standard defines two more general categories or metadata frames:

- The first one includes all those elements and information that identify and describe a Learning Object, meaning those attributes that distinguish a Learning Object as being separate and include it in a classification system.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

This frame includes the following general categories: General, Meta-metadata, Lifecycle, Technical, Educational, Rights, Relation, Annotation, and Classification.

- The second includes all those elements and information referred in the usage framework, meaning those elements that define its teaching/learning usage.

This frame in turn includes the following categories: Teaching, Relation, and Annotation. The basic Meta-metadata category cannot be included in these two categories as it does not refer to the Learning Object but to the metadata instance.

3.2.2 Shareable Courseware Object Reference Model (SCORM)

The ADLNet (Advanced Distributed Learning Network) is a serious effort by the US Department of Defense to help handle successfully the huge amount of cognitive and learning resources. ADL’s goal is to provide access to higher quality learning goods that have the ability adjust to personalized student needs and also availability independent of the access point and timing [35].
A part of this goal is SCORM [36], which is until today the greatest initiative taken in e-learning standards. In simple words, SCORM is a set of specifications for developing, organizing and sharing learning material. Using SCORM allows reusability, easy approach and timelessness of learning material against frequent technological changes and also helps in interoperability between different e-learning platforms.
Philip Dodds, head architect of the SCORM development likes to liken the development progress as books in a library, with additional books (capabilities) added to the database when needed. Today there are four books. The last edition, SCORM 2004, released on January 2004 was accompanied by the fourth book and is considered to be stable. SCORM 2004 [36, 37] consists of the following books:

1. SCORM Overview: Covers the history and objectives of the ADL initiative and SCORM and also includes the standards and specifications.
2. SCORM (Content Aggregation Model – CAM): Describes the pieces used in a learning experience, how they are packaged in order to exchange from system

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

to system in order to allow searching and discovering and also the way rules are set for the pieces sequencing process.

3. SCORM (Run-Time Environment–RTE): Describes the Learning Management Systems (LMS) requirements for handling the operation environment. This book describes the communication protocol between an LMS and SCO (objects) and the elements of the modeling data standard used to transfer content that is relevant to the student.
4. SCORM (Sequencing and Navigation – SN): Describes the way for SCORM compatible to obtain sequence through a series of events triggered by the user or the system.

The Run Time Environment of a learning content administration system must include the following:

- Ways of transferring and showing the Learning Object under the command of the administration system.
- Communication method (API) for exchanging data between the administration system and the Learning Object. Standardization provides operations by using Javascript functions.
- The Data model for standardizing naming and form of elements registered in the system. More specifically, a set of data is defined and the administration system must have access to it.

The SCORM Content Aggregation Model represents a learning taxonomy, which helps teaching designers and fitters to concentrate learning resources with a goal to distribute the desired learning experiences. These experiences consist of Activities, which are supported by electronic or not learning resources.
An Activity includes creating, discovering and aggregation of resources in their most primitive form (assets) into more complex learning resources and afterwards organizing them in a predefined sequence distribution. Basically, SCORM CAM consists of the following:

- **Content Model**: It is a naming which defines the pieces of the learning experience content.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- **Content Packaging**: A content package that contains Learning Objects that can have one or more ways in which they are organized and these Learning Objects represent the learning design of the package. A package can correspond to a series of subjects, a separate subject, a subject section or may simply be a somehow related collection of Learning Objects. A learning content package consists of multimedia digital files and XML manifest file.

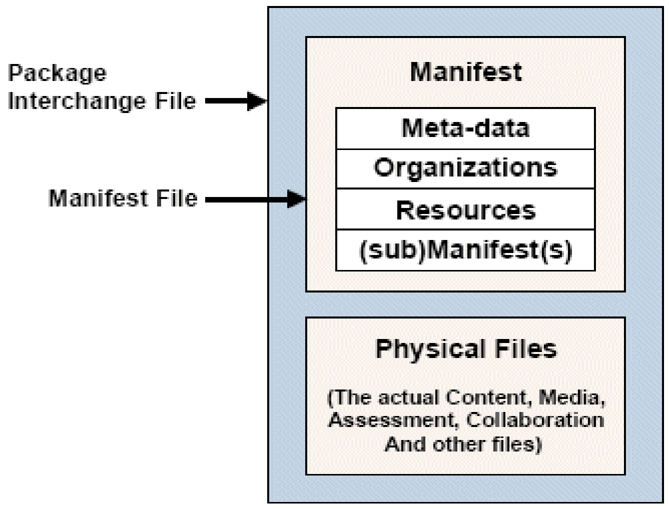


Figure 3-3: Package Structure from the SCORM 1.2 CAM specification [38]

In a XML file the resources are identified, which is the digital multimedia files included in the package, metadata and the learning design that is included in the organization section. The file can also contain additional similar files of lower level (subManifest). Organization section consists of items. Each item in first level can correspond to a learning activity, which can be constituted by sub-activities, and those, in even lower level can correspond to learning resources. Resources in SCORM standardization can be either Learning Objects (SCO) or separate digital elements (assets).

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

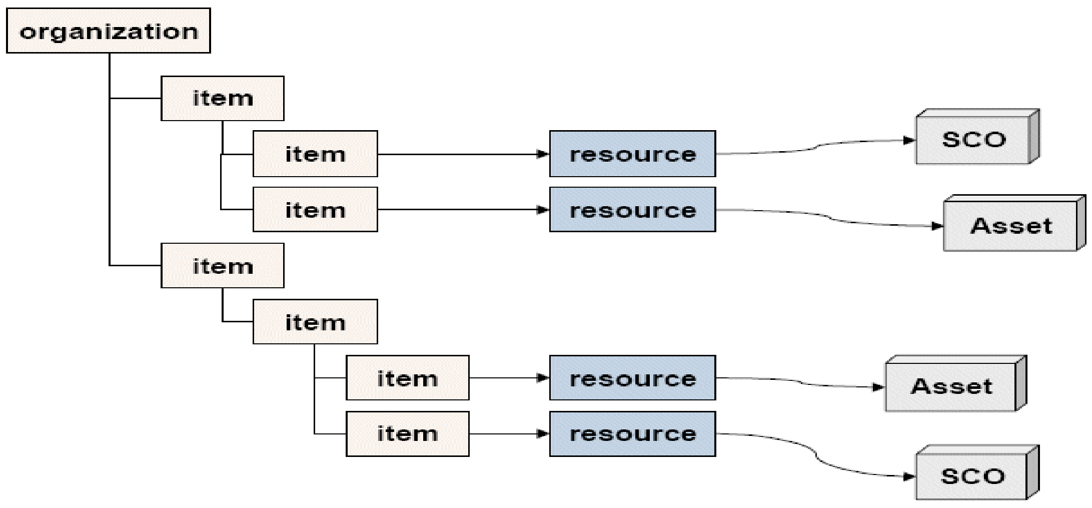


Figure 3-4: Organizations and Items from the SCORM 1.2 CAM specification [39]

Asset: Represents the most basic form of learning resource. Simple resources (assets) are digital representations of media, text, graphics, web pages, evaluation objects or other formats capable of being transferred to an Internet user. An Asset can be described with Asset metadata, so that it is possible to search, discover and reuse it in the data repositories.

SCO: A Sharable Content Object (SCO) represents a collection of one or more resources and is the smallest logical content unit that can be exchanged between LMSs. A SCO can be used in different learning occasions or situations with an ultimate goal of achieving different learning goals. Therefore it is independent of the teaching framework and thus its reusability is enhanced. At the same time, however, completing two or more SCOs creates the necessary background for achieving teaching goals of higher value. A general standard guideline is creating small size SCOs. A SCO can be described by SCO Metadata so that it is made possible to search, discover and reuse it in the data repositories.

- **Meta-data**: A mechanism to describe certain parts of the model content.
- **Sequencing and Navigation**: It is a set of rules that describe the additional sequence and regularization of the activities.

The first section of the SCORM-CAM "book" actually presents the designing and building process of the learning content. Afterwards, the goal is to achieve content

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

availability to students, repositories or LMSs (Learning Management Systems). At this point of the learning resources packaging process, the IMS Content Packaging specifications are applied as well as the IEEE Learning Object Metadata (LOM), so that the learning content is separated from the learning platforms.

3.2.3 Learning Management Systems (LMSs)

Learning Management System (LMS) is the term used to describe a server-based system that is designed to manage learning content and learner interactions. The LMS is a series of software that combine the communications functionality through a computer, the on-line methods of presenting learning material and the administration tools for the learning procedure, altogether binding into a complete Internet learning environment [40]. The LMS is defined as "Software that automates the administration of teaching. A LMS registers users, registers subjects into catalogs, student data and submits reports to the administration. Usually it does not provide authoring capabilities, but focuses on subject administering created by lots of other sources, such as Learning Content Management Systems (LCMS). The LMS is a designed virtual space, which through the lots of multimedia tools provided, succeeds in creating a harmonious and functional cooperation with existing traditional learning environments. In this way they provide the participants the opportunity, not only to come in contact with heterogeneous technologies, but also to incorporate them to their learning course. The LMS users are divided into Learners, Instructors and Administrators, and access to the system is determined by the discrete role that is given to each along with their rights and capabilities. LMSs can be either massively produced for commercial sale, or built to serve certain needs, or they can even be open-source.

Most of the systems that are used widely today in education may be more accurately described as Learning Content Management Systems (LCMS) because they also provide tools to deliver and manage instructor-led synchronous and asynchronous online training based on Learning Object methodology.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

3.3 The CROP View

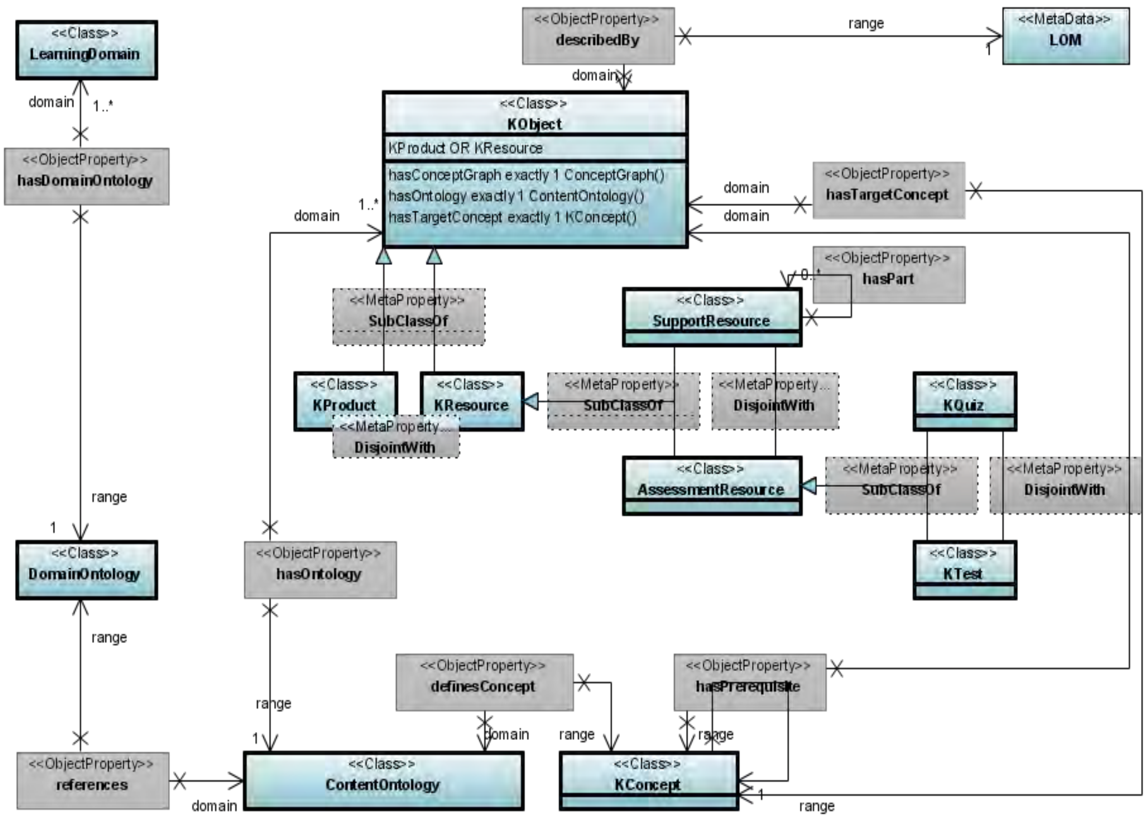
3.3.1 CROP Learning Objects

The CROP (Concept, Resource, Order and Product) is reference architecture for Learning Objects proposed by C. Hartonas [1] in order to support on-the-fly composition of Learning Objects (to suit the needs of a user), resulting from Learning Service communication and exchange of Learning Objects. Contrary to current Learning Object specifications which are quite guiding (they determine and build the Learning Objects content according to certain learning strategies and certain student attributes in order to deliver lessons to a beginner on a certain student goal), the philosophy behind CROP is part of the creating process of a standard whose internal Learning Object Semantic Web structure will not take these certain teaching theories, learning styles and student attributes into account.

Towards this direction the CROP architecture specifies both a formal Learning Object theory (set of axioms and rules specified in Horn Description Logic) and a compositional Learning Object model. Our research constitutes a part of the CROP project, which is a large project with enough concepts under discussion and a number of open problems. In our research defining the static structure of a student model component and trying determine the reasoning issues that results from the interaction between the student and the system we give more emphasize and we try to clarify the KOrder class of CROP architecture.

The CROP [1] reference architecture is centered around four notions, these of Concept, Resource, Order and Product that is indicated by the names of classes' KConcept, KResource, KOrder and KProduct which are presented in following Figure 3-5. The K in front of each notion implies the word Knowledge.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



<<Class>> LearningDomain
<<ObjectProperty>> describedBy
<<MetaData>> LOM
<<Class>> KObject
KProduct OR KResource
hasConceptGraph exactly 1 ConceptGraph()
hasOntology exactly 1 ContentOntology()
hasTargetConcept exactly 1 KConcept()
<<ObjectProperty>> hasDomainOntology
<<ObjectProperty>> hasTargetConcept
<<ObjectProperty>> hasPart
<<MetaProperty>> SubClassOf
<<Class>> SupportResource
<<Class>> KProduct
<<Class>> KResource
<<MetaProperty>> DisjointWith
<<Class>> KQuiz
<<Class>> AssessmentResource
<<Class>> KTest
<<Class>> DomainOntology
<<ObjectProperty>> hasOntology
<<ObjectProperty>> definesConcept
<<ObjectProperty>> hasPrerequisite
<<ObjectProperty>> references
<<Class>> ContentOntology
<<Class>> KConcept
domain
range
1..*
1
0..*

Figure 3-5: CROP Learning Objects and their Content Ontology [41]

CROP's philosophy begins with a KConcept class instance which can be anything that fits the Learning Object requirements. Therefore, it is the learning goal that the user is interested to learn. According to the CROP, each Learning Object provides a Target concept (which represents the exact concept that the user wants to learn) and other concepts whose learning is required to reach the final goal. A concept is a repeated series of concepts that are likely to include other concepts as members. The *part-of* relationship between concepts suggests a logical sense of priority or dependence between them. Thus, we can say that concepts exist which can be the actual prerequisite of other concepts. Furthermore, the CROP model for Learning Object includes the idea of "GroundConcept". A concept is considered as GroundConcept if it does not include any other concepts as members. So, it does not need to satisfy any prerequisite at all. The GroundConcepts of a Learning Object are exactly those that are either knows to the user, or those which the Learning Object keeps as indecomposable to simpler concepts. To summarize, we can infer that the sum of concept items of a Learning Object along with their *part-of* relationships (prerequisites) are the backbone of the ConceptGraph of a Learning Object. Each Learning Object has a ConceptGraph

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

which consists of the concepts the user should have in order to reach the final learning target.

Relationships between concepts of a CROP Learning Object are represented with the help of an ontology, a structured class of concepts and restrictions which describes the concepts of a cognitive space and the relationships amongst them i.e. the content-ontology of the Learning Object. The representation of this knowledge takes place with the OWL language and more specifically the OWL-DL, which, as we have already explained uses the Description Logic language philosophy. Therefore, the concepts are binary correlations between instances are defined by OWL-DL. By taking into account the fact that OWL-DL provides complex concept constructors, including union, a Learning Object's ontology can always have a top concept (Target Concept), which represents exactly the concept the user wants to learn.

On this current architecture the CROP Learning Objects are treated as Knowledge Objects (KObjects). Knowledge Objects come in two different object types, 1) Knowledge Resources (KResources Instances) or 2) Knowledge Products (KProducts instances).

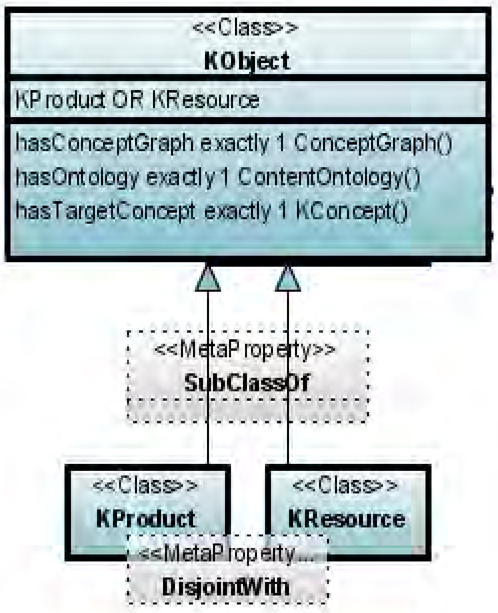


Figure 3-6: Knowledge Objects Structure [41]

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

As shown in Figures 3-5 and 3-6, a CROP KObject is defined as the disjoint union of its subclasses KProduct and KResource. Every KObject has a unique target concept. This is the concept that the KObject is designed to teach to the learner. Associated to a KObject may also be a number of prerequisite concepts. No teaching support is offered to the learner for these concepts. To the contrary, they are assumed to be known to the learner. Each KObject also, is described by its metadata. That is, structured and encoded data which describe KObjects' features, aiming to match, recognize, discover, evaluate and administer the described objects. In the current research we adopt the LOM standard structure for describing metadata and we present the LOM ontology in the following section 4.1.

KResources are atomic KObjects. This means that they are not executable in themselves; they may be atomic resources (text files, images etc) or documents, i.e. collections of atomic resources sequentially structured in a rich document. A KResource item is a learning resource, associated to a file that contains the actual learning material. Every KResource is declared to have a unique target concept (which may nevertheless be an "aggregate" concept, i.e. a union of perhaps interrelated concepts), through the *hasTargetConcept* relation. Further, a KResource item has its own LOM, like KProducts, which captures all metadata information pertaining to the resource (such as technical requirements, in the *Technical* element of the LOM, versioning information, in the *LifeCycle* element, or authorship information, in the *MetaMetaData* element). KResource items are divided further into AssessmentResource and SupportResource depending on whether their role is to provide teaching support or to evaluate student's degree of understanding of the teaching material.

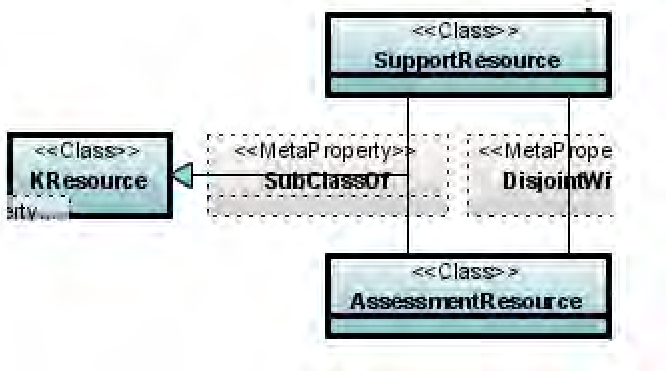


Figure 3-7: Knowledge Resources Structure [41]

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

KProducts are the only main component of a CROP Learning Object (except for its domain ontology). At this point we must say that the terms CROP Learning Object and Knowledge Product (KProduct) have exactly the same meaning in the CROP architecture. However, we use the term KProduct to refer to a Learning Object which is a sub-object of the discussed Learning Object. KProduct is an executable item, contrary to KResource which lacks internal structure, and it consists of the following components:

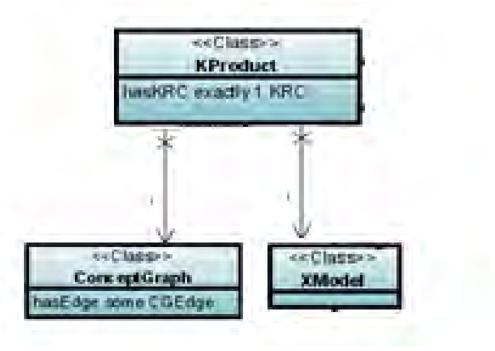


Figure 3-8: Knowledge Products Structure [41]

In the case of a KProduct, in-between the target and the prerequisite concepts there may exist in the content ontology a number of concepts that need to be taught, for successful learning (eventually) of the target concept. Concepts in the content ontology may be related by the *hasPrerequisite* relation. For example, the concept of complex number subsumes the concept of real number and teaching a learner the concept of complex number requires that the learner has acquired the concept of real number. This should not mislead the reader to confuse concept dependence with concept subsumption. The concepts of the content-ontology, together with the *hasPrerequisite* relation, form a graph, the ConceptGraph of the KProduct.

The ConceptGraph is a directed, rooted graph, where the root of the graph (LearningObjective - TargetConcept) is the final node of the graph (source). The ConceptGraph is the mandatory component of a KProduct and it contains two components stated below:

- The *TeachingAct* which includes all the nodes of the graph and which in turn includes another component:
    - The *KObjectList* component which contains the supporting material of the *TeachingAct* node.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- The *TeachingStep* component which constitutes the graph’s edges and also shows the relations between concepts.

For example, to teach learner photography, it is a prerequisite that the concept of camera-part should be first taught, but there is no sub-assumption relation in this example between the concepts of photography and camera-part. Figure 3-9 shows the ConceptGraph for a Learning Object teaching photography.

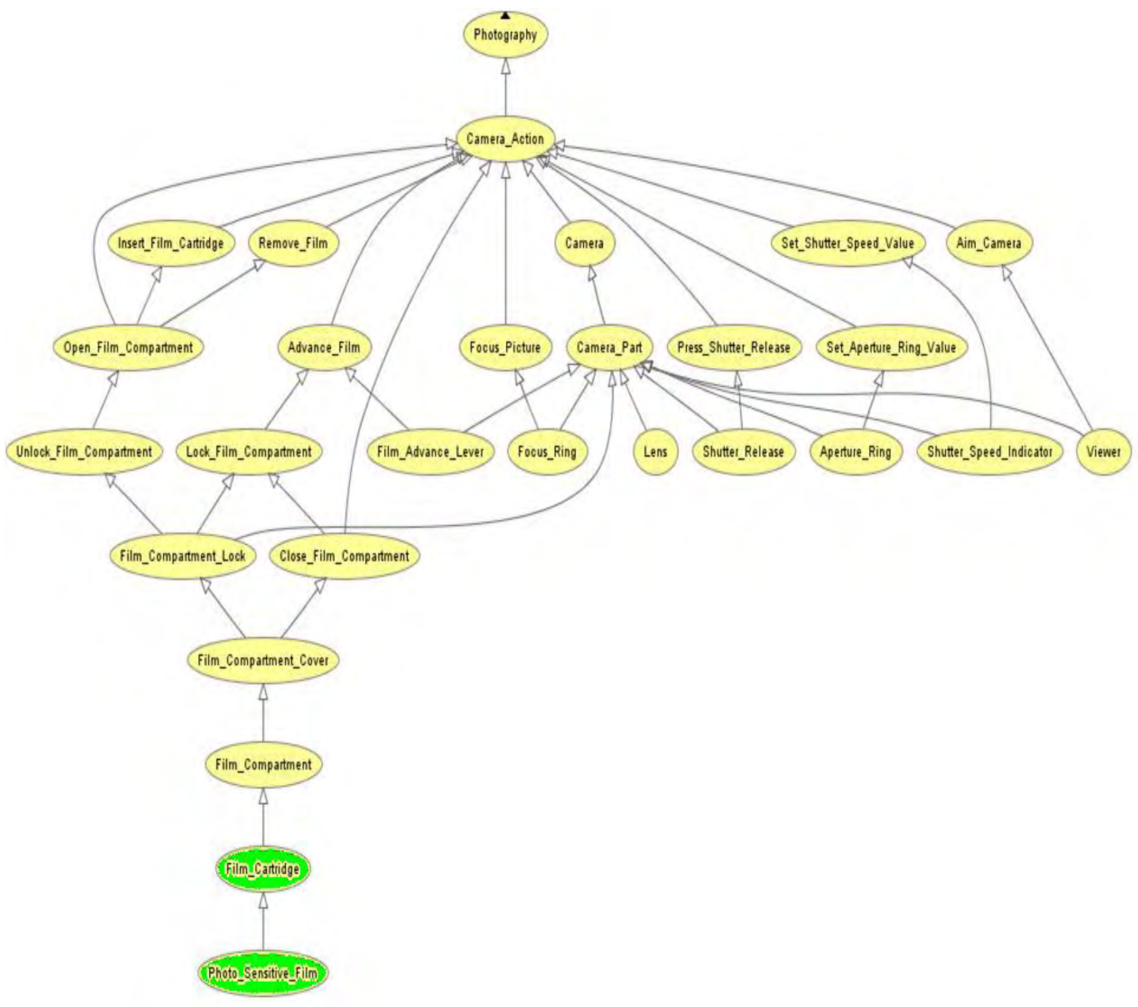


Figure 3-9: The Concept Graph of a Learning Object for Photography [41]

The GroundConcepts (nodes) of the ConceptGraph are the concepts that the author of the Learning Object decides to be the starting points of instruction. These may have prerequisites. In CROP, the ConceptGraph of a KResource may consist of several prerequisites and otherwise a single ground node which is at the same time the top node (and target concept) of the resource. Figure 3-9 shows an example of a Concept

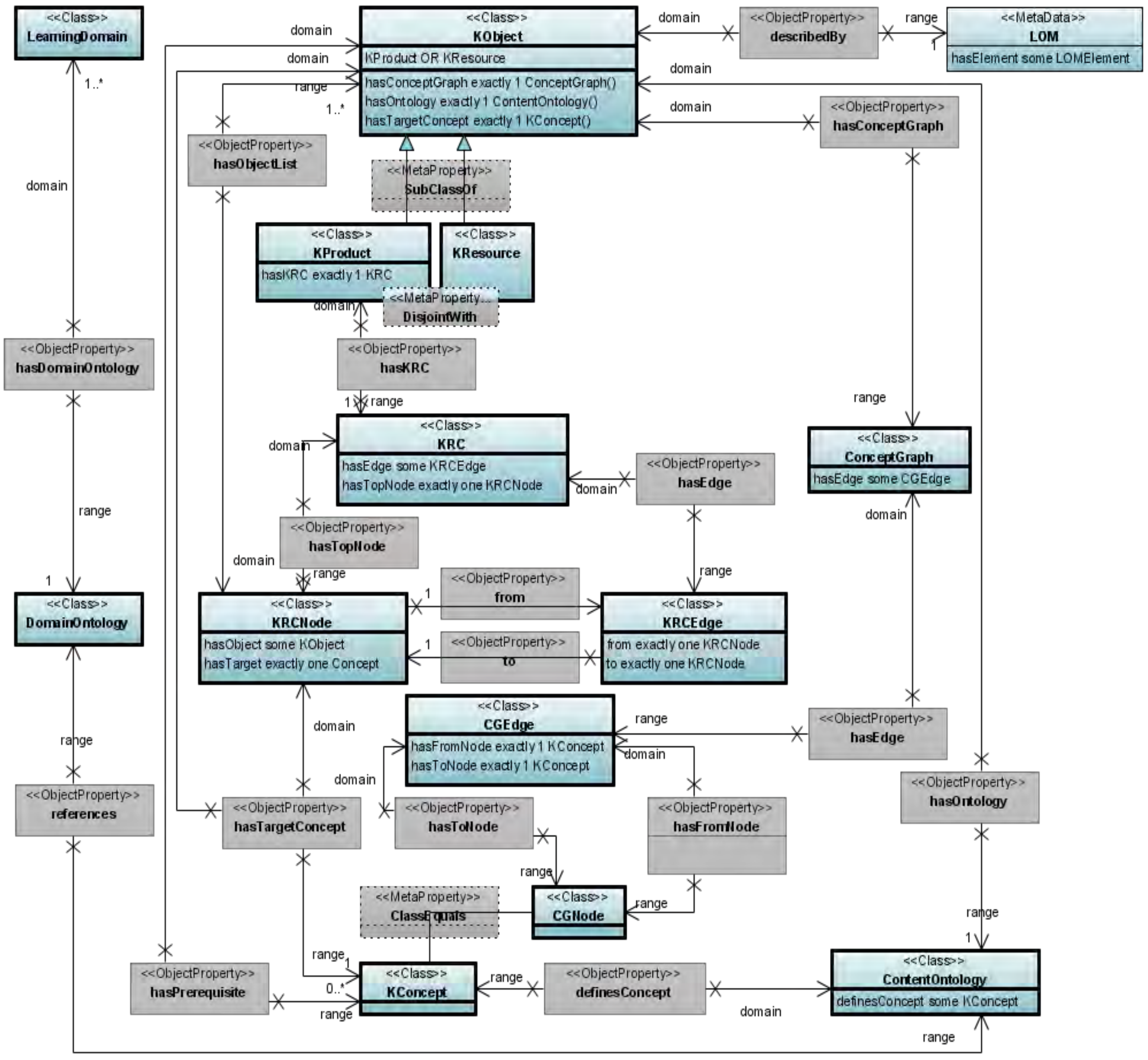
Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Graph for a KProduct where all the yellow leaf nodes, together with the node "Film_Compartment", are considered ground nodes. The set of all prerequisite concepts of the ground concepts of a Learning Object are precisely the prerequisite concepts of the Learning Object itself. In the example of Figure 3-9, these are the concepts "Film_Cartridge" and "Photo_Sensitive_Film". Hence a Learning Object can be abstractly specified as the pair of (prerequisite concepts, target concept). The ConceptGraph determines, in addition, a set of possible Sequencing Rules, relating to possible ways to linearly traverse the graph, while respecting the *prerequisite-of* relation.

Each *TeachingAct* node includes additional supporting material of the concept it is responsible to present. This material is included on the *KObjectList* item whose members are either KProduct or KResource items. The *KObjectList* component contains supporting or evaluating material needed to execute the *TeachingAct* node (executing the ConceptGraph node and teaching the target concept of the node). Adding of a *KObjectList* component on top of the ConceptGraph of a Learning Object gives us the *KRC Graph* of the Learning Object.

A *KRCNode* is defined by specifying a set of KObjects (we refer to this as the *KObjectList* of the *KRCNode*, or the Node Type) whose target concept is the concept of the *KRCNode* and whose prerequisite concepts are amongst the predecessors of the current concept in the ConceptGraph of the object. Figure 3-10 completes the graphical presentation of the CROP ontology, adding the *KRC graph* in the picture.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



<<Class>> LearningDomain
<<Class>> KObject
KProduct OR KResource
hasConceptGraph exactly 1 ConceptGraph()
hasOntology exactly 1 ContentOntology()
hasTargetConcept exactly 1 KConcept()
<<ObjectProperty>> describedBy
<<MetaData>> LOM
hasElement some LOMElement
<<ObjectProperty>> hasConceptGraph
<<ObjectProperty>> hasObjectList
<<MetaProperty>> SubClassOf
<<Class>> KProduct
hasKRC exactly 1 KRC
<<Class>> KResource
<<MetaProperty>> DisjointWith
<<ObjectProperty>> hasDomainOntology
<<ObjectProperty>> hasKRC
<<Class>> KRC
hasEdge some KRCEdge
hasTopNode exactly one KRCNode
<<ObjectProperty>> hasEdge
<<Class>> ConceptGraph
hasEdge some CGEdge
<<ObjectProperty>> hasTopNode
<<Class>> DomainOntology
<<Class>> KRCNode
hasObject some KObject
hasTarget exactly one Concept
<<ObjectProperty>> from
<<ObjectProperty>> to
<<Class>> KRCEdge
from exactly one KRCNode
to exactly one KRCNode
<<Class>> CGEdge
hasFromNode exactly 1 KConcept
hasToNode exactly 1 KConcept
<<ObjectProperty>> hasEdge
<<ObjectProperty>> references
<<ObjectProperty>> hasTargetConcept
<<ObjectProperty>> hasToNode
<<ObjectProperty>> hasFromNode
<<ObjectProperty>> hasOntology
<<MetaProperty>> ClassEquals
<<Class>> CGNode
<<ObjectProperty>> hasPrerequisite
<<Class>> KConcept
<<ObjectProperty>> definesConcept
<<Class>> ContentOntology
definesConcept some KConcept
domain
range
1..*
1
0..*

Figure 3-10: KRC Graph Structure of Learning Object [41]

A *KRCNode* is a self-contained *TeachingAct*, targeting a single KConcept item in the object’s content ontology, and it will be executed provided all of its predecessor nodes in the *KRC* have been executed (unless the user wishes to override the default execution model). CROP Learning Objects can be composed by including one in the node type (the *KObjectList*) of the other, provided the target concept of the component Learning Object coincides with the target concept of the node and that its prerequisite concepts are included amongst the concepts taught at predecessor nodes and the prerequisites of the host object.

Executing a KProduct means the same as executing its *KRC graph*. And this means traversing the graph in a sequential way, respecting the *prerequisite-of* relation, and executing each *KRCNode* in the graph. A *KRCNode* is executed by executing (or

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

displaying, in the case of KResources) all the KObjects in the node's *KObjectList*, following sequencing rules determined in the Execution Model (XModel) of the object. The ExecutionModel is the one model that defines the order in which the ConceptGraph's nodes will be executed during the interaction with the user. The XModel component of the KProduct consists of the following:

- The XGraph, which is built upon the *KRC Graph*. It adds the Control, Dialogue and "Sequencing Rules" nodes, and it is one of the two founding elements of the XModel which in turn consists of XNodes and XEdges. XNodes are the individual kinds that have exactly one *KRCNode* or Dialoge Node or ControlNode component and exactly one XNodeManager. The XNodeManager of an XNode combined with a *KRCNode* are responsible for applying the sequencing rules when executing the *KRCNode's* objects on the *KObjectList*.
- The XGraphManager.

The above learning process requires presence of an administrator, who will handle the interaction between the KProduct and the learner. This role in CROP model goes to the KOrder class instance. A KOrder instance is created for each KProduct and user of the product. A KOrder instance uses an execution model (XModel instance) for executing its associated KProduct. The KOrder also processes any available information on the Learner knowledge level, cognitive profile and preferences, and constructor updates an instance of the LearnerProfile. Further, it monitors the KProduct - Learner interaction and notifies appropriately the LearningService. Such notifications, when the subject of notification is observed learning difficulties or failures, are adaptation triggers for the Learning Service, which may seek to compensate for such shortcomings.

Summarizing, we can infer that the key aspect of a Learning Object is its ontology. This ontology is named content-ontology of the Learning Object. Besides the subsumption relationships, the concepts are related to a *hasPrerequisite* relation who supplies the content-ontology with a graph structure. Additionally, it is possible for closely related concept clusters of the ontology to exist as these come up by the distinct objects or subjects which can be the content-ontology of the Learning Object.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

3.3.2 CROP Learning Services and Learning Domains

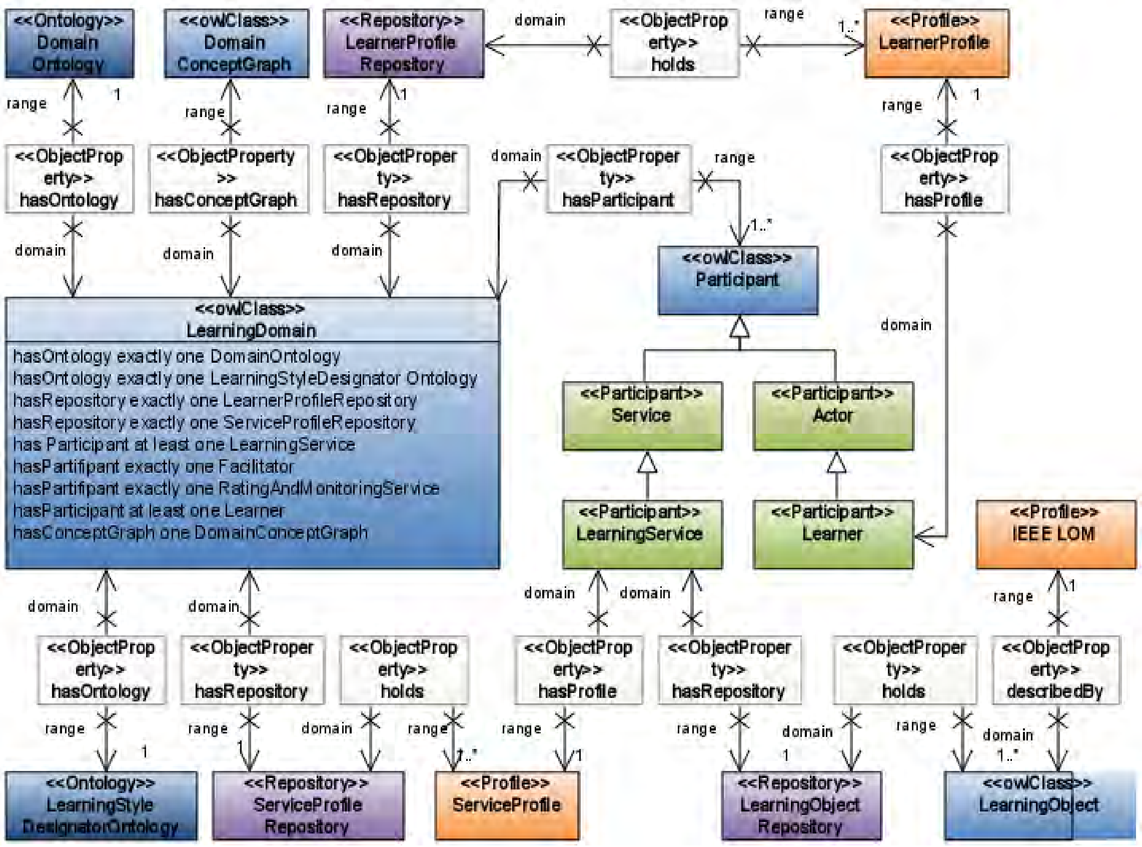
According to the [5], Learning Services conceived as Web Services that provide Learning Objects are organized in Learning Domains and cooperate so as to dynamically compose (at run-time) or modify CROP Learning Objects that meet (a) user features (learning style preferences etc, choices from the user profile), as well as (b) dynamically detected needs or issues (i.e. low test scores, user requested additional explanatory material) during run time.

Services are part of the network (Learning Domain) in which they grow and collaborate. This collaboration of Services in order to construct and adjust Learning Objects implies the existence of reference architecture for the Learning Domain. That architecture for us is the CROP Reference Architecture that we have already explained. It models Learning Objects in a way that allows for typical Object description, as well as composing (dynamic construction) new ones to serve the needs of each student.

The specifications of structure and operation for Learning Domains for the current research were made by developing a Role Model, assigning relationships between two roles (peer roles) and also assigning roles as subjects or behavior sets (behaviors, processes), local (internal) as well as interactive. Interactive behavior, according to the Internet Services standard, consists of message exchanges whose content is facts drawn in an appropriate language. Participants in a Learning Domain are modeled as role sets.

Some typical ontologies of a Learning Space (Figure 3-11) are: Participants (as role sets), (Global) Domain Concept Graph, Ontologies and Repositories.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



<<Ontology>> Domain Ontology
<<owlClass>> Domain ConceptGraph
<<Repository>> LearnerProfile Repository
<<ObjectProperty>> holds
<<Profile>> LearnerProfile
<<ObjectProperty>> hasOntology
<<ObjectProperty>> hasConceptGraph
<<ObjectProperty>> hasRepository
<<ObjectProperty>> hasParticipant
<<ObjectProperty>> hasProfile
<<owlClass>> Participant
<<owlClass>> LearningDomain
<<Participant>> Service
<<Participant>> Actor
<<Participant>> LearningService
<<Participant>> Learner
<<Profile>> IEEE LOM
<<ObjectProperty>> hasOntology
<<ObjectProperty>> hasRepository
<<ObjectProperty>> holds
<<ObjectProperty>> hasProfile
<<ObjectProperty>> hasRepository
<<ObjectProperty>> holds
<<ObjectProperty>> describedBy
<<Ontology>> LearningStyle DesignatorOntology
<<Repository>> ServiceProfile Repository
<<Profile>> ServiceProfile
<<Repository>> LearningObject Repository
<<owlClass>> LearningObject
domain
range
1
1..*

hasOntology exactly one DomainOntology
hasOntology exactly one LearningStyleDesignator Ontology
hasRepository exactly one LearnerProfileRepository
hasRepository exactly one ServiceProfileRepository
has Participant at least one LearningService
hasPartifipant exactly one Facilitator
hasPartifipant exactly one RatingAndMonitoringService
hasParticipant at least one Learner
hasConceptGraph one DomainConceptGraph

Figure 3-11: Learning Domain [5]

There are two Repository types: i) Service Profile Repository and ii) Learner Profile Repository. In the first, Services deposit their own teaching capabilities, so as to be traceable by the Students and by other Services. In the second, the model (student) features are being registered.

Global ontologies clarify concepts that are common in the Learning Domain, such as: a) Domain Ontology, which includes the merging of all Content Ontologies of the Learning Objects in the Domain, and b) Learning Style Ontology, as described in section 4.2.2. In the same way, c) Global ConceptGraph, this includes (as sub-graphs) all the ConceptGraphs of the Learning Domain Objects.

The Participants in the Learning Space are Services and Actors. We can also distinguish the following as important amongst others: Learning Services, Facilitator, Rating & Monitor Services, Learners, and Authors.

The main functions in a Learning Domain are i) search, ii) composition or adjustment and usage (execution) of Learning Objects. Searching is done by students as well as

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

by Learning Services. The first submit a search request for an Object that serves a desirable teaching goal (target concept). Services launch a new search request to respond either to a search request made by a student, or another Service (which at first they don't accomplish), or to adjust an Object which they have, before or during runtime by a Student (after receiving a report from Order). The Service can choose to construct a new Object by adding new objects of pre-requisite concepts unknown to the Student to already existing Objects. In case the Service does not have the pre-requisite Objects, it launches a new search in the Learning Domain, in order to find a Service that will give them to it (collaboration between Learning Services).

Composing or adjusting an object can also take place at runtime of Learning Objects usage, for example if the Student fails to gather the necessary grades when studying an intermediate concept. The Service is updated about this happening by the Order, which monitors the learning process, and tries to re-adapt the Object by modifying related *KRC node*. For example it might offer an additional Object for studying or a new exercise. These new Objects can be provided by the Service itself, otherwise it will have to launch a new search in the Learning Domain, so as to find a Service that provides it.

Collaboration between Services to construct Learning Objects requires the existence of reference architecture for the Learning Domain. One in which the ways of interacting between roles created by the participants in the Learning Domain are configured.

According to the roles model [42] of a Learning Domain, roles are distinguished as shown in Table 3-1.

Role	**Participants Utilizing**
LearningObjectRequestorRole	Learner, Learning Provider
LearnerRole	Learner
FacilitatorRole	Facilitator
AuthorRole	Author
LearningProviderRole	Learning Provider
RatingRole	Monitoring and Rating

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

MonitorRole	Monitoring and Rating
Profile Manager Role	
(Sub) Learning Object Profile Manager Role	Learning Provider
(Sub) Learner Profile Manager Role	Learner, Learning Provider
(Sub)ServiceProfileManagerRole	Learning Provider

Table 3-1: The roles and participants that implement them [42]

In Figure 3-12 is presented a part of the Domain's Roles and their correlations.

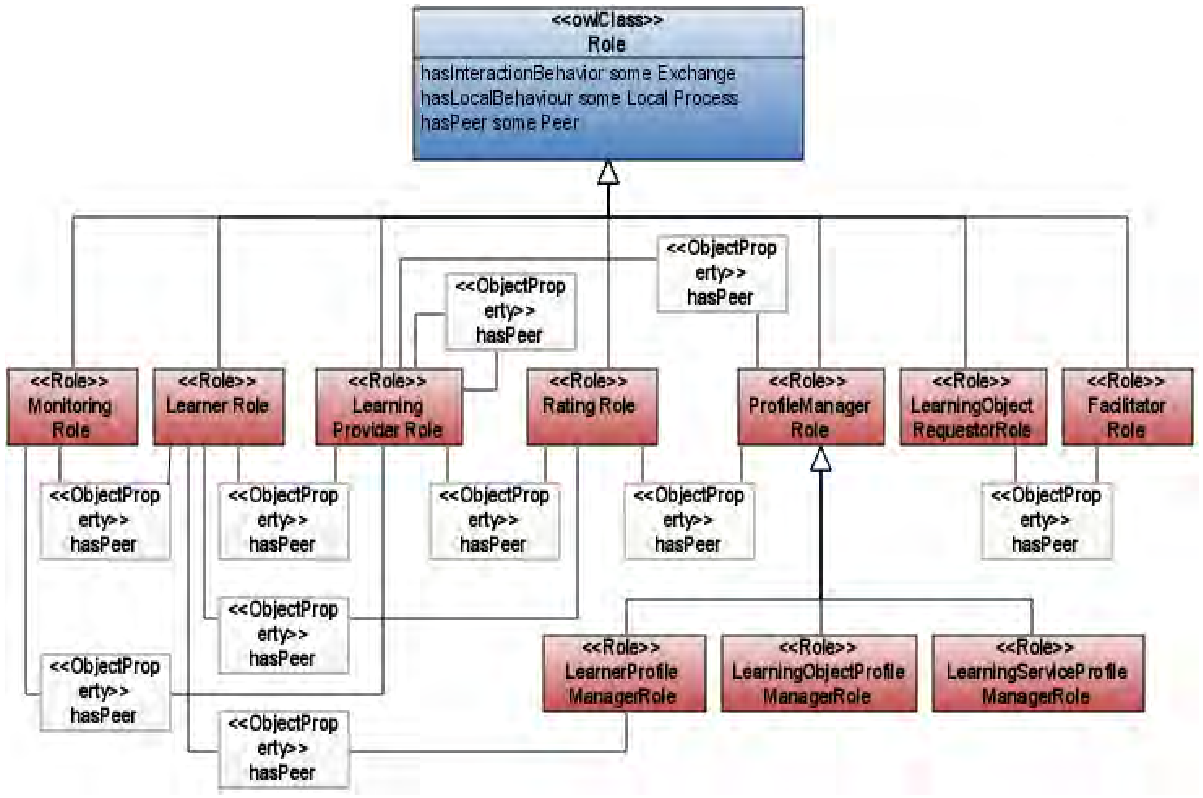

<<owlClass>> Role
hasInteractionBehavior some Exchange
hasLocalBehaviour some Local Process
hasPeer some Peer
<<ObjectProperty>> hasPeer
<<Role>> Monitoring Role
<<Role>> Learner Role
<<Role>> Learning Provider Role
<<Role>> Rating Role
<<Role>> ProfileManager Role
<<Role>> LearningObject RequestorRole
<<Role>> Facilitator Role
<<Role>> LearnerProfile ManagerRole
<<Role>> LearningObjectProfile ManagerRole
<<Role>> LearningServiceProfile ManagerRole

Figure 3-12: Roles in a Learning Domain and their correlations [42]

An elemental interaction scenario between roles in a Learning Domain, the Requesting Role and the Learning Service Provider role is described below.

Let's assume that the first role is realized by a Learning Service that searches for a Learning Object in order to include it in the object it currently constructs.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

The search process is a composite behavior. At first the Requesting role executes the Learning Object Available eXchange query process (asks whether there are available Objects with the requested concept as a target and the desirable pre-requisites, and also other features like duration and density). The Provider receiving the request executes an internal process to determine whether it can provide such Object. In this process, we assume an Object with the desired concept target but without the pre-requisites exists. If it is true, the sub-graph of the global ConceptGraph of the Learning Domain that is defined by the target concept and the pre-requisite concepts is extracted. The Service Provider can then be transformed to a Requesting service (role change), so as to complete an appropriate object. Upon positive outcome it responds positively by executing the Inform Object Available XChange process.

3.4 Related Web Learning Projects

Before we continue with the presentation of our approach we consider advisable to record certain some of the most representative work in the field.

ELENA project [43] started as a European research initiative funded under the IST programme of the European Commission from September 2002 to May 2005. The research team of this project followed the vision of creating Smart Spaces for Learning. These are open environments that support learners in optimizing their learning management. The ELENA project’s goal is to provide personalized access to distributed e-learning repositories, taking advantage of the semantic technologies and metadata description standards. The explicit descriptions of Learning Objects use RDF correlations in LOM and DC while those of the users use RDF Schemas from the PAPI and IMS LIP Standards that provide “questions and answers” capabilities to the P2P structure of Edutella. This approach is based on matching rules between Learning Objects and user descriptions in order to achieve service sentence producing or distributed objects and customized access and teaching.
In this current research our objectives (adaptable learning services in the Semantic Web) are similar to the objectives of the ELENA project. But they are more extensive.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

They expand from a new architecture for Learning Objects to a new architecture for learning management systems, while preserving SCORM compliance. The ELENA project has drawn the attention on providing learning support in a Semantic Web setting. It has done this successfully but to the best of my knowledge it has not resulted in specifying and engineering a full blown product.

RELOAD [44] is a project funded under the JISC Exchange for Learning Programme. The project focuses on the development of tools that are based on emerging learning technology interoperability specifications. The primary aims of this project are to:

- facilitate the creation, sharing and reuse of Learning Objects and services
- enhance the range of pedagogical approaches realizable through the use of lesson plans

These aims will be achieved through the production of a suite of software tools for authoring and delivery of standard-compliant Learning Objects incorporating comprehensive user guides and exemplar resources.

3.5 Learning Styles

The term “Learning Style” is used to describe the personal differences at learning. It is based on the admittance that each person has a discrete way of learning, which means to collect process and organize information. There are lots of definitions given in Bibliography about «learning style», some of which are referenced below. A learning style could be:

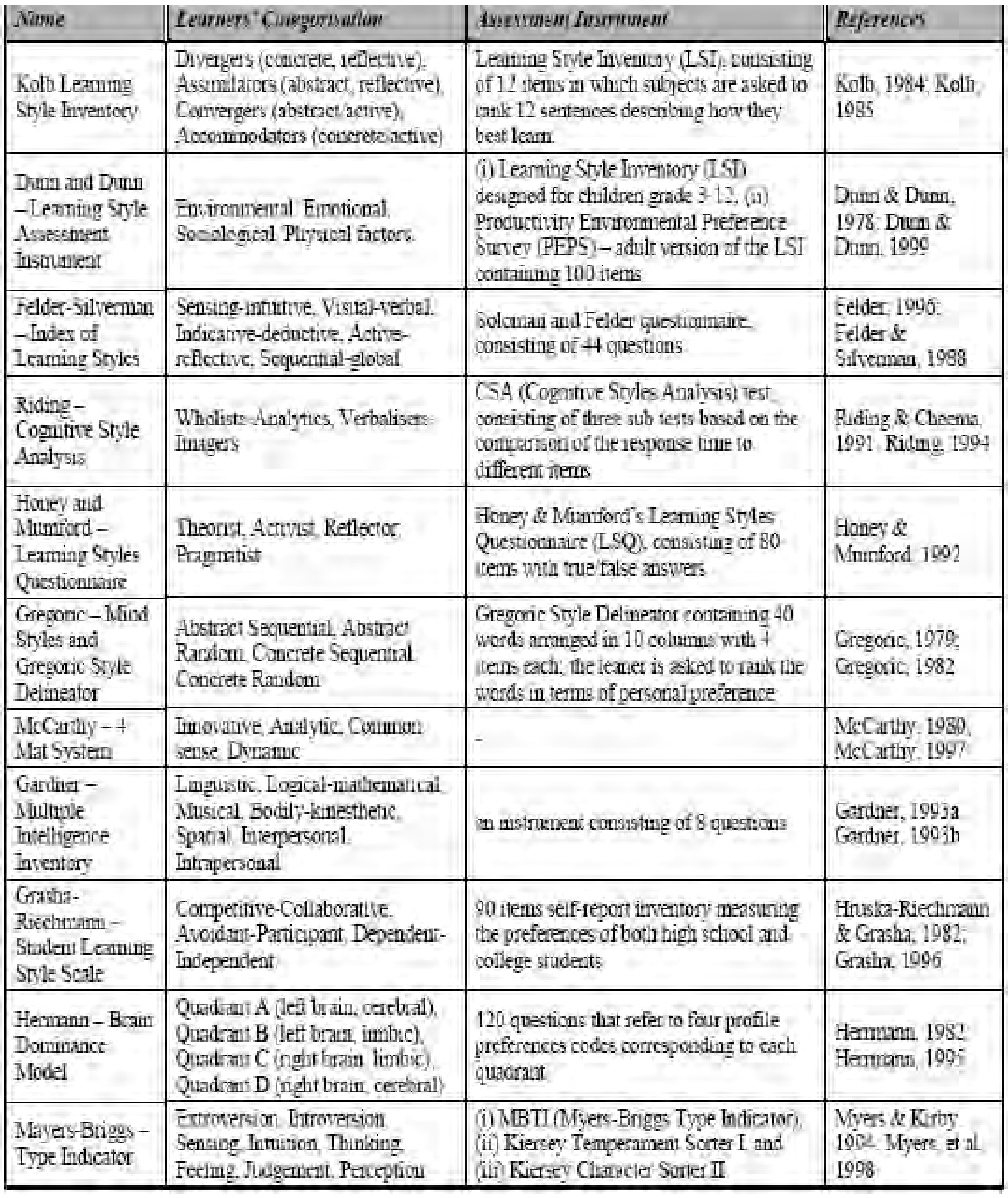
- «The way people understand process, store and recall information [45] ».
- « Discrete behaviors that are used like an index as to how a person learns from and adjusts to his environment, and it also provide indications regarding the way his brain works [46] ».
- « Views and behaviors that define the preferred learning style of a person [47] ».

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Each student is defined by a certain learning style. Most students don't know their exact learning style by they do know undefined ways of information representation that suit them and help them learn more or are more familiar with [45]. So, some students prefer and benefit in learning from information given in picture form, while others respond better to verbal information (narration or text). In a similar way, some students prefer data and facts, while others want theoretical knowledge.

Determining the students learning styles provides information about their special preferences, which can be used to design, develop and distribute the learning material, so that we encourage and stimulate the students at maximum level at acquiring knowledge on the subject's objective, in an effort of teaching personalizing.
Understanding of learning styles can also enhance the design, creation and using of teaching experienced, so that these can more easily be adjusted to the students' expectations, in an effort to enhance learning and the ability of acquiring and recalling their knowledge. Learning style diagnosing can be done according to existing bibliography with one of the models mentioned on the Table 3-2. As we can see from the Table 3-2, there are more than 70 diagnosis models are mentioned in [48]. But all these theories only consists in a (large) number of distinct theories, each advancing claims that, by general consensus, seem not to be sufficiently supported by appropriate evidence, with no mapping between different theories to facilitate comparative understanding and no generally agreed taxonomic approach that would collect similar theories into a few distinct clusters. So, it results in the need of a global model of learning styles, by elaborating a systematic classification of the learning styles dimensions proposed in various models, uncovering relationships of concept identity or subsumption and distinguishing between base and definable concepts**.** The writers in [49] worked towards this direction.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



Name	Learners' Categorisation	Assessment Instrument	References
Kolb Learning Style Inventory	Divergers (concrete, reflective), Assimilators (abstract, reflective), Convergers (abstract/active), Accommodators (concrete/active)	Learning Style Inventory (LSI), consisting of 12 items in which subjects are asked to rank 12 sentences describing how they best learn.	Kolb, 1984; Kolb, 1985
Dunn and Dunn –Learning Style Assessment Instrument	Environmental, Emotional, Sociological, Physical factors.	(i) Learning Style Inventory (LSI) designed for children grade 3-12, (ii) Productivity Environmental Preference Survey (PEPS) – adult version of the LSI containing 100 items	Dunn & Dunn, 1978; Dunn & Dunn, 1999
Felder-Silverman –Index of Learning Styles	Sensing-intuitive, Visual-verbal, Inductive-deductive, Active-reflective, Sequential-global	Soloman and Felder questionnaire, consisting of 44 questions	Felder, 1996; Felder & Silverman, 1988
Riding – Cognitive Style Analysis	Wholists-Analytics, Verbalisers-Imagers	CSA (Cognitive Styles Analysis) test consisting of three sub tests based on the comparison of the response time to different items	Riding & Cheema, 1991; Riding, 1994
Honey and Mumford – Learning Styles Questionnaire	Theorist, Activist, Reflector, Pragmatist	Honey & Mumford's Learning Styles Questionnaire (LSQ), consisting of 80 items with true/false answers	Honey & Mumford, 1992
Gregorc – Mind Styles and Gregorc Style Delineator	Abstract Sequential, Abstract Random, Concrete Sequential, Concrete Random	Gregorc Style Delineator containing 40 words arranged in 10 columns with 4 items each; the leaner is asked to rank the words in terms of personal preference	Gregorc, 1979; Gregorc, 1982
McCarthy – 4 Mat System	Innovative, Analytic, Common sense, Dynamic	-	McCarthy, 1980, McCarthy, 1997
Gardner – Multiple Intelligence Inventory	Linguistic, Logical-mathematical, Musical, Bodily-kinesthetic, Spatial, Interpersonal, Intrapersonal	an instrument consisting of 8 questions	Gardner, 1993a Gardner, 1993b
Grasha-Riechmann – Student Learning Style Scale	Competitive-Collaborative, Avoidant-Participant, Dependent-Independent	90 items self-report inventory measuring the preferences of both high school and college students	Hruska-Riechmann & Grasha, 1982; Grasha, 1996
Hermann – Brain Dominance Model	Quadrant A (left brain, cerebral), Quadrant B (left brain, limbic), Quadrant C (right brain, limbic), Quadrant D (right brain, cerebral)	120 questions that refer to four profile preferences codes corresponding to each quadrant	Hermann, 1982; Hermann, 1995
Mayers-Briggs – Type Indicator	Extroversion, Introversion, Sensing, Intuition, Thinking, Feeling, Judgement, Perception	(i) MBTI (Myers-Briggs Type Indicator), (ii) Kiersey Temperament Sorter I and (iii) Kiersey Character Sorter II	Myers & Kirby, 1994; Myers, et al, 1998

Table 3-2: Learning styles models [48]

A proposal for a unified learning styles model, with the suggestion that "instead of arguing over the best learning style, it is undoubtedly better to take the best of each model and use a complex of features, each with its own importance and influence" has

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

been advanced in [50]. As we understand, this approach is restricted purposefully "gathering characteristics from various learning styles proposed in the literature".

As mentioned in [51], in most theories, learning styles result as combinatorial constructs, based on a number of linear, bipolar base dimensions, such as concrete vs. abstract and active vs. reflective in Kolb's model. In Kolb's model, taking all four combinations results in four types:

- concrete/active (accommodating style),
- concrete/reflective (diverging style),
- abstract/active (converging style) and
- Abstract/reflective (assimilating style).

Note also that each of the opposing terms in a dimension can be regarded, in itself, as a Style Descriptor.

Researchers in the field often propose models that overlap with existing models, adopting a different terminology for their own proposal. The number of base dimensions proposed in a model constitutes the foundation for the extensional definition of styles. Learning styles theorists, however, will also typically propose an intentional definition, characterizing a style by a number of distinguishing traits. The main observation in this regard made in [51] is that even though the content of the intentional definition of style may often present itself as speculative and conjectural, it nevertheless provides the means to establish connections amongst style descriptors originating in distinct theories.

In [51] a global learning styles model is presented as the result of a systematic and rigorous classification of the concepts involved in determining the poles of learning types dimensions, originating in distinct theories. Figure 3-13 depicts the classification (coded in Protégé) of the style dimension designators. The authors of [51] emphasize that this mapping is not imposed on arbitrary grounds, but that it results by systematically examining implicitly stated connections that appear within the models considered, when styles are described and explicated intentionally.
Based on [51] we have elaborated a simple Learning Styles Ontology, as dictated by the needs of our research. The ontology is presented in Section 4.2.2 on Learner Information.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

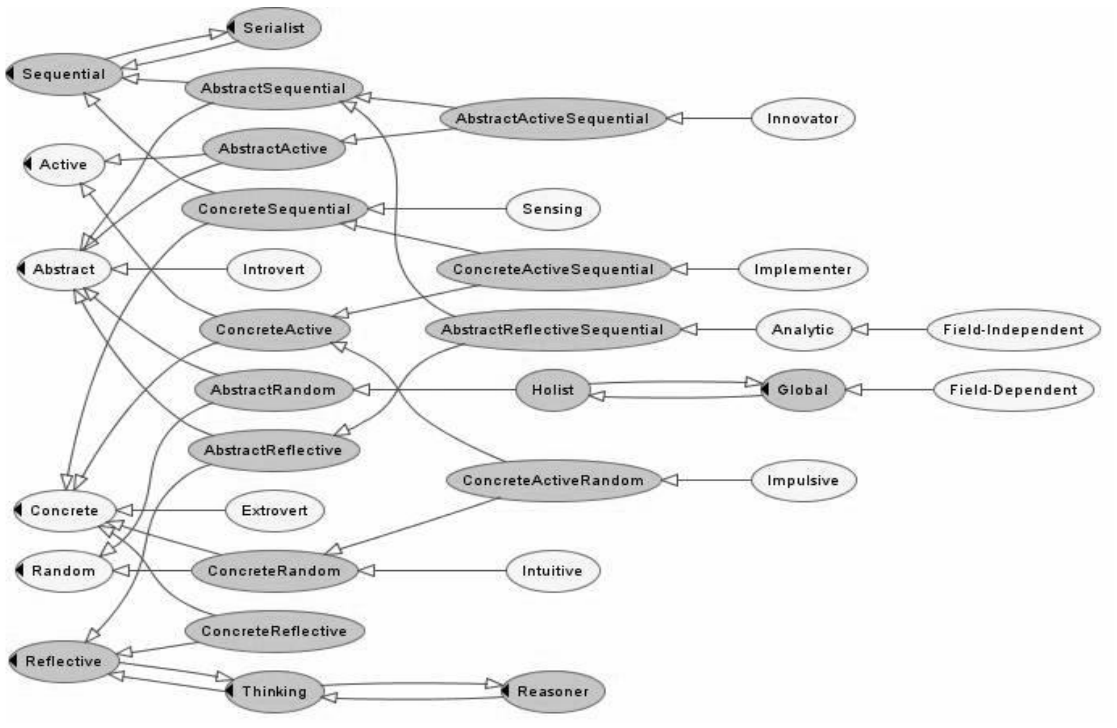


Figure 3-13: The Taxonomy of the Learning Style Dimension Designators [51]

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 4:

Learner & Learning Object Ontologies

Before presenting the Learner and Learning Object ontologies, we give the main principles that have been followed for its development. Ontologies developed in this section are based on existing research [41] but in this current thesis they are upgraded to a level above. According to what we already mentioned in previous chapters about Learning Objects, it is obvious that in order to be able to use them on the Semantic Web framework, it is necessary to develop Learning Object Metadata ontology.

The purpose of this ontology is to provide an owl binding for the IEEE Learning Object Metadata standards which will be used in order to reasoning about Learning Objects, in a semantic web setting. This ontology has been elaborated in the context of the CROP Reference Architecture. Similarly, an ontological specification of CROP Learning Objects (the KObject ontology) has been elaborated in the same context [1].

To meet our research expectations, we developed an OWL binding for Learning Objects (not necessarily CROP objects) and to the Learner respectively. As we already know, the Learner is an ontology that interacts with Learning Objects in order to teach some Object, therefore it is absolutely crucial to develop an ontology which collects some of its important features. We are going to divide this important information regarding the Learner in two ontologies:

- The IMS Learner Information Package ontology and
- The Learning Style ontology.

The IMS Learner Information Package (LIP) ontology is based on the IMS LIP Specification which records all the necessary information of the student or the author. The purpose of this ontology is to record information of the students or authors such as identification data, contact info, goals, qualifications, certifications, licences, competencies, interests, etc. The Learning Style ontology follows the idea of Currys Onion Model. In the last part of this section we develop the Learner Model ontology.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

The goal of this ontology is to create the necessary semantic structure in which to store information about the Learner in format that's appropriate for reasoning. The Learner Model ontology consists of the following ontologies:

The LearningObject ontology, that specifies the minimum requirements for an entity to be a Learning Object.

- The KObject ontology specifies Learning Objects according to the CROP Architecture.

The KConcept ontology, is introduced in CROP so that we can deal with second order relations. (Instances of the KConcept class correspond to classes (Concepts) in the Domain ontology of the Learning Object).
The Graph ontology, is an elementary base ontology for graphs, to be used as an import in subsequent work on ontological specifications of CROP Learning Objects and Learning Domains.
The ConceptGraph ontology, is an ontology whose instances are Concept Graphs which is to say a graph whose nodes are concepts and whose edges are instances of the *Prerequisite* relation.
The LearningBehavior ontology, attempts to capture features of behavioral traits of the learner, during teaching interaction with a Learning Object.

4.1 An Owl Binding for Learning Object Metadata

In this section we present the LOM ontology as developed in the context of the CROP Reference Architecture [41]. The LOM ontology follows precisely, the structure of the classes that are presented in the official specification of the IEEE LOM [33]. For this reason the restriction axioms that are introduced capture exactly the description of the elements given in IEEE LOM. The LOM ontology is expected to be of use in reasoning about Learning Objects, in a semantic web setting. So, the LOM Element classes for each of them are presented below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

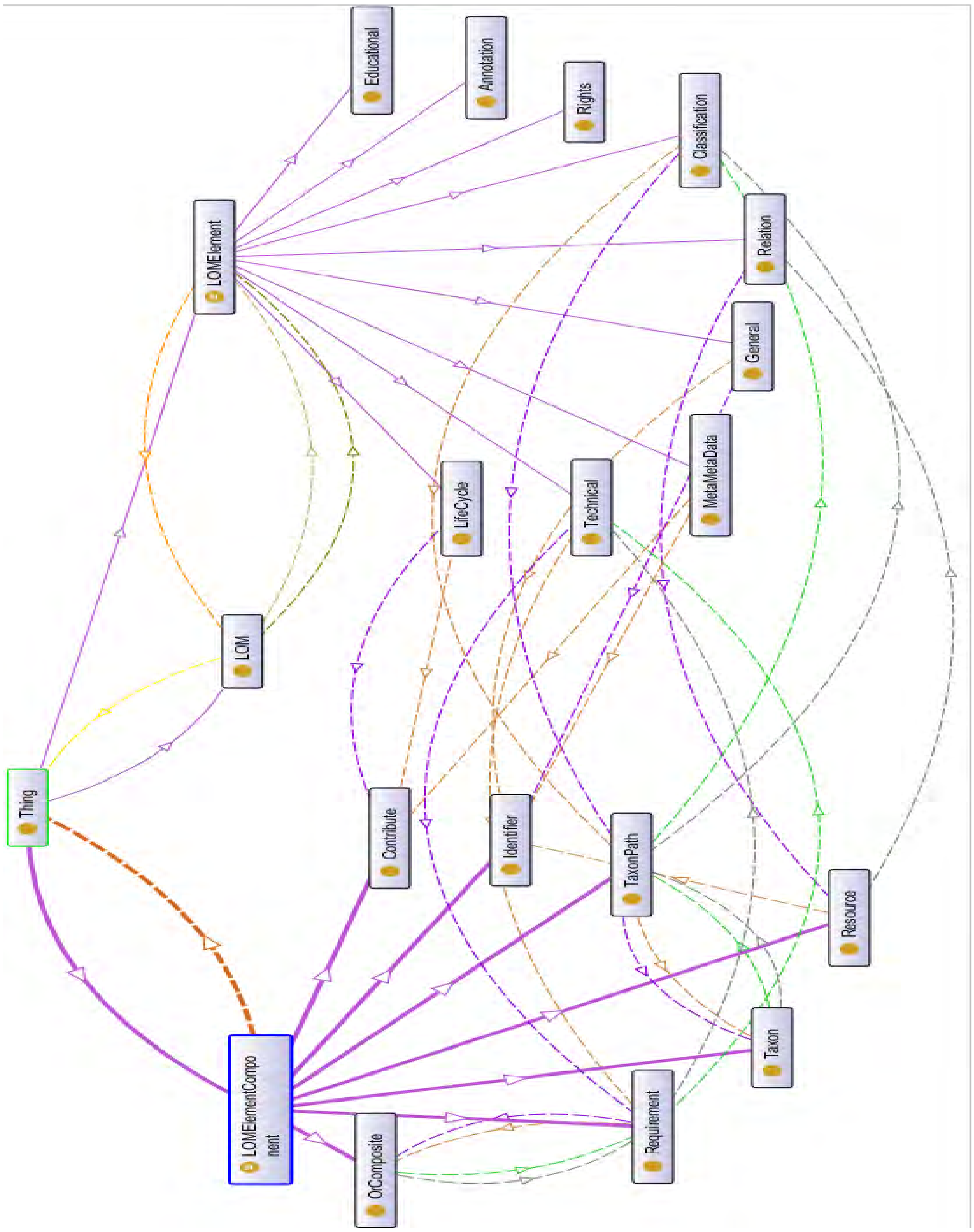


Figure 4-1: LOM ontology ontograph diagram

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

In Figure 4-1 is presented the ontograph diagram of LOM ontology and the relations among its elements. The hierarchy of classes that appears in the Figure 4-1 is depicted to Protégé with the following Figure 4-2.

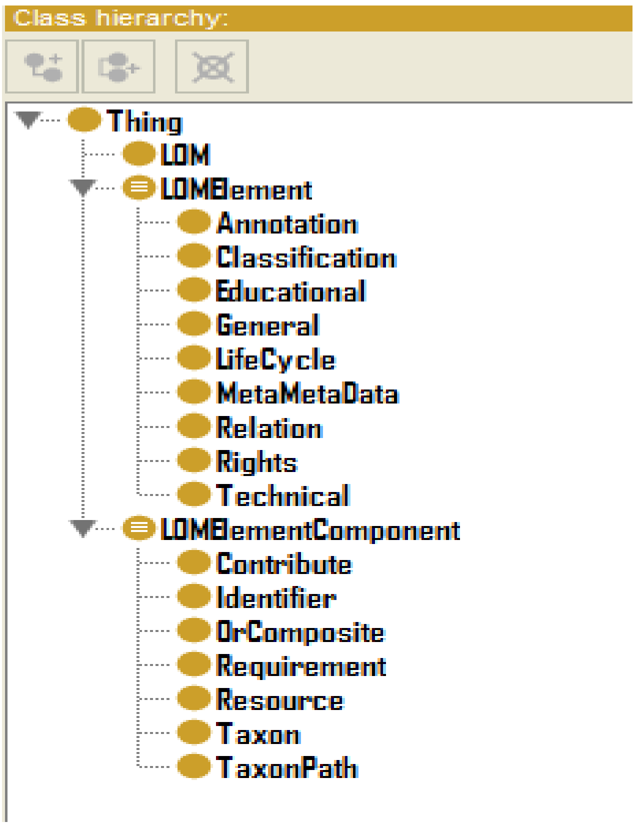


Figure 4-2: Class hierarchy of LOM ontology

The LOM ontology consists of three classes:

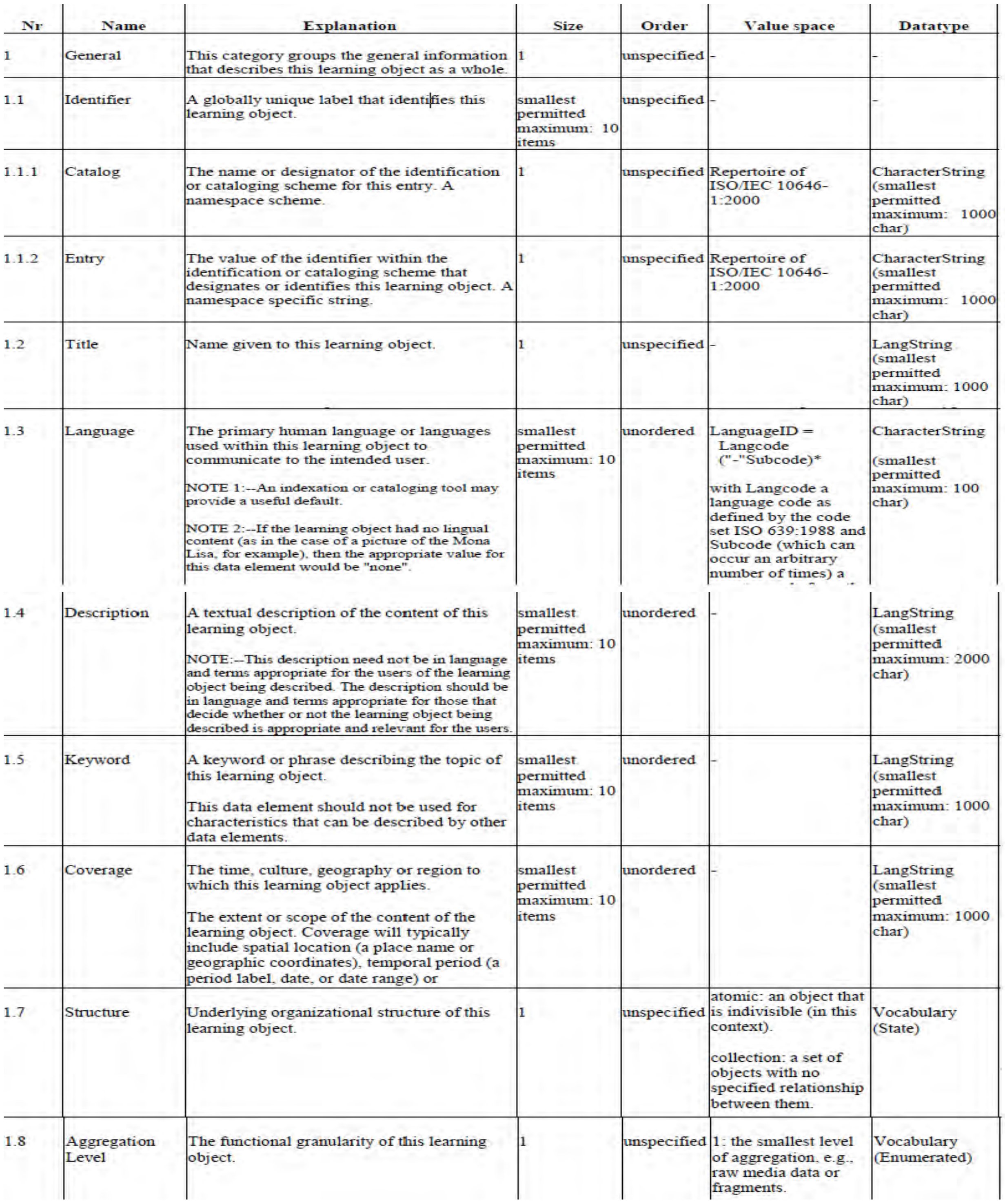
- *LOM class*, which has elements some LOMElement,
- *LOMElement class*, which is data elements that describe a Learning Object and are grouped under categories. There are 9 such categories: general, lifecycle, meta-metadata, technical, educational, rights, relation, annotation, and classification. These categories are absolutely identified with the official IEEE LOM specification.
- *LOMElementComponent class*, which contains some main data elements of the above categories.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Let’s see now with more details the content for each of these categories:

- **General:** This category groups the general information that describes a Learning Object as a whole.

The LOM base schema structure for the General class is depicted below.

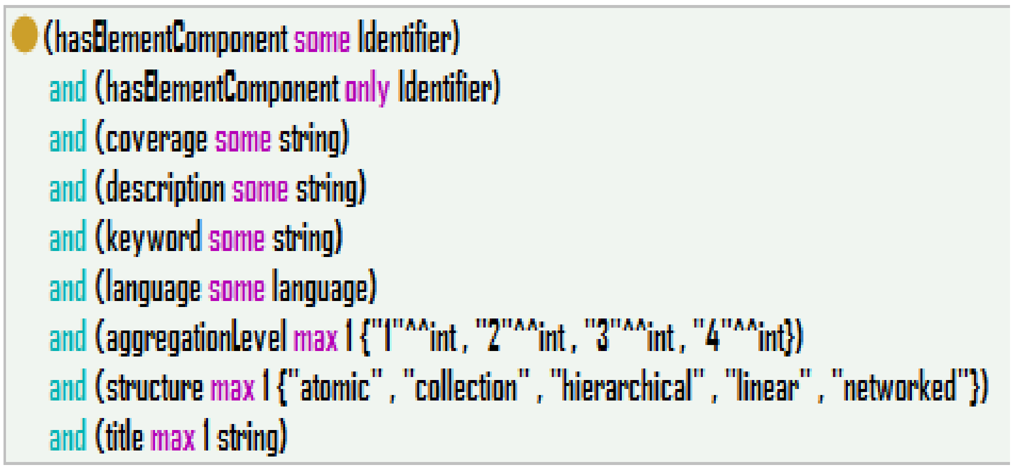


Nr	Name	Explanation	Size	Order	Value space	Datatype
1	General	This category groups the general information that describes this learning object as a whole.	1	unspecified	-	-
1.1	Identifier	A globally unique label that identifies this learning object.	smallest permitted maximum: 10 items	unspecified	-	-
1.1.1	Catalog	The name or designator of the identification or cataloging scheme for this entry. A namespace scheme.	1	unspecified	Repertoire of ISO/IEC 10646-1:2000	CharacterString (smallest permitted maximum: 1000 char)
1.1.2	Entry	The value of the identifier within the identification or cataloging scheme that designates or identifies this learning object. A namespace specific string.	1	unspecified	Repertoire of ISO/IEC 10646-1:2000	CharacterString (smallest permitted maximum: 1000 char)
1.2	Title	Name given to this learning object.	1	unspecified	-	LangString (smallest permitted maximum: 1000 char)
1.3	Language	The primary human language or languages used within this learning object to communicate to the intended user. NOTE 1:--An indexation or cataloging tool may provide a useful default. NOTE 2:--If the learning object had no lingual content (as in the case of a picture of the Mona Lisa, for example), then the appropriate value for this data element would be "none".	smallest permitted maximum: 10 items	unordered	LanguageID = Langcode ("-"Subcode)* with Langcode a language code as defined by the code set ISO 639:1988 and Subcode (which can occur an arbitrary number of times) a	CharacterString (smallest permitted maximum: 100 char)
1.4	Description	A textual description of the content of this learning object. NOTE:--This description need not be in language and terms appropriate for the users of the learning object being described. The description should be in language and terms appropriate for those that decide whether or not the learning object being described is appropriate and relevant for the users.	smallest permitted maximum: 10 items	unordered	-	LangString (smallest permitted maximum: 2000 char)
1.5	Keyword	A keyword or phrase describing the topic of this learning object. This data element should not be used for characteristics that can be described by other data elements.	smallest permitted maximum: 10 items	unordered	-	LangString (smallest permitted maximum: 1000 char)
1.6	Coverage	The time, culture, geography or region to which this learning object applies. The extent or scope of the content of the learning object. Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or	smallest permitted maximum: 10 items	unordered	-	LangString (smallest permitted maximum: 1000 char)
1.7	Structure	Underlying organizational structure of this learning object.	1	unspecified	atomic: an object that is indivisible (in this context). collection: a set of objects with no specified relationship between them.	Vocabulary (State)
1.8	Aggregation Level	The functional granularity of this learning object.	1	unspecified	1: the smallest level of aggregation, e.g., raw media data or fragments.	Vocabulary (Enumerated)

Table 4-1: General Category Base Schema Structure from [33]

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

The above information about General category is implemented in Protégé as follows:

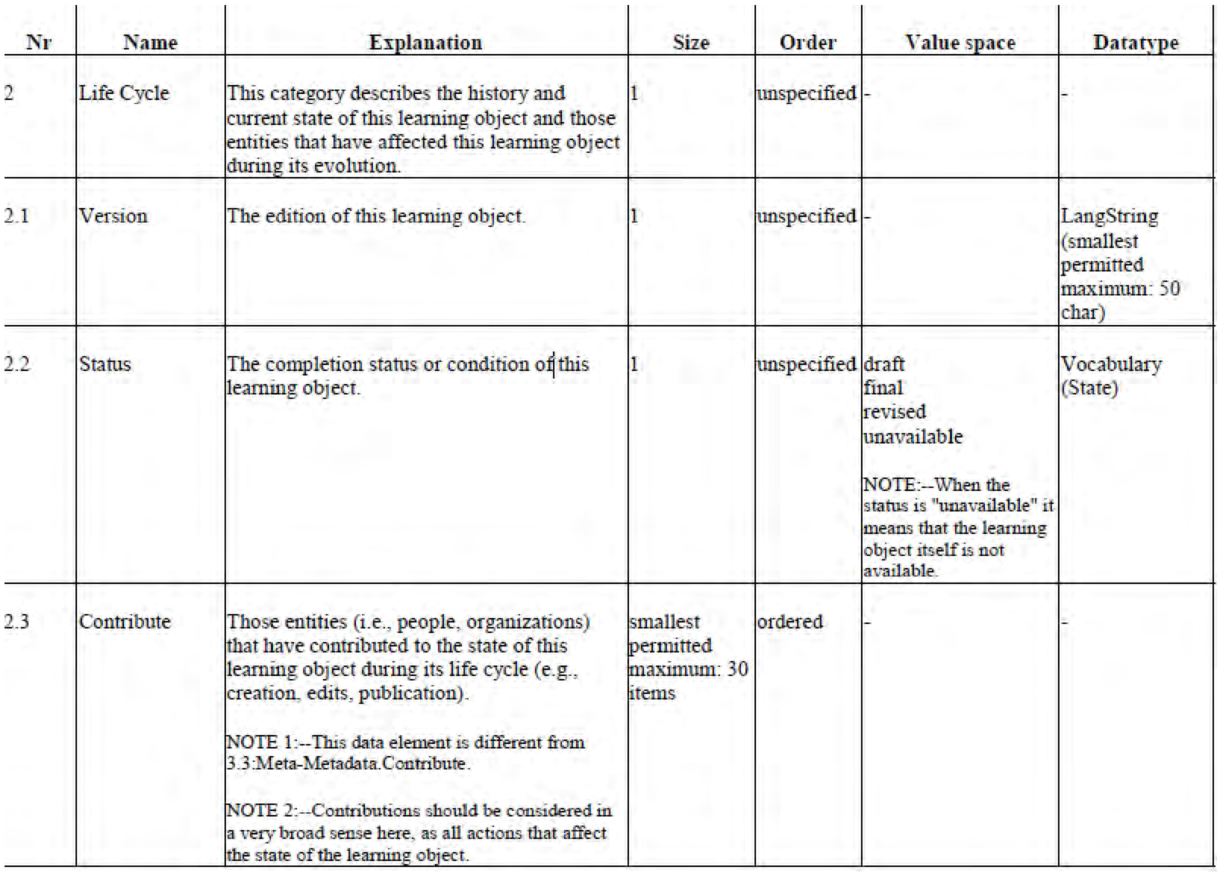


```
(hasElementComponent some Identifier)
  and (hasElementComponent only Identifier)
  and (coverage some string)
  and (description some string)
  and (keyword some string)
  and (language some language)
  and (aggregationLevel max 1 {"1"^^int , "2"^^int , "3"^^int , "4"^^int})
  and (structure max 1 {"atomic" , "collection" , "hierarchical" , "linear" , "networked"})
  and (title max 1 string)
```

Figure 4-3: General class structure

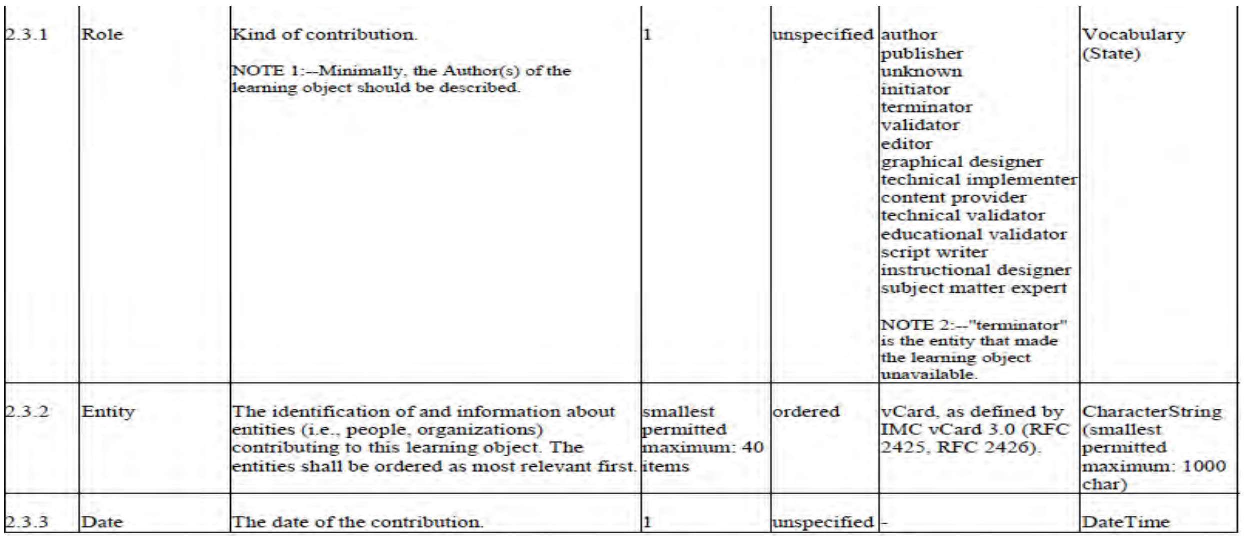
- **LifeCycle:** This category describes the history and current state of a Learning Object and those entities that have affected a Learning Object during its evolution.

The LOM base schema structure for the LifeCycle class is depicted below.



Nr	Name	Explanation	Size	Order	Value space	Datatype
2	Life Cycle	This category describes the history and current state of this learning object and those entities that have affected this learning object during its evolution.	1	unspecified	-	-
2.1	Version	The edition of this learning object.	1	unspecified	-	LangString (smallest permitted maximum: 50 char)
2.2	Status	The completion status or condition of this learning object.	1	unspecified	draft final revised unavailable NOTE:--When the status is "unavailable" it means that the learning object itself is not available.	Vocabulary (State)
2.3	Contribute	Those entities (i.e., people, organizations) that have contributed to the state of this learning object during its life cycle (e.g., creation, edits, publication). NOTE 1:--This data element is different from 3.3:Meta-Metadata.Contribute. NOTE 2:--Contributions should be considered in a very broad sense here, as all actions that affect the state of the learning object.	smallest permitted maximum: 30 items	ordered	-	-

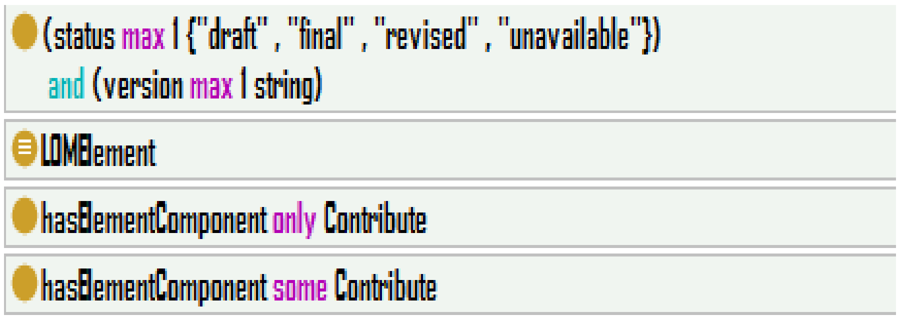
Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



2.3.1	Role	Kind of contribution. NOTE 1:—Minimally, the Author(s) of the learning object should be described.	1	unspecified	author publisher unknown initiator terminator validator editor graphical designer technical implementer content provider technical validator educational validator script writer instructional designer subject matter expert NOTE 2:—"terminator" is the entity that made the learning object unavailable.	Vocabulary (State)
2.3.2	Entity	The identification of and information about entities (i.e., people, organizations) contributing to this learning object. The entities shall be ordered as most relevant first.	smallest permitted maximum: 40 items	ordered	vCard, as defined by IMC vCard 3.0 (RFC 2425, RFC 2426).	CharacterString (smallest permitted maximum: 1000 char)
2.3.3	Date	The date of the contribution.	1	unspecified	-	DateTime

Table 4-2: LifeCycle Category Base Schema Structure from [33]

The corresponding implementation in LOM ontology is the following:



(status max 1 {"draft" , "final" , "revised" , "unavailable"})
and (version max 1 string)

LOMElement

hasElementComponent only Contribute

hasElementComponent some Contribute

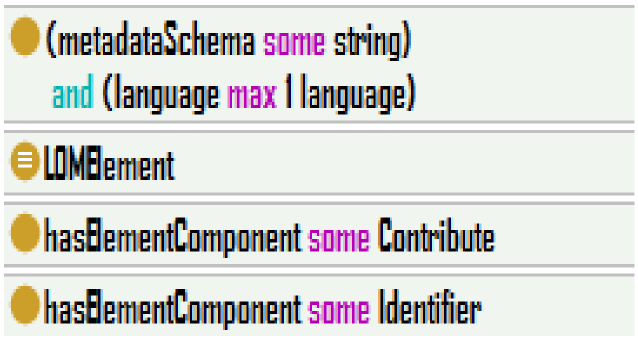
Figure 4-4: LifeCycle class structure

As we observe in current LOM ontology is implemented only some elements (status, version) of the official IEEE LOM specification and not all.

- **Meta-metadata:** This category describes this metadata record itself (rather than the Learning Object that this record describes).

According to the LOM base schema structure for the Meta-metadata category, the Meta-metadata class is depicted below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

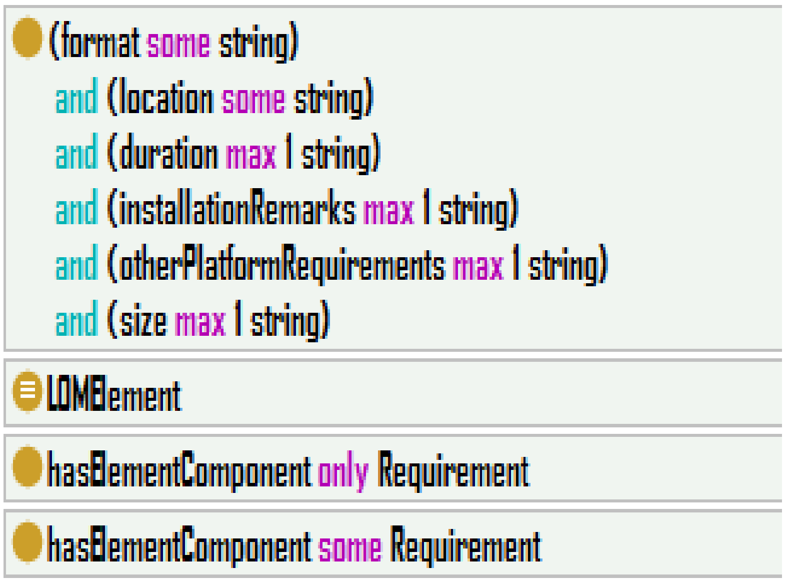


```
(metadataSchema some string)
 and (language max 1 language)
LOMElement
hasElementComponent some Contribute
hasElementComponent some Identifier
```

Figure 4-5: Meta-metadata class structure

- **Technical:** This category describes the technical requirements and characteristics of a Learning Object.

According to the LOM base schema structure for the Technical category, the Technical class is depicted below.



```
(format some string)
 and (location some string)
 and (duration max 1 string)
 and (installationRemarks max 1 string)
 and (otherPlatformRequirements max 1 string)
 and (size max 1 string)
LOMElement
hasElementComponent only Requirement
hasElementComponent some Requirement
```

Figure 4-6: Technical class structure

- **Educational:** This category describes the key educational or pedagogic characteristics of a Learning Object.

According to the LOM base schema structure for the Educational category, the Educational class is depicted below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

(**context** some {"**high school**" , "**higher education (doctoral)**" , "**higher education (graduate)**" , "**higher education (postdoc)**" , "**higher education (undergraduate)**" , "**other**" , "**primary school**" , "**secondary school**" , "**training**"})

and (**description** some **string**)

and (**intendedEndUserRole** some {"**author**" , "**learner**" , "**manager**" , "**teacher**"})

and (**language** some **language**)

and (**learningResourceType** some {"**collection**" , "**diagram**" , "**exam**" , "**exercise**" , "**experiment**" , "**figure**" , "**graph**" , "**index**" , "**lecture**" , "**narrative text**" , "**problem statement**" , "**questionnaire**" , "**self assessment**" , "**simulation**" , "**slide**" , "**table**"})

and (**typicalAgeRange** some **string**)

and (**difficulty** max **1** {"**difficult**" , "**easy**" , "**medium**" , "**very difficult**" , "**very easy**"})

and (**interactivityLevel** max **1** {"**high**" , "**low**" , "**medium**" , "**very high**" , "**very low**"})

and (**interactivityType** max **1** {"**active**" , "**expositive**" , "**mixed**"})

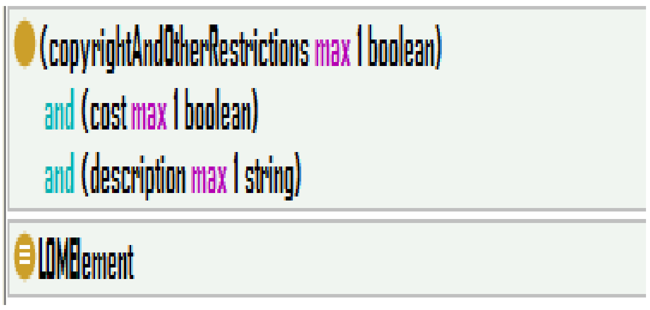
and (**semanticDensity** max **1** {"**high**" , "**low**" , "**medium**" , "**very high**" , "**very low**"})

and (**typicalLearningTime** max **1** **string**)

- **Rights:** This category describes the intellectual property rights and conditions of use for a Learning Object.

According to the LOM base schema structure for the Rights category, the Rights class is depicted below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



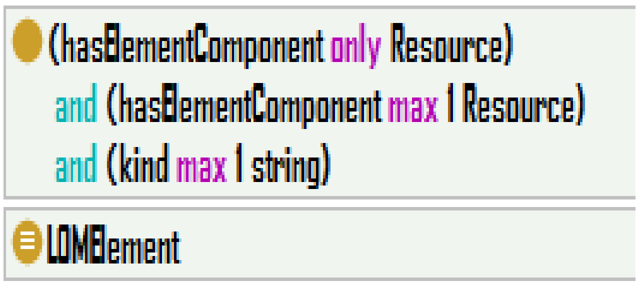
(copyrightAndOtherRestrictions max 1 boolean)
and (cost max 1 boolean)
and (description max 1 string)

LOMElement

Figure 4-7: Rights class structure

- **Relation:** This category defines the relationship between a Learning Object and other Learning Object, if any.

According to the LOM base schema structure for the Relation category, the Relation class is depicted below.



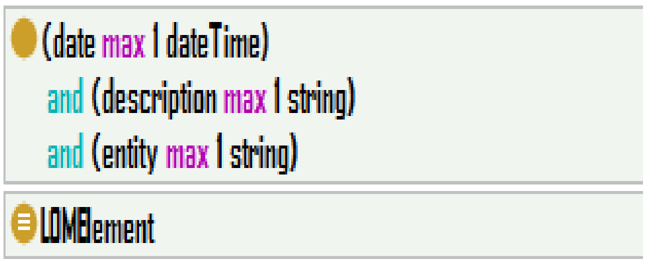
(hasElementComponent only Resource)
and (hasElementComponent max 1 Resource)
and (kind max 1 string)

LOMElement

Figure 4-8: Relation class structure

- **Annotation:** This category provides comments on the educational use of a Learning Object, and information on when and by whom the comments were created.

According to the LOM base schema structure for the Annotation category, the Annotation class is depicted below.



(date max 1 dateTime)
and (description max 1 string)
and (entity max 1 string)

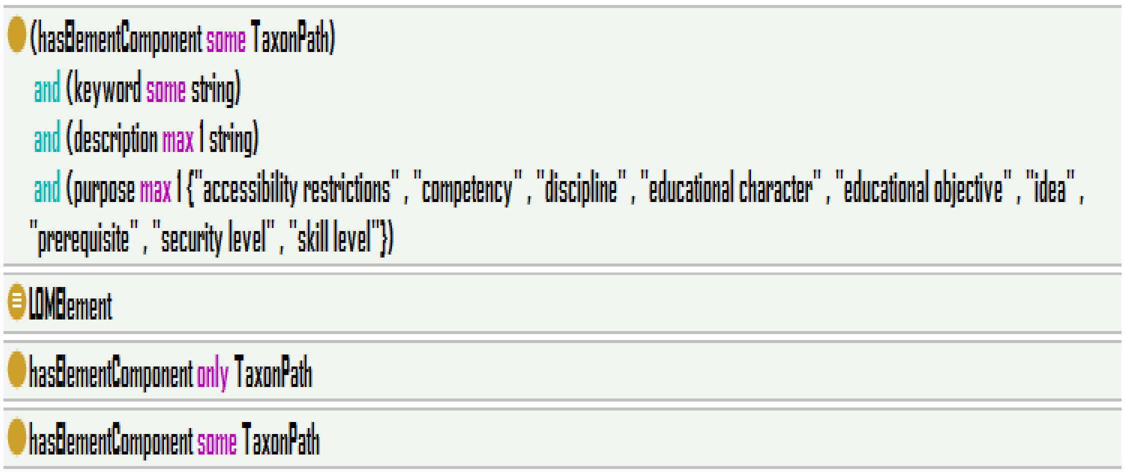
LOMElement

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Figure 4-9: Annotation class structure

- **Classification:** This category describes where a Learning Object falls within a particular classification system.

According to the LOM base schema structure for the Classification category, the Classification class is depicted below.



```
(hasElementComponent some TaxonPath)
 and (keyword some string)
 and (description max 1 string)
 and (purpose max 1 {"accessibility restrictions" , "competency" , "discipline" , "educational character" , "educational objective" , "idea" ,
"prerequisite" , "security level" , "skill level"})
LOMElement
hasElementComponent only TaxonPath
hasElementComponent some TaxonPath
```

Figure 4-10: Classification class structure

The vocabulary that is used in this ontology is the controlled vocabulary of IEEE LOM specification. A vocabulary is a recommended list of appropriate values. Other values, not present in the list, may be used as well. However, metadata that rely on the recommended values will have the highest degree of semantic interoperability, i.e., the likelihood that such metadata will be understood by other end users or systems are highest.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

4.2 An Owl Binding for Learner Information

4.2.1 An ontology for the IMS LIP Package

As we mentioned in a previous section 2.7, the IMS LIP Specification [22], is a collection of data for a student or a learning content creator where user data is defined by a data model which consists of 11 data categories. An XML binding is also provided by IMS.

In our approach we use the IMS LIP Specification for recording necessary information of the student or the author. Our model should begin by recording some explicit information given by the student regarding his personal details such as: name, address, demographic information and contact details as well as information related to his learning level and special preferences. At first, this information is deposited to the student model by the student himself through LIP. Afterwards, while the student interacts with the system, he becomes active in the learning process and some of his data changes; the student model gets updated in the appropriate (changed) fields of LIP. From an implementation perspective, this is done with an OWL binding, creating a LIP Ontology. In our case, we did not include all 11 categories contained in its official specifications in the LIP Ontology, but only some of these for proof of concept purposes.

In our LIP ontology we follow the same approach as the LOM ontology taking into account the IMS LIP specification [22]. So, the restriction axioms that are introduced in this ontology capture exactly the description of the elements given in IMS LIP. The LIP Element classes for each of them are presented below in the corresponding ontograph diagram.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

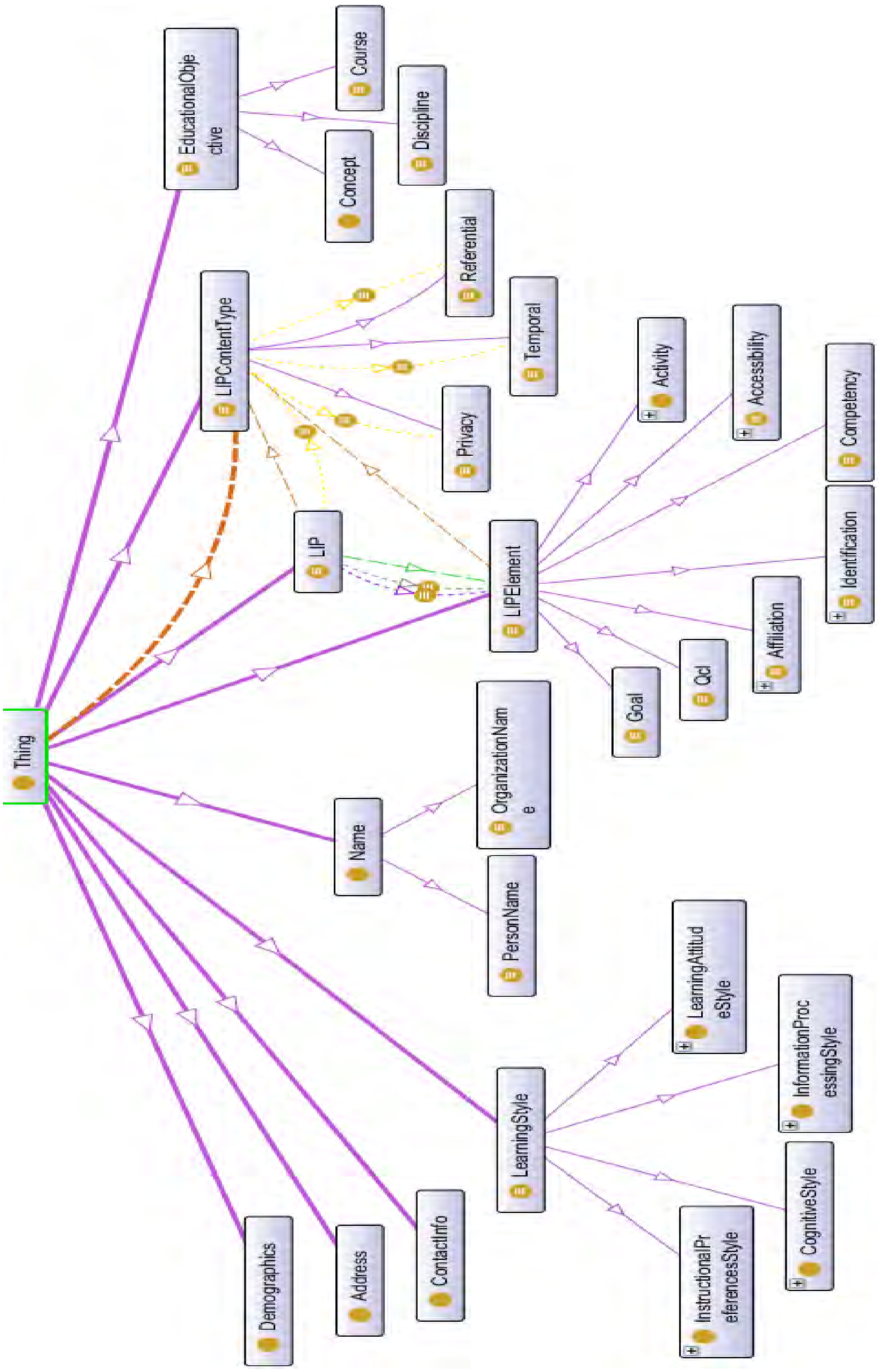
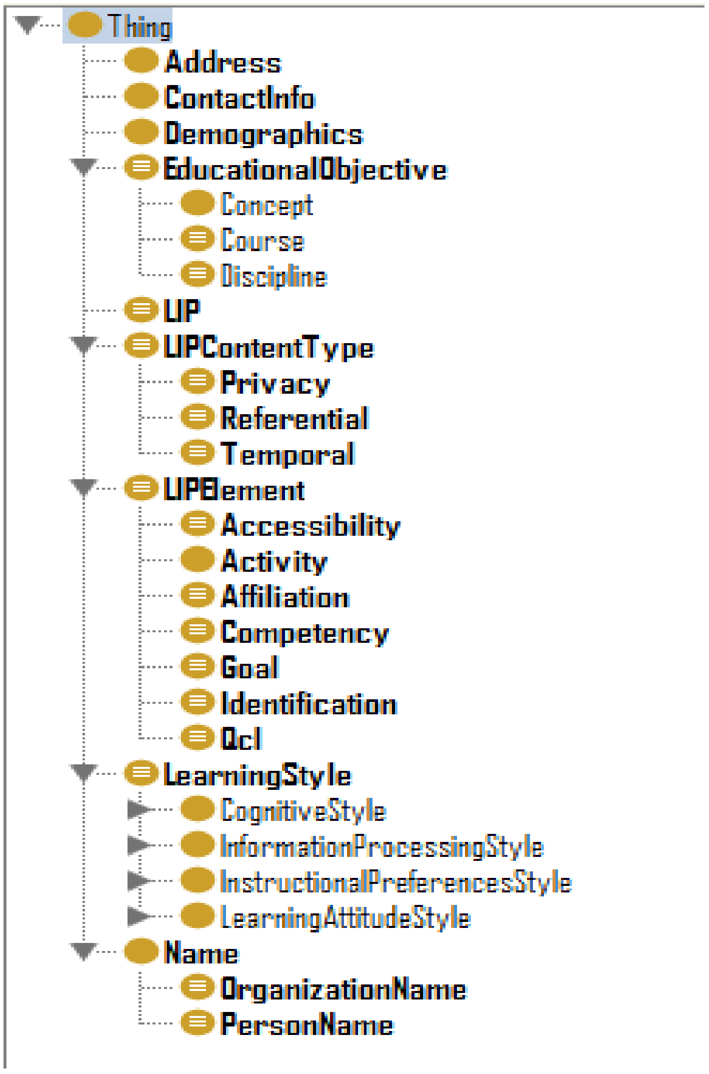


Figure 4-11: LIP ontology ontograph diagram

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

In the Figure 4-11 we present the ontograph diagram of LIP ontology and the relations among its elements. The hierarchy of classes that appears in the Figure 4-11 is depicted to Protégé with the following Figure 4-12.



- Thing
  - Address
  - ContactInfo
  - Demographics
  - EducationalObjective
    - Concept
    - Course
    - Discipline
  - LIP
  - LIPContentType
    - Privacy
    - Referential
    - Temporal
  - LIPElement
    - Accessibility
    - Activity
    - Affiliation
    - Competency
    - Goal
    - Identification
    - Qcl
  - LearningStyle
    - CognitiveStyle
    - InformationProcessingStyle
    - InstructionalPreferencesStyle
    - LearningAttitudeStyle
  - Name
    - OrganizationName
    - PersonName

Figure 4-12: Class hierarchy of LIP ontology

The LIP ontology consists of nine classes:

- **Address:** The detailed address of the individual or organization.

The LIP base schema structure for the Address class is depicted below.

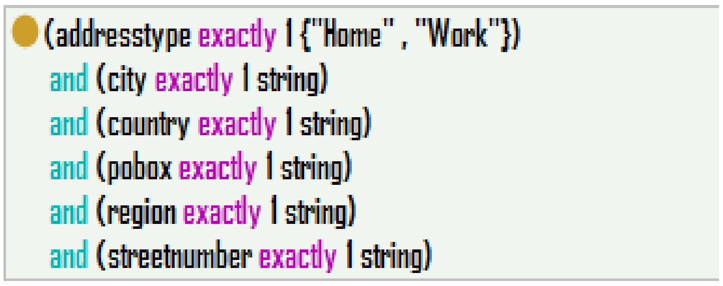
Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

address	The detailed address of the individual or organisation.	O	n		A separate entry is used per address.
typename	The type of address.	M		As per structure 13.4 (Table 6.13). The domain type will be defined from an appropriate vocabulary.	
comment	As per structure 13.2 (Table 6.13).				
contentype	As per structure 13.3 (Table 6.13).				
pobox	Post Office Box number.	O		String. 1-32 chars.	
langtype	The default language used for the PO Box.	As per structure 13.2 (Table 6.13).			
street	The street part of the address.	O			
nonfieldedstreetaddress	Unformatted street address.	O		String. 1-256 chars.	
langtype	The default language used for the non-fielded street address.	As per structure 13.2 (Table 6.13).			
streetnumber	The street number.	O		String. 1-8 chars.	
langtype	The default language used for the street number.	As per structure 13.2 (Table 6.13).			
streetprefix	Street prefix e.g. 'St.'.	O		String. 1-8 chars.	
langtype	The default language used for the street prefix.	As per structure 13.2 (Table 6.13).			
streetname	Street name.	O		String. 1-128 chars.	
langtype	The default language used for the street name.	As per structure 13.2 (Table 6.13).			
streetype	The type of street e.g. 'Road', 'Avenue' etc.	O		String. 1-32 chars.	
langtype	The default language used for the street type	As per structure 13.2 (Table 6.13).			
streetsuffix	Street suffix.	O		String. 1-8 chars.	
langtype	The default language used for the street suffix.	As per structure 13.2 (Table 6.13).			
apttype	Apartment type.	O		String. 1-32 chars.	
langtype	The default language used for the apartment type.	As per structure 13.2 (Table 6.13).			
aptnumprefix	Apartment number prefix.	O		String. 1-8 chars.	
langtype	The default language used for the apartment number prefix.	As per structure 13.2 (Table 6.13).			
aptnumber	The apartment number.	O		String. 1-8 chars.	
langtype	The default language used for the apartment number.	As per structure 13.2 (Table 6.13).			
aptnumsuffix	Apartment number suffix.	O		String. 1-2 chars.	
langtype	The default language used for the apartment suffix number.	As per structure 13.2 (Table 6.13).			

Table 4-3: Address Class Base Schema Structure

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

We implement the above information of the Address class in Protégé as follows:

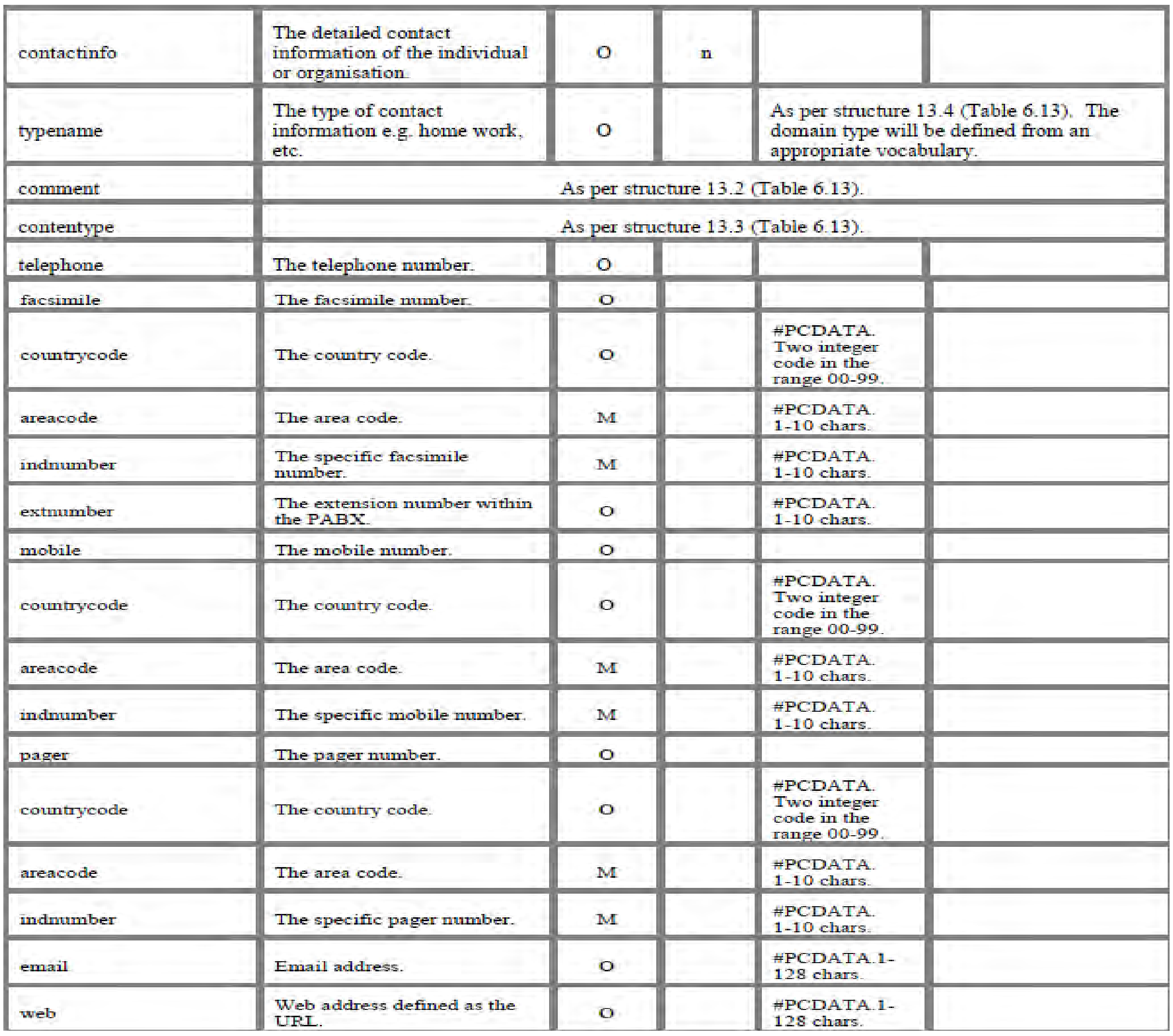


```
(addresstype exactly 1 {"Home" , "Work"})
 and (city exactly 1 string)
 and (country exactly 1 string)
 and (pobox exactly 1 string)
 and (region exactly 1 string)
 and (streetnumber exactly 1 string)
```

Figure 4-13: Address class restriction axioms

- **ContactInfo:** The detailed contact information of the individual or organization.

The LIP base schema structure for the ContactInfo class is depicted below.

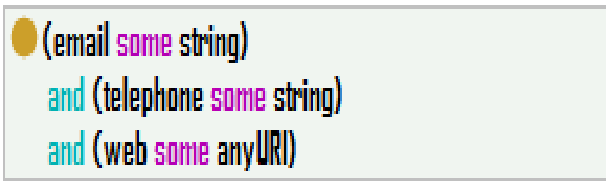


contactinfo	The detailed contact information of the individual or organisation.	O	n		
typename	The type of contact information e.g. home work, etc.	O		As per structure 13.4 (Table 6.13). The domain type will be defined from an appropriate vocabulary.	
comment	As per structure 13.2 (Table 6.13).				
contentype	As per structure 13.3 (Table 6.13).				
telephone	The telephone number.	O			
facsimile	The facsimile number.	O			
countrycode	The country code.	O		#PCDATA. Two integer code in the range 00-99.	
areacode	The area code.	M		#PCDATA. 1-10 chars.	
indnumber	The specific facsimile number.	M		#PCDATA. 1-10 chars.	
extnumber	The extension number within the PABX.	O		#PCDATA. 1-10 chars.	
mobile	The mobile number.	O			
countrycode	The country code.	O		#PCDATA. Two integer code in the range 00-99.	
areacode	The area code.	M		#PCDATA. 1-10 chars.	
indnumber	The specific mobile number.	M		#PCDATA. 1-10 chars.	
pager	The pager number.	O			
countrycode	The country code.	O		#PCDATA. Two integer code in the range 00-99.	
areacode	The area code.	M		#PCDATA. 1-10 chars.	
indnumber	The specific pager number.	M		#PCDATA. 1-10 chars.	
email	Email address.	O		#PCDATA.1-128 chars.	
web	Web address defined as the URL.	O		#PCDATA.1-128 chars.	

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Table 4-4: ContactInfo Class Base Schema Structure

The ContactInfo class can be implemented in Protégé as in the following partial example:

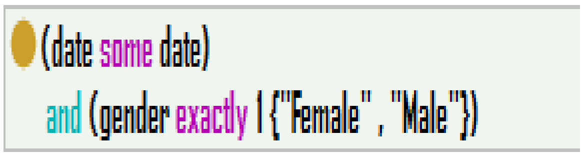


```
(email some string)
 and (telephone some string)
 and (web some anyURI)
```

Figure 4-14: ContactInfo class restriction axioms

- **Demographics:** The mechanisms by which the individual can be recognized for learning.

According to the LIP base schema structure for Demographics element, our Demographics class is depicted below.



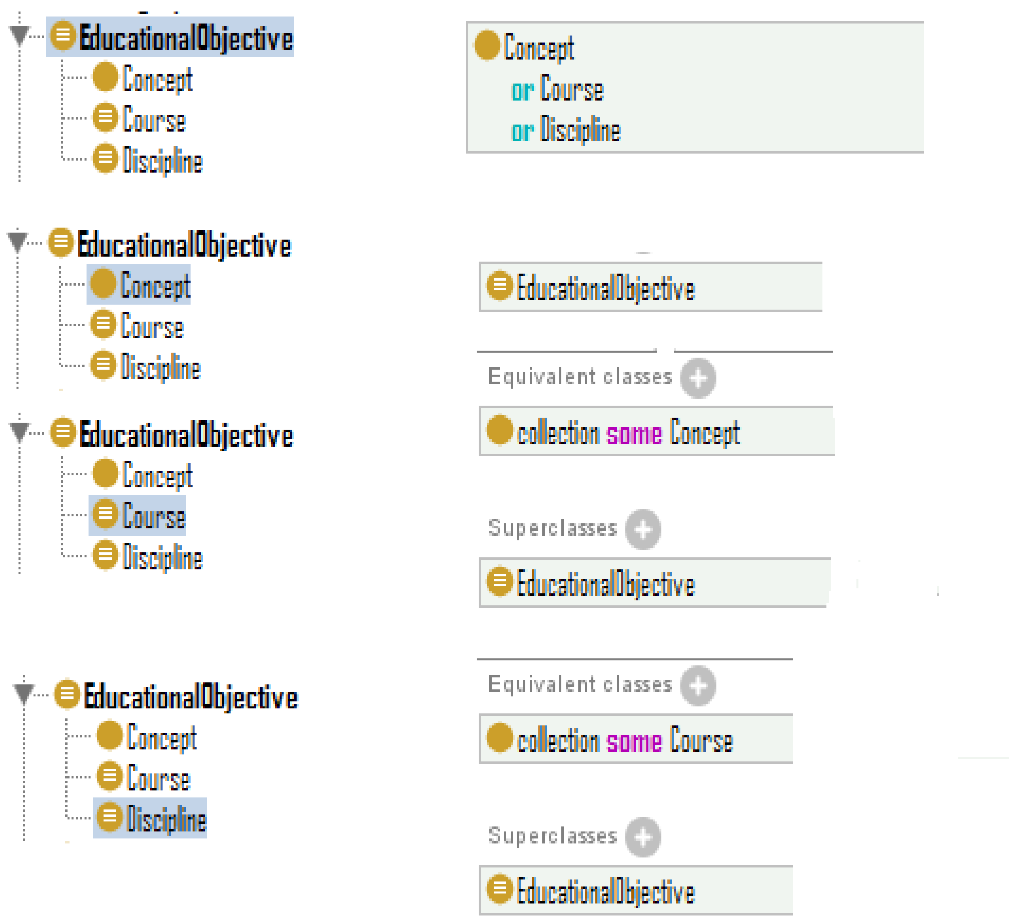
```
(date some date)
 and (gender exactly 1 {"Female" , "Male"})
```

Figure 4-15: Demographics class restriction axioms

- **Educational Objective:** The educational objective of the user. This objective can be a concept, course or discipline.
  - *Concept*: Knowledge Concept, a unit teaching/educational objective.
  - *Course*: A set of concepts, considered to be related in their content. For example, a course "Theory of Computation" might consist of the concepts 'automata', 'formal languages', 'Turing machines', 'recursive functions', 'insolvability', 'time complexity', 'space complexity'.
  - *Discipline*: A set of courses, typically leading to some kind of formal certificate, or degree. For example, "Physics", or "Zoology".

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

In our LIP ontology this is implemented as follows:



EducationalObjective
- Concept
- Course
- Discipline

Concept
or Course
or Discipline

EducationalObjective

Equivalent classes
collection some Concept

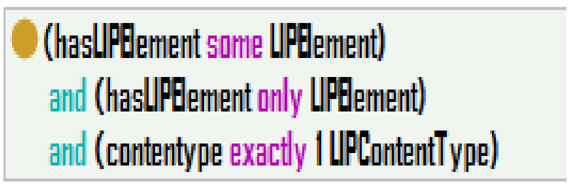
Superclasses
EducationalObjective

Equivalent classes
collection some Course

Superclasses
EducationalObjective

Figure 4-16: Educational Objective class restriction axioms

- **LIP Class:** Contains LIPElement and LIPContenType.



(hasLIPElement some LIPElement)
and (hasLIPElement only LIPElement)
and (contentype exactly 1 LIPContentType)

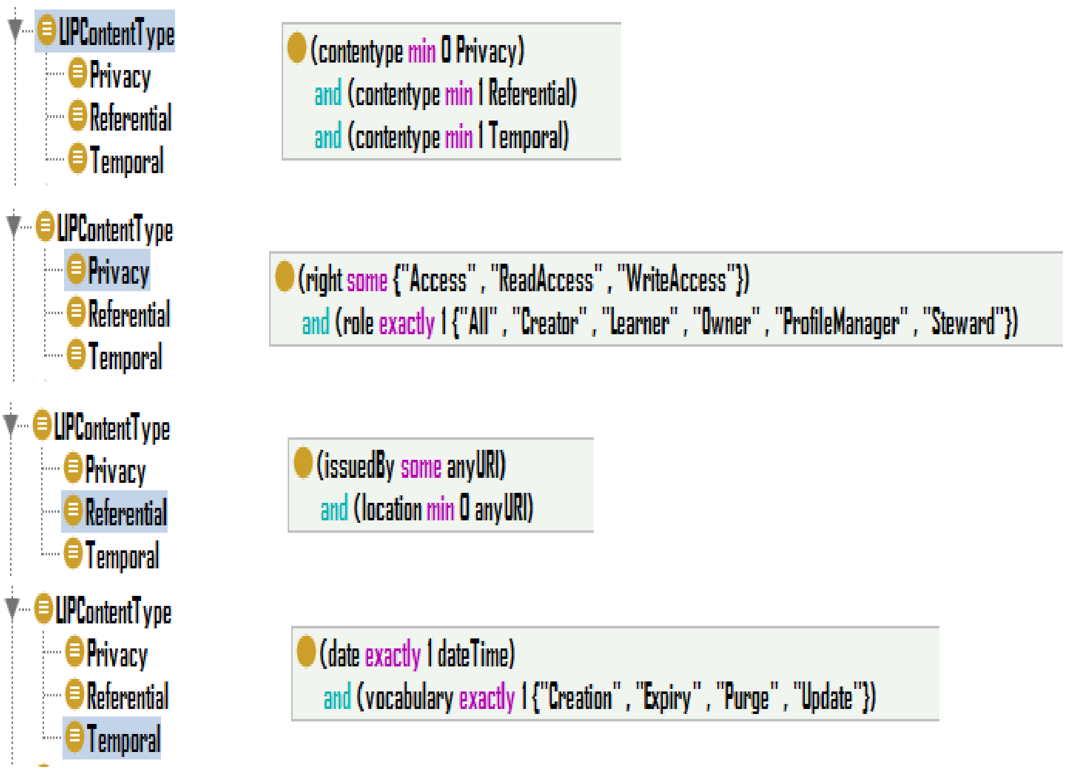
Figure 4-17: LIP class restriction axioms

- **LIPContentType Class:** The content information, contentype, consists of:

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- Privacy: Data that is to be used to describe the access to and to ensure the integrity of the learner information.
- Temporal: Data describing time-based information about the data structure e.g. time of creation, date of expiry, etc. If the expire date is undefined then the information is assumed to have an infinite period of validity. Several different temporal definitions may be defined for a structure e.g. time of creation, and expiry.
- Referential: Reference information that is used to uniquely identify the Learner Information Package and the data structures within it. It consists of an issuing source and (optionally) a location (URI) the LIP file or LIPElement in question.

In our LIP ontology this class is implemented as follows:



```
(contentype min 0 Privacy)
  and (contentype min 1 Referential)
  and (contentype min 1 Temporal)
```

```
(right some {"Access" , "ReadAccess" , "WriteAccess"})
  and (role exactly 1 {"All" , "Creator" , "Learner" , "Owner" , "ProfileManager" , "Steward"})
```

```
(issuedBy some anyURI)
  and (location min 0 anyURI)
```

```
(date exactly 1 dateTime)
  and (vocabulary exactly 1 {"Creation" , "Expiry" , "Purge" , "Update"})
```

Figure 4-18: LIPContentType class restriction axioms

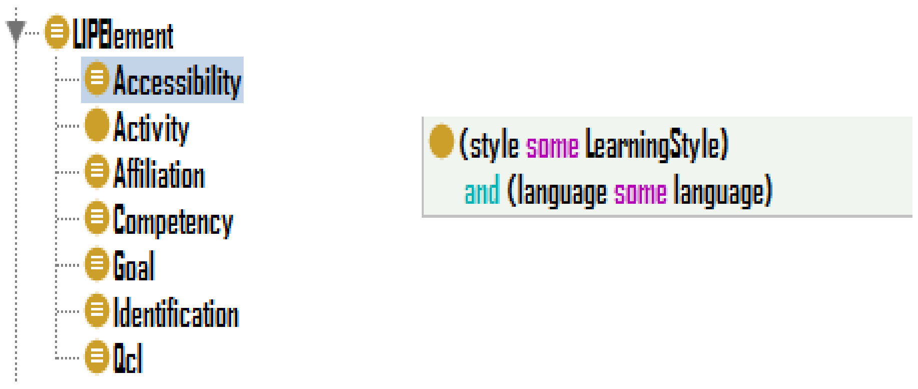
Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- **LIPElement Class:** Contains data elements that describe a learner and are grouped under categories. There are 11 categories in the official specification of IMS LIP but for our needs we can use only the seven from them. These categories are: Accessibility, Activity, Affiliation, Goal, Competency, Qcl and Identification.

Let's see with more details the content for each of them:

- **Accessibility:** General accessibility to the learner information as defined through language capabilities, disabilities, eligibilities and learning preferences including cognitive preferences (e.g. issues of learning style), physical preferences (e.g. a preference for large print), and technological preferences (e.g. a preference for a particular computer platform).

According to the LIP base schema structure for the Accessibility element, our Accessibility class is depicted below.



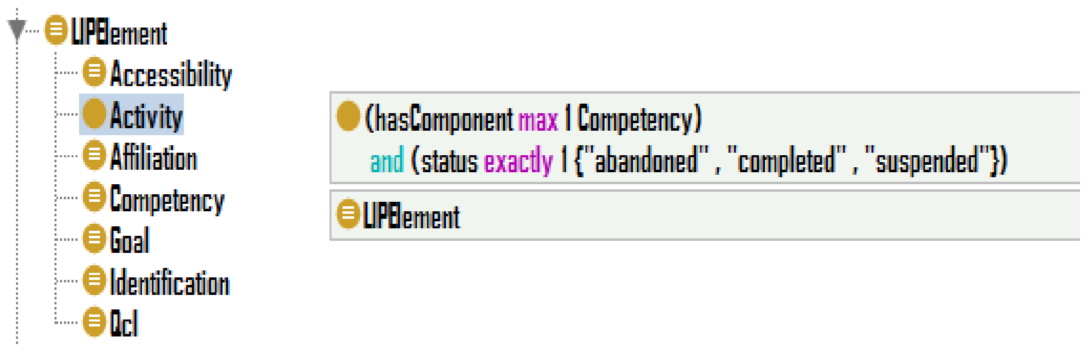
(style some LearningStyle)
and (language some language)

Figure 4-19: Accessibility LIPElement restriction axioms

- **Activity:** Any learning-related activity in any state of completion. Could be self-reported. Includes formal and informal education, training, work experience, and military or civic service.

According to the LIP base schema structure for the Activity element, our Activity class is depicted below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



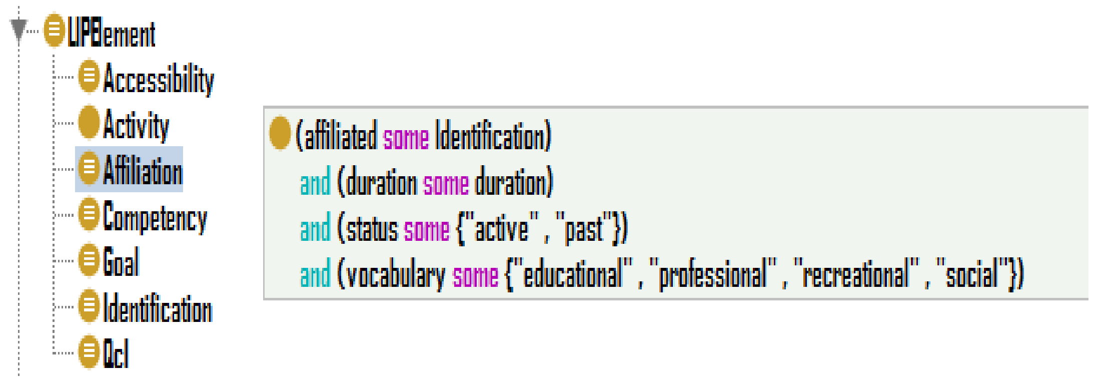
(hasComponent max 1 Competency)
and (status exactly 1 {"abandoned" , "completed" , "suspended"})

LIPElement

Figure 4-20: Activity LIPElement restriction axioms

- **Affiliation:** The affiliation learner information is used to store the descriptions of the organization affiliations associated with the learner. These affiliations may include education groups e.g. classes, cohorts, etc. but it is expected that these will be exchanged using the IMS Enterprise specification technique.

According to the LIP base schema structure for the Affiliation element, our Affiliation class is depicted below.



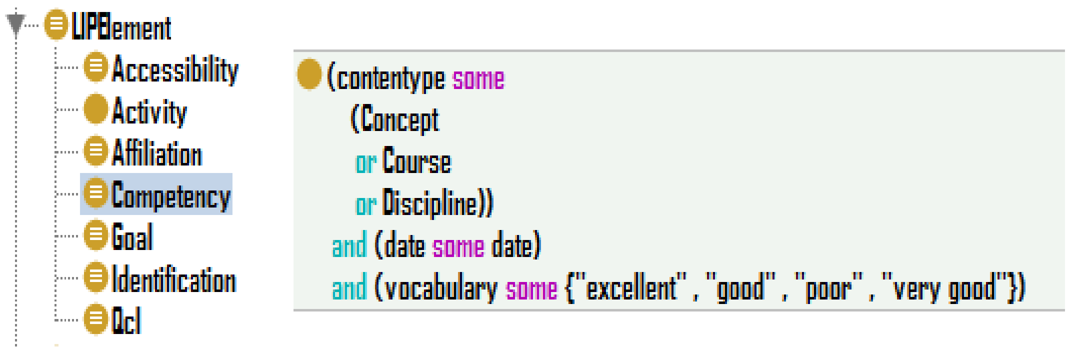
(affiliated some Identification)
and (duration some duration)
and (status some {"active" , "past"})
and (vocabulary some {"educational" , "professional" , "recreational" , "social"})

Figure 4-21: Affiliation LIPElement restriction axioms

- **Competency:** Skills, knowledge, and abilities acquired in the cognitive, affective, and/or psychomotor domains.

According to the LIP base schema structure for the Competency element, our Competency class is depicted below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

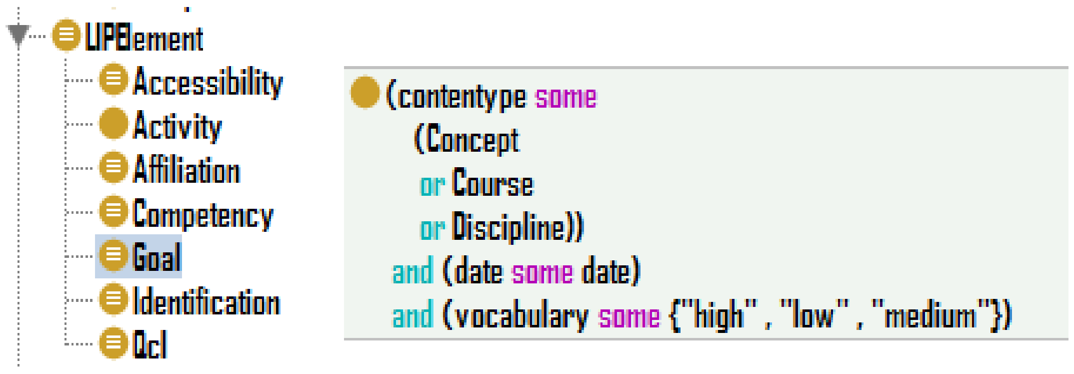


```
(contentype some
    (Concept
     or Course
     or Discipline))
  and (date some date)
  and (vocabulary some {"excellent" , "good" , "poor" , "very good"})
```

Figure 4-22: Competency LIPElement restriction axioms

- **Goal:** Learning, career and other objectives and aspirations.

According to the LIP base schema structure for the Goal element, our Goal class is depicted below.



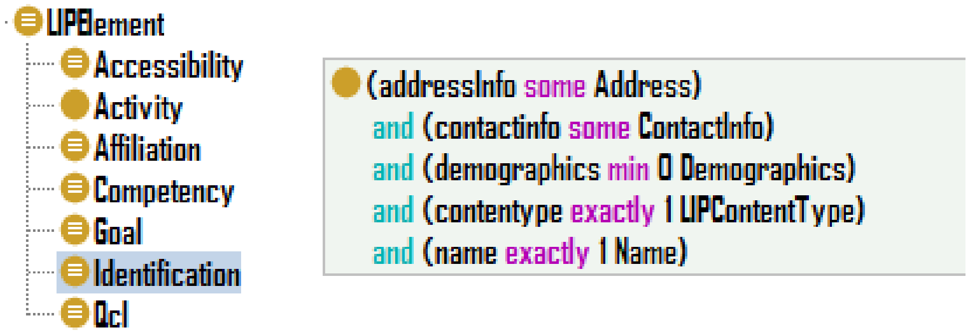
```
(contentype some
    (Concept
     or Course
     or Discipline))
  and (date some date)
  and (vocabulary some {"high" , "low" , "medium"})
```

Figure 4-23: Goal LIPElement restriction axioms

- **Identification:** The identification learner information contains all of the data for a specific individual or organization. This includes data such as: name, address, contact information, agent and demographics.

According to the LIP base schema structure for the Identification element, our Identification class is depicted below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

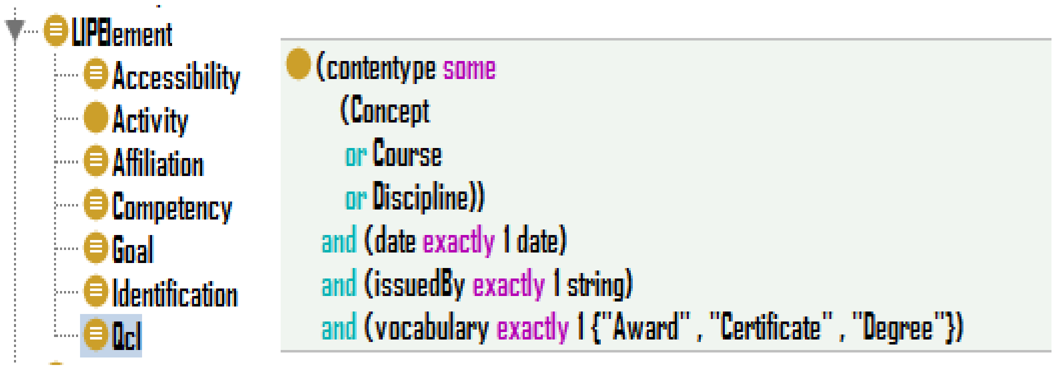


```
(addressInfo some Address)
 and (contactinfo some ContactInfo)
 and (demographics min 0 Demographics)
 and (contentype exactly 1 LIPContentType)
 and (name exactly 1 Name)
```

Figure 4-24: Identification LIPElement restriction axioms

- **Qcl:** The qcl learner information consists of the qualifications, certifications and licenses awarded to the learner i.e. the formally recognized products of their learning and work history. This includes information on the awarding body and may also include electronic copies of the actual documents. A different 'qcl' structure will be used for each qualification, etc.

According to the LIP base schema structure for the Qcl element, our Qcl class is depicted below.

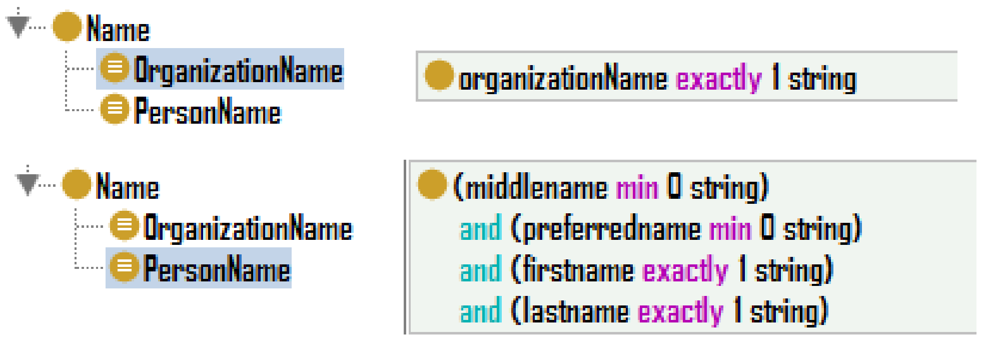


```
(contentype some
   (Concept
    or Course
    or Discipline))
 and (date exactly 1 date)
 and (issuedBy exactly 1 string)
 and (vocabulary exactly 1 {"Award" , "Certificate" , "Degree"})
```

Figure 4-25: Qcl LIPElement restriction axioms

- **Name Class:** The name of the individual or organization.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



organizationName exactly 1 string

(middlename min 0 string)
and (preferredname min 0 string)
and (firstname exactly 1 string)
and (lastname exactly 1 string)

Figure 4-26: Name restriction axioms for Name class

4.2.2 An Ontology for Learning Style

The IMS LIP Accessibility element mentions the need to capture information about Learning Style, without proposed any particular structure to hold such information. What we do here is, to construct a Learning Style ontology following [51] which we present in the sequel. The Learning Style ontology is imported of a sub-ontology of the LIP ontology and its classes together with the ontograph diagram are presented in the Figures 4-27 and 4-28.

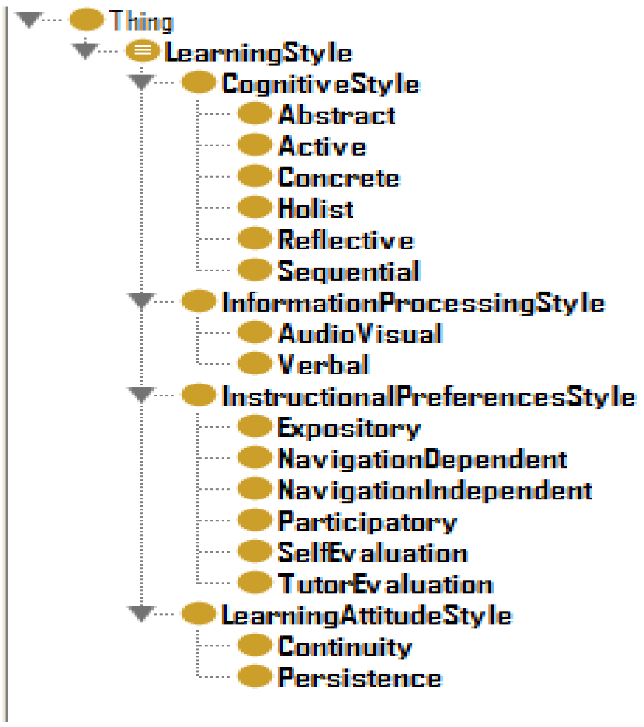


Figure 4-27: Class hierarchy of LearningStyle ontology

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

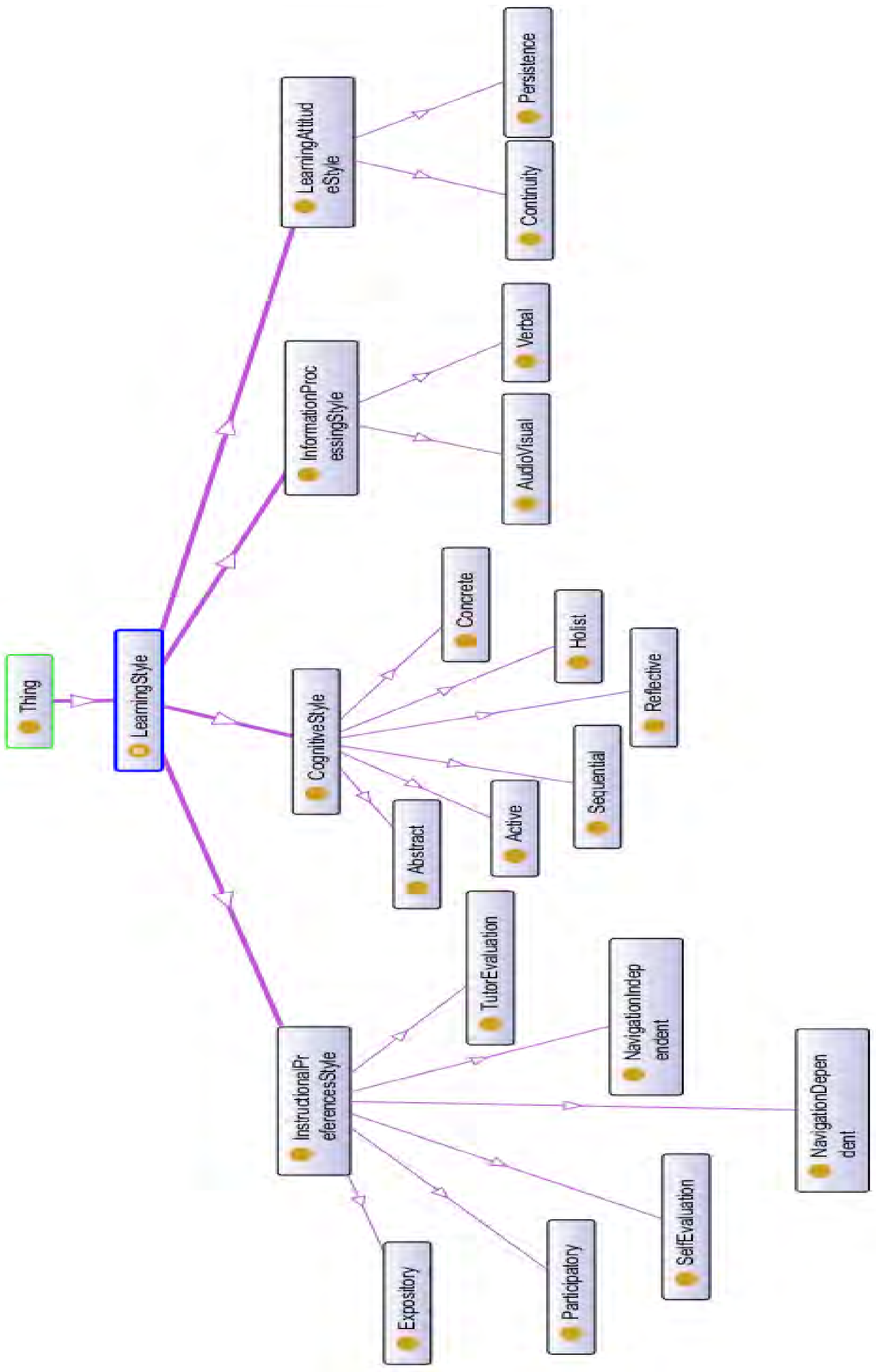


Figure 4-28: LearningStyle ontology ontograph diagram

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

In the Learning Style ontology we follow the idea of Curry Onion Model [52]. Curry proposed that nine of the major learning style measures can be organized into four layers resembling those of an onion. In the Curry Model each layer focuses on different aspects of the learner and how they learn. Curry's onion model was developed with four layers: at the centre, we will find the basic personality traits, and the outer layers include information-processing, social interaction, and, finally, instructional preference.

- Personality learning theories define the influences of basic personality on preferences to acquiring and integrating information. Models used in this theory include Myers-Briggs Type Indicator, which measures personality in dichotomous terms and the Keirsey Temperament Sorter, which classifies people as rationales, idealists, artisans, or guardians.
- Information processing theories encompass individuals' preferred intellectual approach to assimilating information, and includes David Kolb's model of information processing, which identifies two separate learning activities: perception and processing.
- Social learning theories determine how students interact in the classroom and include Reichmann's and Grasha's types of learners: independent, dependent, collaborative, competitive, participant, and avoidant.
- Multidimensional and instructional theories address the student’s environmental preference for learning and include the Learning Style Model of Dunn and Dunn and the multiple intelligences theory of Howard Gardner.

According to the above Curry Onion Model theory, our Learning Style ontology consists of four classes (CognitiveStyle class, InformationProcessingStyle, InstructionalPreferencesStyle, LearningAttitudeStyle):

- **CognitiveStyle class:** As we describe in the Learning Style section 3.5 we adopt the concept of designators. At this layer learner type designators are more resistant to change. Also, this class contains the following subclasses:

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

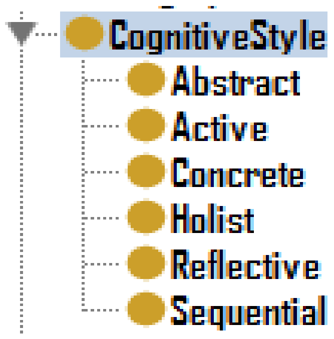


Figure 4-29: CognitiveStyle class and its subclasses

- <u>Abstract:</u> The learner tending and able to conceive general form regardless of particular content, internal structure rather than specific ways of representation. (Opposite: Concrete). The abstract-concrete dimension is present in both Gregorc's and in Kolb’s theories of learning types.

- <u>Active:</u> The learner learning by testing/ experimenting, tending to do things, ‘make it happen', rather than pause and contemplate. Directly engages the problem rather than reflect on it. (Opposite: Reasoner/Reflective/Thinking). Present in both Kolb's and Honey and Mumford's theories. In this ontology we identify 'Active' and Jackson's 'Initiator', despite some difference in the nuances of the two notions.

- <u>Concrete:</u> The learner focusing on specific content rather than form, on specific way of representation of this content. (Opposite: Abstract). The abstract-concrete dimension is present in both Gregorc's and in Kolb's theories of learning types.

- <u>Holist:</u> Holist, or Global. The learner tending to prefer getting an overview of the subject before attempting to tackle details.

- <u>Reflective:</u> Reflective, or Thinking, or Theorist. The learner tending to think through the meaning of the subject, understand its structure and relations to other subjects.

- <u>Sequential:</u> Sequential, or Serialist. The learner tending to take things step by step, in a linear manner, in a bottom up approach.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

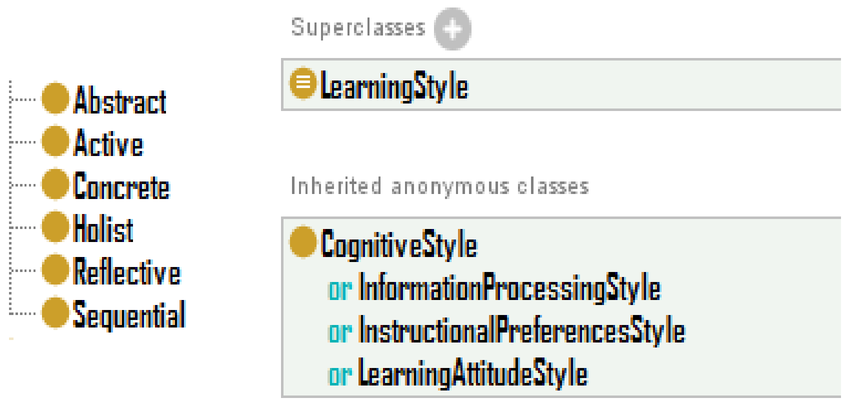


Figure 4-30: Abstract, Active, Concrete, Holist, Reflective and Sequential class structure

- **InformationProcessingStyle class:** This class focusing on the processes, by which information is obtained, sorted, stored and utilized. It contains the following subclasses:

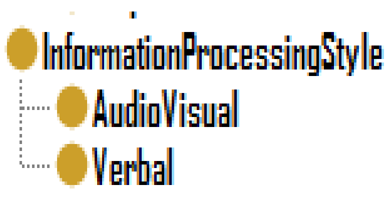
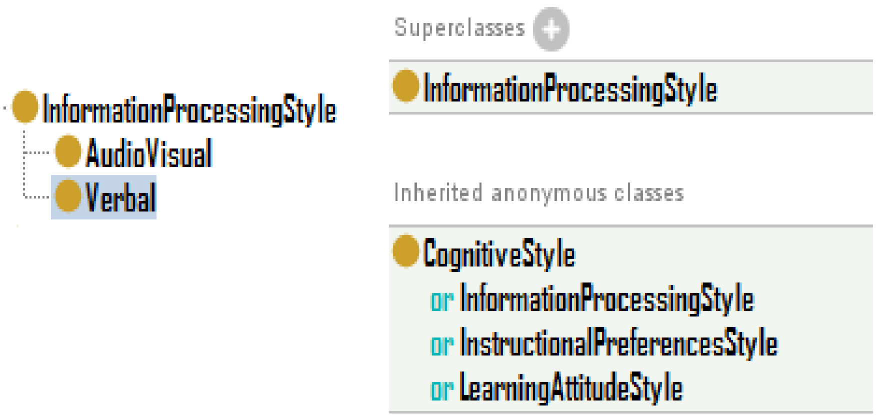


Figure 4-31: InformationProcessingStyle class and its subclasses

– <u>AudioVisual</u>: The learner tending to prefer audio-visual material for learning.

– <u>Verbal:</u> The learner tending to prefer written exposition.


InformationProcessingStyle
AudioVisual
Verbal
Superclasses
InformationProcessingStyle
Inherited anonymous classes
CognitiveStyle
or InformationProcessingStyle
or InstructionalPreferencesStyle
or LearningAttitudeStyle

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Figure 4-32: AudioVisual and Verbal class structure

- **InstructionalPreferencesStyle class:** This class focusing on the most observable traits of a learner, examples of which are the following subclasses:

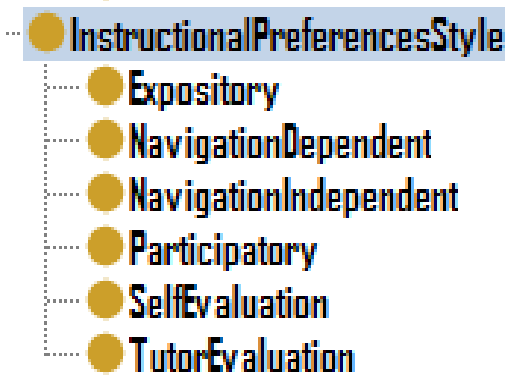


Figure 4-33: InstructionalPreferencesStyle class and its subclasses

- <u>Expository:</u> The learner prefers to follow an exposition.
- <u>NavigationDependent:</u> The learner relies on navigation decisions made by the tutor.
- <u>NavigationIndependent:</u> The learner intervenes and modifies navigation sequence, making own choices.
- <u>Participatory:</u> The learner prefers to engage in interaction, participate in some sort of collaborative, or just active endeavor.
- <u>SelfEvaluation:</u> The learner relies on his/her meta-cognitive self-assessment skills.
- <u>TutorEvaluation:</u> The learner tends to rely on and seek external evaluation to feel reassured on progress made, or get warnings on observed shortcomings.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

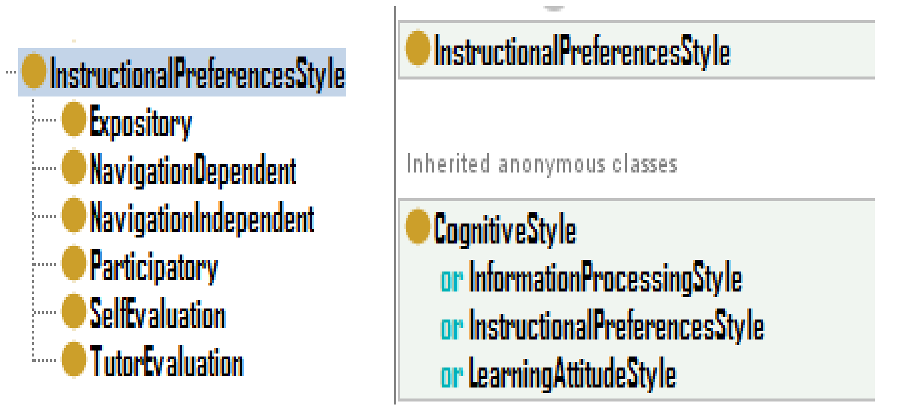

InstructionalPreferencesStyle
Expository
NavigationDependent
NavigationIndependent
Participatory
SelfEvaluation
TutorEvaluation
InstructionalPreferencesStyle
Inherited anonymous classes
CognitiveStyle
or InformationProcessingStyle
or InstructionalPreferencesStyle
or LearningAttitudeStyle

Figure 4-34: Expository, NavigationDependent, NavigationIndependent, Participatory, SelfEvaluation and TutorEvaluation class structure

- **LearningAttitudeStyle class:** This class contains the following subclasses:

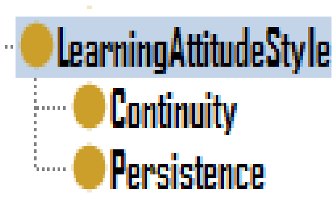


Figure 4-35: LearningAttitudeStyle class and its subclasses

- <u>Continuity:</u> The learner tends to study continuously over extended periods of time.
- <u>Persistence:</u> The learner persists in his/her efforts to achieve the goal (a good performance on the assessment material). Tends to make repeated attempts if original ones fell short of self-imposed objectives.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

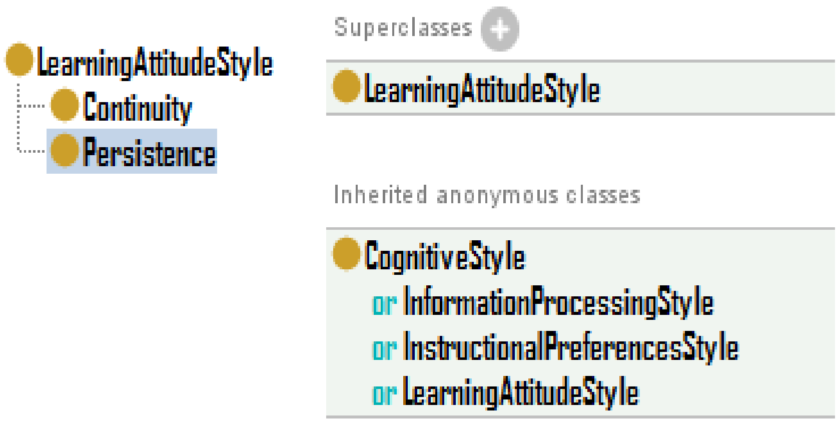


Figure 4-36: Continuity and Persistence class structure

4.3 A Learner Model Ontology

In this section we present our Learner Model ontology. As we mentioned in section 2.1 the main issue with ITS's and the student models they have is that they are specific to the subject taught by the ITS. By contrast, in the CROP Learning Domain view [5] the Learner Model is global. It is the same model that needs to be maintained and updated for each and every subject the learner undertakes to study. In the diagram below we present our view for the structure of the Learner Model.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

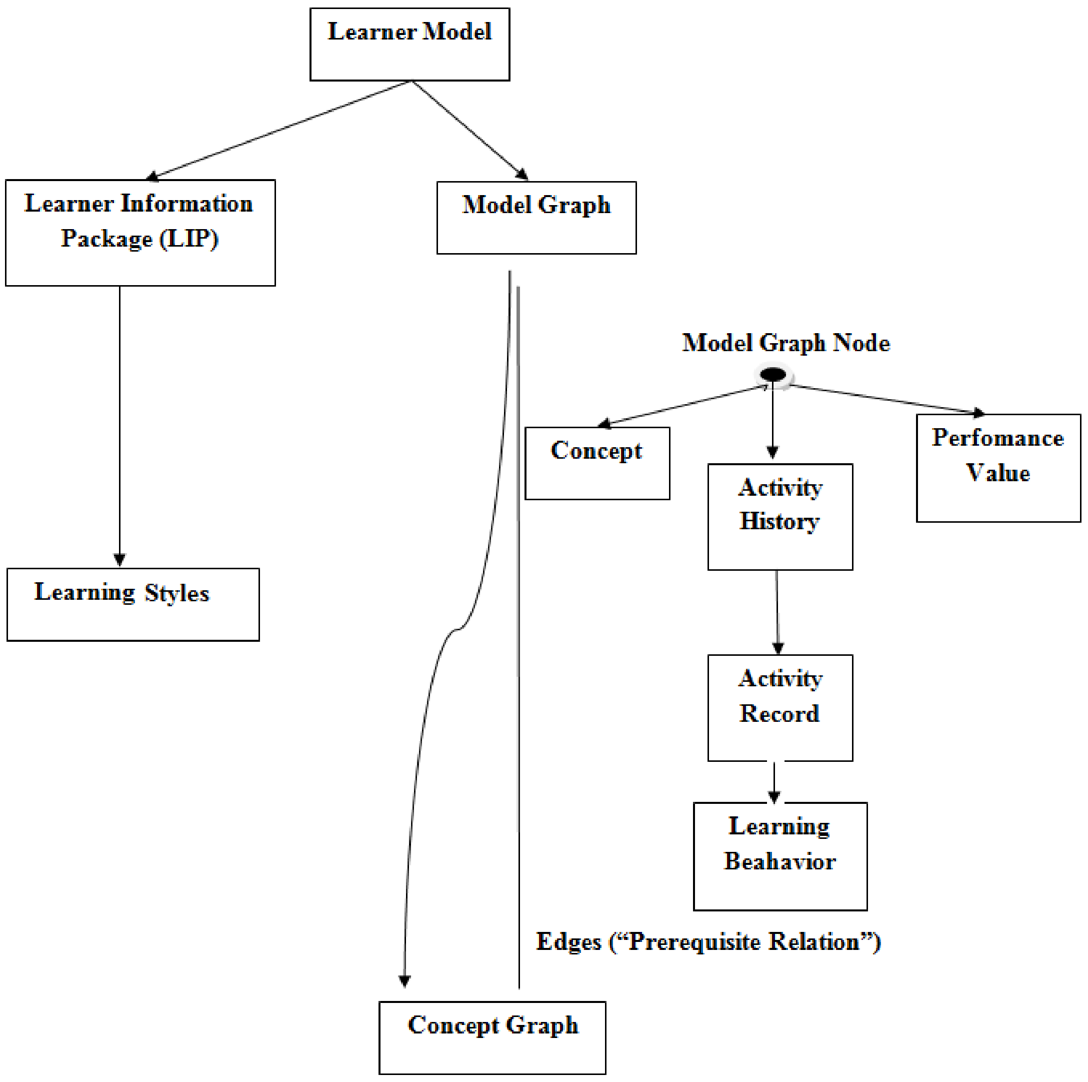


Figure 4-37: Diagram of Learner Model structure

As shown in Figure 4-37 the Learner Model that we propose consists of the IMS LIP (Learner Information) and a Model Graph, all implemented as ontologies. This graph is much like a concept graph, except for the association of additional entities on each node (other than a concept), such as performance measure, duration, request type, rate etc.

This structure is implemented in the Learner Model ontology. The ontograph diagram of the classes of this ontology is presented in Figure 4-38.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

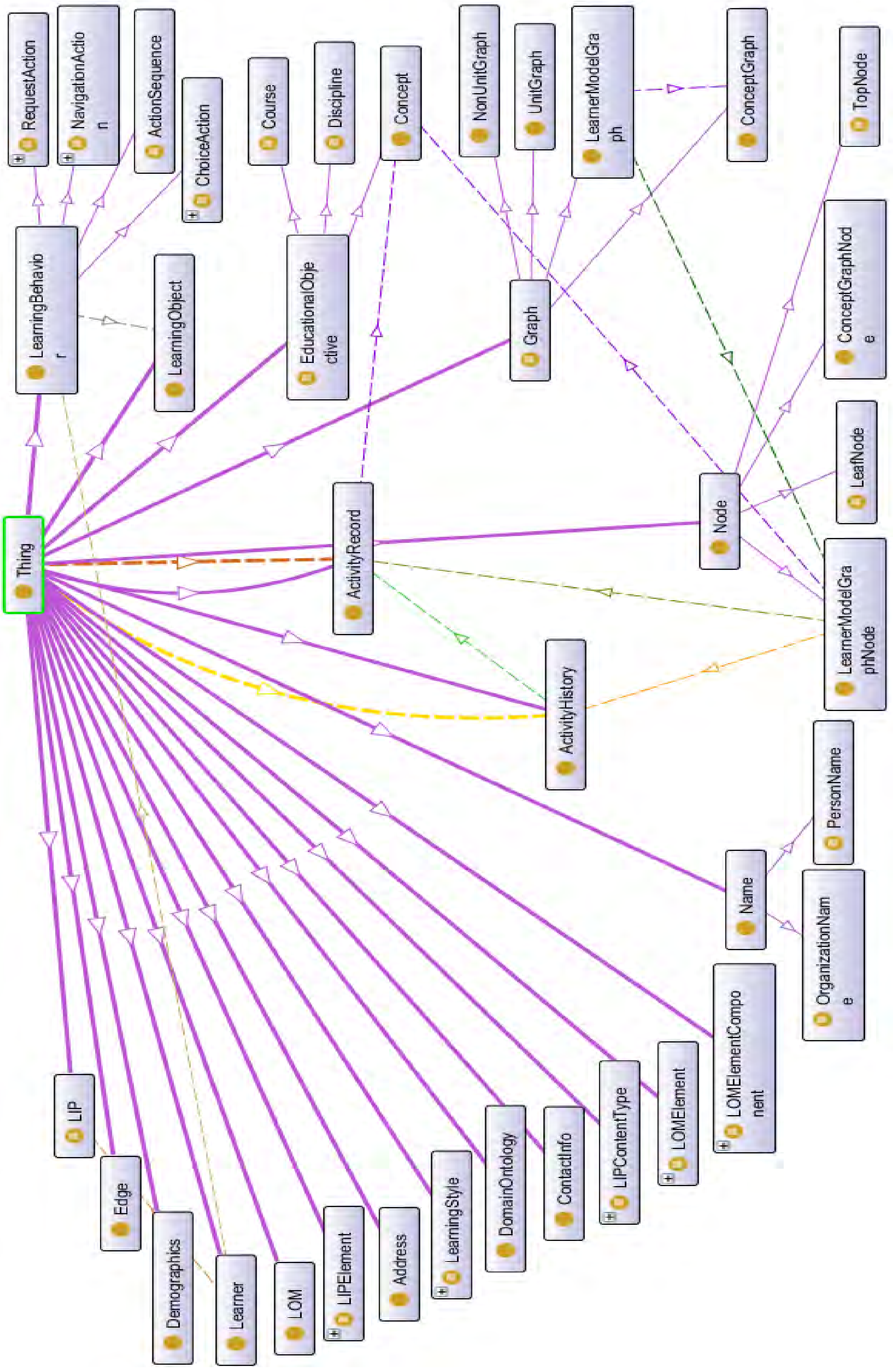


Figure 4-38: Learner Model ontology ontograph diagram

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Figure 4-39 presents the hierarchy of classes of the Learner Model.

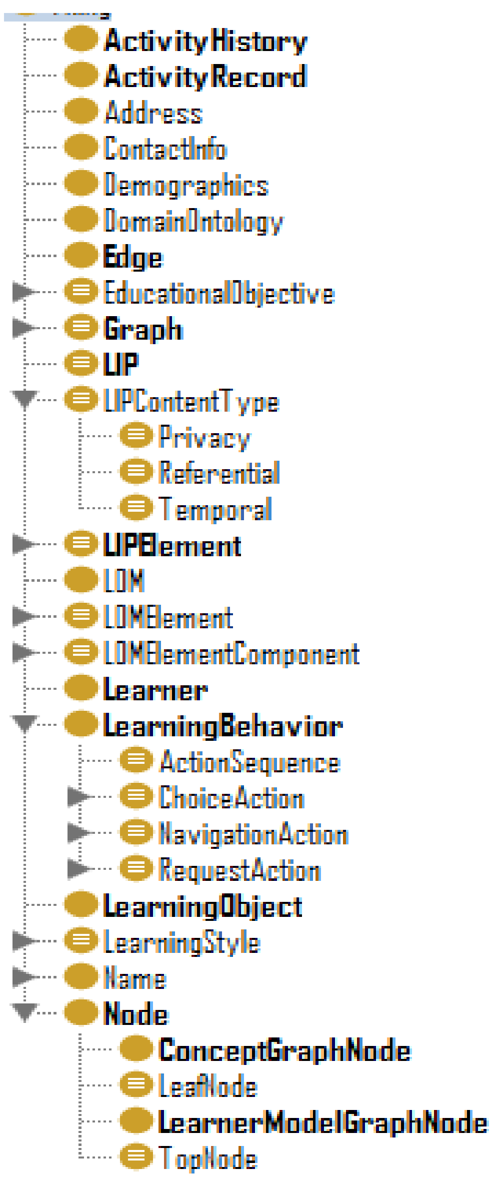
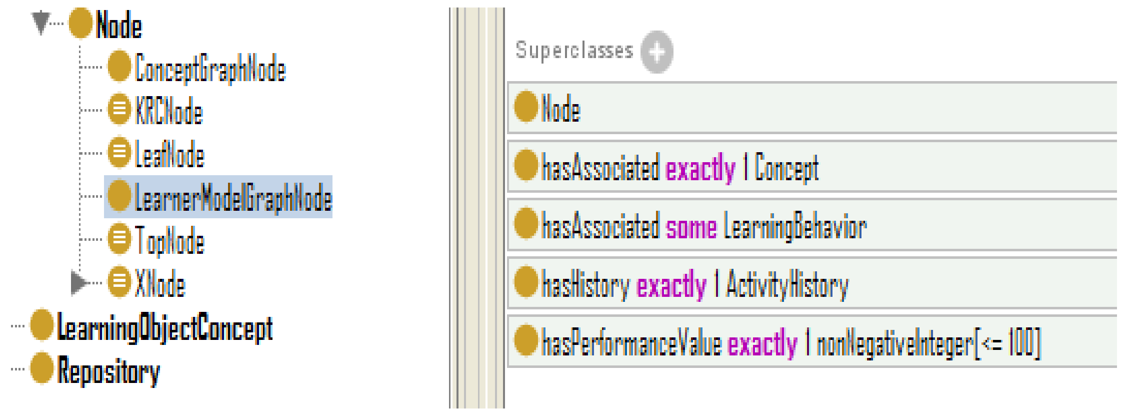


Figure 4-39: Class hierarchy of Learner Model ontology

Our Learner Model ontology is the main ontology of our research. This ontology contains all classes related to student information and uses a number of import ontologies and it is designed to also represent knowledge about the interaction of the learner with the educational material.

The most significant class in this ontology is the LearnerModelGraphNode class, shown in the Figure 4-40.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



Node
ConceptGraphNode
KRCNode
LeafNode
LearnerModelGraphNode
TopNode
XNode
LearningObjectConcept
Repository

Superclasses

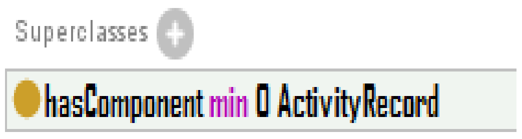
Node
hasAssociated exactly 1 Concept
hasAssociated some LearningBehavior
hasHistory exactly 1 ActivityHistory
hasPerformanceValue exactly 1 nonNegativeInteger[<= 100]

Figure 4-40: The Learner Model Graph node class

Associated to each node of the Model Graph (as captured in the ontology, see Figure 4-40) are the following:

- A Concept (an instance of the KConcept class),
- A LearningBehavior (an imported subontology presented in section 4.3.2),
- An Activity History (consisting of a number of ActivityRecords),
- A Performance Value (indicated the degree of understanding of the concept associate to the node by the student).

The ActivityHistory class is a set of ActivityRecords.



Superclasses

hasComponent min 0 ActivityRecord

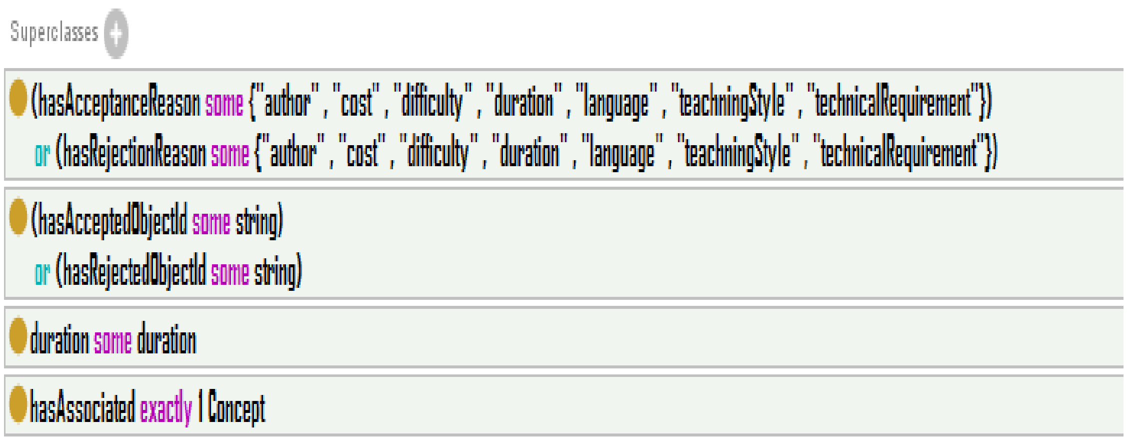
Figure 4-41: ActivityHistory class structure

In terms of the ActivityRecord class, an instance of this class may be for example a record of a request made by the Learner for teaching material targeting a specified (by the Learner) educational objective. Such a record also contains information on a specific response (offer) received by the user, proposing a learning object to him/her. It further contains information on the decision made by the user, possibly with additional explanations as to the reason why the offer was rejected (or accepted), e.g. preference

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

(or lack thereof) for a specific author, rating of the proposed material (offered by some Rating Service), cost, expected duration etc. It further contains, if the offer of the record was eventually accepted and the learning object used, the rating that the learner assigned to the used material.

The implementation of the ActivityRecord class is presented below.


Superclasses

```
(hasAcceptanceReason some {"author" , "cost" , "difficulty" , "duration" , "language" , "teachningStyle" , "technicalRequirement"})
 or (hasRejectionReason some {"author" , "cost" , "difficulty" , "duration" , "language" , "teachningStyle" , "technicalRequirement"})

(hasAcceptedObjectId some string)
 or (hasRejectedObjectId some string)

duration some duration

hasAssociated exactly 1 Concept
```

Figure 4-42: ActivityRecord class structure

The Learner Model ontology imports some other ontologies that help to specify all its necessary concepts. The imported ontologies are:

- Direct Imports:
  - LearningObject.owl
  - LearnerInformation.owl
  - LearningBehavior.owl
  - ConceptGraph.owl
- Indirect Imports:
  - KConcept.owl
  - LearningStyle.owl
  - Graph.owl
  - LearningObjectMetadata.owl

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

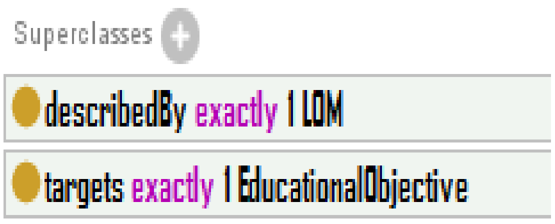
Several of these ontologies, including the KObject ontology (not listed above), have been discussed in previous sections. In the sequel we discuss the Learning Object and Learning Behavior ontologies.

4.3.1 The Learning Object Ontology

This is an ontology intended to capture a general notion of Learning Object and not necessarily a CROP Object. A CROP Learning Object (a Learning Object in the CROP Reference Architecture) is a subspecies of a Learning Object. Non-CROP Learning Objects may be incorporated in the construction of CROP Objects, by treating then as KResources (objects with no internal structure). The current ontology specifies the minimum requirements for an entity to be a Learning Object. It has two imported ontologies:

- The KConcept ontology and
- The LearningObjectMetadata ontology

The main concept of this ontology is the LearningObject class which is described by exactly one LOM and targets exactly one educational objective.



describedBy exactly 1 LOM

targets exactly 1 EducationalObjective

Figure 4-43: LearningObject class structure

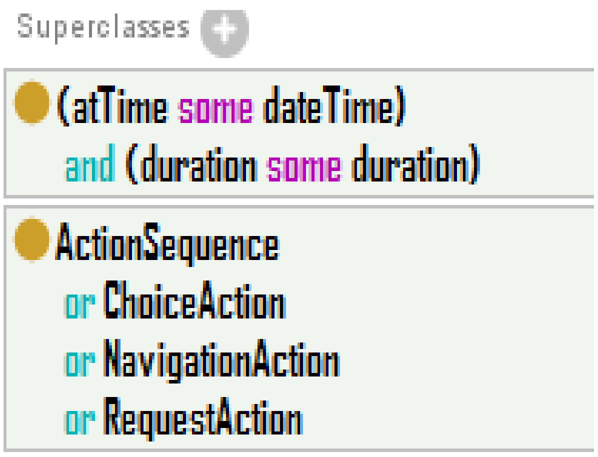
Our view of student models and their use are part of a general view of Learning Domains developed in the context of the CROP Reference Architecture. Hence, we consider Learning Objects as typically, though not exclusively, CROP Objects. The KObject and KConcept ontologies are presented in chapter 3.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Moreover, the LearningObjectMetadata ontology is described in detail in section 4.1 and the LearnerInformation ontology is described in detail to section 4.2.

4.3.2 The Learning Behavior Ontology

This ontology attempts to capture features of behavioral traits of the learner, during teaching interaction with a learning object. The main class of this ontology is the LearningBehavior class and consists of four subclasses which represent the different movement of the learner in the learning domain and also this class records the exactly time and duration of the total interaction.


Superclasses

● (atTime some dateTime)
and (duration some duration)

● ActionSequence
or ChoiceAction
or NavigationAction
or RequestAction

Figure 4-44: LearningBehavior class structure

The Figure 4-45 shows these classes and their relations among them.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

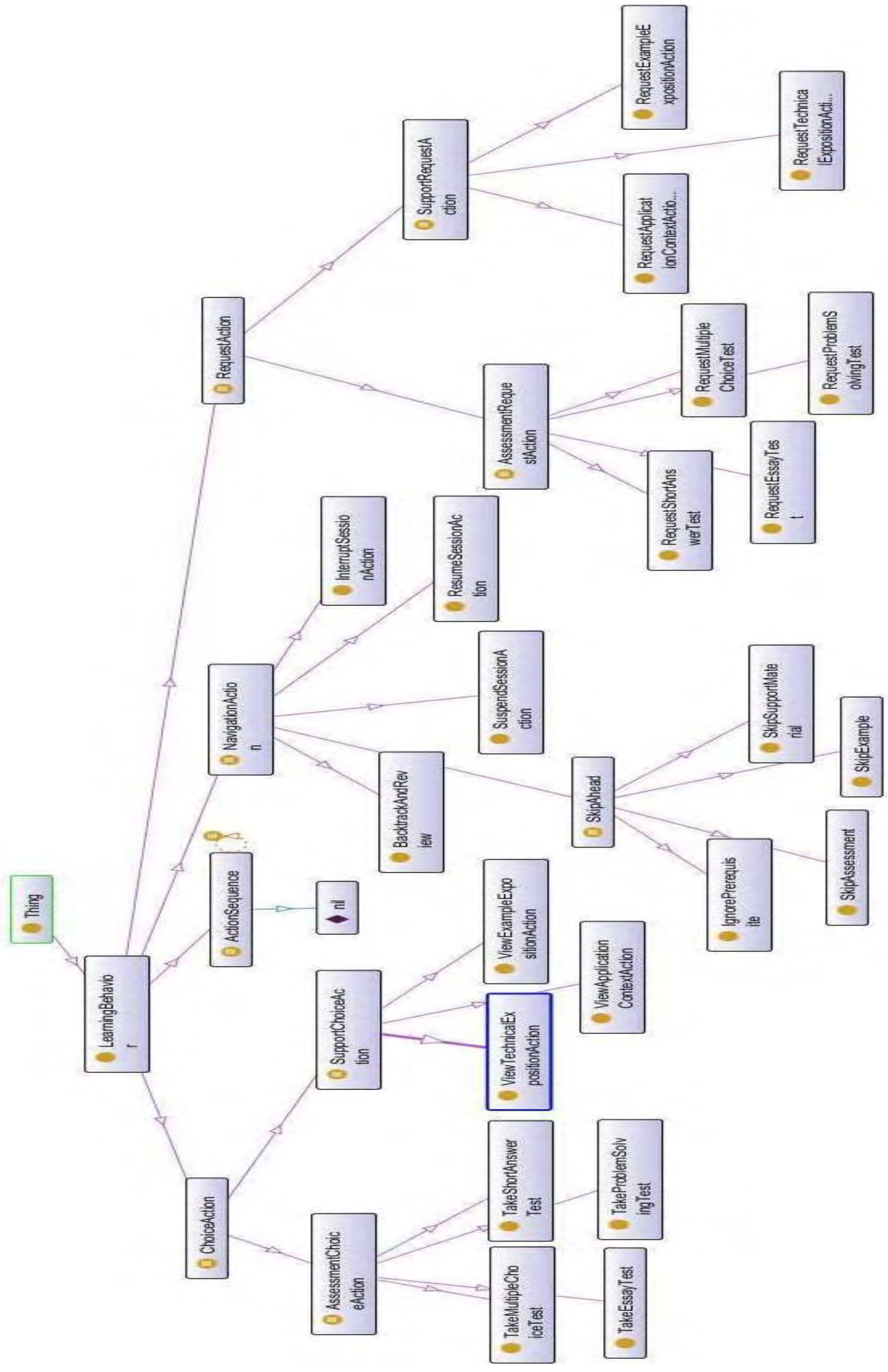


Figure 4-45: LearningBehavior ontology ontograph diagram

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

As we observe from the above diagram LearningBehavior class has the following subclasses:

- **ActionSequence** class: This class presents the sequences of actions that intended to capture repeated action patterns.

```
(first some
    (ChoiceAction or NavigationAction or
     RequestAction))
 and (next some ActionSequence)
```

- **ChoiceAction** class: This class expresses the choice of learner for some support or assessment teaching material. It consists of subclasses :
  - *AssessmentChoiceAction* class and
  - *SupportChoiceAction* class

AssessmentChoiceAction class constitutes from subclasses:

- TakeEssay Test
- TakeMultipleChoice Test
- TakeProblemSolving Test
- TakeShortAnswer Test

and show us what kind of test the learner choose to answer in order to assess his knowledge on a subject.

SupportChoiceAction class constitutes from subclasses:

- ViewApplicationContextAction
- ViewExampleExpositionAction
- ViewTechnicalExpositionAction

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

and give the learner a broad spectrum of choices/options in order to advances in the educational process.

- **NavigationAction** class: Default Navigation of the concept graph is defined by the Execution model. The learner makes decisions and actions that may simply carry out the predefined sequence, or not. The discrepancy between predefined and self-determined navigation may be informative as to the student style and possible adaptation response.

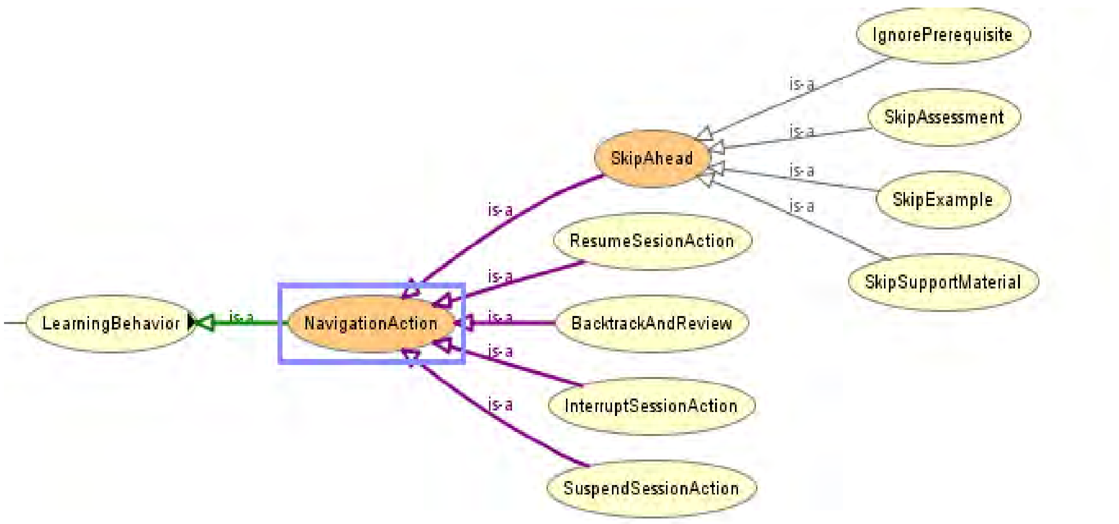


Figure 4-46: OwlViz diagram from NavigationAction class

This class consists of subclasses:

- *BacktrackAndReview class*, this means that backtrack in the concept hierarchy and revisit a learning objective already visited. Frequency of such actions may be relevant in deducing needs for memory aids.
- *InterruptSessionAction* class, interrupt a learning session. Interaction with the same learning object may resume at some future time, or not.
- *ResumeSessionAction* class, resume a suspended session. Session suspension time may be informative as to the student's commitment to continuous studying of lengthy material.
- *SuspendSessionAction class*, pause studying. Suspend the current session.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- *SkipAhead class*, contains the following actions :
  - IgnorePrerequisite class, skip ahead and choose to study a concept though not all of its prerequisites have been studied.
  - SkipAssessment class, while studying a concept skips the (or some) assessment material.
  - SkipExample class, while studying a concept skip (some) example material.
  - SkipSupportMaterial class, while studying a concept skip (some) example material.

- **RequestAction** class: A request submitted by the student, during interaction with a learning object and through a dialogue system. Such a request takes place when the available material already packaged for student use appears to be insufficient for the student, who then requests for additional material.

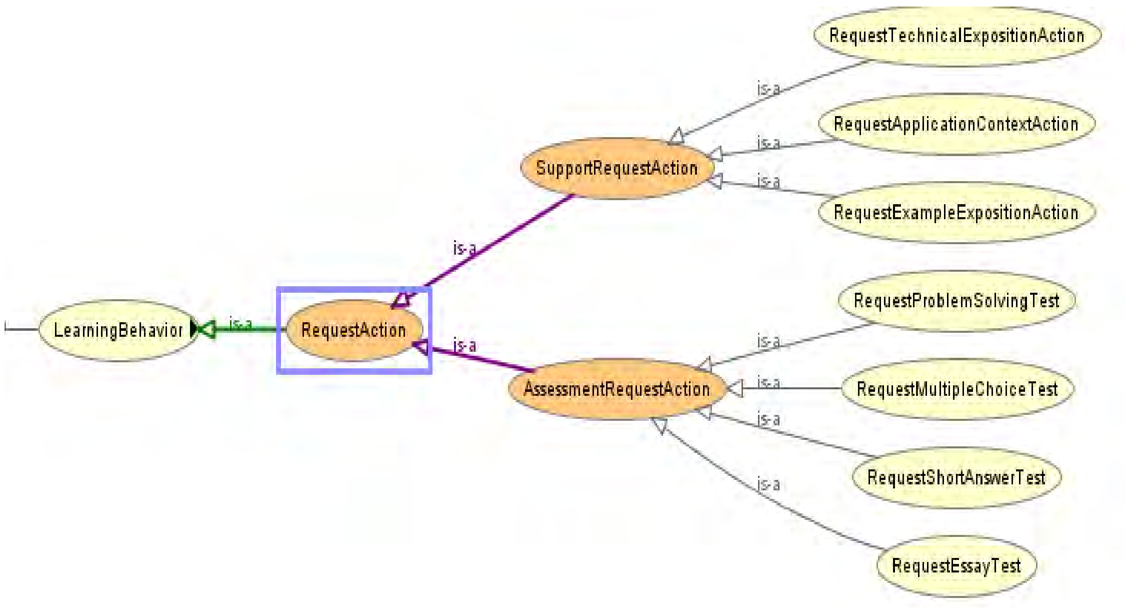

RequestTechnicalExpositionAction
RequestApplicationContextAction
RequestExampleExpositionAction
SupportRequestAction
is-a
LearningBehavior
RequestAction
AssessmentRequestAction
RequestProblemSolvingTest
RequestMultipleChoiceTest
RequestShortAnswerTest
RequestEssayTest

Figure 4-47: OwlViz diagram from RequestAction class

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

As we see from the Figure 4-47 RequestAction class is divided to SupportRequestAction class and AssessmentRequestAction class.

- *AssessmentRequestAction class*, provide request for additional assessment material. This class constitutes from subclasses:
    - RequestEssay Test class,
    - RequestMultipleChoice Test class,
    - RequestProblemSolving Test class,
    - RequestShortAnswer Test class.
- *SupportRequestAction class*, provide request for additional support material (technical explanations, examples, solved problems, illustrations of some kind etc. This class constitutes from subclasses:
    - <u>ViewApplicationContextAction class</u>, request for a presentation or analysis pertaining to the actual or potential applications of the material under study.
    - <u>ViewExampleExpositionAction class</u>, request for an example, a solved problem, and illustration of some kind etc.
    - <u>ViewTechnicalExpositionAction class</u>, request for a technical definition and/or explanation, a definition etc.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 5:

Reasoning Issues

In this chapter we will discuss reasoning issues about Learning Objects, Learner and finally reasoning issues that result from interaction between the Learning Object and the Learner. Generally, the reasoning theory is a very important algorithmic procedure which forms the link between information and knowledge. Therefore, reasoning is the process in which we use an existing knowledge to draw conclusions or extract something that we know in our field of interest. In the current chapter we will attempt to analyze the reasoning issues by using a process that includes the following stages:

1. At first we will need to trace the useful information on which we will process the reasoning. In other words we have to understand what kind of LO or Learner relevant information we need to do the reasoning.
2. The next step is to check on which elements of the LOM (about LO) or the LIP (about Learner) respectively this information is registered, if any.
3. After that, we need to record this information using the right format, so that it can be used for reasoning. In the current research, recording of this information is done ontologically by the aid of an ontology that was created in a previous chapter. As stated before, we have created our ontologies with the help of protégé 4.0.2.
4. The next stage is where we have to use some query language (in our scenario we use the one provided by Protégé and OWL), through which we impose several queries and the applications should be in a position to process the appropriate reasoning and return the desired results.

In Figure 5-1, we present a scenario that includes the above stages and in essence describes the operation of the whole procedure through which reasoning is completed. The scenario works as shown:

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

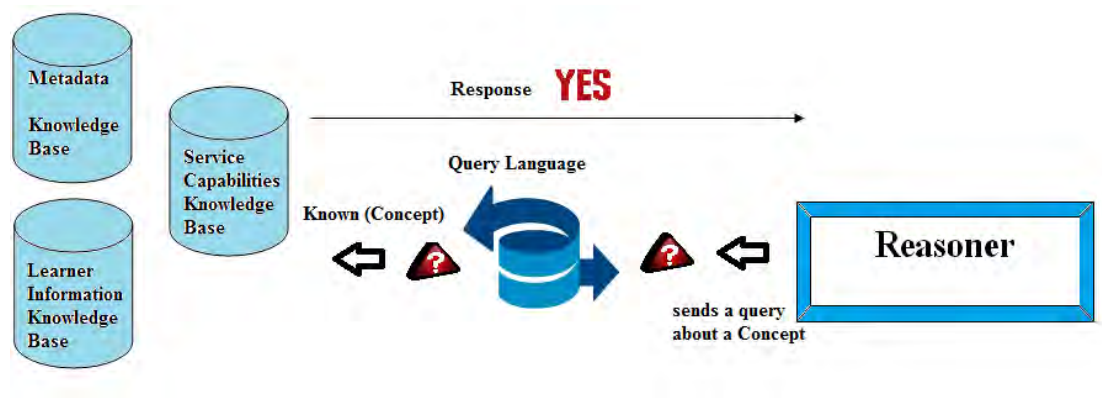

Metadata Knowledge Base
Learner Information Knowledge Base
Service Capabilities Knowledge Base
Response YES
Query Language
Known (Concept)
Reasoner
sends a query about a Concept

Figure 5-1: Reasoning Process

As shown on the Figure 5-1, we have three knowledge bases that include the necessary information relevant to the Learning Object Metadata, Learner and Services correspondingly, which are needed to do the reasoning (1st stage). This information is stored in the appropriate fields of LOM or the LIP for the LO or the Learner respectively (2nd stage). Also, this information is included in the base in the form of an ontology, which means that, the base includes the respective ontologies of LOM and the LIP file (3rd stage). Since this information is registered on the knowledge base, they should be usable by a reasoner that will do automatic reasoning. This will be done with the aid of a query language (4th stage) in the following way:

- The reasoner wants to know whether a concept (the “Oscillations” concept for example) is known to the Learner to decide which LO it should return.
- In order for the Learner to figure that issue, it sends a query to the knowledge base through a query language.
- From its part, the query language sends the query set by the reasoner to the knowledge base and then the base checks for the answer.
- In turn, the base replies to the reasoner about whether the specific concept is known to the Learner or not.

For example, if the answer is positive (in our case, if the Lerner already knows the “Oscillations” concept), then the reasoner can conclude that it is able to send a “Waves” concept to the Learner, since learning about Waves assumes that he already knows about Oscillations. This, in short is the process used to complete the reasoning procedure. Preliminarily, we begin with some basic observations regarding using and updating the Learner Model.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

5.1 Using and Managing the Learner Model

5.1.1 Extending the Student Model Graph

In Section 4.3 we presented our view on a Learner Model and we proposed representing knowledge about the Learner by means of a Learner Model Ontology. This Learner Model contains a Model Graph which is updated each time the Learner uses some Learning Object. A critical issue is extending the Model Graph based on the Concept Graph of the Learning Object. This is because each Learner Model Graph has an underlying Concept Graph (the graph obtained by restricting the node associations to KConcept instances only). There are two ways we can discern for this to be done:

- The first one is to copy the whole Concept Graph of the Learning Object to the Learner Model and thus directly extend the underlying Concept Graph inside the model.
- The second option is, every time that a Concept's study is completed, we could add a new node to the Learner Model Graph so as to gradually extend the underlying Concept Graph.

In the first scenario, it is quite obvious that by copying the Learning Object's underlying Concept Graph from the Domain, we would accumulate a large number of nodes that Learner has yet not visited and therefore are of zero features and information. A disadvantage of this outlook is that the Learner may begin dozens of Learning Objects which he will never complete and as a result he will overload the Learners Model's underlying Concept Graph with a large number of Concepts that in essence have no content at all. However, this issue can be dealt with some kind of garbage collection policy. The big advantage here is that we are solving the ontologies merging issue once and for all (section 3.1.2).

Moreover, it is obvious that some merging problems will be presented when we should merge the model graph ontology with a Concept Graph ontology in order to extend the model graph but we don't deal with such issues in this research. We can trace problems of this kind but it is not part of our research to study ontology merge problems any further.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

5.1.2 Monitoring Performance, Actions and Actions-Sequences

After the learner completes the study of a concept while using a Learning Object he typically goes through an evaluation procedure (executing appropriate assessment resources). Upon completing the tests, the model must be updated so that it contains the new data about the new knowledge state of the learner. Regarding the process of updating the model, there some issues that arise:

1) What entities responsible for performing the update procedures?
2) What is actually “updated”?

The answer to the first question is actually given in section 3.3.2 (Learning Services and Learning Domain). As mentioned in the corresponding section, the Learning Domain consists of roles that are actually a set of behaviors (local and interactive) and it is utilized by some Participant, Service or Actor. Therefore this role is responsible for updating the learner model. In other words it is some entity that plays the role of StudentProfileManager.

With regard to what is to be updated, the answer is that we insert the learner’s performance in several tests that he takes on a concept upon completing a learning object. The learner’s performance on those tests is recorded by the LearnerModelGraphNode of LearnerModel ontology. As we refer in the corresponding section 4.3, the LearnerModelGraphNode doesn’t store only the taught concept, but also keeps additional information in order to serve the learner’s needs. Such additional information in the LearnerModel Graph Node constitutes the data type PerfomanceValue which is implemented in our LearnerModel ontology with the following relation:

hasPerformanceValue exactly 1 nonNegativeInteger[<= 100]

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

The learner model, in our proposal, can also record learner’s actions, corresponding to navigations choices or requests (identified by appropriate button actions the learner executes). Detailed observation of the sequence of these actions may lead to useful conclusions regarding the student's learning style. The capturing of the learner’s actions is the responsibility of a Monitor entity in the Learning Domain. Further, a Profile Manager entity proceeds to update the model. In our research knowledge relating to learner actions is captured by means of the LearningBehavior ontology (section 4.3.2).

5.2 Reasoning about Learning Objects

All reasoning about Learning Objects must be based on the MetaData of Learning Objects. MetaData entries can be regarded as *statements of properties* of a Learning Object. To clarify this point we consider in the sequel some of the elements of a LOM file and the intended use of the information they contain. An arising question here is about which of the LOM file’s elements are necessary to a degree that their registration is made obligatory regarding reasoning. Furthermore, supposing we registered those elements, how can we store them in such way that they are exploitable and usable? These are the questions we will try to answer on this section.

Beginning with the question of which are the right LOM file elements to choose, we must discuss about what kind of information would be useful to the Learner and which of his learning needs should he fulfill to select a specific Learning Object. At this point, we must note that on this section we actually examine the reasoning issue about a Learning Object from the Learner’s side, not the Service’s side. This is happen because some type of information contained in the LOM, for example information contained in the “LifeCycle” element (version/status component) maybe is relevant to reasoning performed by the Learning Service and not by the Learner.

We propose that the basic information relevant to reasoning about properties of a Learning Objects is as follows:

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- The Target Educational Objective of the LO, that is to say what concept, course or discipline the LO teaches,
- The Learning Object ontology, namely the set of concepts taught by this LO,
- The Prerequisites concepts required to be known by the user of the LO,
- The Educational Context of Use / Educational Level that this LO is appropriate for,
- The Language(s) in which the actual teaching material is composed,
- The Typical Duration of the learning experience with the LO,
- The Author(s) of the LO,
- Its Coverage (Historical/Cultural/Geographical),
- Its Date of Construction (how recent it is).

➢ Its Rating, direct or indirect (success levels of users), or by explicit recommendation,

➢ The (Frequency of) use by users,

➢ The (Frequency of use) by authors (in other LOs),

➢ The Cost, that is to say the amount that the Learner should pay in order to use the LO.

▪ The Difficulty / Density level of the LO.

❖ The Information about Illustration Type Material (Figure,diagram,video,image etc),

❖ The InteractivityType/Level, namely the predominant mode of learning supported by this LO.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

In a Learning Domain the above information is used by a Broker / Facilitator participant who realizes the search of a LO according to each request of the Learner. The Broker submits a proposal in Learner for one or more LOs with communication elements for the Learning Service that owns it. At this point, the first stage is completed. It's the stage of tracing the useful information based on which we will proceed with the reasoning process. We move on the next stage which is recording this information so that is accessible. This recording is done with the help of the Learning Object Metadata ontology. At this stage, in a way, we will assign this information that we traced before to several elements of Learning Object Metadata ontology. This will be done in the following way:

The target educational objective of the LO, consists of information that can be retrieved by the "purpose" (subelement of the "classification" element). The classification category at LOM scheme describes the learning object in relation to a particular classification system. The classification element is partially defined by the following restriction:

(**hasElementComponent** some **TaxonPath**)

and (**hasElementComponent** only **TaxonPath**)

and (**keyword** some **string**)

and (**description** max 1 **string**)

and (**purpose** max 1 {"**accessibility restrictions**" , "**competency**" , "**discipline**" , "**educational character**", "**educational objective**" , "**idea**" , "**prerequisite**" , "**security level**" , "**skill level**"})

Multiple classification elements can be defined and this basically depends on the *purpose* of the classification. The IEEE LOM standard introduces a controlled vocabulary for the specification of the purpose of the classification, shown above. For

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

example, assume the objective of the Learning Object is to teach a learner the subject of learning object metadata. The subject falls within both the Computer Science discipline and the Education discipline. We will then have a classification item including, for example, the information below,

purpose (“**discipline**”)

hasElementComponent(“http://www.cs.teilar.gr/ontologies/ScienceDisciplines.owl#**ComputerScience**”)

hasElementComponent(“http://www.cs.teilar.gr/ontologies/ScienceDisciplines.owl#**Education**”)

Another classification item would be required to specify the teaching objective of the learning object.

purpose (“**educational objective**”)

hasElementComponent (“http://www.cs.teilar.gr/ontologies/lom.owl#**LOM**”)

More classification elements can be created to declare the prerequisites of the learning object, for example

purpose (“**prerequisite**”)

hasElementComponent(“http://www.cs.teilar.gr/ontologies/metadata.owl#**MetaData**”)

The IEEE LOM standard assumes that appropriate classification schemes for the other “purpose” entries exist. The proposed vocabulary can be certainly modified appropriately to encompass other legitimate classification purposes and needs.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

At this point we should clarify that all information about a learning object can be found in its LOM while its ontology can be found in the learning domain. In the learning domain, there is an ontology repository where ontologies are stored. These include the global ontology of the learning domain as well as its subontologies, the LO domain (content) ontologies. In terms of the prerequisites of the LO, we can retrieve them, also, from the "classification" element of the LOM file. This can be achieved with "purpose" (subelement of the "classification" element).

The educational "context" of use (IEEE LOM subelement 5.6) / educational level, "language(s)" (IEEE LOM subelement 5.11), "typical duration" (IEEE LOM subelement 5.9), "difficulty" (IEEE LOM subelement 5.8), "density" (IEEE LOM subelement 5.4), "interactivity type" (IEEE LOM subelement 5.1) and "interactivity level" (IEEE LOM subelement 5.3) of the LO are information which can be retrieved by the corresponding subelements of the "educational" element (described in detail to section 3.7). In many typical situations we can observe some similarity in the content of "context" and "TypicalAgeRange" subelements capture the same information (to show the principal environment within which the learning and use of this learning object is intended to take place). There are also cases where the two subelements may include no identical information.

In the case of the information about illustration type material (figure, diagram, video, image etc) we can retrieve the necessary information through the "relation" element and the "has part" relation one can retrieve information about all LOs whose "structure" (subelement of the "general" element) is atomic and its "LearningResourceType" a (subelement of the "educational" element) is a figure, diagram, table, etc.

From the "general" element of the LOM file we can recover information relates to "coverage (historical/cultural/geographical)" subelement (IEEE LOM subelement 1.6).

Information like the author and the date of construction of the LO is contained in the "lifecycle" element through the "contribute" subelement (IEEE LOM subelement 2.3). This subelement includes information about the entities that have contributed to the state of this learning object during its life cycle, the role under which they made their contribution and the date of the contribution.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Regarding the rating of the LO, we should note that a rating service exists, but its analysis is beyond the scope of this thesis. The rating of an LO includes information such as,

- Frequency of use by users,
- Frequency of use by authors (in other LOs).

Finally, the cost that an LO may have is contained in the "rights" element through the "cost" subelement (IEEE LOM subelement 6.1).

On the fourth and final stage of analysis procedure for reasoning issues about LOs, we apply a query language. There have been several query languages developed(RQL, SquishQL, RDFQL, SPARQL, VERSA) for searching and data acquiring in an ontology as discussed in section 3.1.3.

5.3 Reasoning about the Learner

As for reasoning about Learning Objects, reasoning about the Learner is based on the Learner Model. In the CROP view, the Learner Model consists of

(a) a Learner Information Package, the IMS LIP
(b) a record of learner behaviors and
(c) a learner model graph

(Each of the above has been presented in a previous chapter 3)

Following the first stage, the main question that results logically also in the learner's reasoning case is, what kind of information about the learner do we need to base our reasoning on? In the LO's reasoning we presented a list of information which is used by any entity (such as the Broker) constructing a proposal to the Learner for one or more LOs. Constructing a proposal for a learner also depends on information about the learner which will discuss below,

❖ The basic information regarding the identification of the learner:

- The age of the learner,

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- The gender of the learner (if he/she is male or female),
- What is the educational level of the learner (if he/she is a high school student or higher education student),
- What is his/her affiliation (if student, this may be school attended),
- What language(s) the student speaks,
- What qualifications, certifications and licenses (qcl) the learner has (official Qualifications: his/her target concepts and dates),
- What is the rated knowledge that learner has, of concepts, courses, or disciplines that he/she studied and which is the information about the performance that he/she acquired in these,
- The abilities (competencies) that the learner has,
- What is the (current) goal (educational objective/ target concept) for learning,
- The LOs usage (of past), for what reasons the learner accepted or rejected a LO at the past.

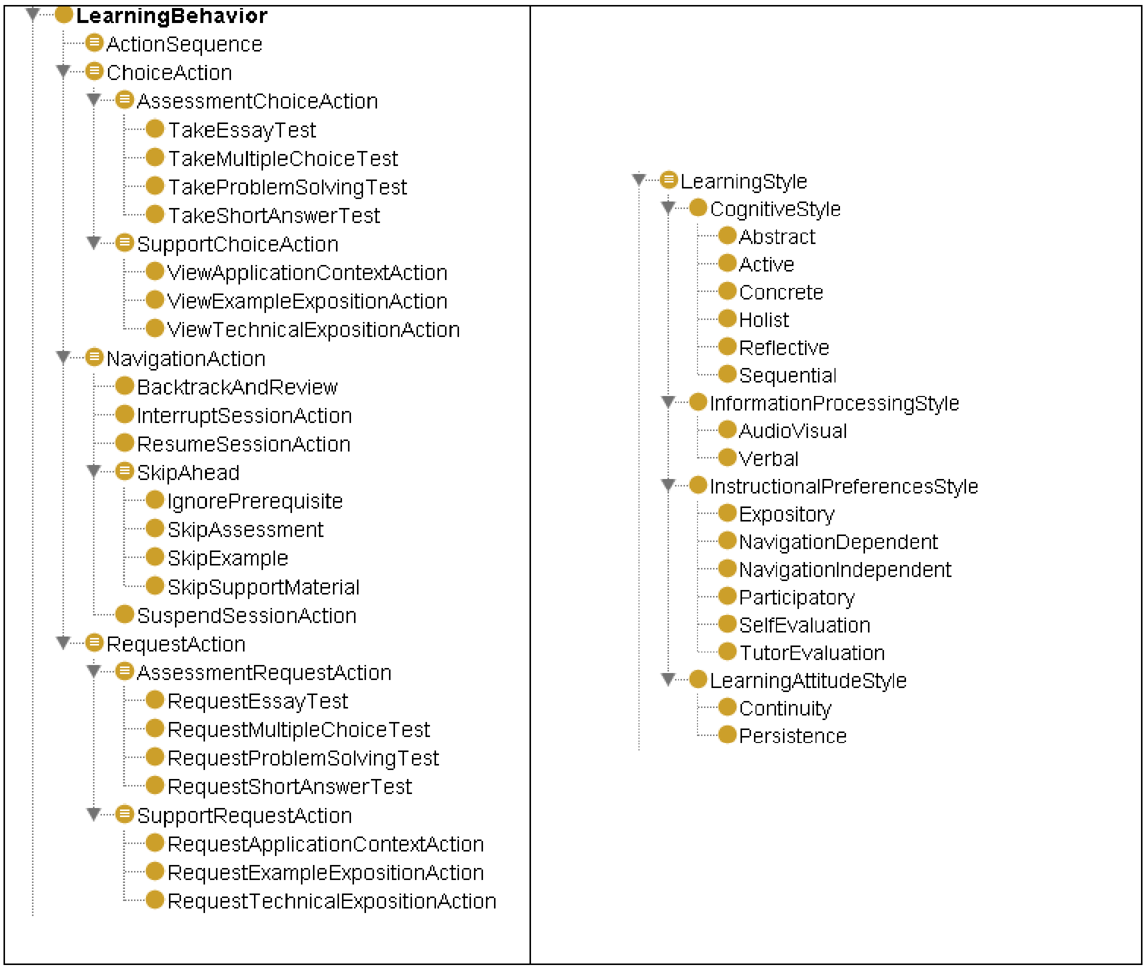
❖ The behavior and style of the learner:

The main issue here is to infer the learner's learning style from

- o The behavior patterns in interaction with the system,
- o The performance on certain types of test (tests about data and facts as opposed to tests about abstract ideas, principles and theories).

In the Table 5-1 we see in the left column the LearningBehavior ontology while in the right column the LearningStyle ontology. The key issue here is to relate the behavior of the learner and the learning type characteristics of his/her.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76



- LearningBehavior
  - ActionSequence
  - ChoiceAction
    - AssessmentChoiceAction
      - TakeEssayTest
      - TakeMultipleChoiceTest
      - TakeProblemSolvingTest
      - TakeShortAnswerTest
    - SupportChoiceAction
      - ViewApplicationContextAction
      - ViewExampleExpositionAction
      - ViewTechnicalExpositionAction
  - NavigationAction
    - BacktrackAndReview
    - InterruptSessionAction
    - ResumeSessionAction
    - SkipAhead
      - IgnorePrerequisite
      - SkipAssessment
      - SkipExample
      - SkipSupportMaterial
    - SuspendSessionAction
  - RequestAction
    - AssessmentRequestAction
      - RequestEssayTest
      - RequestMultipleChoiceTest
      - RequestProblemSolvingTest
      - RequestShortAnswerTest
    - SupportRequestAction
      - RequestApplicationContextAction
      - RequestExampleExpositionAction
      - RequestTechnicalExpositionAction

- LearningStyle
  - CognitiveStyle
    - Abstract
    - Active
    - Concrete
    - Holist
    - Reflective
    - Sequential
  - InformationProcessingStyle
    - AudioVisual
    - Verbal
  - InstructionalPreferencesStyle
    - Expository
    - NavigationDependent
    - NavigationIndependent
    - Participatory
    - SelfEvaluation
    - TutorEvaluation
  - LearningAttitudeStyle
    - Continuity
    - Persistence

Table 5-1: LearningBehaviour and LearningStyle ontology

The issue pointed out above is under intense current research and many researchers in the e-learning field are working to make progress in this subject. In the sequel, we will see some behavioral patterns that seem to be related to style. As we discussed in the previous section 3.5 (about learning styles) we faced the learning style issue as Global Learning Styles, shifting the emphasis from "style" to "style designator".

In the following we discuss the behavioral patterns that pointing to style designators take into account the [53].

<u>Behavioral patterns pointing to style designators</u>

Active

Active learning refers to students that do more than simply listen to a lecture. Students are *doing* something including discovering, processing, and applying information.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Active learning "derives from two basic assumptions: (1) that learning is by nature an active endeavor and (2) that different people learn in different ways.

Indicative Active Learner's Behaviour

- discuss (in forums or chat rooms/ group learning case),
- expected to perform more self-assessment tests, more exercises, spend more overall time on assessment,
- finds it difficult to sit in lectures just taking notes; requires interaction,
- expected to spend less (little) time on examples (since they prefer doing things by themselves).

Reflective

Reflective learning is highly individual. It refers to a great or deeper degree of processing of material to be learned. Reflective students prefer to spend more time collecting information (which means searching one's memory as well as external sources) and analyzing its relevance to the solution before offering a response.

Indicative Reflective Learner's Behaviour

- expected to spend more time on reading material
- tend to take longer on assessment, tend to take time on reflecting on assessment results
- unlikely to answer incorrectly the same test (they had taken time to reflect on first-time results)
- prefer working alone
- retain and understand information best by thinking about it first

Sensing (Concrete)

Concrete learning refers to students that absorb information through direct experience, by doing, acting, sensing, and feeling. Sensing learners tend to like learning facts; they get inference by analyzing performance on assessment about facts and data.

Indicative Concrete Learner's Behaviour

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- tends to prefer examples (of visits/requests and time spent on examples)
- likes to solve problems based on *standard* procedures (*how do you detect that??*) – indicated by a high interest in examples and solved problems (in order to get to grips with the approach/method/procedure)
- likes to take tests (in order to check understanding)
- tend to be more practical and careful
- do not like courses that have no apparent connection to the real world

Intuitive (Abstract, Theorist)

Intuitive learning refers to students that like to discover relationships and possibilities. They dislike hearing the same thing over and over again – they like fresh, new, interesting ideas. They often are better at learning abstract concepts than they are at mastering concrete facts.

Indicative Intuitive Learner's Behaviour

- less time spent on examples
- like innovation and dislike repetition
- tend to work faster and are more innovative but may be careless
- do not like courses that involve a lot of memorization and routine calculations
- like challenges

Visual

Visual learning utilizes graphical ways of working with ideas and presenting information. The visual learners remember best what they see - pictures, diagrams, flow charts, time lines, films, and demonstrations.

Indicative Visual Learner's Behaviour

- prefers/requests pictures, photos, videos, diagrams, maps, charts etc as appropriate,
- benefits from information obtained from textbooks and class notes,

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

- tend to like to study on their own in a quiet room,
- Often see information "in their mind's eye" when they are trying to remember something.

Verbal

Verbal learning refers to students that get more out of words - written and spoken explanations.

Indicative Verbal Learner's Behaviour

- prefers/requests test-based exposition,
- try to work in groups where members explain concepts and ideas,
- when studying, write summaries or outlines of course material in their own words

Sequential

Sequential learning refers to students that tend to gain understanding in linear steps, with each step following logically from the previous one.

Indicative Sequential Learner's Behaviour

- tends to depend on predefined navigation sequence,
- gain understanding in small sequential, logical steps,
- May not understand material fully but are still able to solve problems and pass tests.

Global

Global learning refers to students that tend to learn in large jumps, absorbing material almost randomly without seeing connections and then suddenly "getting it."

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

Indicative Global Learner's Behaviour

- skip ahead actions/ backtrack and resume (maybe repeated cycle),
- make own decisions on navigation,
- may have difficulty in explaining their knowledge.

Moving on to the second stage of recording we will see with which way we will store all above information about the learner in our learner model. As we referred in above section 3.8.1 when we spoke for the learner we had said that we use the IMS LIP Specification for recording necessary information of the learner or the author. Hence, we will use the LearnerInformation Ontology in order to store all information about the learner. Let's see which elements of LearnerInformation Ontology are useful for this process.

Beginning with the demographics elements of learner related to age and gender, they are contained to "Demographics" element of Learner Information Ontology (IMS LIP subelement 2.7).

(**date** some **date**)

and (**gender exactly 1** {"**Female**" , "**Male**"})

The educational level that the learner possesses can be inferred from information contained on the "Qcl" and "Competency" elements on the LIP. This means that if someone (the Learning Service for example) sees the certifications and knowledge attributes that a Learner has declared, plus those that he has gained during the learning process (Competency element), it can infer about what is the educational level of the learner. This inference should be done automatically (automated reasoning) according to the queries set to the knowledge base by the process described in the beginning.

The affiliation information can be found in the "Affiliation" element (IMS LIP element 6.3.9). The affiliation learner information is used to store the descriptions of the organization affiliations associated with the learner.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

The language(s) that the learner speaks are recorded in the “Accessibility” element (“language” subelement, IMS LIP subelement 3.3).

The qualifications, certifications and licenses of the learner are recorded in the “qcl” element (IMS LIP element 6.3.5)

```
(contentype some
   (Concept
    or Course
    or Discipline))
 and (date exactly 1 date)
 and (issuedBy exactly 1 string)
 and (vocabulary exactly 1 {"Award" , "Certificate" , "Degree"})
```

and consists of the qualifications, certifications and licenses awarded to the learner. This element contains the learner’s official qcl’s (his/her target concepts) and the date that these qcls are obtained.

Regarding the abilities that the learner has acquired during learning, these are referred to “Competency” element (IMS LIP element 6.3.7) and this contains the educational objective (concept/course/discipline) that the learner has acquired, as well as the date and performance that the learner has when he/she is obtained this competency.

In the LearnerInformation ontology also is stored in the “goal” element (IMS LIP element 6.3.4) the current goal (educational objective/ target concept) that the learner wants to obtain in the learning.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

(**contentype** some

(**Concept**

or **Course**

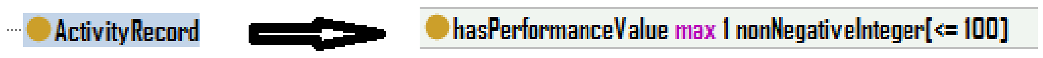
or **Discipline**))

and (**date** some **date**)

and (**vocabulary** some {"**high**" , "**low**" , "**medium**"})

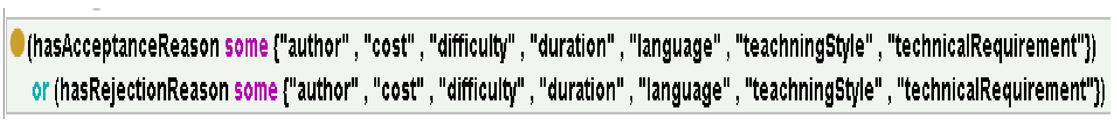
In this element we also measure in which degree (high, low, medium) the learner wants to acquire each educational objective.

On the subject of the rated knowledge that learner has, to concepts, courses, or disciplines that studied and which is the information about the performance that he/she required in these, we can take this information from the "ActivityHistory" class which is contained in the LearnerModel ontology. This class includes all "ActivityRecords" of the learner, hence, includes information about all LOs that the learner has used and the performance that he/she has acquired in each of them.


ActivityRecord

hasPerformanceValue max 1 nonNegativeInteger[<= 100]

At last, in terms of the LOs usage (of past), that is to say for what reasons the learner accepted or rejected a LO at the past we use the "ActivityRecord" class of LearnerModel ontology. This class contains the accepted and rejected reasons of a LO as well as the identification of accepted or rejected LO and the duration of this interaction.


(hasAcceptanceReason some {"author" , "cost" , "difficulty" , "duration" , "language" , "teachningStyle" , "technicalRequirement"})
or (hasRejectionReason some {"author" , "cost" , "difficulty" , "duration" , "language" , "teachningStyle" , "technicalRequirement"})

These elements are necessary for reasoning about the usage of LOs.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

On the fourth and final stage of analysis procedure for reasoning issues about the Learner, we must apply a query language to be able to acquire the right data that we need to apply reasoning with a reasoner.

5.3.1 Inferring Style Characteristics from User Actions

On this subsection, it is worth mentioning that there is an issue of inferring style characteristics from user actions. This means that must be in a position to extract conclusions about the style and other information about the Learner by actions done by him. We are aware of this problem but we don't deal in this research with such issues.

5.4 Reasoning about the Learner – Learning Object Interaction

In the last section of our discussion about reasoning issues, we will speak for reasoning about the learner – Learning Object interaction. There are a number of issues here, in which we should pay attention.

❖ <u>An important issue is matching a LO to a Learner Request</u>

In this section we will discuss identification done by an LO to a request sent by the Learner to the system. It is important to note that during this identification (matching) we should execute reasoning both for the LO and the Learner simultaneously, in order to achieve the best possible matching. Therefore, our goal here is to show on what occasions it is required to do the combined reasoning, meaning the need to know something both about the Learner and the LO, and what we should do with this combined information.

Before we start the analysis of these issues, let's take a look at what happens to another important issue, the request sent by the Learner. It is important to note that the Learner's request is not a simple string search based on the Learner's request string. The learner's string must be matched to an ontology concept. Of course, some may find many alternative ways to deal with the Learner's request, but we will only concentrate on this one below.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

At first the Learner submits a request for an educational objective which can either be a concept, a course, or a discipline. We wouldn't want the system to handle this educational objective as simple words used in a simple string search to match objects, but instead we want to give the name of a concept, a name which is included inside an ontology. For example, if the Learner sends a request for OWL language basics, the system should not simply search inside pages that use this word, but instead it should look inside a knowledge base (Domain Ontology) to find this/those ontology/ontologies that include this concept. In that case the Learner must know the full path of the concept's location plus the ontology that it belongs to, so in essence it will know the taxon and taxon path sub-elements of the LOM. This means that the Learner will know the ontology, in the sense that he will be given the opportunity to define it. The system behind the Learner will know what he defined so it will return a tree of options which includes what he would like to learn.

It is a composite scenario regarding the Learner's request and we can safely conclude that it is not so handy.

In the sequel, beginning with a Learner that sends a request for an educational objective, a raised issue regarding the appropriate LOs returned has to do with its perquisites. The system sends a response to the Learner that includes some LOs, and for each of them the Learner receives a list of its prerequisite concepts. In the following a Profile Manager searches in the student model to find-out whether the prerequisites are known to the learner. At this point there are two possible cases:

- If all prerequisites are known to the learner then, filter the list of LO's using other criteria (we see these later on below paragraph).
- If some prerequisite is not known to the learner, then notify the Learning Service (the Service may either ignore the notification or decide to compensate for the problem by constructing a larger LO, one that teaches the prerequisites of the first LO – the new LO will have its own prerequisites and the Broker needs to find out if these are known to the learner etc).

In addition, except from the prerequisites, a number of other criteria such as cost, typical duration, author, language etc should be taken under consideration in order to achieve the right matching. This information today can be retrieved either by a

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

question form completed by the user during his registration, or by directly asking the user for information. These methods, however, do not always have the desired results as it is not sure that they will be sufficient to cover all the needed information gaps that the system might have. Because of this, we need to find a way to infer some crucial needed information by interacting with the system. For example, the system should be in a position to infer that the Learner does not want an LO whose duration is 8 hours but instead lasts only 4 hours, enough to give him a general picture of the subject. The system should be able to do that by doing the appropriate reasoning both at the LO and the Learner simultaneously. Or, it should give the Learner only LOs written in Greek, if the Learner has declared that he only knows Greek. Therefore, there is a plethora of information that concerns some generic Learner features, and by doing the appropriate inference, they can correctly return the right LOs to the Learner.

Another crucial issue is matching a Learning Object to a Learner according to style information. In the previous section 4.2 we discussed the issue of inferring style from behavior. If style information is available, then the correlations developed in the previous section 4.2 can be re-used here to establish criteria for matching an LO to a learner style. For example, we discussed that the Indicative Behavior for the Concrete style designator is as follows

- tends to prefer examples (of visits/requests and time spent on examples),
- Likes to solve problems based on *standard* procedures (*how do you detect that??*) – indicated by a high interest in examples and solved problems (in order to get to grips with the approach/method/procedure),
- likes to take tests (in order to check understanding),
- tend to be more practical and careful,
- do not like courses that have no apparent connection to the real world.

Therefore, the LO can be searched (i.e. its LOM can be searched) to find out if it does indeed contain plenty of examples and related problems and tests).

❖ <u>Reasoning about the Learner and adaptive response at run time</u>

Finally, there is an issue that has to do with reasoning about the Learner and the process of adaptation at run time. At run time, the model gets updated (by the Learner Profile Manager, who receives the relevant information from the Learning Monitor).

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

There are many ways at this point that can trigger the need for reasoning during execution of the LO. Some triggering examples for this need are the following:

- The educational process of LO execution by the Learner is controlled by a “monitor” (controls interaction between LO and Learner), which checks for wrong results or deficiencies (in specific events or series of events), that have to deal with the Learner’s progress. Because of these side-results, a need to review the teaching process of a concept is raised. For example, the Learner’s low score upon completion of the teaching material is a case that requires correction. Supposing that the Learner answers all the questions on the tests given to him by the LO and does very badly. He then uses the additional teaching material but the results remain as bad as his performance is still under the acceptable level set by the LO’s author. Upon such “observation”, the monitor (KOrder) will send a modification message to the service in the form of a report. This report would be a message in a specific format. There are many different formats for messages that the KOrder can send to the service, depending on the event that took place or caused the need to send the message to the service. Reports are written in a specific language. A system has got a message format which includes information for the problem, claiming for example that in that concept, in that LO, after X attempts, the performance is X% and the teaching material has been exhausted. Upon receiving this message, the service’s response would be to provide additional material. The service must send an adaptive response which will adapt the existing running LO to the new formed situation. The question is: what kind of material would that additional material sent to the Learner be? This is the point where the system must do the appropriate reasoning for user information, to gather and process important information to return the right additional teaching material (LO). It is required to gather information from the Learner about the way he learns (i.e. give him abstract definitions, examples, etc.), or what kind of material to give (visual, audio, textual, etc.). All this, is information that requires evaluation by the service to find out what kind of new material it should offer to the Learner. Another useful source of information is the action sequence ran by the Learner while using the LO. For example, did he often ask for examples, or bypass the general explanations? This information allows us to

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

know the type of material chosen by the Learner. It is very useful because even if we are not using style information, the action sequences help us draw useful conclusions.

- Each LO has a plurality of execution models. This means that there are many predefined ways running through it. So the LO starts its execution but at run time this execution path might need to be reconstructed. For example we begin with a completely interactive model which, before each action, gives us some options and asks us to choose from them, or we declare that we want the execution model to make the choices on behalf of ours. If we always choose the option of the execution model choosing what's best, it is obvious that we don't need the interactive model. We go straight to the default execution model of the system and that chooses for us. We could draw some useful conclusions about the style of Learner from that fact.
- Furthermore, by observing the performance of other students in that specific LO, the service could draw some useful conclusions about the LO itself. For example, if the performance of many students is low, it could mean that the LO is not that competitive in the market, so certain changes must be made to it. The question here is: what kind of changes should be made? The answer lies on the action sequences the Learners did, the requests they submitted for execution and several others.
- In addition, by observing (reasoning on the recorded actions) the user request actions and availability of the type of requested material, future requests may be anticipated that cannot be accommodated by the running LO and an LO update process may be triggered. For example, we have an LO with 10 registered Learners. We observe requests from all of them and if all ten Learners request for the same thing then it means that even more Learners will ask for it. It is high probable for that to happen but not certain.

Also on this section, the last stage of analysis involves using a query language in order to gain data and use it for reasoning.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 6:

Conclusions and Further Research

We have investigated in this research issues relating to Student Modeling in a Semantic Web Setting. The underlying conception we have adopted is that of a CROP Learning Domain, where Semantic Learning Services (the owners of Learning Objects) provide services to Learners. This approach essentially implies that what needed to be investigated was a notion of a *global student model*, capturing information about a number of distinct interactions of the Learner with Learning Objects.

The next issue we had to deal with in this current research was developing a *student model* component which is not a student model specially designed for a specific Learning Object as usual in current application, but instead to create global *student model.* This fact raises some problems not only about how to update to the specific information such as performance, rate, etc., but also the extension of the student model as the Learner goes through the learning process. Also, another given point is (in the case of our research) the fact that data acquired during interaction of the Learner with other Learning Objects can be used to extract some useful conclusions. These conclusions can be of help when choosing a new Learning Object for an Object the Learner has supposedly requested, or in case we want to modify an existing Learning Object to better match the Learner’s needs.

Therefore, in order to achieve in creating these functions (best possible utilization of the student model to enable the supporting of such functions) we need to take some actions. These actions were to determine what kind of information the *student model* should include so that it would be able to support the functions we desire. The data which should be included in our student model are in excess of the data of the Learner Information Package (LIP) according to IMS. This happens because although our student model includes 100% of data according to IMS LIP, it also includes some additional data needed to service our needs.

This additional data could be, for example, the Model Graph concept which is inserted so as to clearly show the co-relations between concepts, the prerequisite relations that

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

show dependency relations between concepts, the ActivityRecord concept which includes records with explanations as to the reason why an offer was rejected (or accepted), e.g. preference (or lack thereof) for a specific author, rating of the proposed material (offered by some Rating Service), cost, expected duration etc.

Also, another additional and very important data to record to our student model is the user’s behavior. Recording of this information happens ontologically by creating the LearningBehavior ontology. This ontology registers the Learner’s interactive behaviors, meaning the order in which he chose to run the objects, what kind of actions did he take etch. The goal of this registration was to come to some conclusions about special preferences of each Learner so as to provide him with appropriate Learning Objects that correspond to his behaviors. Also, this information (Learning Behavior) can be used extract some useful conclusions about the Learning Style of the Learner as well. Of course, in our current research we did not deal with this subject, as it is outside the scope of our research. Towards that direction though, we only stored several Style indices in Learning Style ontology. A further research on conclusions correlations (behaviors on specific style indices) remains to be done.

Therefore, what we did on the first stage was to trace the kind of information to register on our *student model*. At this point, it is necessary to stress the fact that we did not just register the existing data from the IMS LIP specification, but also add extra information, so that we can safely talk about a *global student model*. On the first stage, however, we only deal with the *student model*’s static structure.

On the second phase, taking advantage of information gathered from the previous stage, we researched several reasoning issues for Learning Objects, the Learner and their interactions. However, in order for the reasoning issue to be raised, all this information about Learning Objects, Learner and their interactions has to be represented in a format that supports reasoning. And speaking about Semantic Web, this form is the ontologies. Therefore we realized ontological knowledge representation by developing the corresponding ontologies of the Learning Object, the Learning Object Metadata’s and the Learner’s Model’s etc. Of course, development of the ontologies was based on current research of [41]. In chapter 5 we set forth all the reasoning issues we dealt with and analyzed.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

A late issue that was raised but we did not have enough time to study further as that of the query languages that we have to use to be able to recover information from the ontologies. In fact, the querying process lies before that of reasoning.

Another important matter that we did not have time to deal with was implementation of the proposed research/theory. It still remains then for someone to implement a Learning Domain with a Learning Object based on CROP, to take advantage of the student model's structure according to the specifications that we proposed.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

REFERENCES

CHAPTER 1

[1] C. Hartonas and E. Gana. Learning Objects and Learning Services in the Semantic Web. In Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies, pages 584/586. IEEE Computer Society, 2008.

[2] Wikipedia. Learning Object. [Online] Available at: http://en.wikipedia.org/wiki/Learning_object. [Accessed 01 05-2010].

[3] McGreal, R., Ed. (2004). Online Education Using Learning Objects. Open and Distance Learning Series. London, Routledge/Falmer. Page 36.

[4] U. D. Ehlers *&* J. M. Pawlowski (Eds.), (2006) Handbook on quality and standardization in e-learning (pp. 109 124). Berlin, Heidelberg and *New* York*:* Springer.

[5] M. Tsiakmaki, E. Gana, C. Hartonas, Adaptation as an Emergent Property of Learning Service Composition in the Semantic Web (in greek), Proceedings of the 2nd Conference on Computer Assisted Education, Patras, Greece, 2011.

[6] A.P. Ayala. Student Modeling Based on Ontologies. First Asian Conference on Intelligent Information and Database Systems, pages 392/397. IEEE 2009.

[7] Zhou, Y. 2000. Building a new student model to support adaptive tutoring in a natural language dialogue system. Unpublished Master’s Thesis.

[8] Vladan Devedžić (2006). Semantic Web and education Integrated series in information systems Springer eBooks collection: Computer science, Springer.

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

CHAPTER 2

[9] Brusilovsky, P. and Weber, G. (1996) Collaborative example selection in an intelligent example-based programming environment. In: D. C. Edelson and E. A. Domeshek. Proceedings of International Conference on Learning Sciences, ICLS'96, Evanston, IL, USA, AACE, pp. 357-362.

[10] Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. Kunstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, 4, 19-25.

[11] Seridi H., Sellami M. (2001). Design of an intelligent tutoring system on the www to support interactive learning". International Conference on Engineering Education. August 6 - 10, 2001 Oslo, Norway. http://cblis.utc.sk/cblis-cd-old/2001/text/doc/b6.doc.

[12] Retalis S., Papasalouros A., Skordalakis E., (2003) "Formalizing the design process of web-based adaptive educational applications using an object oriented design model", proceedings of AH2003: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, at the ACM Hypertext Conference, Nottingham, UK, 26 August 2003. http://wwwis.win.tue.nl/ah2003/proceedings/paper16.pdf.

[13] Brusilovsky, P., Schwarz, E., & Weber, G. (1998). Web-based Education for All: A Tool for Development Adaptive Courseware. Published in: Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998) pp 291-300.

[14] Brusilovsky, P., A. Kobsa, and J. Vassileva: (1998), "Adaptive Hypermedia and Hypertext". Kluwer Academic Publishers.

[15] Brusilovsky, P. 1994. "ISIS-Tutor: An Intelligent Learning Environment for CDS/ISIS Users." In Proceedings of the Interdisciplinary Workshop on Complex Learning in Computer Environments (CLCE'94), Joensuu, Finland, pp 29-33. http://www.cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

[16] [Online] Available at: http://www.cs.mdx.ac.uk/staffpages/serengul/Overlay.student.models.htm [Accessed 01-02-2011].

[17] [Online] Available at: http://www.cs.mdx.ac.uk/staffpages/serengul/Differential.student.models.htm[Accessed 01-02-2011].

[18] Πρέντζας Δ. & Χατζηλυγερούδης Ι,(2001). Προσαρμοστικά Εκπαιδευτικά Υπερμέσα: Αρχές και Υπηρεσίες, 1ο Πανελλήνιο Συνέδριο στην Ανοικτή και Εξ' Αποστάσεως Εκπαίδευση, Πάτρα, Μάιος 2001. http://www.aegean.gr/culturaltec/c_karagiannidis/AEH/prentzas2001.pdf

[19] Virvou, M., Tsiriga, V. & Moundridou, M. (2001): Adaptive navigation support in a web-based software engineering course. In: Spyrou, C. (ed.): Proceedings of the second International Conference on Technology in Teaching and Learning in Higher Education, National and Kapodistrian University of Athens, Athens, pp 333-338. http://education.aspete.gr/users/mariam/papers/SoftEng.pdf

[20] Judy Kay (2006): Stereotypes, student models and scrutability, Basser Dept of Computer Science Madsen F09, University of Sydney Australia.

[21] De Bra, P., Eklund, J., Kobsa, A., Brusilovsky, P., Hall, W. (1999). Adaptive Hypermedia: Purpose, Methods and Techniques. *10th ACM Conference on Hypertext and Hypermedia*, 1999, 199-200.

[22] IMS Learner Information Package Information Model Specification v1 Revision: March 2001. [On-line] Available: http://www.usit.uio.no/prosjekter/eSU/eSU-revisjon/rapporter-referater/eksterne-rapporter/lip/lipv1/lipinfo01.pdf

CHAPTER 3

[23] Berners-Lee, T., Hendler, J. & Lassila, O. (2001). The Semantic Web. Scientific American, 284, 34-43.

[24] World Wide Web Consortium, (W3C). [Online] Available at: http://www.w3.org [Accessed 15-03-2011].

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

[25] R. Studer 1998, πρωτότυπος ορισμός από τον T. Gruber το 1993

[26] [Online] Available at:
http://i2c.engineering.utoronto.ca/I2C/Data/Lattice/Problems.aspx
[Accessed 05-01-2011].

[27] OWL Web Ontology Language Overview. [Online] Available at:
http://www.w3.org/TR/owl-features/ [Accessed 06-01-2011].

[28] OWL Reasoning Examples and Hands-On Session. [Online] Available at:
http://owl.man.ac.uk/2005/07/sssw/people.html [Accessed 04-04-2011].

[29] Eric Prud'hommeaux - W3C, Andy Seaborne - Hewlett Packard Laboratories. "SPARQL Query Language for RDF." *W3C Candidate Recommendation*, 6 April 2006.

[30] Martin O'Connor, Amar Das, SQWRL: a Query Language for OWL, Stanford Center for Biomedical Informatics Research, Proceeding of OWL: Experiences and Directions 2009 (OWLED 2009)

[31] Lehman, Rosemary (2007), *"*Learning Object Repositories*", New Directions for Adult and Continuing Education, No 113, pp. 57-66*, [Online] Available at: http://www.interscience.wiley.com

[32] Μαυρομμάτης, Γεώργιος (2005), "Δια Βίου Αναζήτηση", Ανακοίνωση στο Πανελλήνιο Επιστημονικό Συνέδριο *Νέες Τεχνολογίες στη δια Βίου Μάθηση,* ΤΕΙ Λαμίας, 16-17 Απριλίου, [Online] Available at: http://cosy.ted.unipi.gr/NTdiabiou2005/media/papers/P33.doc].

[33] IEEE LOM (2002). Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, Sponsored by the Learning Technology Standards Committee of the IEEE, 15 July 2002 [Online] Available at: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

[34] IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata [Online] Available at:
http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html

[35] Advanced Distributed Learning (ADL) (2005). About ADL. [Online] Available at: http://www.adlnet.org/aboutadl/index.cfm

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

[36] SCORM (2004), Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 Overview, 2004.

[37] SCORM (2004a). Sharable Content Object Reference Model (SCORM) 2004 2nd edition, Overview. [Online] Available: http://www.adlnet.org/scorm/index.cfm. [Accessed 08-11-2010].

[38] [Online] Available at: http://www.vsscorm.net/2009/09/30/step-32-scorm-1-2-cam-content-packaging/ [Accessed 22-01-2011].

[39] [Online] Available at: http://www.vsscorm.net/2009/09/30/step-32-scorm-1-2-cam-content-packaging/ [Accessed 08-02-2011].

[40] Britain Sandy, Liber Oleg, (1999). A Framework for Pedagogical Evaluation of Virtual Learning Environments. JTAP, JISC Technology Applications. Report 41. University of Wales – Bangor

[41] C. Hartonas, private communication

[42] Agathoklis Kritsimalis. (2010). Role Modeling for CDL Specifications of Web Services, Unpublished Master's Thesis.

[43] [Online] Available at: http://www.elena-project.org [Accessed 22-10-2010].

[44] [Online] Available at: http://www.reload.ac.uk/ [Accessed 27-10-2010].

[45] Gregoric A. (1979), "Learning /teaching styles: Their names and effects.", In J Keefe (Ed.), Student learning styles : Diagnosing and prescribing programs, pp. 19-26, VA: National Association of Secondary Schools Principals.

[46] Honey P., & Mumford A. (1992), "The manual of Learning Styles", 3rd edition, Maindenhead, Berkshire.

[47] Sampson D. & Karagiannidis C. (2002), "Accommodating Learning Styles in Adaptation Logics for Personalized Learning Systems", In Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2002, pp. 1715-1726, Chesapeake, VA: AACE.

[48] Ferber R. & Silverman L. (1988), "Learning and Teaching Styles in Engineering Education", Engineering Education Journal, 78(7), pp. 674-681

[49] Grasha A. (1996), "Teaching with style", Pittsburgh, PA: Alliance Publishers, ISBN: 0-6745071-1-0

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76

[50] Popescu, E., Trigano, P. and Badica, C., “Towards a Unified Learning Style Model in Adaptive Educational Systems”, Proc. ICALT07, 2007.

[51] C. Hartonas and E. Gana. Adaptivity for Knowledge Content in the Semantic Web, in Proc. KGCM'08, PeRLA Research Project, ARCHIMEDES II

CHAPTER 4

[52] Curry, L. (1983). “An Organization of Learning Styles Theory and Constructs”, ERIC Document Reproduction Service No ED 235 185.

CHAPTER 5

[53] Sabine Graf, Kinshuk, Tzu-Chien Liu, Identifying Learning Styles in Learning Management Systems by Using Indications from Students’ Behavior, Eighth IEEE International Conference on Advanced Learning Technologies 2008 IEEE

Institutional Repository - Library & Information Centre - University of Thessaly
03/09/2024 08:15:35 EEST - 3.135.192.76