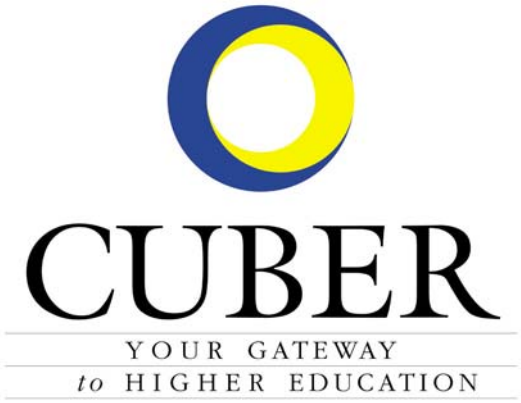

PERSONALISED CURRICULUM BUILDER IN THE FEDERATED VIRTUAL

UNIVERSITY OF THE EUROPE OF REGIONS

Contract No: IST-1999-10737

FINAL VERSION OF METADATA SPECIFICATION

Deliverable D3.3

Authors:
Päivi Nykänen and Kirsi Pelto-Aho
University of Helsinki (Finland)

The CUBER project is supported by the European Community in the IST Programme of the Fifth Framework. Nevertheless the content of this document does not represent the opinion of the European Community, nor is the European Community responsible for any use that might be made of data appearing therein.

Administrative Data Summary

Project Number:	IST-1999-10737
Project Title	CUBER – Personalised Curriculum Builder in the Federated Virtual University of the Europe of Regions
Deliverable Type:	Public
Deliverable Number:	D3.3
Due Date of Deliverable:	24 October 2001
Title of Deliverable:	Final version of metadata specification
Contributing Workpackage(s):	WP 3
Nature of Deliverable:	Report
Author(s):	Päivi Nykänen and Kirsi Pelto-Aho, University of Helsinki (Finland) in collaboration with Juha Puustjärvi, Helsinki University of Technology (Finland)

Abstract	The report includes the final metadata specification of CUBER. This specification consists of five separate schemas: the metadata schema itself, an ontology for the metadata, a conceptual model of the metadata in UML, a database schema of the metadata in SQL, and a XML schema with a DTD of the metadata. A short report of the metadata specification precedes the schemas. During the process of defining the metadata specification, the previous metadata specification of CUBER was studied and evaluated, and information was gathered from other relevant metadata projects in order to gain experience from using educational metadata. In addition, the requirements and the information needs of the other CUBER partners were investigated and considered. Furthermore, the demands of the IEEE's Learning Object Metadata (LOM) standard were taken into consideration. This metadata specification is based on the LOM, but CUBER has introduced some extensions and changes to the original LOM schema. Nevertheless, special attention was paid to making the CUBER metadata conform to the LOM standard for the interoperability issues. This report provides the metadata specification for the implementers of CUBER system. The report also provides the implementers with an SQL schema and an XML/DTD schema that can be utilised in the implementation of the metadata in the CUBER system.
Key words:	Metadata, IEEE/LOM, educational standards, ontology, conceptual model, UML, database schema, SQL, XML schema, DTD

Distribution List:	CUBER Project BSCW system, European Commission
Issue	1.2
Reference:	D3.3metadata.doc
Total Number of Pages:	137

TABLE OF CONTENTS

1 Introduction

1.1 Scope of the document

The aim of the deliverable 3.3 is to define the final metadata specification for CUBER. The report contains the following specifications and schemas as appendices:

- Metadata specification 2.2 (MDS 2.2)
- Conceptual model of metadata in UML
- Ontology for the metadata
- Relational database schema in SQL
- XML/DTD

The structure of this report follows the progress of the work on work package 3 (WP3) at the University of Helsinki (UH). This report consists of 6 chapters. The first chapter outlines the content of this report, and the second and third chapters concentrate on metadata. Chapters 4 and 5 deal with the ontology and the conceptual model of CUBER metadata. Chapter 6 deals with the implementation of metadata as a relational database and as an XML schema. The metadata specification, ontology, the class diagrams, SQL definition, and XML schemas will be found in the Appendices section.

1.2 The General Architecture of CUBER

Technically, CUBER consists of three main components: a Knowledge Base for standardised course descriptions and domain knowledge, a Search Engine for finding the courses and generating study packages, and an Authoring Interface for entering and maintaining course metadata. The Knowledge Base includes a lexical database, where the standardised vocabularies and classifications are entered. The Knowledge Base also includes standardised metadata and an ontology which defines the semantics of the metadata and the vocabularies used in the system.

CUBER is designed to become a broker system that supports the search for study courses from European universities. CUBER facilitates the access of various kinds of learners to a vast collection of higher education courses offered by European course providers, in particular by distance teaching universities.

Although CUBER has been planned to offer different learners study courses that can be studied on the Internet, other kinds of courses can be integrated into the system as well. As CUBER provides learners with a possibility to compare courses from different providers, it enables them to find the matches that correspond best to their the personal educational aims. CUBER will also take learner's previous knowledge, qualification objectives, and schedule constraints into account. CUBER gives each learner the possibility to generate a complete curriculum plan by offering information on how the courses can be combined, and whether they lead to a degree or certificate. Offering information on the recognition and international approval of the degrees or certificates is also one of the goals of the CUBER system.

CUBER has decided to use metadata to serve the purpose of facilitating search, comparison and selection of study courses, packages and programmes offered by the course providers, i.e. universities and other higher education institutions. Thus metadata is of central significance within the CUBER project, providing the system with a basis for many of its services. The CUBER system also includes elements of an ontology that defines the common semantics of the metadata. However it should be noted that the ontology of the CUBER metadata provides only a partial semantic for the specific purposes of the CUBER system, not a full-scale semantic system.

1.3 The Metadata Specification Process

In spring 2001, the entire WP3 team at the University of Helsinki was re-established. The new leader of WP3, BA (Educ.) Päivi Nykänen, started her work in March, and in May Kirsi Pelto-Aho, a researcher and a PhD student from the Helsinki University of Technology (HUT), joined the team. Her

supervisor at HUT, PhD Juha Puustjärvi from HUT, gave the WP3 team valuable guidance and advice during the metadata specification process. All members of the team were involved with all aspects of the metadata specification work.

WP 3 has been defining a new version of the metadata specification for the CUBER project since March 2001. During this definition process, CUBER's previous metadata specification 1.1 was studied and evaluated in depth. In addition, other relevant metadata developments and projects were reviewed in order to gain ideas and experiences in using educational metadata. Other requirements and information needs of Course Providers, Search Engine and Knowledge Base of CUBER have been investigated and considered. Furthermore, the demands of the emerging Learning Object Metadata (LOM) have been taken into consideration. Since CUBER metadata specification is an application profile of LOM, the WP3 team had to define the necessary changes to LOM in order to adapt it to the needs of CUBER. The primary challenge in defining the metadata has been the attempt to find a compromise between the various and partly conflicting requirements and expectations.

In May 2001, a new version of the CUBER metadata specification (MDS 2.0) was released. Most partners of CUBER (including WP4, WP5, WP7, and WP9) evaluated this metadata specification, and on this basis the next metadata specification (MDS 2.1) was developed. MDS 2.1 was released in the beginning of August 2001. That metadata specification introduced a significant improvement compared to the previous versions, as it provided Conceptual Models of metadata in UML, and an Ontology. Since the release of MDS 2.1, WP3 has defined a XML/DTD and a relational database schema in SQL for the implementation of metadata, in addition to some other improvements.

2 The Background and Purpose of Metadata in CUBER

2.1 Definition of Metadata

Put simply, *metadata* is descriptive and classifying *information about an object*. It describes certain important characteristics of its target. Metadata plays a central role in improving searching and categorising objects within a defined context of use. In order to be able to use metadata efficiently across different contexts and systems, the metadata scheme should be standardised. There is a growing interest in using metadata in the field of education, too. [Britain et al.]. Metadata can also be defined as *"data about data"*. In this sense metadata describes a data set and the format of this data. In addition, metadata can be described by a set of *meta-metadata*. Meta-metadata is descriptive information on the metadata record itself. [Wason]

Educational metadata may describe any class of educational objects, such as study courses. The pedagogical features of the course, the contents, special target groups, and the technical requirements of the study course can be described with the help of a metadata schema. Educational metadata can be utilised by educational and pedagogical professionals, by the institutions offering education, and by the students searching for education. Well-designed and sufficient metadata aid the decision making process of the students and help the educational institutions to provide suitable information about their educational offerings. [Lamminaho] A more detailed analysis of the concept of metadata was provided by the WP3 in the Deliverable D3.1 in June 2000.

2.2 Metadata in CUBER

The purpose of metadata in CUBER is to facilitate search, comparison and selection of study courses, packages and programmes offered by the course providers, i.e. universities and other educational institutions. The goals of CUBER are, in short:

- To make it possible to compare courses from different providers and to find the best matches in view of one's personal educational goals
- To take learner's previous knowledge into account
- To take learner's qualification objectives into account
- To take learner's schedule constraints into account
- To make it possible to generate a complete curriculum plan
- To offer courses that can be studied on the Internet, however other kind of courses can be integrated to the system as well
- To provide information on how the courses are integrated together
- To provide information on whether the studies lead to some kind of degree or certificate
- To provide information on the recognition and international approval of the degrees or certificates
- To take various target groups into account
- To facilitate searching according to personal preferences and keywords
- To enable the learner to create an own user profile and to save it for later searches

The goals of CUBER have direct consequences for the metadata. It is possible to deduce most of the metadata elements needed by exploring the goals and aims of CUBER. The following questions have been considered when specifying metadata in order to take the goals of CUBER into account:

- Which elements will be used for describing the courses? What elements do the course providers need to insert into the system to make the courses distinguishable?
- Which elements will be used for searching the courses / packages / programmes?
- Which metadata elements are needed

- to generate packages and programmes?
- to generate whole curriculum plans?
- to compare the courses?
- to tell which study elements are connected together and in what way?
- to take the learner's educational needs and previous education into account?
- to take the schedule constraints of the learner into account?
- to take different target groups and their needs into account?
- to take learner's educational aims and interests into account?
- to generate user profiles?

3 Learning Object Metadata and CUBER metadata

CUBER metadata has been designed to be an application profile of IEEE's Learning Object Metadata Standard 6.1 (LOM). Some extensions were made, for LOM was insufficient for the specific purposes of CUBER. The original LOM metadata elements have not been replaced or abused. They have been taken into CUBER as they were defined in the LOM Draft 6.1. However, not all of the LOM metadata elements are used in CUBER, because they were of little significance for the goals of CUBER.

All the metadata categories, metadata elements and sub-elements adopted from LOM 6.1 are used as such; they have not been changed because of the notes of conformance in LOM. According to LOM, there can be extension elements, but none of the LOM elements or sub-elements can be replaced or transformed in some way. In other words, the LOM metadata allow extensions, but only if the original LOM elements are retained as they were defined. The only exception is the possibility to use other values in the Value space than the values defined in the Vocabulary of the Data type of that metadata element. [IEEE/LTSC]

3.1 Overview of the Metadata model

The metadata model of CUBER consists of nine categories that contain metadata elements. The metadata categories do not carry any information. Instead, they function only as rubrics under which the related metadata elements are gathered. The metadata elements, for their part, can carry information as values of the Value space. They can also include sub-elements that carry more detailed information about the study elements described. In this latter case, the metadata elements containing sub-elements cannot themselves have values directly. The sub-elements carry the values in question, and the metadata elements themselves can have their values only indirectly through their sub-elements. [IEEE/LTSC]

The metadata schema is organised hierarchically in the form of a tree. The metadata categories are placed in the top of the hierarchy, and the metadata elements are right below them. One category can have several metadata elements. The metadata sub-elements are subordinated to the metadata elements, and one metadata element can have one or several sub-elements. [IEEE/LTSC]

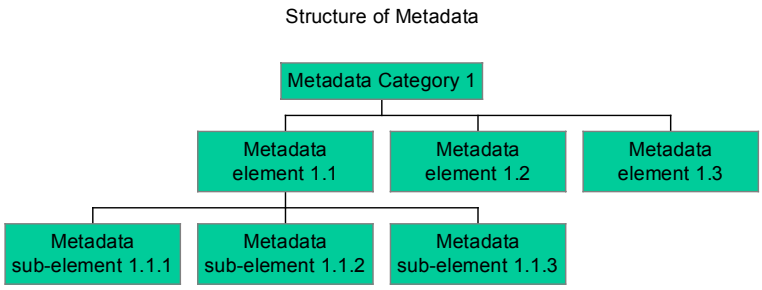

Figure 1 The structure of CUBER metadata

3.2 Metadata scheme overview: Metadata Categories

There are nine categories in LOM and they all are used in CUBER metadata. The metadata categories of LOM [IEEE/LTSC] and CUBER are presented in Table 1.

Table 1 Categories of CUBER metadata

1. General	The general information that describes the study elements of CUBER as a whole.
2. Lifecycle	The features that describe the history and current state of the study element. Information about the contributors.
3. Meta-metadata	Information on this metadata record itself.
4. Technical	The technical characteristics and requirements for use of the study element.
5. Educational	The educational and pedagogical characteristics of the study element.
6. Rights	The intellectual property rights and the conditions of use for the study element.
7. Relation	Defines the relationships between given study element and other targeted study elements.
8. Annotation	Provides comments on the educational use of the study element. Information on the commentators and the comments themselves.
9. Classification	Describes where the study element falls within a particular classification system.

3.3 The CUBER Metadata Extensions

There are three kinds of *metadata elements* in CUBER metadata. 1) LOM metadata elements, 2) LOM metadata elements with alternative vocabulary, and 3) CUBER extension metadata elements. All *metadata categories* defined in LOM 6.1 are used in CUBER, but some of the metadata elements defined in LOM 6.1 were considered to be of little or no relevance for the purposes of CUBER, and they have not been used. They have not been removed or transformed due to the notes of compatibility in LOM 6.1. Instead, these LOM elements will be left empty in CUBER metadata when exporting or importing data.

The *CUBER extensions* to the LOM metadata have been added as complementary elements in order to be able to describe all the characteristics of the CUBER study elements and to be able to provide the functions of the CUBER system. The extensions include also additions to the vocabularies of LOM. According to LOM the vocabularies can be extended, although it is strongly recommended to use the vocabularies defined in LOM. However, the metadata elements and their meanings and attributes cannot be changed. There

are three ways to adapt the metadata schema to particular needs: 1) using Classification systems in Category 9 Classification, 2) creating extensions to the metadata schema that do not override the original LOM elements, and 3) changing the vocabulary used in the LOM elements. [IEEE/LTSC] All these methods have been used in the metadata specification of CUBER. The CUBER extensions have been added to the LOM Base Scheme as independent metadata elements and sub-elements. In addition, vocabularies have been modified and added to meet the goals of CUBER, but the vocabularies of LOM have been utilised wherever possible to ensure as high a level of interoperability and conformity as possible.

The CUBER system has decided to use the data type Langstring in two metadata elements: 1.2 General.Title and 1.5 General.Description. This means that these two metadata elements may contain information with the same contents in several languages. The information of the metadata element is two-piece. In the beginning of the information there is a sign that specifies the language used. This sign is followed by the actual information that is presented in the language specified by the sign.

> For example, 1.2 General.Title can contain (en, C++ Programming) (fi, C++ ohjelmointi), which means that the first title is in English and the second title is in Finnish. The abbreviations “en” and “fi” stand for the languages English and Finnish. The second part of the information in the metadata element contains the actual title of the study element in the chosen language.

Because of the decision to implement Langstring in the CUBER system, there is no need to define a separate metadata elements for the title in English and the title in the original language.

The aggregation levels of CUBER have been defined in a *separate* metadata element that is different from the 1.9 General.Aggregation of LOM, which should help CUBER avoid conflicts with LOM. The CUBER system will use the aggregations defined in the metadata element 1.11 General.CUBER_Aggregation for the internal purposes, and the aggregations defined in LOM will be used for transferring data between CUBER and other

systems. A mapping between the two aggregation elements will be provided in chapter 3.4 of this report.

In order to facilitate internal identification of the study elements within the CUBER system, a CUBER Identifier (MD element 1.10 General.CUBER_Identifier) was introduced. An unequivocal identifier for the study elements of CUBER is needed in order to make the study elements distinguishable and in order to make the relations among the study elements. The solution is to use a CUBER identifier for the internal purposes of the system. This identifier will be used for defining the targets of the relations in the metadata element 7.2.3 Relation.CatalogEntry.

The vocabulary of the metadata sub-element 2.3.1 LifeCycle.Contribute.Role has been extended in order to meet the needs of the CUBER system. The vocabulary of the metadata element consists of two lists that will be used at different aggregation levels. The first list that is defined in LOM standard will be used only at the Material level. The second list that is defined in CUBER will be used at the other aggregation levels (Course, Package and Programme).

The date defined in LOM 2.3.3 is not used in CUBER because of the need to express the begin and end dates of the study elements, which would not have been possible with the LOM 2.3.3 Date. The CUBER extension elements include information the important dates (MD element 2.5 LifeCycle.Date with sub-elements Begin, End and Kind) of and information on the recurrence of the study elements (MD element 2.4 LifeCycle.Recurrence) are also added to the schema.

Two additional metadata elements have been added to the category 4 Technical; 4.8 Technical.Material_Size will be used to describe the size of any kind of study material, and 4.9 Technical.Description will be used to describe the technical characteristics of any study element. The metadata element 4.2 Technical.Size will not be used in CUBER because this element allows to

express only the digital size of a material, which does not serve the needs of CUBER.

The greatest number of changes were made within metadata Category 5 Educational, because of the special need of CUBER to describe the educational characteristics of the study elements. First of all, the vocabulary of MD element 5.6 Educational.Context was changed to meet CUBER's needs. The teaching activities, examination, ECTS-credits, study guidance, enrolment, dedication to studying, pre-requisites, and related official degrees can be described with the additional metadata elements. In addition, the cost of the study element in EURO (MD element 6.4 Rights.Cost_in_EURO), and the possibilities for the student to obtain financial aid (MD element 6.5 Rights.Financing) can be announced. Metadata Category 7 Relation was also extended by adding one element, 7.3 Relations.Dependencies. This element can be used to describe the dependencies among the study elements and the conditions under which they can be combined into a larger entity. This description will be given as free-text input, instead of using relations, in an attempt to avoid creating too complicated an information system that would be unnecessarily troublesome to maintain and update.

3.4 Aggregation levels

CUBER has decided to have four aggregation levels, viz. study material, study course, study package and study programme. Study material is the smallest unit in CUBER, followed in the hierarchy by study course. After study package, the largest unit in CUBER is the study programme. These aggregations are used in CUBER because *the study elements* to be described in CUBER differ from each other in their scope, activities, purposes, and other significant characteristics. The aggregation levels are used to describe the differences and the relationships between the study elements in CUBER. Aggregation levels are organised hierarchically, and they have both common and individual features. This classification of the aggregation levels intends to describe the existing classification system in higher education, though there are significant discrepancies among the European countries involved in CUBER.

The *descriptions* of the study elements in the different aggregation levels vary according to the individual level. The metadata schema offers a possibility to use free-text descriptions in some of its elements, and the level of abstraction in these descriptions should be compatible with the aggregation level of the study element. For example, the description of the contents of a course must be more detailed and concrete than the description of programme's contents. The description is most detailed in courses and materials. The level of generalisation is intermediate in packages, and the programmes are described in general terms. The description of learning targets and outcomes, teaching methods, and examinations moves gradually from concrete and specific towards abstract and general according to the level of aggregation. The aggregation levels may also have some specific metadata elements with specific vocabulary that differs from those of the other aggregation levels. More detailed instructions on the use of metadata elements and descriptions can be found in Appendices 1 and 2 (Metadata Specification and Ontology).

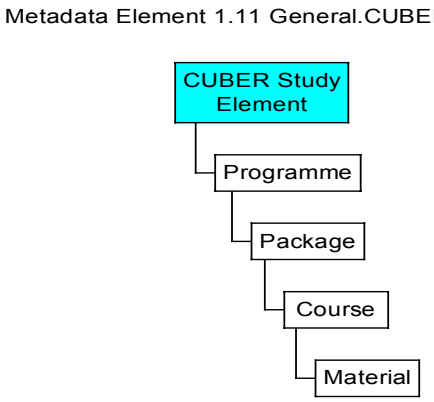

Figure 2 Metadata Element 1.11 General.CUBER_Aggregation with Ontology

Table 2 Ontology of CUBER_Aggregation

Word	Meaning
Material	The term study **material** refers to any kind of resource (digital or non-digital) that contains information used in education. Study **material** includes no teaching activity; its function is to serve as information source for a study course.
Course	The term study **course** refers to a complete unit of instruction that provides the learners with the knowledge or skills required for competence in a subject matter. A study **course** is any academic or vocational course arranged by a course provider. This is the lowest level that can offer credits or recognition within an educational institution. Study **course** usually includes teaching activity and examination.
Package	The term study **package** refers to a collection of study courses. Study packages can offer credits but no official degrees nor certificates, i.e. a study **package** has internally visible outcomes. A study **package** can be part of a study programme.
Programme	The term study **programme** refers to a collection of study courses and/or study packages, and it can lead to an official university degree or a certificate of competence, i.e. a study **programme** has externally visible outcomes.

CUBER 1.11 metadata element CUBER_Aggregation is introduced because the aggregations of LOM are slightly different from the aggregations defined in CUBER. First, the level of abstraction is more detailed in LOM than in CUBER. LOM aggregations are much more atomic, when compared to the aggregations of CUBER. Second, the content and context of the CUBER_Aggregations is different from those of the aggregations of LOM. Instead, CUBER needs to describe its Learning objects or Study elements in a much wider context that enables the description of larger Study elements, such as study courses, packages and programmes. This context of description must also include the specific nature and characteristics (such as examinations, counselling etc.) of these Study elements. That is why CUBER metadata defines its own aggregation levels. The main functionality of the CUBER system is based on the aggregation levels defined in CUBER, and it would be very problematic to integrate the aggregations of CUBER into the aggregations of LOM. Thus the

aggregation levels that are defined in the LOM standard will not be used internally in the CUBER system. Instead, the CUBER system will use the aggregations defined in the metadata element 1.11 General.CUBER_Aggregation for the internal purposes, and the aggregations defined in LOM will be used for transferring data between CUBER and other systems.

In the case of importing LOM-conformant data from outside into the CUBER system, the aggregations of LOM (1.9) will be mapped into the aggregations of CUBER (1.11). Accordingly, in the case of exporting CUBER data into a system that conforms to LOM standard, the CUBER_Aggregations (1.11) will be mapped into the aggregations of LOM (1.9). This is how data can be exchanged between CUBER and other systems that use LOM standard. All the information that is transferred can be handled, and the interoperability of the systems will be ensured. This mapping also improves the conformance of CUBER metadata to the LOM standard.

Table 3 The mapping of CUBER_Aggregations and LOM aggregations

1.11 CUBER_Aggregation	**1.9 LOM aggregation level**
0 = Material is any kind of resource that is used in education as a source of information.	2 = a collection of atoms, e.g. a document or a lesson
1 = Course is a unit of instruction that offers credits.	3 = a collection of level 2 learning objects or a course
2 = Package is a collection of courses but it does not lead to a degree.	4 = the largest level of granularity, a set of courses that lead to a certificate
3 = Programme is a collection of packages and/or courses and it leads to a degree.	4 = the largest level of granularity, a set of courses that lead to a certificate

3.4.1 Common metadata elements for the aggregation levels

Despite the fact that the Study elements on different aggregation levels differ significantly from each other, they still have some common characteristics and common metadata elements. These elements have the same meanings on each

level and they must be used coherently in order to avoid misinterpretations and dysfunction of the CUBER system. If a metadata element has a vocabulary, this vocabulary has to be used coherently on each aggregation level. The instructions for using the metadata elements with regard to the aggregation levels are defined in the MDS document's column "Aggregation levels". The more accurate definitions of the terms used in this report can be found in Appendices 1 and 2 (Metadata Specification and Ontology). The following paragraphs and chapters intend to introduce and describe the aggregation levels and their central characteristics on a general level only.

All the aggregation levels included in CUBER have a certain language. They all have a title, which can be both in the local language and in English. All the levels can be subject to charge, and the amount of the fee or payment is given in Euro. The aggregation levels may require some previous skills, knowledge or education, and CUBER will refer to these as prerequisites. The prerequisites can be academic achievements, work experience or specific skills, such as the ability to use certain software. Each aggregation level can have an intended target audience. In addition they have difficulty levels. Content is a common element too, for all levels contain some kind of information or activity. The levels are located somewhere. Material can be found e.g. in libraries or on the Web. The courses, packages and programmes are usually located in the country of the provider, albeit they do not have to be located in a certain physical environment; they can be studied virtually as distance learning. All the study elements have providers, which can be academic or other educational institutions. Providers have names and contact information, and the providers can be persons and institutions. These study elements can be joined together by using the relations defined in MDS. Relation *Has_part* defines the larger entity that the study element belongs to. Relation *Requires* defines the study elements that must have been completed successfully before entering this particular study element. The different *versions* of the study elements can be expressed by using the metadata element 2.1 LifeCycle.Version. There is also a possibility to express the version of a study course by implementing the Course_Occasion class in the Class diagram of CUBER metadata. (See Appendices 2 Ontology and 3 Class diagram.)

3.4.2 Material

The term study material refers to any kind of digital or non-digital resource that contains information used in education. [IEEE/LTSC2] Material can be attached to particular a course and it can be reused in a number of courses. Material includes no teaching activity; its function is to serve as an aid or as a source of information in a course. Study material is the lowest level in the hierarchy, and there cannot be any levels or study elements below it.

Material has to have some kind of content but the content can vary according to the intended usage of the material. Also the type of material can vary. Material can consist of e.g. books, journals, articles, or HTML pages, downloadable files, CD-ROMs, or TV programmes, videos, and DVDs. As someone has created the material, it always has an author or many authors. The author or some organisation may own the copyrights in the material, and there can be conditions or limits for using the material, such as for teaching purposes only. Material has usually been created in some kind of organisation or academic institution and the author may have an official role in that organisation, such as professor or researcher. Material can be developed; i.e. it can be revised or re-edited. Thus several versions or editions of the original material can exist.

3.4.3 Course

The term study course refers to a complete unit of instruction that provides the learners with the knowledge or skills required for competence in a subject. A course may consist of lessons and tests together with associated learning objectives. Study course can be any academic course arranged by a course provider. Study course is the lowest level capable of offering credits or recognition. Study course is also the lowest level to offer teaching activity. Nevertheless, a course cannot offer degrees or certificates, but it can be part of a study programme that offers a degree or a certificate. In addition, a course can be part of a study package. It is should be noted that course can be an independent element; it does not have to belong to any entity. The time frame or

the duration of a course can vary from days to months, but cannot exceed one semester or study year.

The learning targets are quite precise, and they can be described as specific skills or pieces of knowledge. The learning outcomes are rather narrow, which means that they must be described in a concrete and detailed manner. The learning outcomes are evaluated, usually through exams. One or more teaching methods can be used in a course, but they are more limited in number than e.g. in a programme. The teaching methods used can be described in detail or at least they can be announced more accurately than in packages or programmes.

The learner needs to enrol in a course, and he needs the enrolment dates and methods in order to be able to enrol. In addition to enrolment procedures, there may be limitations for entering the course, such as limitations to the number of participants. Furthermore, prerequisites can limit the target audience remarkably. Courses can be unique, i.e. courses can be arranged only once, however they can also be repeated periodically. If the course is repeated, the version has to be announced. Each course belongs to some discipline or at least to some more general subject matter taught at universities. The learner has to study to complete a course successfully, and the study load can be indicated in terms of hours of work required of the learner.

3.4.4 Package

The term study package refers a collection of courses. There must be more than one course in a package. A study package can offer credits or other official recognition, but it is not capable of offering any degrees or certificates. Nevertheless, it can be part of a study programme, which does lead to a certain degree or a certificate. In other words, a study package can offer internally visible outcomes.

Since there are several courses in a package, the learning targets can include many kind of specific skills and pieces of knowledge. Thus the learning targets

must be described in a more abstract manner and the description has to refer to all the main issues of the courses included in the package in question. Accordingly, the learning outcomes are described on a more general level than in courses. The description of learning outcomes must mention the skills and knowledge gained on a general level or as classified into categories.

Packages can have prerequisites, just as courses can; i.e. the package requires some previous studies, experience or skills. The teaching methods can vary remarkably, because there can be many kind of courses in one package. Thus the teaching methods cannot be described in detail but as a generalisation of the main methods. The methods of evaluation, such as exams, can vary within a study package due to the variety of courses contained in the package.

As a unit of instruction, a package is longer than a course. The duration of a package can vary from months to one year. The amount of work required of the learner is announced in hours. Enrolment is required for admission, but there can be other requirements or limits as well. The package can belong to a certain discipline, and the courses of the package can represent individual or various subject matters. The packages offered by the course providers can be arranged only once or they can be repeated. In the latter case the version of the package must be announced, e.g. the starting semester of the package.

3.4.5 Programme

The term study programme refers to a collection of courses and/or packages that gives an official degree or a certificate, i.e. an externally visible outcome. In other words, a programme always requires courses or packages as its building blocks. The study programme is the largest unit in CUBER, and there cannot be any entities above it. Study programmes have the longest time frame; they can last for several years. The amount of work required from the learner cannot be announced in hours because of the length of studying. Instead, the official or average time needed to complete the programme can be announced. The programmes can include courses in one or more disciplines and on different levels of difficulty, but the degree itself can be only in one particular discipline.

The entire programme can have its own level of difficulty, such as the Master's or doctoral level.

The learning targets are on a rather general level, and the learning outcomes are broader than in a course or in a package, and they should be described in general terms. The teaching methods can vary remarkably within a programme, for which reason the description should allow using generalised terms. Due to the length and complexity of a programme, the methods used to evaluate students can be diverse. Consequently, only the main alternatives should be mentioned here. Programmes can be offered only once, or they can be repeated. If the programme is repeatable, the versions of the programmes must be distinguishable, because the structure and the content may change from year to year.

3.5 Metadata structure

Metadata is composed of *categories and metadata elements*, which can have *sub-elements*. The structure of CUBER metadata is *hierarchical.* The elements represent the general level of description, and the sub-elements are more detailed in the information they convey. The categories exist only for grouping the related metadata elements together. [IEEE/LTSC]

3.5.1 Data elements

Each data element in the metadata schema has been defined in terms of name, explanation, size, order, value space, data type and example. These terms define the characteristics, values and purposes of use that are allowed for the metadata element.

- Name - the name by which the metadata element is referenced
- Explanation - the written definition of the metadata element
- Size - the number of values allowed for the metadata element
- Order - whether the order of the values is significant or not
- Value space - the set of allowed values, e.g. a vocabulary

- Data type - a set of distinct values, e.g. char, string
- Example - an illustrative example of the metadata element

Notice that both the Size and Data type columns in the metadata specification can include smallest permitted maximum values (explained later). Only sub-elements and metadata elements without sub-elements can carry information as values. Categories and metadata elements that include sub-elements cannot have values. These metadata elements can have values indirectly through their sub-elements. [IEEE/LTSC]

CUBER has added two extra data elements, or columns, to the original LOM schema:

- **Aggregation levels** - defines the aggregation levels on which the metadata element should be used
- **Mandatory** - defines whether it is obligatory to use this metadata element to describe the study element.

These determinants are to be used within the CUBER system only for the special purposes of CUBER.

3.5.2 Definitions

There are some terms in the metadata specification that need definition in order to be used in a coherent manner. The following list defines these terms according to LOM, and they have to be used in CUBER as described here.

- *A category* is a rubric for a group of related metadata elements gathered together. Categories cannot have values; i.e. they do not carry any information.
- *A CUBER metadata element* is a data element for which the name, explanation, size, order, value space and data type are defined. An example can also be provided. Metadata elements can have values; i.e. they can carry information.

- *A value space* defines the set of values for a given metadata element. In CUBER the value space is usually defined by referring to a given standard or vocabulary. The value space can be enumerated outright as well.
- *A Langstring* is a specific LOM data type that represents phrases in one or several human languages. Multiple semantically equivalent phrases can be included, e.g. translations and alternative descriptions.
- *A smallest permitted maximum* defines the smallest permitted maximum value the application must support for that data type.
- *A reserved data element* is a data element that is not present in data instances.
- *A taxonomy* is a hierarchy of terms arranged from general to specific. It describes and defines a particular classification system in a specific field.
- *A data type* defines a set of distinct values, characterised by the properties of those values and by the operations on those values.
- *A vocabulary* is a list of values that define the value space of a metadata element. The use of the vocabulary is recommended in order to guarantee high interoperability. [IEEE/LTSC]

3.5.3 List values

In some instances, a metadata element may contain a list of values, rather than a single value. A list of values must contain at least one value; in other words, the list cannot have a zero-length. A list of zero length cannot be distinguishable from no value, and if a value is intended to be present, a list of zero length cannot be valid as a final value. However, lists of zero length can be used for internal operations of an implementation. If a metadata element with sub-elements contains a list of values, then each of these values shall be a tuple of sub-elements. This means that the value of that metadata element is a list of pairs of the form (sub-element1, sub-element2). [IEEE/LTSC]

There can be two kinds of lists: ordered and unordered. In an ordered list the order of the values in the list is significant. For example, the more important attribute can be mentioned first in the list, and a hierarchical classification system proceeds from general towards specific. In an unordered list the order of the values bears no meaning, and values of the list can appear in any order without any loss of information. [IEEE/LTSC]

3.5.4 Vocabularies

A vocabulary is *a recommended list of appropriate values for a given metadata element.* Vocabularies have been defined for some of the CUBER metadata elements. Although the vocabulary is meant to be used as defined in the metadata schema, other values that are not present in the vocabulary list may be used as well. [IEEE/LTSC] In most cases it is preferable to use the CUBER vocabularies, for these values have the highest semantic interoperability, which ensures common understanding of the metadata. Some vocabularies of CUBER have been adopted from LOM as such, and some vocabularies have been created specifically for CUBER in order to be able to express the intended meanings and dimensions of the study elements. The vocabularies of LOM have been used whenever possible. In some cases the vocabularies of LOM had to be accompanied with extra values that will be used consistently within CUBER.

3.5.5 Smallest permitted maximum values

The intention of the smallest permitted maximum value is to cover more than 99% of all cases of the values of the data type of the metadata element. Smallest permitted maximum values can be used for two determinants of the metadata elements: size and data type. In addition, there are two cases for which the smallest permitted maximum has a slightly different meaning. First, for the metadata elements with a list value, the applications must support at least that number of entries for the list. The maximum number of entries to be supported imposed by the application must not go under the smallest permitted maximum defined for the metadata element. Second, for the metadata elements with data type Characterstring or Langstring, all applications must support at least that length for the Characterstring value, either directly or contained in the Langstring. The maximum number of characters to be supported must not go

under the smallest permitted maximum for the data type of that metadata element. [IEEE/LTSC]

3.5.6 Character sets

The aims of LOM and CUBER are to define a conceptual structure for metadata related to learning objects and education. The metadata scheme specifies some external standards to which any Characterstring representation should conform. In the case of non-restricted Characterstring values, reference is made to the repertoire of ISO/IEC10646-1. The decisions to be made that deal with representation shall be taken with a view to support multiple languages. [IEEE/LTSC]

3.6 Conformance to LOM

CUBER metadata has been designed to conform to LOM 6.1. Some extensions have been made, for LOM was insufficient for the specific purposes of CUBER. The original LOM metadata elements have not been replaced or abused. They have been taken into CUBER as they were defined in the LOM Draft 6.1. However, not all of the LOM metadata elements are used in CUBER, because they were of little significance for the goals of CUBER. These metadata elements have been left empty, but they have not been removed. The CUBER extensions are added to the LOM Base Scheme as independent metadata elements and sub-elements. Also vocabularies have been modified and added to meet the goals of CUBER, but the vocabularies of LOM have been utilised wherever possible to ensure as high interoperability and conformity as possible.

3.7 Data exchange between CUBER and other systems

When exchanging data between the CUBER system and other systems that conform to the LOM standard, a model for transferring and handling the data will be needed. Because of the CUBER extensions there are several metadata elements in the CUBER system that are not present in the original LOM schema. In addition, not all LOM metadata elements are used in the CUBER system. When exporting data from CUBER, a solution is needed to define how the CUBER extension elements are handled in the receiving information system.

The fact is that the other information systems may not understand or be able to handle at all the metadata elements that are not defined in the LOM standard. Accordingly, when importing data into the CUBER system, a solution is needed to define how the LOM elements are mapped into CUBER metadata or handled otherwise in the CUBER system.

Basically, there are three kinds of possibilities to handle the data. The basic assumption is that both information systems use a metadata schema that is conformant to the LOM standard. The systems may have own extensions to the LOM, and they do not need to use every metadata element of LOM. Although it is required to use the LOM metadata as defined by LOM standard; no abuse of the meaning or structure of the metadata elements is accepted. Thus the LOM standard functions as a common language between different information systems. Nevertheless, variations that are internal to the systems are allowed.

1) The data is handled as such and in the same metadata elements in the both sides of the transfer. This means that the particular metadata element is used by both systems and the data contained in this metadata element is corresponding in every aspect. This kind of handling of the metadata requires that the both systems use the same metadata specification, namely the LOM standard.

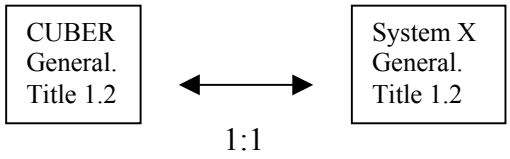

Figure 3 Direct data exchange between CUBER and other systems.

2) The data is exchanged but not handled by the receiving system. This is the case when CUBER exports its extension elements and when CUBER imports LOM elements (or any other metadata elements) that are not used in the CUBER system. This "extra" information will not be used, but will be ignored by the system.

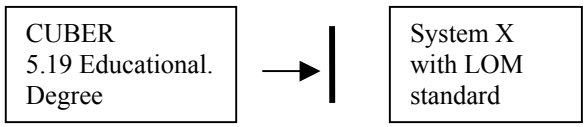

Figure 4 a. Other systems will not handle CUBER's metadata extensions.

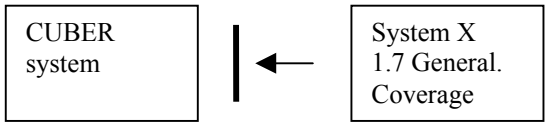

Figure 4 b. CUBER will not handle LOM (or other) elements that are not used in CUBER.

3) The data is exchanged and a mapping will be performed after which data will be used by the system. This is the case when CUBER uses a different kind of metadata for its internal purposes and LOM elements for the import/export functions. A specific mapping will be provided for the mapping.

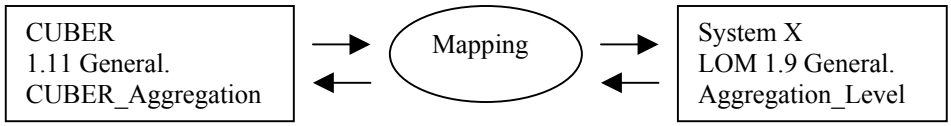

Figure 5 Mapping between CUBER extensions and LOM elements

The figures are intended to illustrate the data exchange between the CUBER system and other systems that use a metadata schema that conforms to LOM standard. There are three groups of metadata elements and the metadata elements are listed and grouped according to the categorisation described above. The specific lists of metadata elements for the data exchange can be found in the Appendix X.

4 The need for an ontology

A definite problem emerged during the process of metadata specification in spring 2001. The equivocal terms and concepts used within CUBER caused many misunderstandings and misinterpretations. Because there are many nationalities involved in CUBER, there are as many interpretations of the concepts that are central for the metadata specification. The inevitable need for commonly understood and unambiguous definitions of concepts and the relationships between them was recognised during the process of metadata specification.

In the CUBER project, research was done on a European virtual university. One of the most important issues has been the lack of a common understanding among the participating universities and countries. To transfer data seamlessly and transparently in the virtual university, there has to be a standard way for both people and computers to communicate all necessary knowledge, with both people and computer systems. The objective of our research was to enable this communication by providing a conceptual model and relevant ontology for the purpose of building an information infrastructure for a virtual university.

4.2 What is an ontology

In philosophy, an ontology is *a theory about the nature of existence, of what types of things exist; ontology, as a discipline, studies such theories* [Berners-Lee et al.]. We do not use the word ontology in that sense. With an ontology we mean *a set of formally specified metadata structures consisting of commonly agreed concepts that bear a limited sense of meaning with them* [Jokela, Turpeinen]. An ontology can have different *dimensions*. For example, an ontology for a document can have *author*, *subject* and *location* as dimensions. In essence, an ontology gives semantics to the metadata. Thus, the metadata structures are just the dimensions of the ontology. The meanings of concepts have to be defined before the semantic web of the virtual university can be useful. Here, by semantic web we mean *an extension of the current Web in which information is given a well-defined meaning that makes it easier for computers and people to co-operate efficiently* [Berners-Lee et al.]. The goal of ontology building is to simplify reality while retaining fidelity [Lamminaho]. In

other words, an ontology provides a vocabulary for users and programmes, with which to communicate about knowledge [Farquhar et al.], and it is essential for the development and use of intelligent systems.

Ontologies can vary greatly in their internal structures. There are ontologies that have none and they are called flat ontologies. In them, every concept is equal and there are no relationships between them. An example of such would be an ontology dimension concerning the authorship of some educational material. There cannot be any hierarchy in the set of names. Sometimes, however, there is indeed an inherent hierarchy. A good example of a *hierarchical ontology* would be location. The world can be divided into continents and further subdivided into countries and areas and cities. There are some cases where even this is not enough. For example, if we had an ontology dimension describing the subject of a course, it is easy to think of a situation where a simple hierarchy would not suffice. Here is a very simplified example (Figure).

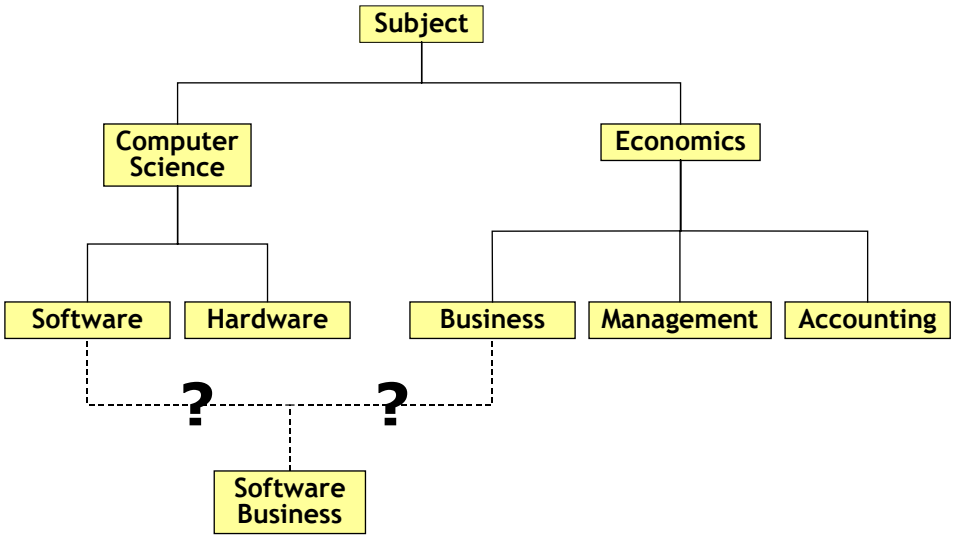

Figure 6 An example of a hierarchical ontology

Assume that there are two subjects in the hierarchy, *Computer Science* and *Economics*. Under Computer Science there are Software and Hardware. In

Economics there are Business, Management and Accounting. What if we were to introduce a course on Software Business? Such a course certainly does not fit only in Computer Science and Software, because it is clearly business-related. But it cannot be filed exclusively under Economics and Business either, because of its software content. In this case, we have to use a directed acyclic graph (DAG) and allow the course to be filed under both.

A directed acyclic graph is a graph where all transitions are unidirectional and there are no loops. The need for ontologies of this sort has been found also in other studies, for example in the Ontolingua Server [Farquhar et al.], and also in the MARC research [Lamminaho]. In a typical case, a course or other object in the virtual university can cover several slightly different subjects. It is practical to be able to tell the system exactly *how much* a course deals with a certain subject. In this way, we make it possible for user queries to be more accurate and we generate better results. The metadata can be given weights, so that it can be partly about one subject area and partly about others.

4.3 Ontology in CUBER

Building an ontology is reportedly extremely difficult, and by its nature it is often an iterative task [Jokela, Turpeinen]. Still, it has been clearly shown that there is a definite need for semantics [Domingue, Motta]. During the course of the CUBER project, we have analysed relevant information and developed a conceptual model and a meta ontology. This meta ontology describes all of the ontology dimensions and their internal structure and data. These are represented as a class diagram. Aside from the trivial cases where the ontology clearly had no structure and was flat, both hierarchical and acyclic structures were considered. Each of the meta ontology dimensions was discussed with domain experts, and the most proper ontology structure was chosen. Typically, many of the ontologies were flat, several had some hierarchy in them and one of them was best implemented as a DAG. In other research in the area of ontologies for educational use [Crampes, Ranwez], the focus has been mainly in the

information system structure, whereas we have tried to find content and structure for the ontologies and a proper conceptual model for the area.

There are many tools to choose from when defining and implementing an ontology. Logic languages can be used, data models can be drawn, for example with UML, a combination of XML and RDF is feasible, or the ontology can be described with some existing ontology building system, such as Ontolingua [Farquhar et al.]or the (KA)2-system [Benjamins et al.]. At the core of CUBER there is a database, not a Web server (or several). This is why we have not chosen to define ontologies as XMLs and DTDs, but rather as UML-diagrams and a database schema generated from them. The main concern was easy implementation and good communication with the work package for database installation.

The advantages and drawbacks of different types of ontologies were studied. On every occasion, the most suitable form of ontology structure was chosen. Approximately half of the ontology dimensions have a flat structure, the other half a hierarchical structure, and some individual dimensions actually had a graph structure. In the CUBER project, user profiles were considered but found redundant with the search engine. The main reason for this is that rather than actually deriving courses from the resources and matching them to the user requirements, courses are only listed, not constructed. Essentially, the user might want to take different kinds of courses over time, rather than only going to courses on the same topic over and over again. Thus, the search form tells more about the user's intentions than about the courses he or she might already have taken. Of course, there is the issue of prerequisites, but that is a slightly different issue, and it is tackled in the CUBER conceptual model. As for the learner information and profile, we have found it sufficient to store some basic attributes about the learner (e.g. name, language and contact information) and allow the user to save any searches he or she might make. If considered necessary, despite the difficulties, it is quite easy to add the weighted learner interest mechanism into the model. Just as there are weights between learner and subjects, a similar relationship can be added between learner and subject.

In addition, if the profile were used to check the user's level, several difficulties would arise. In this case, the CUBER system would have to have access to the study registers of all its partners. If this were the case, who would be responsible for keeping the data up-to-date and consistent? All transactions should be atomic, and all data very robust. How would it be ensured that all systems are compatible? Nevertheless, additional information about the learner can be added as attributes. Of course, if this information is about the learner's qualifications or some similar attribute, it cannot be retrieved from outside systems, instead the learner will have to be trusted to give accurate input. The use of these attributes is, of course, entirely within the search engine. Attributes, relations, and various kinds of data structures can be defined every which way, but in the end these structures come in handy only if they are exploited in the applications (search engine and database front-end).

5 Meta ontology and conceptual model

The defined class diagram consists of two parts: the *conceptual model* and the *meta ontology*. These are discussed in the following chapters.

5.1 Meta ontology

The meta ontology shows the structure of the ontology in the class diagram. All appropriate characteristics to be defined about the domain have been introduced as dimensions of the ontology. These dimensions are attached to relevant concepts in the conceptual model with associations. Each dimension of the meta ontology is represented as a class with or without an association to itself. If the class does not have an association to itself, the ontology is considered *flat*. When the association is present, the structure is either hierarchical or a directed acyclic graph depending on the multiplicities of the association. A many-to-many association represents a graph structure, whereas a one-to-many association represents a tree. The unlikely occasion of a one-to-one association is just an exception of the tree structure, i.e. a tree without any branches.

Where applicable, the actual content of the ontology dimensions is presented as separate graphs, or external authorised ontologies are used.

5.2 Conceptual model

The conceptual model defines the essential concepts of the domain and their attributes. These concepts are also attached to the meta ontology with associations to the relevant ontology dimensions. The whole conceptual model has been divided into parts in order to provide clarity and ease of understanding. The modelling has been done with UML (Unified Mark-up Language) and realised with Rational Rose Enterprise Edition. It has to be noted that some additional notations have been used, mainly to provide information not available within the Rational Rose notation. This information is vital when the schema is implemented into a relational database schema, as is the case with CUBER. The diagrams contain a few basic elements. A very simple example of these can be seen in Figure 7. The elements are *classes*, *attributes*, *associations* and *association classes*.

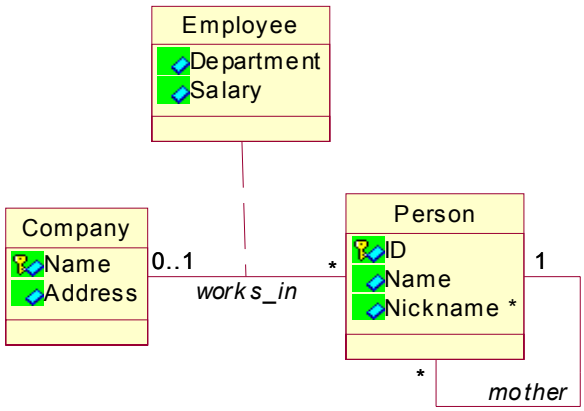

Figure 7 An example of the notation

Classes represent entities of the modelled domain. In the conceptual model of CUBER, these are, for example, *Person*, *Course* and *Institute*. Classes have **attributes**, which are basically just relevant information about the class. In the example, a class *Person* has the attributes *ID*, *Name* and *Nickname*. Relationships between classes are called **associations**. The class *Person* has an association to itself. This means that two instances of this class can have an

association. In the example this association has been named *mother*. So one person can be the mother of another. Note the **multiplicities** at the ends of the association. These mean that every person has exactly one mother, but the mother can have an undefined number of children. A person has also an association with the class *Company* that has been named *works_in*. Again, a person can work in zero or one company, whereas a company has many employees. The multiplicity marked with an asterisk can really be any number, including zero! The association *works_in* has also some information that cannot be attached to either *Company* or *Person*. Therefore, an association class *Employee* has been defined. This class and its attributes depend on an association between a certain company and a certain person. This is shown with a dashed line between the association class and the association to which it is attached.

In addition, we have used the two following notations that are not found in a standard UML:
A ***key***: When there is a small key in front of an attribute, this means that the attribute is an identifier of the class. This is mainly relevant in the relational database schema, where this attribute is marked as a *primary key*. The value of this attribute has to be unique, i.e. no two instances of this class can have the same value for it.
An ***asterisk***: An asterisk following an attribute declaration means that this attribute has *multiple values*. Whereas an attribute usually has one value for each attribute, an attribute with multiple values can have many.

In our example in Figure 7, the ID is the key of the class, i.e. no two persons can have the same ID. A person has exactly one name, but can have several nicknames.

6 Implementation of the class diagram

This chapter describes how we have transformed the class diagram (conceptual model and ontology) into a database schema and how the metadata schema has been transformed into an XML schema with a DTD file.

6.1 Database schema and SQL

Actually, the class diagram with its classes and relations is quite close to a database schema. The transformation is made in a very straightforward manner. In the field of database management, there are established manners by which to do this. The result is a set of database tables with attributes, and all the information in the class diagram is found in the database schema. This schema is then converted into a suitable normal form. In CUBER, we have come to the conclusion that third normal form (3NF) is appropriate.

After normalisation, the tables are converted into SQL and can be directly imported into a relational database, such as the one used by the work package for the CUBER database. All information in the conceptual model and the ontology has been made into `CREATE TABLE` commands in SQL (Appendix 4 The SQL definition). This way, the SQL definition is directly executable in a relational database, and all the necessary tables in the conceptual model are created. They contain all the information included in the metadata definition. As the SQL is written in standard SQL, some slight alterations may be necessary before installation because of the different 'dialects' of SQL used by different database systems. The contents of the ontology dimensions have been added into the SQL definition, except for two: Subject and Discipline. These two are classifications rather than ontologies. They are references outside the conceptual model, and are dealt with in the application using the database, not in the database itself.

6.2 XML schema of the metadata

In addition to the database solution, a description schema in XML language has been made of the metadata for possible future use. This consists of a DTD file, in which the structure and the tags of the XML file are introduced. This way, the XML can be validated. The whole schema is closed within `<metadata></metadata>` tags. The content has been divided into the nine categories present in the metadata definition with `<category></category>` tags. In the categories, there are `attributes` (in tags accordingly), which in turn contain the same information as the metadata definition found in Appendix 1, all within tags: `number`, `name`, `explanation`, `size`, `order`, `valuespace`, `datatype`, `example`, `aggregationlevels`, `mandatory`. The tags are exactly the same column titles found in the metadata definition. The XML schema contains the metadata information in a structured form. Further constraints to the metadata content could be made with XML Schemas technology. In XML Schemas data types can be defined. The XML definition of the metadata can be used by any application capable of reading the information in the XML. This could be, for example, a Web application, where the semantics of the metadata tags is defined. Thus, XML enables the metadata to be used in a Web environment where a standard set of tags is understood.

A new DTD of the metadata will be defined in the near future, as soon as the Knowledge Base of the CUBER system is mature enough to enable the data exchange between CUBER and other systems. This DTD to be developed will describe the structure of the metadata and can be used to produce instances of Study element descriptions in XML notation. Developing the DTD for the CUBER metadata requires exploring the emerging XML-bindings of LOM, such as the IMS XML-binding. At the same time, during the implementation stage of the CUBER metadata and the Knowledge Base, the final refinements to the metadata specification are expected to emerge. The possible changes in the metadata and the most recent development of the LOM-XML –bindings will be taken into consideration when defining the CUBER metadata DTD.

References

[Benjamins et al.] Benjamins, V. Richard, Fensel, Dieter, Decker, Stefan and Gómez Pérez, Asunción. 1999. *(KA)^2: building ontologies for the Internet: a mid-term report*. Int. J. Human-Computer Studies (1999) 51. Pp. 687-712.

[Berners-Lee et al.] Berners-Lee, Tim, Hendler, James and Lassila, Ora. 2001. *The Semantic Web*. Scientific American, May 2001. http://www.sciam.com/2001/0501issue/0501berners-lee.html [9.5.2001]

[Britain et al.] Britain, Sandy & Liber, Oleg. 1998. *A Framework for Pedagogical Evaluation of Virtual Learning Environments.* JISC Technology Application Programme.

http://www.jisc.ac.uk/jtap/htm/jtap-041.html

[Crampes, Ranwez] Crampes, Michel, Ranwez, Sylvie. 2000. *Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web*. 9 p.

[Domingue, Motta] Domingue, John, Motta, Enrique. 2000. *PlanetOnto: From News Publishing to Integrated Knowledge Management Support*. IEEE Intelligent Systems, May/June 2000. Pp. 26-32.

[Farquhar et al.] Farquhar, Adam, Fikes, Richard and Rice, James. 1997. *The Ontolingua Server: a tool for collaborative ontology construction*. Int. J. Human-Computer Studies (1997) 46. Pp. 707-727.

[IEEE/LTSC] IEEE Learning Technology Standards Committee (LTSC)/ IEEE P1484.12 Learning Object Metadata Working Group. 2001. *Draft Standard for Learning Object Metadata.*

http://ltsc.ieee.org/doc/wg12/LOM_WD6-1_1.doc

[IEEE/LTSC2] IEEE Learning Technology Standards Committee (LTSC)/ IEEE P1484.3 Glossary Working Group. 2001. *Draft Standard for Information Technology -- Learning Technology – Glossary.* http://ltsc.ieee.org/doc/wg3/Glossary-20010309.doc

[Jokela, Turpeinen] Jokela, Sami and Turpeinen, Marko. 2000. *Ontology Development for Flexible Content*. HICSS-33, Minitrack on System Support For Electronic Business On Internet. 26 p.

[Lamminaho] Lamminaho, Virva. 2000. *Metadata specification: Forms, Menus for Description of Courses and All Other Objects.* CUBER project: Deliverable D3.1.

[Wason] Wason, Tom. 2001. *Dr. Tom's Metadata Guide.* IMS Global Learning Consortium. http://www.imsproject.org/drtommeta.html

[Weinstein] Weinstein, Peter C. 1998. *Ontology-Based Metadata: Transforming the MARC Legacy*. 10 p.

Appendices

Appendix 1 The CUBER metadata

CUBER METADATA SCHEMA version 2.3 **27 November 2001**

Based on LOM 6.1. Authors: Päivi Nykänen & Kirsi Pelto-Aho, WP3 (UH)

Note: The metadata elements written in normal font are original LOM elements.
The metadata elements written in *italic* font are original LOM elements, but NOT used in CUBER.
The metadata elements written in **bold** font are CUBER's extensions to the LOM.

CUBER has also added two new columns to the table of LOM schema: "Aggregation levels" and "Mandatory". These will be used only internally for the CUBER system.

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
1	General	General descriptive information about the learning object as a whole.	1	N/A	-	-	-	-	-
1.1	Identifier	A globally unique label for identifying the learning object.	1	N/A	-	Reserved.	Not Used.	-	-
1.2	Title	Name of the L.O.	1	N/A	-	Langstring (smallest permitted max: 1000 char)	-	All	Yes
1.3	CatalogEntry	Defines an entry within a catalog assigned to this L.O.	Smallest permitted max. 10 values	No.	-	-	-	-	-
1.3.1	Catalog	The name of the catalog (i.e. the listing identification	1	N/A	Repertoire of ISO/IEC 10646-1	Characterstring (min-max: 1000 char)	ISBN, ARIADNE	All	No

		system).							
1.3.2	Entry	Actual string value of the entry within the catalog defined in 1.3.1.	1	N/A	-	Langstring (smallest permitted max: 1000 char)	-	All	No
1.4	Language	The primary human language(s) used within this L.O.	Smallest permitted max: 10 items	No	LanguageID =Langcode (´-´Subcode)*, ISO 639, ISO 3166	Characterstring (smallest permitted max: 100 char)	"en" "en-GB" "de" "fr-CA" "it"	All	Yes
1.5	Description	A textual description of the content of this L.O.	Smallest permitted max: 10 items	No	-	Langstring (smallest permitted max: 2000 char)	-	All	Yes for Course, Package and Programme
1.6	Keywords	Keywords or phrases describing this L.O.	Smallest permitted max: 10 items	No	-	Langstring (1000 char)	-	All	Yes
1.7	*Coverage*	*The span of such things as time, culture, geography or region that applies to this L.O.*	*Smallest permitted max: 10 items*	*No*	-	*Langstring (smallest permitted max: 1000 char)*	*(en, Circa, 16th century France)*	*NOT USED IN CUBER!*	-
1.8	Structure	Underlying organisational structure of this L.O.	1	N/A	Collection Mixed Linear Hierarchical Networked Branched Parcelled Atomic	Vocabulary; see Ontology	-	Material Package Programme	No
1.9	*Aggregation Level*	*The functional granularity of the L.O.*	*1*	*N/A*	*1=the smallest level of aggregation 2=a collection*	*Vocabulary*	-	*Not used internally for CUBER; used only for import and export of data.*	-

					of atoms *3=a collection of level 2 Los or a course* *4=the largest level of granularity, set of courses that lead to a certificate*				
1.10	**CUBER Identifier**	**A label for identifying the L.O. Valid only within CUBER.**	**1**	**N/A**	**Repertoire of ISO/IEC 10646-1**	**Characterstring (min-max: 100 char)**	**-**	**All**	**Yes**
1.11	**CUBER Aggregation**	**The functional granularity of the L.O.s included in CUBER.**	**1**	**N/A**	**0=material 1=course 2=package 3=programme**	**Vocabulary; see Ontology**	**-**	**All**	**Yes**

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
2	LifeCycle	This category describes the history and current state of this L.O. and its contributors.	1	N/A	-	-	-	-	-
2.1	Version	The edition of this L.O.	1	N/A	-	Langstring (smallest permitted max: 50 char)	3.0, 3.1, 1.2 alpha	Material Course	No
2.2	Status	The state or condition of this L.O.	1	N/A	Draft Final Revised Archived Unavailable	Vocabulary; see Ontology	-	All	Yes
2.3	Contribute	This element	Smallest	No	-	-	-	-	-

		describes the people and organisations that have affected the state of this L.O.	permitted maximum 30 items						
2.3.1	Role	Kind of contribution. The roles of the persons or organisations involved in providing the L.O. *At least the author(s) of the L.O. should be described*. Notice: List 1. is defined by LOM and it will be used for the Material only. List 2. is defined by CUBER and it will be used for Course, Package and Programme.	1	N/A	List 1 -Author -Editor -Publisher -Content provider -Graphical designer -Instructional designer -Initiator -Terminator -Technical implementer -Educational validator -Technical validator -Script writer -Unknown **List 2** **-Administrator** **-Advisor** **-Assistant** **-Contact person** **-Examiner** **-Lecturer** **-Teacher** **-Tutor** **-Provider institution**	Vocabulary; **see Ontology**	-	Material (list 1) **Course, Package and Programme (list 2)**	Yes

2.3.2	Entity	Information about the people and organisations contributing to this L.O.	Smallest permitted max: 40 items	Yes	V-card	Characterstring (smallest permitted max: 1000 chars)	University of X, Professor Nevgi	All	Yes
2.3.3	*Date*	*The date of the contribution.*	*1*	*N/A*	-	*Date*	-	*NOT USED IN CUBER!*	-
2.4	**Recurrence**	**This element indicates whether the L.O. is unique or repeated periodically.**	**1**	**N/A**	**Repeated One-time study element Every 3 or 6 months Every year Last occasion**	**Vocabulary; see Ontology**	**-**	**All**	**No**
2.5	**Date**	**The time span or important dates of the L.O.**	**N**	**-**	**-**	**-**	**-**	**-**	**-**
2.5.1	**Begin**	**The begin date of the L.O.**	**1**	**N/A**	**-**	**Date**	**-**	**All**	**Yes**
2.5.2	**End**	**The end date of the L.O.**	**1**	**N/A**	**-**	**Date**	**-**	**All**	**No**
2.5.3	**Kind**	**The nature of the contribution or action required with regard to the dates announced.**	**1**	**N/A**	**Enrolment Exam period Publishing time Study period Other**	**Vocabulary; see Ontology**	**-**	**All**	**Yes**

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
3	MetaMetadata	This category describes the M.D. record itself.	1	N/A	-	-	-	-	-
3.1	Identifier	A globally unique label that identifies	1	N/A	-	Reserved	-	-	-

		this M.D. record.							
3.2	CatalogEntry	This element describes an entry within a catalog given to the M.D. instance.	Smallest permitted max: 10 items	No	-	-	-	-	-
3.2.1	Catalog	The name of the catalog (i.e. listing identification system).	1	N/A	Repertoire of ISO/IEC 10646-1	Characterstring (smallest permitted max: 1000 char)	ARIADNE	All	No
3.2.2	Entry	Actual string value of the entry in the catalog.	1	N/A	-	Langstring (smallest permitted max: 1000 char)	(en,KUL532)	All	No
3.3	Contribute	Describes the people and organisations that have affected the state of this M.D.	Smallest permitted max: 10 items	Yes	-	-	-	-	-
3.3.1	Role	Kind of contribution	1	N/A	Creator Validator	Vocabulary, Ontology	-	All	Yes
3.3.2	Entity	Information about the people and organisations contributing to this M.D. instance.	Smallest permitted max: 10 items	Yes	V-card	Characterstring (smallest permitted max: 1000 char)	-	All	Yes
3.3.3	Date	The date of the contribution.	1	N/A	-	Date	-	All	Yes
3.4	Metadata Scheme	The name and version of the authoritative specification used to create this M.D. instance.	Smallest permitted max: 10 items	No	Repertoire of ISO/IEC 10646-1	Characterstring (smallest permitted max: 30 char)	LOM-1.0	All	Yes
3.5	Language	Language of this	1	N/A	LanguageID=	Characterstring	"en"	All	Yes

		M.D. instance. Default value for all the Langstring values in this M.D. instance.			Langcode (´-´Subcode)*, ISO 639, ISO 3166	(smallest permitted max: 100 char)	Default in CUBER is English.		

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
4	Technical	This category described the technical requirements and characteristics of this L.O.	1	N/A	-	-	-	-	-
4.1	Format	Technical data type(s) of this L.O.	Smallest permitted max: 40 items	No	MIME types based on IANA registration (see RFC2048) or "non-digital"; see Vocabulary & Ontology	Characterstring (smallest permitted max: 500 char)	Video/mpg, Text/html, Application/x-toolbook, Book	Material	Yes
4.2	*Size*	*The size of the digital L.O. in bytes.*	*1*	*N/A*	*ISO 646, but only the digits ´0´…´9´*	*Characterstring (smallest permitted max: 30 char)*	-	*NOT USED IN CUBER!*	-
4.3	Location	A string to access this L.O. Physical location of the L.O. Exact location or method of locating.	Smallest permitted max: 10 items	Yes	Repertoire of ISO/IEC 10646-1	Characterstring (smallest permitted max: 1000 char)	Http://host/id	All	No
4.4	Requirements	Describes the technical	Smallest permitted	No	-	-	-	-	-

		capabilities required to use this L.O.	max: 40 items						
4.4.1	Type	The technology required to use this L.O.	1	N/A	Operating system Browser	Vocabulary; see Ontology	-	Material Course	No
4.4.2	Name	Name of the technology required to use this L.O. Note1: The value of this element can be derived from 4.1 Technical.Format automatically.	1	N/A	If Type =Operating system, then: PC-DOS MS-Windows MacOS Unix Multi-OS None If Type=Browser, then: Any Netscape Internet Explorer Opera	Vocabulary; see Ontology	-	Material Course	No
4.4.3	Minimum Version	Lowest possible version of the required technology to use this L.O.	1	N/A	Repertoire of ISO/IEC 10646-1	Characterstring (smallest permitted max: 30 char)	-	Material Course	No
4.4.4	*Maximum Version*	*Highest version of the technology known to support the use of this L.O.*	*1*	*N/A*	*Repertoire of ISO/IEC 10646-1*	*Characterstring (smallest permitted max: 30 char)*	-	*NOT USED IN CUBER*	-
4.5	*Installation Remarks*	*Description of how to install this L.O.*	*1*	*N/A*	-	*Langstring (smallest permitted max:*	-	*NOT USED IN CUBER*	-

						1000 char)			
4.6	*Other Platform Requirements*	*Information about other software and/or hardware requirements.*	*1*	*N/A*	-	*Langstring (smallest permitted max: 1000 char)*	-	*NOT USED IN CUBER*	-
4.7	*Duration*	*Time a continuous L.O. takes when played at intended speed.*	*1*	*N/A*	-	*Date*	-	*NOT USED IN CUBER*	-
4.8	**Material Size**	**Size of a digital or a non-digital L.O.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**	**150 pages 11700 words 50 kB 1H 15Min**	**Material**	**No**
4.9	**Description**	**Further description on the technical characteristics of the L.O.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**	**Guidelines and commands for using Unix.**	**All**	**No**

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
5	Educational	This category describes the key educational and pedagogical characteristics of this L.O.	1	N/A	-	-	-	-	-
5.1	*Interactivity Type*	*The flow of interaction between the learner and this L.O.*	*1*	*N/A*	*Active Expositive Mixed Undefined*	*Vocabulary*	*Active: Exercises Expositive: Documents*	*NOT USED IN CUBER*	-
5.2	Learning Resource Type	Specific kind of L.O.	Smallest permitted max: 10	Yes	Exercise Simulation Questionnaire	Vocabulary; see Ontology	-	Material	No

		NOTE: This element will be used for material-level only; it makes no sense to use this element for other aggregation levels with this vocabulary.	items		Diagram Figure Graph Index Slide Table Narrative text Exam Experiment Problem-statement Self assessment				
5.3	*Interactivity Level*	*The degree of interactivity between the learner and the L.O.*	*1*	*N/A*	*Very low* *Low* *Medium* *High* *Very high*	*Vocabulary*	-	*NOT USED IN CUBER*	-
5.4	*Semantic Density*	Amount of information conveyed by this L.O. as compared to its size or duration.	*1*	*N/A*	*Very low* *Low* *Medium* *High* *Very high*	*Vocabulary*	-	*NOT USED IN CUBER*	-
5.5	*Intended End User Role*	*Principal users for whom this L.O. was designed.*	*Smallest permitted max: 10 items*	*Yes*	*Teacher* *Author* *Learner* *Manager*	*Vocabulary*	-	*NOT USED IN CUBER*	-
5.6	Context	The principal environment within which the use of this L.O. is intended to take place. The original LOM vocabulary has been replaced by	Smallest permitted max: 10 items	No	**DL0 General studies** **DL1 Basic, Bac.** **DL2 Intermediate, Bachelor** **DL3 Advanced, Master** **DL4 Post-**	Vocabulary **See Ontology**	**-**	**Course** **Package** **Programme**	**Yes for Course, Package and Programme**

		CUBER´s own vocabulary.			**graduate, L/D DL5 Vocational, further education**				
5.7	*Typical Age Range*	*Age of the typical user of this L.O.*	*Smallest permitted max: 5 items*	*No*	-	*Langstring (smallest permitted max: 1000 char)*	-	*NOT USED IN CUBER*	-
5.8	*Difficulty*	*How hard it is for the target audience to work through this L.O. in relation to the educational level*	*1*	*N/A*	*Very easy Easy Medium Difficult Very difficult*	*Vocabulary*	-	*NOT USED IN CUBER*	-
5.9	Typical Learning Time	Typical time it takes to work through this L.O. (e.g. hours, days, weeks, months)	1	N/A	-	Time, Date	Used in CUBER only when ECTS not available.	All	No
5.10	*Description*	*Comments on how this L.O. is to be used.*	*1*	*N/A*	-	*Langstring (smallest permitted max: 1000 char)*	*Teacher guidelines that come with a textbook.*	*NOT USED IN CUBER*	-
5.11	*Language*	*The human language used by the target group of this L.O.*	*Smallest permitted max: 10 items*	*No*	*LanguageID =Langcode (´-´Subcode)*, ISO 639, ISO 3166*	*Characterstring (smallest permitted max: 100 char)*	*"en" "en-GB" "de" "fr-CA" "it"*	*NOT USED IN CUBER*	-
5.12	**Teaching Activity**	**Description of principal teaching activities used for this L.O.**	**1**	**N/A**	**-**	**-**	**-**	**-**	**-**
5.12.1	**Teaching Method**	**This sub-element describes the**	**1**	**N/A**	**Face-to-face Distance (www-**	**Vocabulary; see Ontology**	**-**	**Course Package**	**Yes**

		principal teaching method used for this L.O.			**based) Distance (independent) Mixed face-to face and distance Undefined**			**Programme**	
5.12.2	**Dependence on time**	**This sub-element describes the dependence on time of this L.O.**	**1**	**N/A**	**Given schedule Negotiable schedule No time-restrictions Undefined**	**Vocabulary; see Ontology**	**-**	**Course Package Programme**	**Yes**
5.12.3	**Dependence on place**	**This sub-element describes the dependence on place of this L.O.**	**1**	**N/A**	**Given place Negotiable place No place-restrictions Undefined**	**Vocabulary; see Ontology**	**-**	**Course Package Programme**	**Yes**
5.13	**ECTS Credits**	**This element describes the ECTS credits of this L.O.**	**1**	**N/A**	**-**	**Characterstring (smallest permitted max: 30 char)**	**-**	**Course Package Programme**	**Yes for Course**
5.14	**Dedication**	**This sub-element describes how intensively the learner must work.**	**1**	**N/A**	**Part-time Full-time Mixed (part&full) No time limits Undefined**	**Vocabulary; see Ontology**	**-**	**Course Package Programme**	**No**
5.15	**Evaluation**	**This element describes the principal method(s) and amount of evaluation for this L.O.**	**1**	**N/A**	**-**	**-**	**-**	**-**	**-**
5.15.1	**Assessment**	**This sub-element**	**1**	**N/A**	**Formal**	**Vocabulary; see**	**-**	**Course**	**No**

		describes the assessment related to this L.O.			**assessment Informal assessment Final assessment Continuous assessment Several assessments No assessment *Undefined***	**Ontology**		**Package**	
5.15.2	**Method**	**The principal method(s) of assessment for this L.O.**	**N**	**No**	**Exam with attendance Electric exam in distance Exercises Assignment Participation Presentation Essay Seminar paper Portfolio *Undefined***	**Vocabulary; see Ontology**	**-**	**Course Package**	**No**
5.15.3	**Number**	**The number of tasks or exams that form the basis for evaluation for this L.O.**	**1**	**N/A**	**ISO 646, but only digits ´0´…´9´**	**Characterstring (smallest permitted max: 30 char)**	**-**	**Course Package**	**No**
5.16	**Enrolment**	**This element contains information on the enrolment, e.g. method of enrolment.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**	**-**	**Course Package Programme**	**No**
5.17	**Study Guidance**	**This element describes the**	**1**	**N/A**	**-**	**Langstring (smallest**	**-**	**Course Package**	**No**

		guidance or tutoring provided for the learner.				**permitted max: 1000 char)**		**Programme**	
5.18	**Pre-requisites**	**This element describes the skills required in order to take this L.O.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**	**Good computer literacy; Fluent German.**	**Course Package Programme**	**No**
5.19	**Degree**	**The official degree related to this study element.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**		**Course Package Programme**	**Yes for Programme**

Nr	**Name**	**Explanation**	**Size**	**Order**	**Value Space**	**Data Type**	**Example**	**Aggregation levels**	**Mandatory**
6	Rights	This category described the intellectual property rights and the conditions of use for this L.O.	1	N/A	-	-	-	-	-
6.1	*Cost*	*Whether use of this L.O. requires payment.*	*1*	*N/A*	*Yes* *No*	*Vocabulary*	-	*NOT USED IN CUBER*	-
6.2	Copyright	Copyright and other restrictions on the use of this L.O.	1	N/A	Yes No	Vocabulary	-	Material	No
6.3	Description	Comments on conditions of use of this L.O.	1	N/A	-	Langstring (smallest permitted max: 1000 char)	-	Material	No
6.4	**Cost in EURO**	**The amount of payment in EURO.**	**1**	**N/A**	**-**	**Characterstring (smallest permitted max:**	**100 EURO**	**All**	**Yes**

						100 char)			
6.5	**Financing**	**The financing possibilities or grants available for the learner.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**	**-**	**All**	**No**

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
7	Relation	This category describes the relationships between this L.O. and other L.O.s	Smallest permitted max: 100 items	No	-	-	-	-	-
7.1	Kind	Nature of relationship between this L.O. and the target L.O. NOTE1: Relations based on Dublin Core.	1	N/A	IsPart **HasPart** IsVersionOf HasVersion IsFormat Of HasFormat References IsReferencedBy IsBasedOn IsBasisFor **Requires** IsRequiredBy	Vocabulary; See Ontology	**NOTE2: HasPart and Requires are used in CUBER. Optional, Additional, Compulsory and Ex-changeable can be used too, but only in free text**.	All	Yes for package and programme
7.2	Resource	The target L.O. that this relationship references.	1	N/A	-	-	-	-	-
7.2.1	Identifier	Unique identifier of the target L.O.	1	N/A	-	Reserved	Not used.	-	-
7.2.2	*Description*	*Description of the target L.O.*	*1*	*N/A*	-	*Langstring (smallest permitted max: 1000 char)*	-	*NOT USED IN CUBER*	-

7.2.3	CatalogEntry	Defines an entry within a catalog assigned to this L.O.	Smallest permitted max: 10 items	No	-	-	-	All	Yes for package and programme
7.3	**Dependencies**	**Description of dependencies between the study elements.**	**1**	**N/A**	**-**	**Langstring (smallest permitted max: 1000 char)**	**-**	**Course Package Programme**	**No**

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
8	Annotation	This category provides comments on the educational use of this L.O.	Smallest permitted max: 30 items	No	-	-	-	-	-
8.1	Person	The person who created this annotation.	1	N/A	V-card	Characterstring (smallest permitted max: 1000 char)	Information from the V-card will be defined separately.	All	No
8.2	Date	Date this annotation was created.	1	N/A	-	Date	-	All	No
8.3	Description	The content of this annotation.	1	N/A	-	Langstring (smallest permitted max: 1000 char)	-	All	No

Nr	Name	Explanation	Size	Order	Value Space	Data Type	Example	Aggregation levels	Mandatory
9	Classification	This category describes where this L.O. falls within a particular classification system.	Smallest permitted max: 40 items	No	-	-	-	-	-
9.1	Purpose	The purpose of classifying this L.O.	1	N/A	Discipline Subject Idea Prerequisite Educational objective Accessibility restrictions Educational level Skill level Security level	Vocabulary	Discipline and Subject will be used in CUBER.	Course Package Programme	Yes
9.2	TaxonPath	This element describes a taxonomic path in a specific classification system.	Smallest permitted max: 15 items	No	-	-	-	-	-
9.2.1	Source	The name of the classification system.	1	N/A	Repertoire of ISO/IEC 10646-1	Langstring (smallest permitted max: 1000 char)	(en, ACM) (en, ARIADNE)	Course Package Programme	Yes
9.2.2	Taxon	This element describes a particular term within the taxonomy.	Smallest permitted max: 15 items	Yes	-	-	-	-	-
9.2.2.1	Id	The identifier of the Taxon.	1	N/A	Repertoire of ISO/IEC 10646-1	Characterstring (smallest	320, 4.3.2,	Course Package	Yes

						permitted max: 100 char)	BF180	Programme	
9.2.2.2	Entry	The textual label of the Taxon.	1	N/A	-	Langstring (smallest permitted max: 500 char)	(en, Medical sciences)	Course Package Programme	Yes
9.3	*Description*	*This is the description of the L.O. relative to the Classification Purpose (9.1) of this classification.*	*1*	*N/A*	-	*Langstring (smallest permitted max: 2000 char)*	-.	*NOT USED IN CUBER*	-
9.4	*Keywords*	*The keywords descriptive of the L.O. relative to the Classification Purpose (9.1) of this specific classification.*	*Smallest permitted max: 40 items*	*Yes*	-	*Langstring (smallest permitted max: 1000 char)*	-	*NOT USED IN CUBER*	-

Appendix 2 Ontology for CUBER metadata

Ontology for CUBER metadata specification 2.3

The new version of CUBER metadata has an ontology that aims to define the terms, concepts and vocabularies used in the CUBER metadata. Each metadata element that has an ontology (i.e. a vocabulary to be explained and defined) is referenced in this document with the name and number of the metadata element. Notice that not all metadata elements have an ontology.

The ontology is represented in almost every case both graphically and as a table with written explanations and definitions. Only some vocabularies of metadata elements do not have explanations, because they may be self-explanatory as such. The graphical part of the ontology describes the structure and relationships of the ontology. The written definitions aim to give unambiguous meanings to the vocabularies used in the metadata.

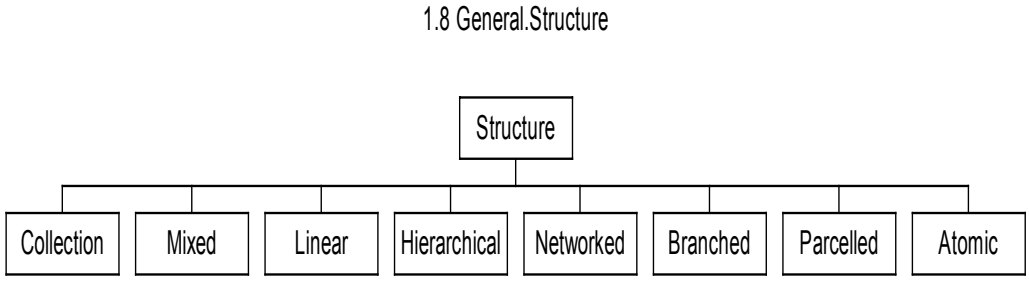

Figure 1. Metadata element 1.8 General.Structure with Ontology.

Definitions of the Vocabulary of Metadata element 1.8 General.Structure according to IMS/T. Wason (2000, Metadata FAQ)

Word	**Meaning**
Collection	A collection is a library or a repository of resources. It may all be packaged in one resource. It has no inherent navigational structure but it may have an index and/or a table of contents. Items that belong to a collection may be accessed on an individual basis.
Mixed	A resource that may be experienced in several different modes, with no single dominant mode, is classified as mixed. This option should be used as the last resort.
Linear	A resource with a single pathway through it has a linear structure. The sequencing through contents is in a well-defined order.
Hierarchical	A tree structure in which one enters at one place and may then select different branches at each succeeding level is called hierarchical. Entrance to the structure exists at a single point. A hierarchy implies well-ordered, differing levels of importance, granularity or specificity.
Networked	A networked resource is characterised by links but no overall structure by design. Entrance may exist at any point in the structure, and navigation may extend beyond the immediate resource.
Branched	A branched resource is characterised by decision points with no inherent “up” or “down” but with a well-designed structure.
Parcelled	A parcelled resource is divided into distinct components or domains. These may be thought of as “rooms” of chat spaces.
Atomic	An atomic resource has no further subdivisions or navigation within the resource. A collection may consist of atomic resources.

1.11 General.CuberAggregation

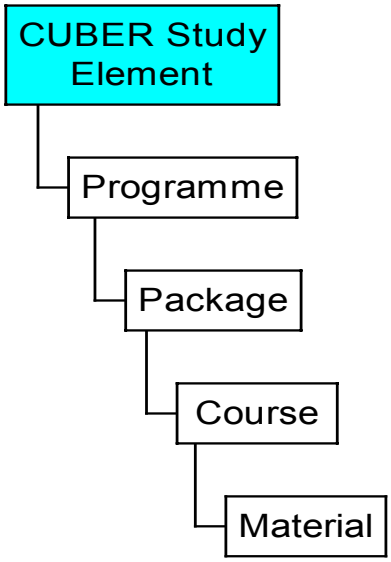

Figure 2. 1.11 General.CuberAggregation with Ontology.

Word	**Meaning**
Material	The term study material refers to any kind of resource (digital or non-digital) that contains information used in education. Material includes no teaching activity; its function is to serve as a source of information for a study course.
Course	The term study course refers to a complete unit of instruction that provides the learners with the knowledge or skills required for competence in a subject matter. A study course is any academic or vocational course arranged by a course provider. This is the lowest level that can offer credits or recognition within an educational institution. A study course usually includes teaching activity and examination.
Package	The term study package refers to a collection of study courses. A study package can offer credits but no official degrees or certificates, i.e. a study package has internally visible outcomes. Packages can be part of study programmes.
Programme	The term study programme refers to a collection of courses and/or packages, and it can lead to an official university degree or a certificate of competence, i.e. a study programme has externally visible outcomes.

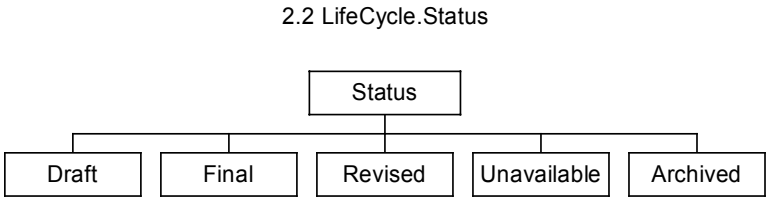

Figure 3. 2.2 LifeCycle.Status with Ontology.

Word	**Meaning**
Draft	A draft is an unpublished and uncompleted version of the study element.
Final	The term final refers to the fact that the study element is complete and published.
Revised	The term revised refers to the fact that the study element has been changed or updated after publication.
Unavailable	The study element cannot be retrieved when it is unavailable.
Archived	An archived study element has been saved for reuse. The study element is an older version of a current study element and may serve only as a resource for the learners of the current study element.

Note: This metadata element refers to the current state of the *study element itself*, not to the metadata record of the study element.

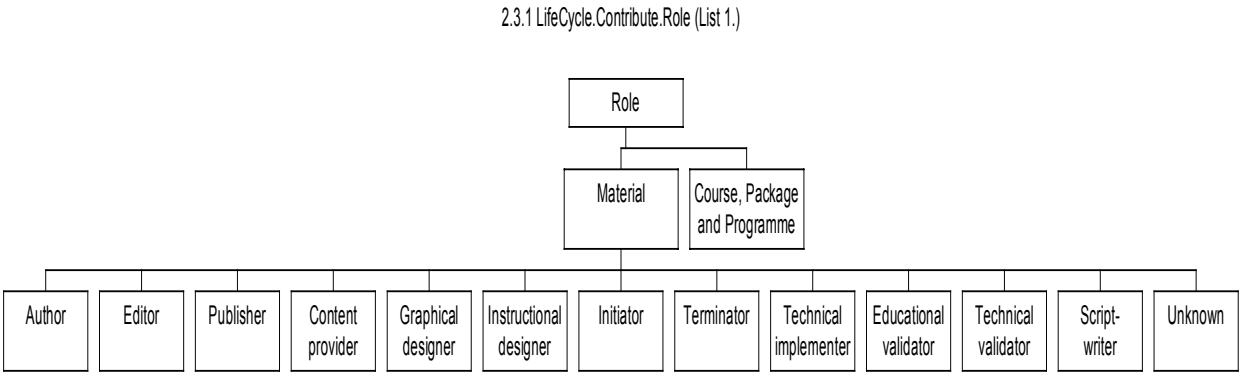

Figure 4. 2.3.1 LifeCycle.Contribute.Role with Ontology, List 1 for Material.

Word	**Meaning**
Author	An author is the creator of the material.
Editor	An editor has collected and reworked the material for publication.
Publisher	A publisher is a person or an institute who has published the material through some channel of communication.
Content provider	A content provider is a person who is responsible for the subject matter or the factual content of the material.
Graphical designer	A graphical designer is responsible for the graphical appearance of the material.
Instructional designer	An instructional designer is responsible for the educational guidelines of the material.
Initiator	An initiator is the person who has taken an initiative in order to contribute to the material.
Terminator	A terminator is a person that has completed or finished the material.
Technical implementer	A technical implementer is responsible for the technical realisation of the study material, e.g. programming.
Educational validator	An educational validator is responsible for the educational and pedagogical quality of the material.
Technical validator	A technical validator is responsible for the correctness and quality of the technical functionality of the study material.
Script writer	A scriptwriter is responsible for the manuscript and the overall planning of the material.
Unknown	The role of the contributor is not known or it cannot be categorised according to the classification of CUBER metadata.

Note: This ontology is used only on the aggregation level Material. The ontology of the metadata element 2.3.1 LifeCycle.Contribute.Role is defined separately for the aggregation levels Course, Package and Programme. (See figure 5.)

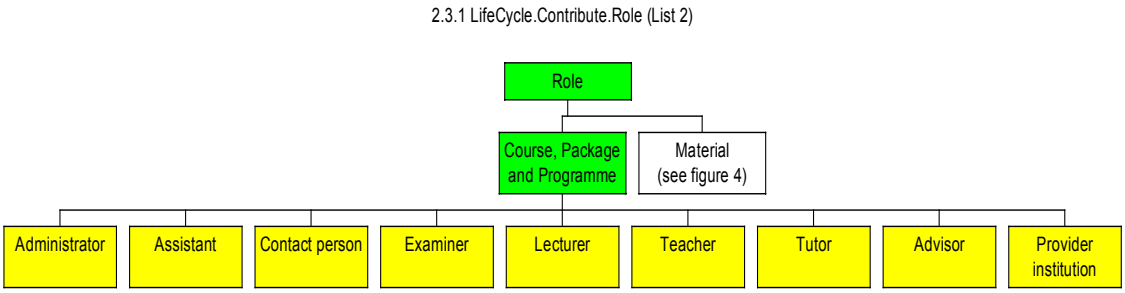

Figure 5. 2.3.1 LifeCycle.Contribute.Role with Ontology, List 2 for Course, Package and Programme.

Word	**Meaning**
Administrator	An administrator is a person who is responsible for the administration of the study element and for its planning.
Assistant	An assistant is a person who assists in the teaching and the administration of the study element.
Contact person	A contact person gives further information to the learners and is a link between the learners and the course providers. Learners should begin by contacting this person in order to gain further information on the study element in question.
Examiner	An examiner is a person who is responsible for the evaluation of student performance in the study element. The examiner assesses the skills and the learning results of the learners.
Lecturer	A lecturer is a person who gives the lectures of the study element.
Teacher	A teacher is a person who is responsible for the teaching activities of the study element.
Tutor	A tutor is a person who guides and supports the learning process of the learners. A tutor is different from a teacher and an assistant, because he or she only facilitates the learning process but does not actually teach.
Advisor	Advisor is a person who gives general guidance related to the studies or the study element in question.
Provider institution	A Provider institution is an educational organization that offers and arranges the study element.

Note: This ontology will be used only on the aggregation levels Course, Package and Programme. The ontology of the metadata element 2.3.1 LifeCycle.Contribute.Role is defined separately for the aggregation level Material. (See figure 4.)

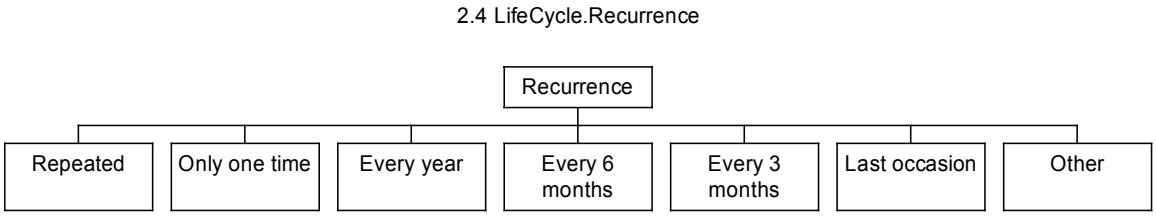

Figure 6. Metadata element 2.4 LifeCycle.Recurrence with Ontology.

Word	**Meaning**
Repeated	The study element can be arranged repeatedly.
Only one time	The study element cannot be repeated, i.e. it is unique.
Every year	The study element is repeated periodically each year.
Every 6 months	The study element is repeated periodically every six months.
Every 3 months	The study element is repeated periodically every three months.
Last occasion	The study element will not be repeated after this time, i.e. this is the last possibility to participate.
Other	The study element may be repeated according to a schedule different from the categorisation of CUBER metadata.

2.5.3 LifeCycle.Date.Kind

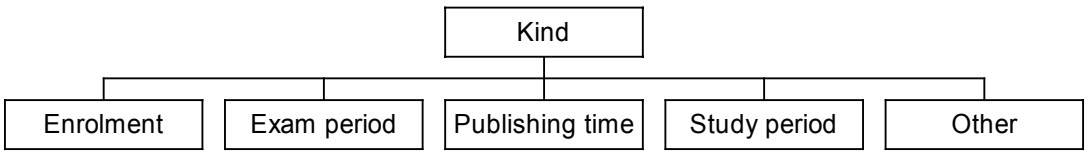

Figure 7. 2.5.3 LiceCycle.Date.Kind with Ontology.

Word	**Meaning**
Enrolment	The term enrolment date refers to the first and the last date(s) of enrolment in the study element.
Exam period	The term exam period refers to the date(s) of the exam(s) or the date(s) of the other kind(s) of assessment(s). When only the begin date is given, this date is the exact date of the exam or the assessment. When both the begin and the end dates are given, the time frame between these dates is an exam or assessment period.
Publishing time	The term publishing time refers to the publication date(s) of the study element. Only one date can be given, or both the begin and end dates, to define a time frame for publishing. In this latter case the publication time frame can announce for example the broadcasting period of a TV programme that is used as teaching material.
Study period	The term study period refers to the date(s) or the time frame of the actual teaching activities of the study element.
Other	Other important date that is not defined in the ontology of the CUBER metadata.

3.3.1 MetaMetadata.Contribute.Role

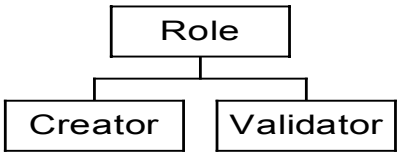

Figure 8. 3.3.1 MetaMetadata.Contribute.Role with Ontology.

Word	**Meaning**
Creator	A creator is a person who creates the metadata record of the study elements. A creator inputs the information of the study element to the CUBER system.
Validator	A validator checks the correctness of the metadata record.

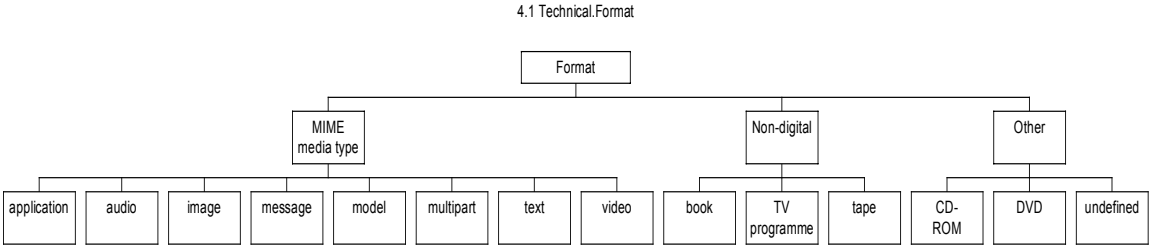

Figure 9. 4.1 Technical.Format with Ontology.

Note: This ontology is based on the Content types of IANA media types specification (RFC2048).

Source: http://www.isi.edu/in-notes/iana/assignments/media-types/media-types

4.4.1 Techical.Requirements.Type

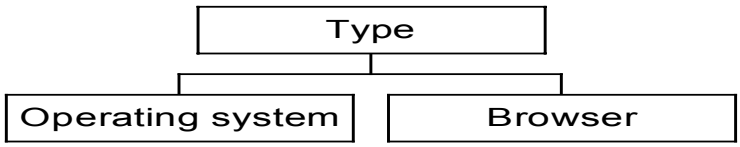

Figure 10. 4.4.1 Techical.Requirements.Type with Ontology.

Word	**Meaning**
Operating system	An operating system is a basic computer program that enables the use of application programs. It is the software that controls the execution of programs and that provides services such as resource allocation, scheduling input/output control, and data management.
Browser	A browser is the applications program that enables viewing WWW-pages on the Internet.

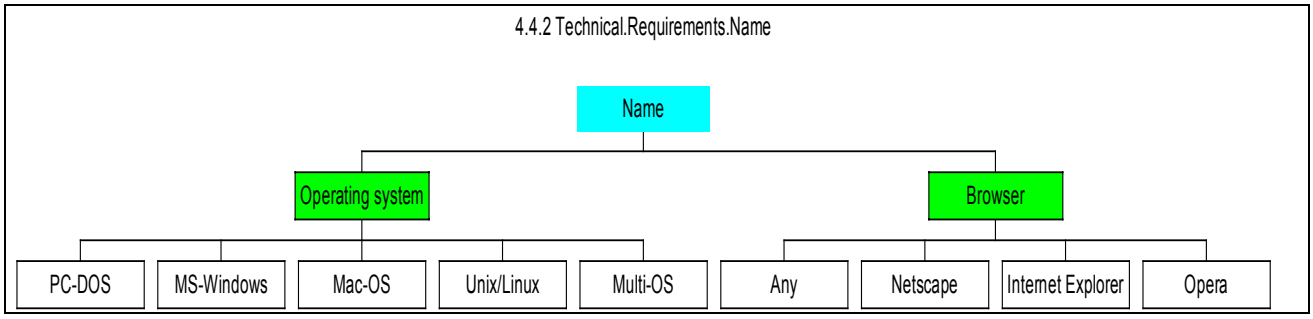

Figure 11. 4.4.2 Technical.Requirements.Name with Ontology.

Note: These names of the Operating systems and Browsers are product names.

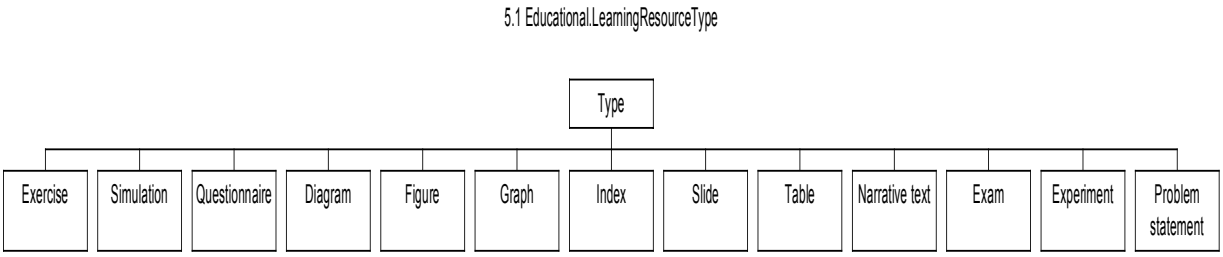

Figure 12. 5.2 Educational.LearningResourceType with Ontology.

Word	**Meaning**
Exercise	An exercise refers to a practice or a rehearsal of a specific topic, or solving a rather small problem.
Simulation	The technique of imitating artificially the behaviour of some real-life situation or process is called a simulation.
Questionnaire	A written inquiry that requires learners to answer questions is a questionnaire.
Diagram	An illustrative figure which, without representing the exact appearance of an object, gives an outline or general scheme of it, so as to exhibit the shape and relations of its various parts, is called a diagram.
Figure	A figure is a representation of a phenomenon that tries to illustrate the message of a written text.
Graph	A graph is a kind of symbolic diagram in which a system of relationships is expressed.
Index	An index is a table of contents prefixed to a book, a brief list or summary of the matters treated in it. Index is also an alphabetical list, placed (usually) at the end of a book, of the names, subjects, etc. occurring in it, with indication of the places in which they occur.
Slide	A slide is a transparency or a slide picture shown with an overhead projector.
Table	A table is an arrangement of numbers, words, or items of any kind, in a definite and compact form in columns and lines.
Narrative text	A narrative text is a text with informative verbal content.
Exam	An exam is an examination that measures the level of competence of the learner in some specific area of knowledge or expertise.
Experiment	An experiment is a test conducted in order to demonstrate or try out some phenomenon under realistic conditions.
Problem statement	An argument or a comment on a specific topic.

Source: http://dictionary.oed.com/

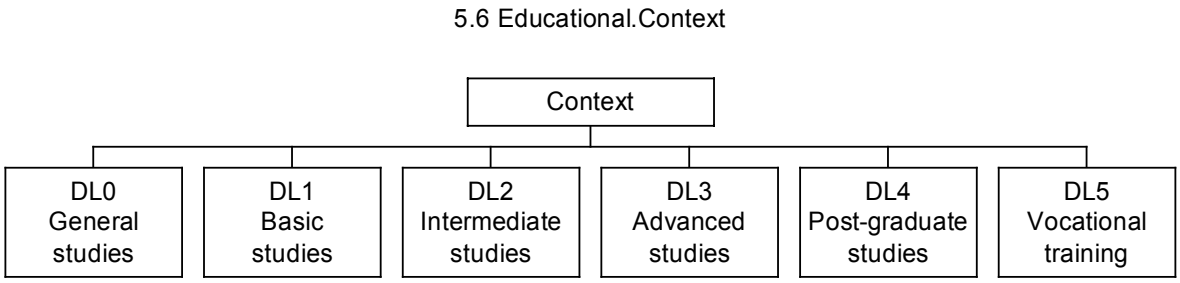

Figure 13. 5.6 Educational.Context with Ontology.

- **DL0 General studies, no official degree or open education**

  These studies can be courses that serve as orientation for the learner to more advanced university studies, or that lead to general competencies that may not be part of any specific degree but are needed for most degrees as prerequisites, such as scientific writing skills or ICT-English.

- **DL1 Basic studies, included in the Bachelor's degree**

  These studies are the first courses to be taken at the university and they are part of the basic university degree, which usually is the Bachelor's degree. These studies give the basic information and skills that are needed in intermediate studies.

- **DL2 Intermediate studies, included in the Bachelor's /Master's degree**

  These studies can usually be taken after the Basic studies, and they can be part of the Bachelor's or Master's degree. These studies are usually required as prerequisites for the Advanced studies.

- **DL3 Advanced studies, included in the Master's degree**

  These studies can be taken after completing Intermediate studies or the Bachelor degree. Advanced studies comprise the courses/packages required only for the Master's degree, not for the Bachelor's degree.

- **DL4 Post-graduate studies, included in the Licentiate/Doctoral degree**

  These studies can be taken after completing a Master's degree, and these studies can lead to a Licentiate or Doctoral degree.

- **DL5 Professional formation, Vocational training**

  These studies do not lead to a university degree. Instead they provide the learners with skills and knowledge that deepen or broaden their professional ability. These studies can include further education or updating training for some specific target groups with some previous degree or work experience, such as IT teachers or Web designers.

5.12.1 Educational.TeachingActivity.TeachingMethod

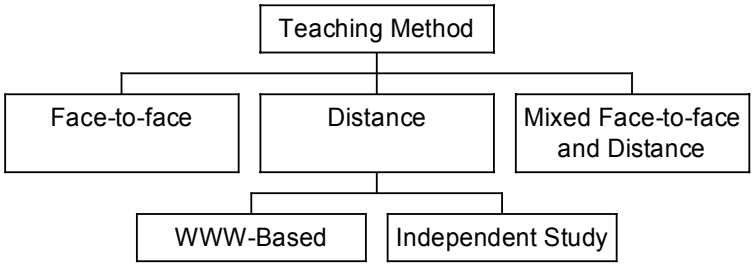

Figure 14. 5.12.1 Educational.TeachingActivity.TeachingMethod with Ontology.

Word	**Meaning**
Face-to-face	Face-to-face teaching is traditional teaching that is held at a certain time and at a certain place. It requires the presence of the teacher and the learners. Usually face-to-face teaching consists of lectures, seminars and exams.
Distance	Distance teaching is independent of place, and it may utilise communication technology to provide instruction at a distance. The learner and the teacher can be in different locations.
WWW-Based	WWW-based studies can be carried out entirely or almost entirely on the Web.
Independent Study	The learner studies independently by completing assignments and tasks. A teacher or a tutor may guide the otherwise self-directed studies of the learner.
Mixed Face-to-face and Distance	Mixed face-to-face and distance teaching consists of both types of teaching. The type of teaching may vary in different stages of studying.

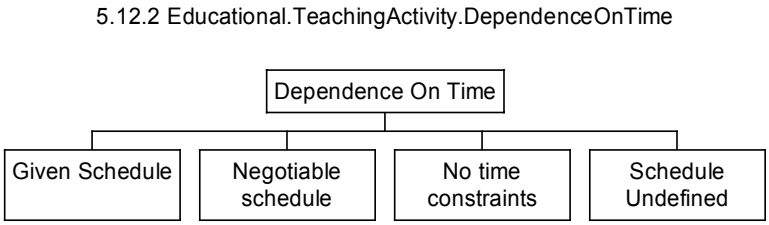

Figure 15. 5.12.2 Educational.TeachingActivity.DependenceOnTime with Ontology.

Word	**Meaning**
Given schedule	The schedule of the study element is determined in advance. The learners have to follow this schedule in their studies.
Negotiable schedule	The schedule of the study element may be determined in advance, but the learners may partially adapt the schedule to their own needs and possibilities to study.
No time constraints	There is no determined schedule for the study element. The learners may plan the timing of their studies completely on their own.
Schedule undefined	The type of the schedule of the study element is not defined according to the preceding categorisation of CUBER metadata.

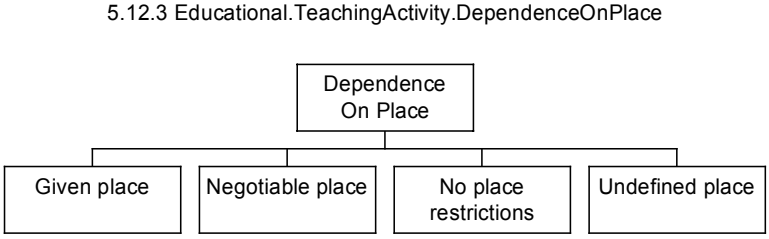

Figure 16. 5.12.3 Educational.TeachingActivity.DependenceOnPlace with Ontology.

Word	**Meaning**
Given place	The location of the study element is determined in advance. The learners have to do their studies on this given location.
Negotiable place	The location of the study element may be determined in advance, but the learners may partially adapt the location to their own needs. There may be some alternative locations for the study element.
No place restrictions	There is no determined place for the study element. The learners may choose the place for the studying completely on their own.
Undefined place	The type of the study element is not defined according to the preceding categorisation of CUBER metadata.

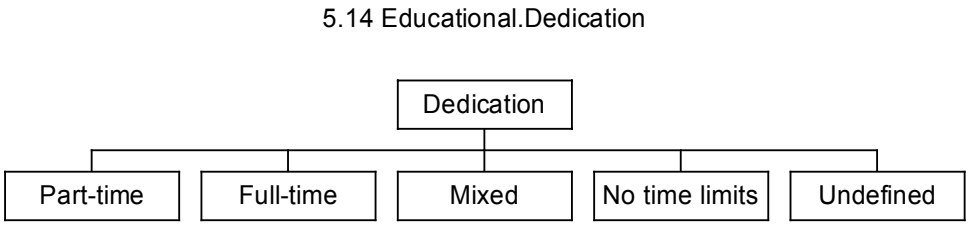

Figure 17. 5.14 Educational.Dedication with Ontology.

Word	**Meaning**
Part-time	Part-time studying requires less than 20 hours of work per week, and it is possible while working full-time.
Full-time	Full-time studies require more than 20 hours of work per week, and the learner must usually consider them as a priority.
Mixed part and full-time	Part-time and full-time studies may alternate during different stages or phases of studying.
No time limits	There are no fixed time limits for the study element, i.e. the learner may choose the rate of studying him/herself.
Undefined	The extent of dedication is not defined for the study element according to the categorisation of CUBER metadata.

5.15.1 Educational.Evaluation.Assessment

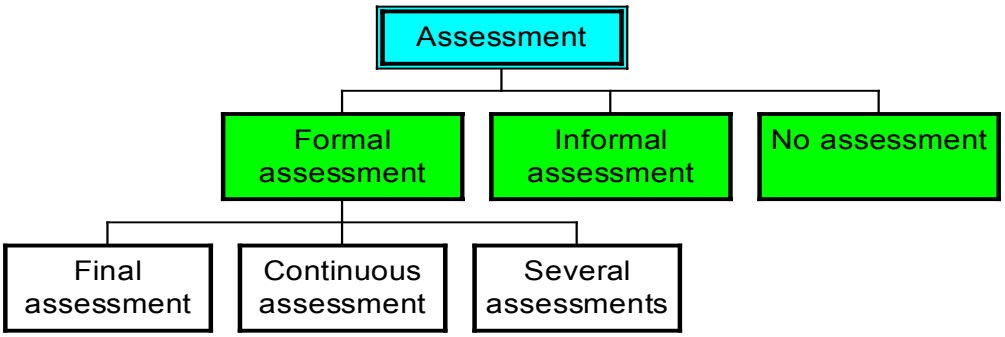

Figure 18. 5.15.1 Educational.Evaluation.Assessment with Ontology.

Word	**Meaning**
Formal assessment	The performance of the learner is measured and evaluated according to some formal criteria of an educational institution or a faculty. Usually a grade or a score is given as feedback, and stored in a record of study attainments.
Informal assessment	The performance of the learner is evaluated informally and some kind of feedback can be given. No official grade or score is given.
No assessment	There is neither measurement nor evaluation of the learner's performance.
Final assessment	The performance of the learner will be measured by one final exam or other method of evaluation at the end of the study element.
Continuous assessment	The performance of the learner will be measured and evaluated continuously throughout the study element.
Several assessments	The performance of the learner will be measured and evaluated at several points during the study element. There will be several exams or other methods of evaluation.

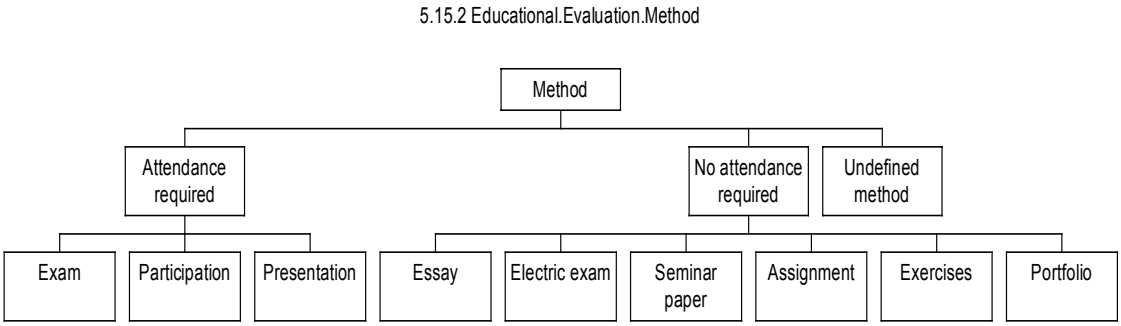

Figure 19. 5.15.2 Educational.Evaluation.Method with Ontology.

Word	**Meaning**
Attendance required	The evaluation of the study element requires attendance at a certain place at a certain moment.
No attendance required	The evaluation of the study element does not require attendance at a certain place at a certain moment.
Undefined method	The evaluation method of the study element is not defined according to the categorisation of CUBER metadata.
Exam	The action of testing or judging that is based on a standard or a rule. The aim of the exam is to measure the competencies and performance of the learner. It usually requires answering some questions in writing.
Participation	The evaluation is based on the learner's active participation in the teaching activities, e.g. seminars and discussions.
Presentation	The evaluation is based on a talk or a presentation prepared and given by the learner on a specific topic.
Essay	The evaluation is based on a written composition on a specific topic produced by the learner.
Electric exam	The examination of the study element can be carried out at a distance: on the Internet or with the help of ICT.
Seminar paper	The evaluation of the study element is based on a written report on some research work. Usually presented for a seminar group.
Assignment	The evaluation is based on a task assigned to the learner on a practical topic or a problem.
Exercises	The evaluation is based on the learner's performance on small tasks, practices, and rehearsals.
Portfolio	A portfolio is a selected collection of learner's accomplishments that includes samples of the learner's the learning results and demonstrations of the learner's competence. The form of the portfolio can vary: the portfolio can be an album, a briefcase, a video film, a diskette, or any other kind of collection of the learner's work.

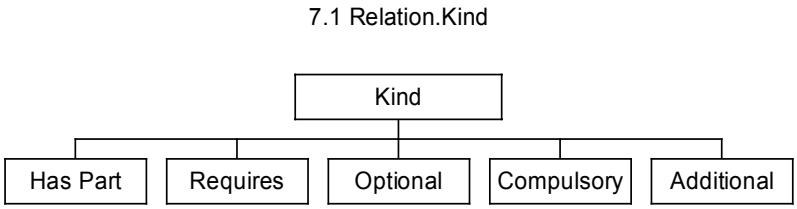

Figure 20. 7.1 Relation.Kind with Ontology.

Word	**Meaning**
Has Part	This relation defines the parts of an entity, e.g. this relation can be used to describe the study courses that belong to a certain study package.
Requires	This relation defines the prerequisites of a study element; e.g. the study courses that have to be taken before a certain more advanced study course.
Optional	This relation defines the study elements that are elective, i.e. the study elements that are not compulsory within a certain more extensive study element.
Compulsory	This relation defines the obligatory parts of a certain study element; e.g. the obligatory study courses within a study package.
Additional	This relation defines the extra study elements that the learner may utilise in the learning process, e.g. some extra study material that supports the learning targets of a certain study course.

Note: Only Has Part and Requires can be presented as relations. Optional, Compulsory, and Additional must be presented in free text due to the complexity of the information system.

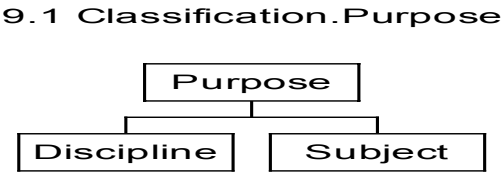

Figure 21. 9.1 Classification.Purpose with Ontology.

Word	**Meaning**
Discipline	General field of science or branch of research, such as Mathematics or Psychology.
Subject	Detailed branch of particular discipline or field of science (or branch of research), such as Algebra or Cognitive Psychology.

CUBER DISCIPLINES for Metadata 9.1 Classification.Purpose.Discipline

I HUMAN & SOCIAL & BEHAVIORAL SCIENCES

Anthropology
Culture and arts
Education
History
Language and literature
Law
Military science
Philosophy
Political science
Psychology
Religion
Social sciences
Welfare

II NATURAL SCIENCE

Agriculture and forestry
Astronomy
Biology
Botany
Chemistry
Geography
Geology
Mathematics
Medicine
Meteorology
Pharmacy
Physics
Statistics
Zoology

III ENGINEERING & ECONOMY

Economy
Accounting, finance and investment
Economics
E-commerce
Enterprise and small business
International business
Logistics and transportation

Marketing and advertising
Organisation and management
Trade

Technology
Aeronautics and astronautics
Architecture
Automation and systems technology
Chemical technology
Civil and environmental engineering
Electrical engineering and electronics
Environmental technology
Handicrafts
Hydraulic engineering
Industrial and manufacturing engineering
Mechanical engineering and machinery
Mining engineering and metallurgy
Motor vehicles
Transport engineering

IV ICT & COMPUTER SCIENCE

Algorithms
Applied computer science
Artificial intelligence and machine learning
Computer hardware
Computer organisation and architecture
Computer mathematics
Computer networks
Computer science (general)
Computer software
Data bases
Data communications
Data structures
Digital signal processing
Distributed systems
Formal methods
Human-computer interaction
Information and communication studies
Information management, storage and retrieval
Information systems
Intelligent systems
Internet application technologies
Neural networks
Media technologies
Mobile and wireless computing
Operating systems
Programming languages

Programming paradigms and techniques
Security and cryptography
Software engineering
Telecommunications
Theory of computing
Utility programs

THIS CLASSIFICATION OF DISCIPLINES IS BASED ON THE FOLLOWING EXISTING CLASSIFICATIONS.

These classifications have been partially integrated and complemented with CUBER's own disciplines.

Source 1) UNESCO Thesaurus http://www.ulcc.ac.uk/unesco/thesaurus.htm

Source 2) Library of Congress Classification Outline

http://lcweb.loc.gov/catdir/cpso/lcco/lcco.html

Source 3) IEEE Internet Computing Classification http://computer.org/Internet/xtras/

Source 4) Computer Science Classification System

http://www.ma.huji.ac.il/~library/classc.htm

Source 5) ACM Computing Classification System

http://www.acm.org/class/1998/overview.html

Source 6) ResearchIndex of the NECI Scientific Literature Digital Library

http://citeseer.nj.nec.com/directory.html

CUBER PROFILES FOR USERS AND COURSE PROVIDERS
Authors: Päivi Nykänen & Kirsi Pelto-Aho, UH/WP3

Provider.Person	**Explanation**	**Data type & Size**
Username	A unique identifier for the Course Provider person that works in a Course Provider Institution.	String (10 char)
V-card	Contact information	String (1000 char)

Provider.Institution	**Explanation**	**Data type & Size**
Identifier	A unique identifier for the Course Provider Institution.	String (15 char)
V-card	Contact information	String (1000 char)

Learner	**Explanation**	**Data type & Size**
Identifier	A unique identifier for the learner, e.g. username.	String (10 char)
Name	Name of the learner.	String (50 char)
Language *	Languages used by the learner.	String (25 char)
Degree *	Academic degree or other certificate of the learner.	String (20 char)
Contact information	Address, telephone numbers, e-mail etc. of the learner.	String (200 char)
Saved search *	Possibility to save old searches to the learner's user profile.	String (1000 char)

* = multiplicity

V-CARD INFORMATION

Name	String (1000 char)
Title/Degree	
Address	
Telephone number	
Fax number	
E-mail address	
URL of Web-page	

Appendix 3 The class diagram

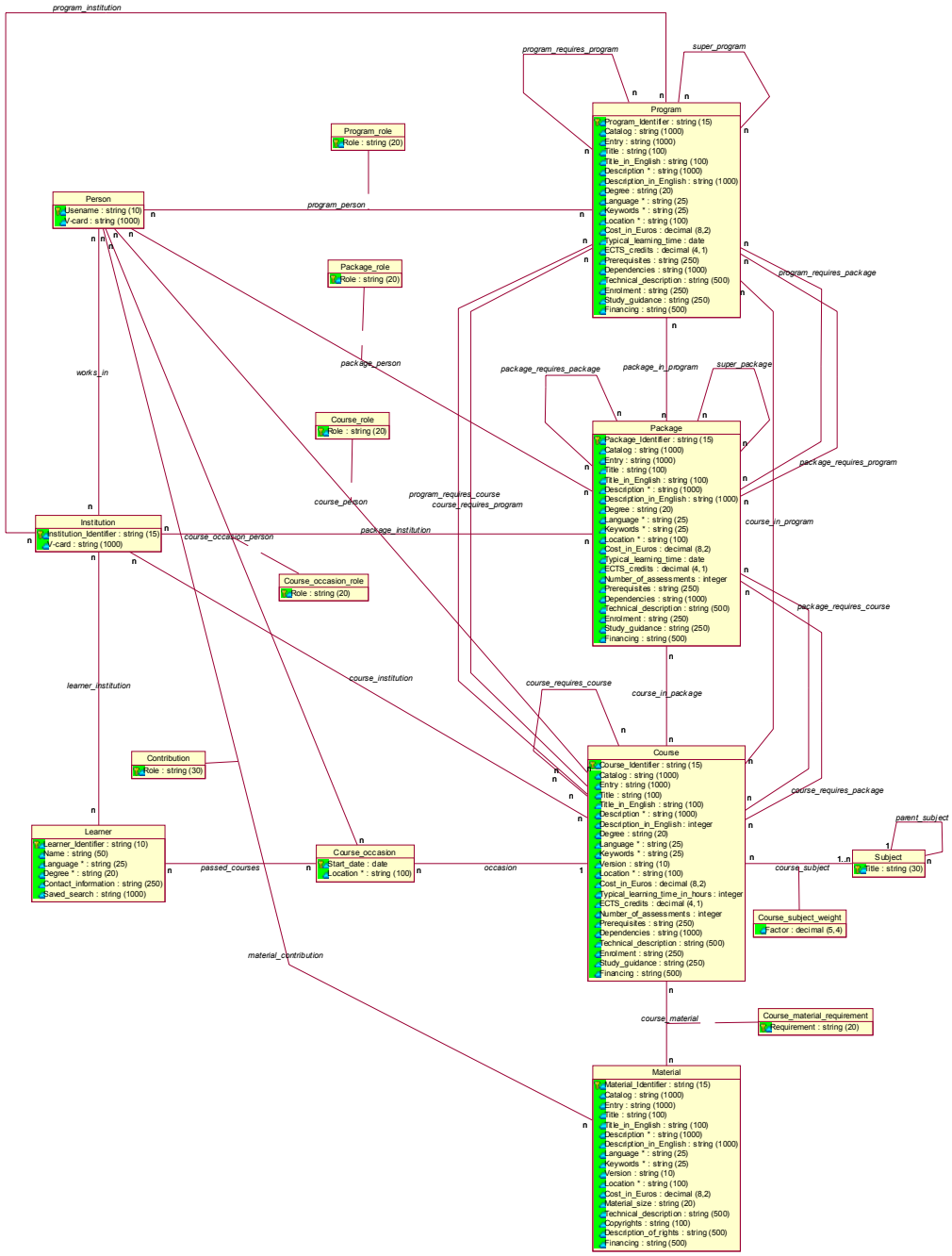

Figure 3 The whole class diagram without the ontology classes

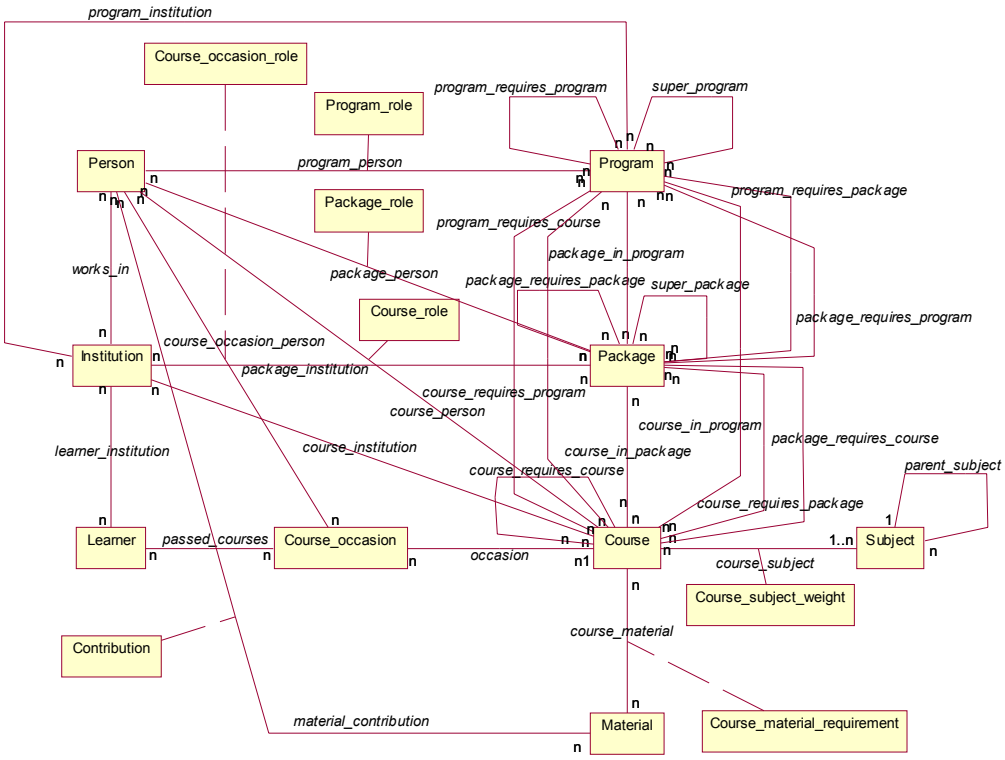

Figure 4 The main classes and relationships

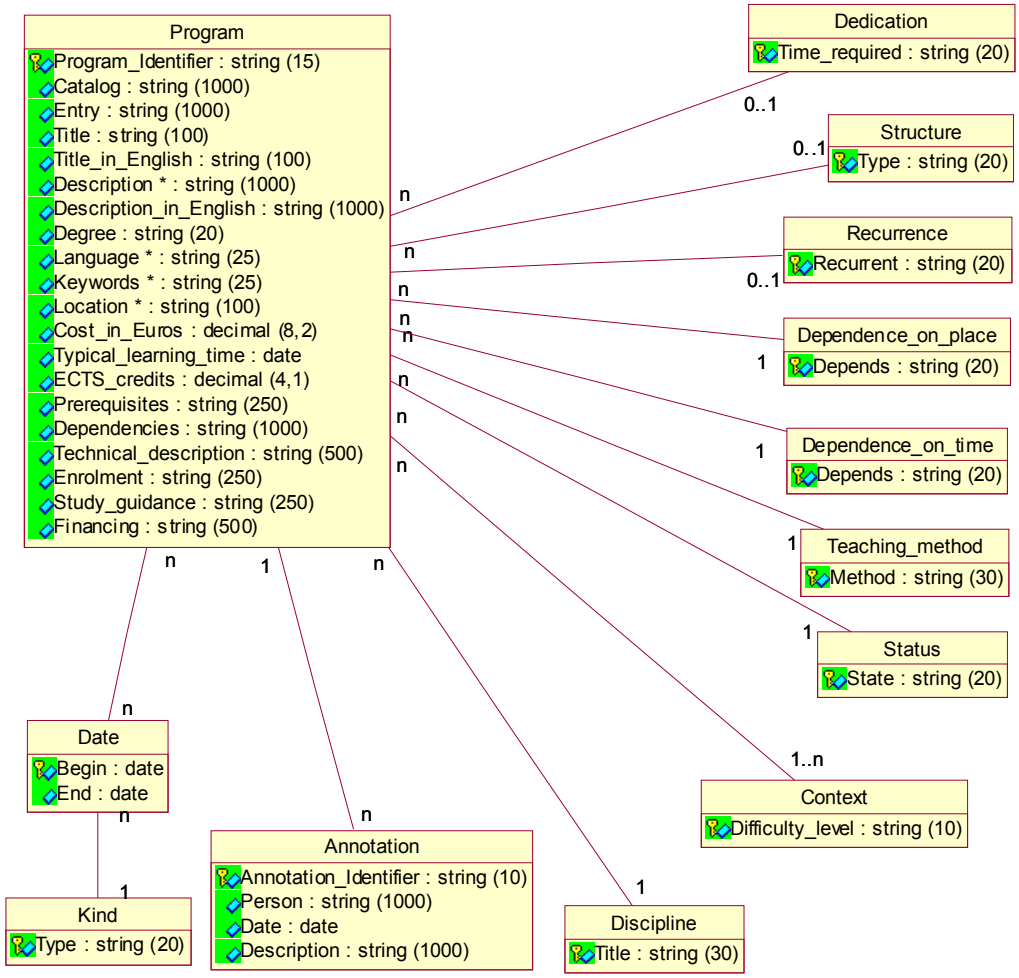

Figure 5 A detailed view of the class 'Program'

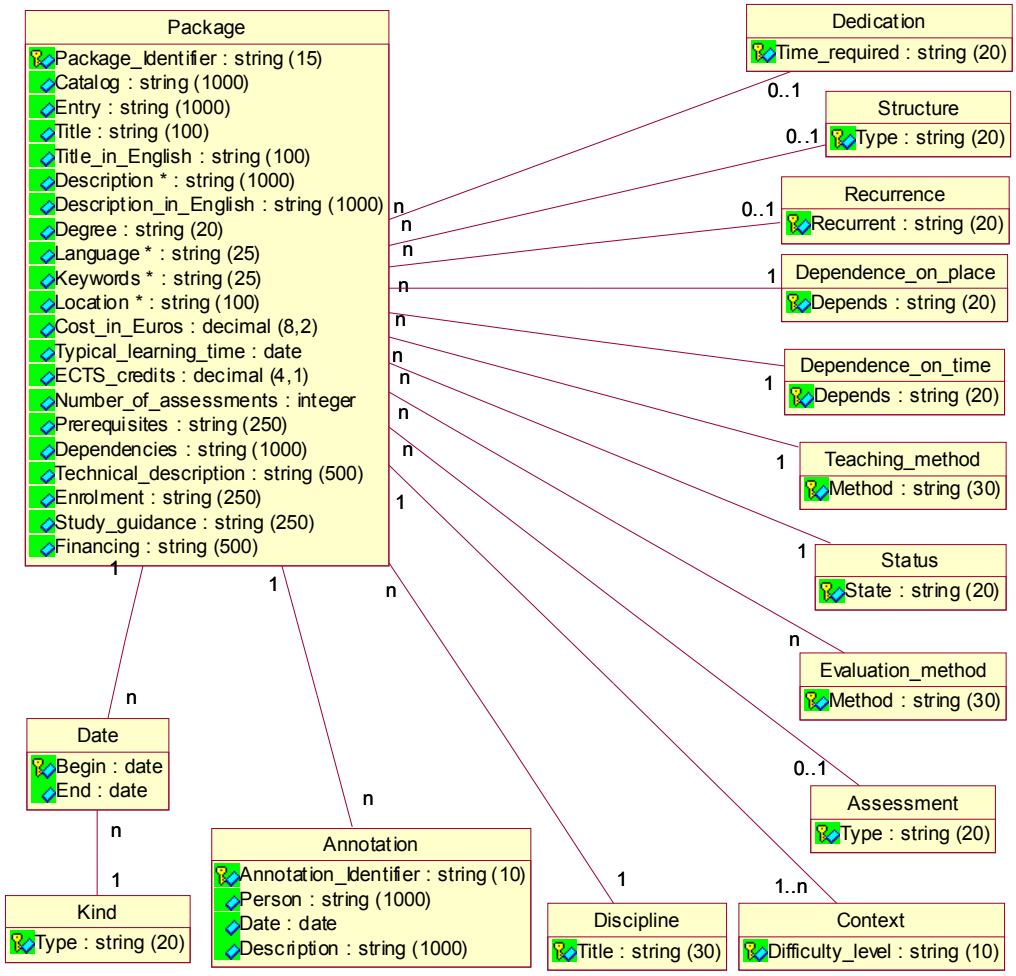

Figure 6 A detailed view of the class 'Package'

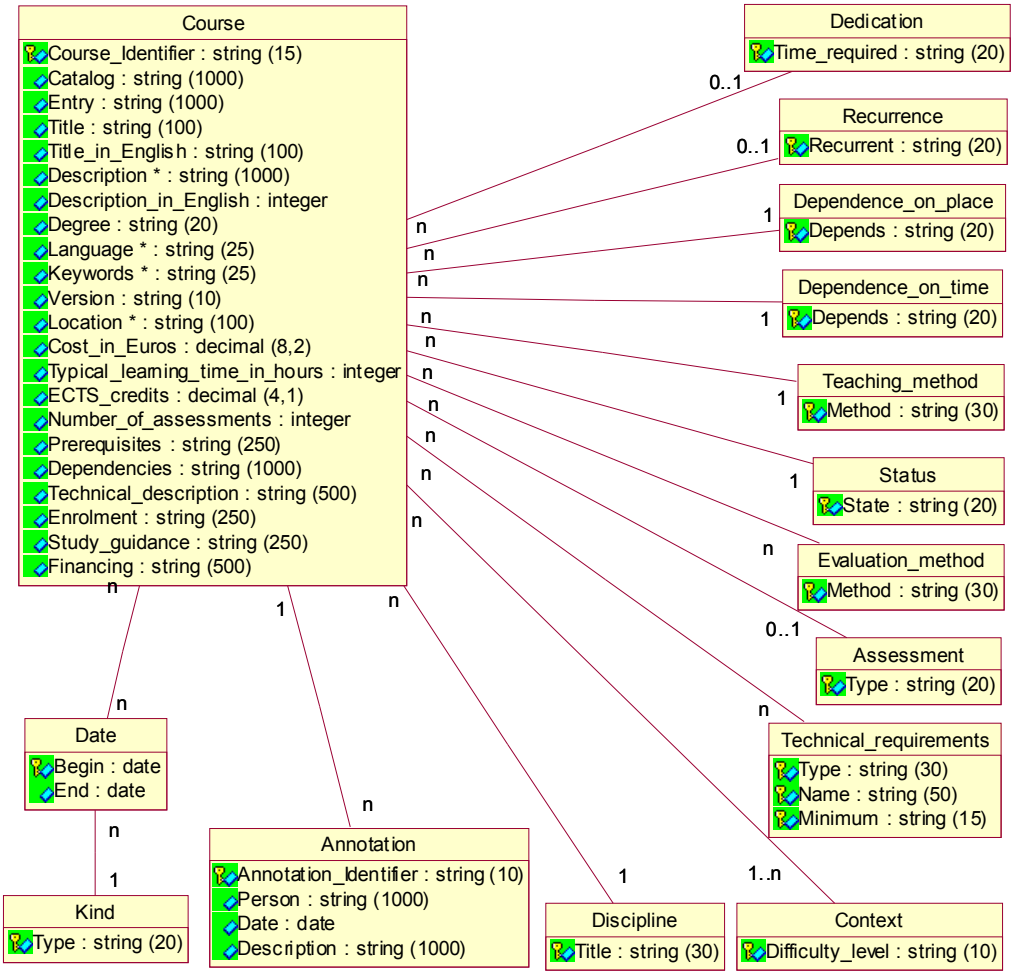

Figure 7 A detailed view of the class 'Course'

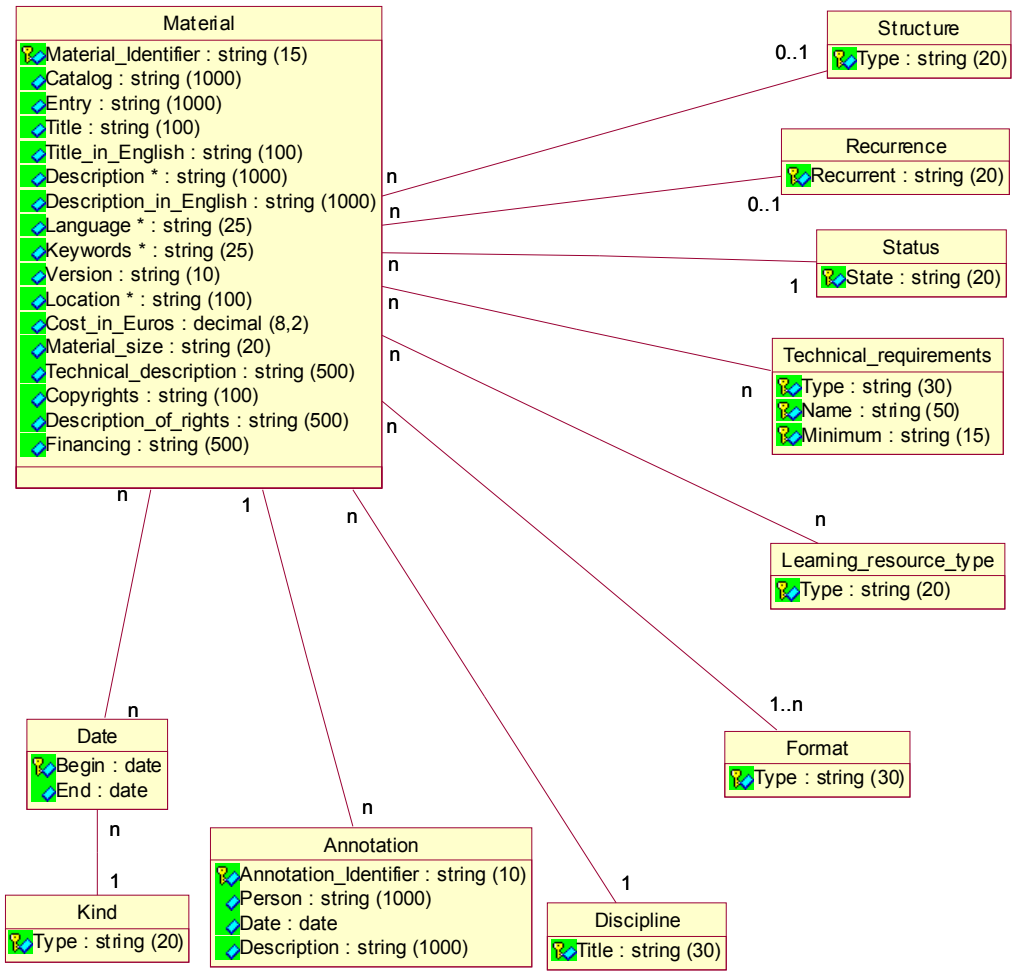

Figure 8 A detailed view of the class 'Material'

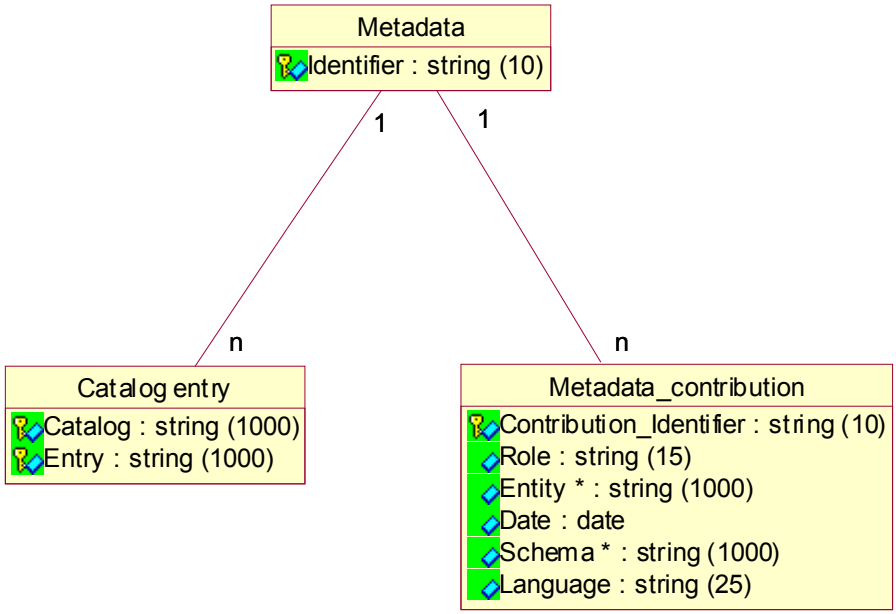

Figure 9 The class for metametadata

Appendix 4 The SQL definition

```
CREATE TABLE Course_occasion_role (
  role varchar(20) CHECK (role IN
('Administrator','Assistant','Contact
person','Examiner','Lecturer','Teacher','Tutor','Advisor')),
  username varchar(10),
  start_date date,
  course_identifier varchar(15),
  PRIMARY KEY (role, username, start_date, course_identifier),
  FOREIGN KEY (username)
    REFERENCES Person,
  FOREIGN KEY (start_date)
    REFERENCES Course_occasion
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE Course_role (
  role varchar(20) CHECK (role IN
('Administrator','Assistant','Contact
person','Examiner','Lecturer','Teacher','Tutor','Advisor')),
  username varchar(10),
  course_identifier varchar(15),
  PRIMARY KEY (role, username, course_identifier),
  FOREIGN KEY (username)
    REFERENCES Person,
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE Package_role (
  role varchar(20) CHECK (role IN
('Administrator','Assistant','Contact
person','Examiner','Lecturer','Teacher','Tutor','Advisor')),
  username varchar(10),
  package_identifier varchar(15),
  PRIMARY KEY (role, username, package_identifier),
  FOREIGN KEY (username)
    REFERENCES Person,
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE Program_role (
  role varchar(20) CHECK (role IN
('Administrator','Assistant','Contact
person','Examiner','Lecturer','Teacher','Tutor','Advisor')),
  username varchar(10),
  program_identifier varchar(15),
  PRIMARY KEY (role, username, program_identifier),
  FOREIGN KEY (username)
    REFERENCES Person,
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE Person (
  username varchar(10),
```

```
  v-card varchar(1000) NOT NULL,
  PRIMARY KEY (username)
);

CREATE TABLE Institution (
  institution_identifier varchar(15),
  v-card varchar(1000) NOT NULL,
  PRIMARY KEY (institution_identifier)
);

CREATE TABLE Learner (
  learner_identifier varchar(10),
  name varchar(50) NOT NULL,
  contact_information varchar(250),
  PRIMARY KEY (learner_identifier)
);

CREATE TABLE Learner_languages (
  language varchar(25),
  learner_identifier varchar(10),
  PRIMARY KEY (language, learner_identifier),
  FOREIGN KEY (learner_identifier)
    REFERENCES Learner
);

CREATE TABLE Learner_degrees (
  degree varchar(20),
  learner_identifier varchar(10),
  PRIMARY KEY (degree, learner_identifier),
  FOREIGN KEY (learner_identifier)
    REFERENCES Learner
);

CREATE TABLE Learner_searches (
  search varchar(1000),
  learner_identifier varchar(10),
  PRIMARY KEY (search, learner_identifier),
  FOREIGN KEY (learner_identifier)
    REFERENCES Learner
);

CREATE TABLE Course_occasion (
  start_date date,
  course_identifier varchar(15),
  PRIMARY KEY (start_date, course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
  );

CREATE TABLE Course_occasion_locations (
  address varchar(100),
  start_date date,
  course_identifier varchar(15),
  PRIMARY KEY (address, start_date, course_identifier),
  FOREIGN KEY (start_date)
    REFERENCES Course_occasion
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE Contribution (
```

```
  role varchar(30) CHECK (role IN
('Author','Editor','Publisher','Content provider','Graphical
designer','Instructional
designer','Initiator','Terminator','Technical
implementer','Educational validator','Technical validator','Script
writer','Unknown')),
  material_identifier varchar(15),
  username varchar(10),
  PRIMARY KEY (role, material_identifier, username),
  FOREGN KEY (material_identifier)
    REFERENCES Material,
  FOREIGN KEY (username)
    REFERENCES Person
);

CREATE TABLE Subject (
  title varchar (30),
  PRIMARY KEY (title)
);

CREATE TABLE parent_subject (
  parent_title varchar (30) NOT NULL,
  child_title varchar (30),
  PRIMARY KEY (child_title)
  FOREIGN KEY (child_title)
    REFERENCES Subject
);

CREATE TABLE Course_subject_weight (
  factor dec(5,4),
  course_identifier varchar(15),
  title varchar(30),
  PRIMARY KEY (course_identifier, title),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (title)
    REFERENCES Subject
);

CREATE TABLE Program (
  program_identifier varchar(15),
  catalog varchar(1000),
  entry varchar(1000),
  title varchar(100) NOT NULL,
  title_in_english varchar(100) NOT NULL,
  description_in_english varchar(1000) NOT NULL,
  degree varchar(20) NOT NULL,
  cost_in_euros dec(8,2) NOT NULL,
  typical_learning_time date,
  ects_credits decimal(4,1),
  prerequisites varchar(250),
  dependencies varchar(1000),
  technical_description varchar(500),
  enrolment varchar(250),
  study_guidance varchar(250),
  financing varchar(500),
  PRIMARY KEY (program_identifier)
);

CREATE TABLE Program_descriptions (
  description varchar(1000),
```

```
  program_identifier varchar(15),
  PRIMARY KEY (description, program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE Program_languages (
  language varchar(25),
  program_identifier varchar(15),
  PRIMARY KEY (language, program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE Program_keywords (
  keyword varchar(25),
  program_identifier varchar(15),
  PRIMARY KEY (keyword, program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE Program_locations (
  address varchar(100),
  program_identifier varchar(15),
  PRIMARY KEY (address, program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE program_requires_course (
  program_identifier varchar(15),
  course_identifier varchar(15),
  PRIMARY KEY (program_identifier, course_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE program_requires_package (
  program_identifier varchar(15),
  package_identifier varchar(15),
  PRIMARY KEY (program_identifier, package_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE program_requires_program (
  program_identifier varchar(15),
  required_program_identifier varchar(15),
  PRIMARY KEY (required_program_identifier),
  FOREIGN KEY (required_program_identifier)
    REFERENCES Program
);

CREATE TABLE Package (
  package_identifier varchar(15),
  catalog varchar(1000),
```

```
  entry varchar(1000),
  title varchar(100) NOT NULL,
  title_in_english varchar(100) NOT NULL,
  description_in_english varchar(1000) NOT NULL,
  degree varchar(20),
  cost_in_euros dec(8,2) NOT NULL,
  typical_learning_time date,
  ects_credits decimal(4,1),
  number_of_assessments int,
  prerequisites varchar(250),
  dependencies varchar(1000),
  technical_description varchar(500),
  enrolment varchar(250),
  study_guidance varchar(250),
  financing varchar(500),
  PRIMARY KEY (package_identifier)
);

CREATE TABLE Package_descriptions (
  description varchar(1000),
  package_identifier varchar(15),
  PRIMARY KEY (description, package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE Package_languages (
  language varchar(25),
  package_identifier varchar(15),
  PRIMARY KEY (language, package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE Package_keywords (
  keyword varchar(25),
  package_identifier varchar(15),
  PRIMARY KEY (keyword, package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE Package_locations (
  address varchar(100),
  package_identifier varchar(15),
  PRIMARY KEY (address, package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE package_requires_course (
  package_identifier varchar(15),
  course_identifier varchar(15),
  PRIMARY KEY (package_identifier, course_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE package_requires_package (
```

```
  package_identifier varchar(15),
  required_package_identifier varchar(15),
  PRIMARY KEY (required_package_identifier),
  FOREIGN KEY (required_package_identifier)
    REFERENCES Package
);

CREATE TABLE package_requires_program (
  package_identifier varchar(15),
  program_identifier varchar(15),
  PRIMARY KEY (package_identifier, program_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE Course (
  course_identifier varchar(15),
  catalog varchar(1000),
  entry varchar(1000),
  title varchar(100) NOT NULL,
  title_in_english varchar(100) NOT NULL,
  description_in_english varchar(1000) NOT NULL,
  degree varchar(20),
  version varchar(10),
  cost_in_euros dec(8,2) NOT NULL,
  typical_learning_time_in_hours int,
  ects_credits dec(4,1) NOT NULL,
  number_of_assessments int,
  prerequisites varchar(250),
  dependencies varchar(1000),
  technical_description varchar(500),
  enrolment varchar(250),
  study_guidance varchar(250),
  financing varchar(500),
  PRIMARY KEY (course_identifier),
);

CREATE TABLE Course_descriptions (
  description varchar(1000),
  course_identifier varchar(15),
  PRIMARY KEY (description, course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE Course_languages (
  language varchar(25),
  course_identifier varchar(15),
  PRIMARY KEY (language, course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE Course_keywords (
  keyword varchar(25),
  course_identifier varchar(15),
  PRIMARY KEY (keyword, course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
```

```
);

CREATE TABLE Course_locations (
  address varchar(100),
  course_identifier varchar(15),
  PRIMARY KEY (address, course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE course_requires_course (
  course_identifier varchar(15),
  required_course_identifier varchar(15),
  PRIMARY KEY (required_course_identifier),
  FOREIGN KEY (required_course_identifier)
    REFERENCES Course
);

CREATE TABLE course_requires_package (
  course_identifier varchar(15),
  package_identifier varchar(15),
  PRIMARY KEY (course_identifier, package_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE course_requires_program (
  course_identifier varchar(15),
  program_identifier varchar(15),
  PRIMARY KEY (course_identifier, program_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE Material (
  material_identifier varchar(15),
  catalog varchar(1000),
  entry varchar(1000),
  title varchar(100) NOT NULL,
  title_in_english varchar(100) NOT NULL,
  description_in_english varchar(1000),
  version varchar(10),
  cost_in_euros dec(8,2) NOT NULL,
  material_size varchar(20),
  technical_description varchar(500),
  copyrights varchar(100),
  description_of_rights varchar(500),
  financing varchar(500),
  PRIMARY KEY (material_identifier),
  );

CREATE TABLE Material_descriptions (
  description varchar(1000),
  material_identifier varchar(15),
  PRIMARY KEY (description, material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material
```

```
);

CREATE TABLE Material_languages (
  language varchar(25),
  material_identifier varchar(15),
  PRIMARY KEY (language, material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);

CREATE TABLE Material_keywords (
  keyword varchar(25),
  material_identifier varchar(15),
  PRIMARY KEY (keyword, material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);

CREATE TABLE Material_locations (
  address varchar(100),
  material_identifier varchar(15),
  PRIMARY KEY (address, material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);

CREATE TABLE Assessment (
  type varchar(20) CHECK (type IN ('Formal assessment','Informal
assessment','No assessment','Final assessment','Continuous
assessment','Several assessments')),
  PRIMARY KEY (type)
);

CREATE TABLE Structure (
  type varchar(20) CHECK (type IN
('Collection','Mixed','Linear','Hierarchical','Networked','Branched',
'Parcelled','Atomic')),
  PRIMARY KEY (type)
);

CREATE TABLE Dedication (
  time_required varchar(20) CHECK (time_required IN ('Part
time','Full time','Mixed part and full time','No time
limits','Undefined')),
  PRIMARY KEY (time_required)
);

CREATE TABLE Evaluation_method (
  method varchar(30) CHECK (method IN ('Attendance required','No
attendance required','Undefined
method','Exam','Participation','Presentation','Essay','Electric
exam','Seminar paper','Assignment','Exercises','Portfolio')),
  PRIMARY KEY (method)
);

CREATE TABLE Teaching_method (
  method varchar(30) CHECK (method IN ('Face-to-
face','Distance','WWW-based','Independent study','Mixed Face-to-face
and Distance')),
  PRIMARY KEY (method)
);
```

```
CREATE TABLE Dependence_on_time (
  depends varchar(20) CHECK (depends IN ('Given schedule','Negotiable
schedule','No time constraints','Schedule undefined')),
  PRIMARY KEY (depends)
);

CREATE TABLE Dependence_on_place (
  depends varchar(20) CHECK (depends IN ('Given place','Negotiable
place','No place restrictions','Undefined place')),
  PRIMARY KEY (depends)
);

CREATE TABLE Recurrence (
  recurrent varchar(20) CHECK (recurrent IN ('Repeated','Only one
time','Every year','Every 6 months','Every 3 months','Last
occasion','Other')),
  PRIMARY KEY (recurrent)
);

CREATE TABLE Status (
  state varchar(20) CHECK (state IN
('Draft','Final','Revised','Unavailable','Archived')),
  PRIMARY KEY (state)
);

CREATE TABLE Context (
  difficulty_level varchar(10) CHECK (difficulty_level IN ('DL0
General studies','DL1 Basic studies','DL2 Intermediate studies','DL3
Advanced studies','DL4 Post-graduate studies','DL5 Vocational
training')),
  PRIMARY KEY (difficulty_level)
);

CREATE TABLE Date (
  begin date,
  end date,
  type varchar(20),
  PRIMARY KEY (begin, type)
  FOREIGN KEY (type)
    REFERENCES Kind
);

CREATE TABLE Kind (
  type varchar(20) CHECK (type IN ('Enrolment','Exam
period','Publishing time','Study period','Other')),
  PRIMARY KEY (type)
);

CREATE TABLE Annotation (
  annotation_identifier varchar(10),
  person varchar(1000),
  date date,
  description varchar(1000),
  PRIMARY KEY (annotation_identifier)
);

CREATE TABLE program_annotation (
  program_identifier varchar(15),
  annotation_identifier varchar(10),
  PRIMARY KEY (annotation_identifier),
```

```
  FOREIGN KEY (program_identifier)
    REFERENCES Program
  FOREIGN KEY (annotation_identifier)
    REFERENCES Annotation
);

CREATE TABLE package_annotation (
  package_identifier varchar(15),
  annotation_identifier varchar(10),
  PRIMARY KEY (annotation_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
  FOREIGN KEY (annotation_identifier)
    REFERENCES Annotation
);

CREATE TABLE course_annotation (
  course_identifier varchar(15),
  annotation_identifier varchar(10),
  PRIMARY KEY (annotation_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
  FOREIGN KEY (annotation_identifier)
    REFERENCES Annotation
);

CREATE TABLE material_annotation (
  material_identifier varchar(15),
  annotation_identifier varchar(10),
  PRIMARY KEY (annotation_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material
  FOREIGN KEY (annotation_identifier)
    REFERENCES Annotation
);

CREATE TABLE Technical_requirements (
  type varchar(50) CHECK (type IN ('Operating system','Browser')),
  name varchar(50) CHECK (name IN ('PC-DOS','MS-Windows','Mac-
OS','Unix/Linux','Multi-OS','Any','Netscape','Internet
Explorer','Opera')),
  minimum varchar(15),
  PRIMARY KEY (type, name, minimum)
);

CREATE TABLE Discipline (
  title varchar(30),
  PRIMARY KEY (title)
);

CREATE TABLE Course_material_requirement (
  requirement varchar(20) CHECK (requirement IN
('Required','Optional','Additional')),
  course_identifier varchar(15),
  material_identifier varchar(15),
  PRIMARY KEY (requirement, course_identifier, material_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);
```

```
CREATE TABLE Learning_resource_type (
  type varchar(20) CHECK (type IN
('Excercise','Simulation','Questionnaire','Diagram','Figure','Graph',
'Index','Slide','Table','Narrative
text','Exam','Experimental','Problem statement','Self-assessment')),
  PRIMARY KEY (type)
);

CREATE TABLE Format (
  type varchar(30) CHECK (type IN
('application','audio','image','message','model','multipart','text','
video','book','TV-program','tape','CD-rom','DVD','undefined')),
  PRIMARY KEY (type),
);

CREATE TABLE works_in (
  username varchar(10),
  institution_identifier varchar(15),
  PRIMARY KEY (username, institution_identifier),
  FOREIGN KEY (username)
    REFERENCES Person,
  FOREIGN KEY (institution_identifier)
    REFERENCES Institution
);

CREATE TABLE program_institution (
  program_identifier varchar(15),
  institution_identifier varchar(15),
  PRIMARY KEY (program_identifier, institution_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (institution_identifier)
    REFERENCES Institution
);

CREATE TABLE package_institution (
  package_identifier varchar(15),
  institution_identifier varchar(15),
  PRIMARY KEY (package_identifier, institution_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (institution_identifier)
    REFERENCES Institution
);

CREATE TABLE course_institution (
  course_identifier varchar(15),
  institution_identifier varchar(15),
  PRIMARY KEY (course_identifier, institution_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (institution_identifier)
    REFERENCES Institution
);

CREATE TABLE learner_institution (
  learner_identifier varchar(10),
  institution_identifier varchar(15),
  PRIMARY KEY (learner_identifier, institution_identifier),
  FOREIGN KEY (learner_identifier)
```

```
    REFERENCES Learner,
  FOREIGN KEY (institution_identifier)
    REFERENCES Institution
);

CREATE TABLE passed_courses (
  course_identifier varchar(15),
  start_date date,
  username varchar(10),
  PRIMARY KEY (course_identifier, start_date, username),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (start_date)
    REFERENCES Course_occasion,
  FOREIGN KEY (username)
    REFERENCES Person
);

CREATE TABLE course_in_package (
  course_identifier varchar(15),
  package_identifier varchar(15),
  PRIMARY KEY (course_identifier, package_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (package_identifier)
    REFERENCES Package
);

CREATE TABLE package_in_program (
  packagecourse_identifier varchar(15),
  program_identifier varchar(15),
  PRIMARY KEY (package_identifier, program_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE course_in_program (
  course_identifier varchar(15),
  program_identifier varchar(15),
  PRIMARY KEY (course_identifier, program_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (program_identifier)
    REFERENCES Program
);

CREATE TABLE type_of_material (
  type varchar(20),
  material_identifier varchar(15),
  PRIMARY KEY (type, material_identifier),
  FOREIGN KEY (type)
    REFERENCES Learning_resource_type,
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);

CREATE TABLE material_format (
  type varchar(30),
  material_identifier varchar(15),
```

```
  PRIMARY KEY (type, material_identifier),
  FOREIGN KEY (type)
    REFERENCES Format,
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);

CREATE TABLE super_package (
  super_package_identifier varchar(15),
  sub_package_identifier varchar(15),
  PRIMARY KEY (super_package_identifier, sub_package_identifier),
  FOREIGN KEY (super_package_identifier)
    REFERENCES Package,
  FOREIGN KEY (sub_package_identifier)
    REFERENCES Package
);

CREATE TABLE super_program (
  super_program_identifier varchar(15),
  sub_program_identifier varchar(15),
  PRIMARY KEY (super_program_identifier, sub_program_identifier),
  FOREIGN KEY (super_program_identifier)
    REFERENCES Program,
  FOREIGN KEY (sub_program_identifier)
    REFERENCES Program
);

CREATE TABLE material_technical_requirements (
  material_identifier varchar(15),
  type varchar(50),
  name varchar(50),
  minimum varchar(15),
  PRIMARY KEY (type, name, minimum, material_identifier),
  FOREIGN KEY (type)
    REFERENCES Technical_requirements,
  FOREIGN KEY (name)
    REFERENCES Technical_requirements,
  FOREIGN KEY (minimum)
    REFERENCES Technical_requirements,
  FOREIGN KEY (material_identifier)
    REFERENCES Material
);

CREATE TABLE course_technical_requirements (
  course_identifier varchar(15),
  type varchar(50),
  name varchar(50),
  minimum varchar(15),
  PRIMARY KEY (type, name, minimum, course_identifier),
  FOREIGN KEY (type)
    REFERENCES Technical_requirements,
  FOREIGN KEY (name)
    REFERENCES Technical_requirements,
  FOREIGN KEY (minimum)
    REFERENCES Technical_requirements,
  FOREIGN KEY (course_identifier)
    REFERENCES Course
);

CREATE TABLE material_date (
  material_identifier varchar(15),
```

```
  begin date,
  type varchar(20),
  PRIMARY KEY (material_identifier, begin, type),
  FOREIGN KEY (material_identifier)
    REFERENCES Material
  FOREIGN KEY (begin)
    REFERENCES Date,
  FOREIGN KEY (type)
    REFERENCES Kind,
);

CREATE TABLE course_date (
  course_identifier varchar(15),
  begin date,
  type varchar(20),
  PRIMARY KEY (course_identifier, begin, type),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
  FOREIGN KEY (begin)
    REFERENCES Date,
  FOREIGN KEY (type)
    REFERENCES Kind,
);

CREATE TABLE package_date (
  package_identifier varchar(15),
  begin date,
  type varchar(20),
  PRIMARY KEY (package_identifier, begin, type),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
  FOREIGN KEY (begin)
    REFERENCES Date,
  FOREIGN KEY (type)
    REFERENCES Kind,
);

CREATE TABLE program_date (
  program_identifier varchar(15),
  begin date,
  type varchar(20),
  PRIMARY KEY (program_identifier, begin, type),
  FOREIGN KEY (program_identifier)
    REFERENCES Program
  FOREIGN KEY (begin)
    REFERENCES Date,
  FOREIGN KEY (type)
    REFERENCES Kind,
);

CREATE TABLE package_evaluation_method (
  package_identifier varchar(15),
  method varchar(30),
  PRIMARY KEY (package_identifier, method),
  FOREIGN KEY (package_identifier)
    REFERENCES Package
  FOREIGN KEY (method)
    REFERENCES Evaluation_method
);

CREATE TABLE course_evaluation_method (
```

```
  course_identifier varchar(15),
  method varchar(30),
  PRIMARY KEY (course_identifier, method),
  FOREIGN KEY (course_identifier)
    REFERENCES Course
  FOREIGN KEY (method)
    REFERENCES Evaluation_method
);

CREATE TABLE program_context (
  program_identifier varchar(15),
  difficulty_level varchar(10),
  PRIMARY KEY (program_identifier, difficulty_level),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (difficulty_level)
    REFERENCES Context
);

CREATE TABLE package_context (
  package_identifier varchar(15),
  difficulty_level varchar(10),
  PRIMARY KEY (package_identifier, difficulty_level),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (difficulty_level)
    REFERENCES Context
);

CREATE TABLE course_context (
  course_identifier varchar(15),
  difficulty_level varchar(10),
  PRIMARY KEY (course_identifier, difficulty_level),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (difficulty_level)
    REFERENCES Context
);

CREATE TABLE program_discipline (
  program_identifier varchar(15),
  title varchar(30) NOT NULL,
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (title)
    REFERENCES Discipline
);

CREATE TABLE package_discipline (
  package_identifier varchar(15),
  title varchar(30) NOT NULL,
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (title)
    REFERENCES Discipline
);

CREATE TABLE course_discipline (
  course_identifier varchar(15),
```

```
  title varchar(30) NOT NULL,
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (title)
    REFERENCES Discipline
);

CREATE TABLE material_discipline (
  material_identifier varchar(15),
  title varchar(30) NOT NULL,
  PRIMARY KEY (material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material,
  FOREIGN KEY (title)
    REFERENCES Discipline
);

CREATE TABLE program_status (
  program_identifier varchar(15),
  state varchar(20) NOT NULL,
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (state)
    REFERENCES Status
);

CREATE TABLE package_status (
  package_identifier varchar(15),
  state varchar(20) NOT NULL,
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (state)
    REFERENCES Status
);

CREATE TABLE course_status (
  course_identifier varchar(15),
  state varchar(20) NOT NULL,
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (state)
    REFERENCES Status
);

CREATE TABLE material_status (
  material_identifier varchar(15),
  state varchar(20) NOT NULL,
  PRIMARY KEY (material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material,
  FOREIGN KEY (state)
    REFERENCES Status
);

CREATE TABLE program_recurrence (
  program_identifier varchar(15),
  recurrent varchar(20),
```

```
  PRIMARY KEY (program_identifier, recurrent),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (recurrent)
    REFERENCES Recurrence
);

CREATE TABLE package_recurrence (
  package_identifier varchar(15),
  recurrent varchar(20),
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (recurrent)
    REFERENCES Recurrence
);

CREATE TABLE course_recurrence (
  course_identifier varchar(15),
  recurrent varchar(20),
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (recurrent)
    REFERENCES Recurrence
);

CREATE TABLE material_recurrence (
  material_identifier varchar(15),
  recurrent varchar(20),
  PRIMARY KEY (material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material,
  FOREIGN KEY (recurrent)
    REFERENCES Recurrence
);

CREATE TABLE program_teaching_method (
  program_identifier varchar(15),
  method varchar(30) NOT NULL,
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (method)
    REFERENCES Teaching_method
);

CREATE TABLE package_teaching_method (
  package_identifier varchar(15),
  method varchar(30) NOT NULL,
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (method)
    REFERENCES Teaching_method
);

CREATE TABLE course_teaching_method (
  course_identifier varchar(15),
  method varchar(30) NOT NULL,
  PRIMARY KEY (course_identifier),
```

```
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (method)
    REFERENCES Teaching_method
);

CREATE TABLE program_dedication (
  program_identifier varchar(15),
  time_required varchar(20),
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (time_required)
    REFERENCES Dedication
);

CREATE TABLE package_dedication (
  package_identifier varchar(15),
  time_required varchar(20),
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (time_required)
    REFERENCES Dedication
);

CREATE TABLE course_dedication (
  course_identifier varchar(15),
  time_required varchar(20),
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (time_required)
    REFERENCES Dedication
);

CREATE TABLE program_dependence_on_place (
  program_identifier varchar(15),
  depends varchar(20) NOT NULL,
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (depends)
    REFERENCES Dependence_on_place
);

CREATE TABLE package_dependence_on_place (
  package_identifier varchar(15),
  depends varchar(20) NOT NULL,
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (depends)
    REFERENCES Dependence_on_place
);

CREATE TABLE course_dependence_on_place (
  course_identifier varchar(15),
  depends varchar(20) NOT NULL,
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
```

```
    REFERENCES Course,
  FOREIGN KEY (depends)
    REFERENCES Dependence_on_place
);

CREATE TABLE program_dependence_on_time (
  program_identifier varchar(15),
  depends varchar(20) NOT NULL,
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (depends)
    REFERENCES Dependence_on_time
);

CREATE TABLE package_dependence_on_time (
  package_identifier varchar(15),
  depends varchar(20) NOT NULL,
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (depends)
    REFERENCES Dependence_on_time
);

CREATE TABLE course_dependence_on_time (
  course_identifier varchar(15),
  depends varchar(20) NOT NULL,
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (depends)
    REFERENCES Dependence_on_time
);

CREATE TABLE program_structure (
  program_identifier varchar(15),
  type varchar(20),
  PRIMARY KEY (program_identifier),
  FOREIGN KEY (program_identifier)
    REFERENCES Program,
  FOREIGN KEY (type)
    REFERENCES Structure
);

CREATE TABLE package_structure (
  package_identifier varchar(15),
  type varchar(20),
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (type)
    REFERENCES Structure
);

CREATE TABLE material_structure (
  material_identifier varchar(15),
  type varchar(20),
  PRIMARY KEY (material_identifier),
  FOREIGN KEY (material_identifier)
    REFERENCES Material,
```

```
  FOREIGN KEY (type)
    REFERENCES Structure
);

CREATE TABLE package_assessment (
  package_identifier varchar(15),
  type varchar(20),
  PRIMARY KEY (package_identifier),
  FOREIGN KEY (package_identifier)
    REFERENCES Package,
  FOREIGN KEY (type)
    REFERENCES Assessment
);

CREATE TABLE course_assessment (
  course_identifier varchar(15),
  type varchar(20),
  PRIMARY KEY (course_identifier),
  FOREIGN KEY (course_identifier)
    REFERENCES Course,
  FOREIGN KEY (type)
    REFERENCES Assessment
);

CREATE TABLE Metadata (
  identifier varchar(10),
  PRIMARY KEY (identifier)
);

CREATE TABLE Catalog_entry (
  catalog string(1000),
  entry string(1000),
  identifier varchar(10),
  PRIMARY KEY (catalog, entry, identifier),
  FOREIGN KEY (identifier)
    REFERENCES Metadata
);

CREATE TABLE Metadata_contribution (
  metadata_contribution_identifier varchar(10),
  identifier string(10) NOT NULL,
  role string(15) NOT NULL CHECK (role IN ('Creator','Validator')),
  date date NOT NULL,
  language varchar(25),
  PRIMARY KEY (metadata_contribution_identifier),
  FOREIGN KEY (identifier)
    REFERENCES Metadata
);

CREATE TABLE Metadata_person (
  metadata_contribution_identifier varchar(10),
  username string(15),
  PRIMARY KEY (metadata_contribution_identifier, username),
  FOREIGN KEY (metadata_contribution_identifier)
    REFERENCES Metadata_contribution,
  FOREIGN KEY (username)
    REFERENCES Person
);

CREATE TABLE Metadata_schema (
  metadata_contribution_identifier varchar(10),
```

```
  schema string(1000),
  PRIMARY KEY (metadata_contribution_identifier, schema),
  FOREIGN KEY (metadata_contribution_identifier)
    REFERENCES Metadata_contribution
);
```

Appendix 5 The XML schema

The DTD file

```
<!ELEMENT metadata (class)>

<!ELEMENT category (attribute+)>

<!ELEMENT attribute
(number,name,explanation,size,order,valuespace?,datatype?,example?,ag
gregationlevels?,mandatory?)>

<!ELEMENT number (#PCDATA)>

<!ELEMENT name (#PCDATA)>

<!ELEMENT explanation (#PCDATA)>

<!ELEMENT size (#PCDATA)>

<!ELEMENT order (#PCDATA)>

<!ELEMENT valuespace (#PCDATA)>

<!ELEMENT datatype (#PCDATA)>

<!ELEMENT example (#PCDATA)>

<!ELEMENT aggregationlevels (#PCDATA)>

<!ELEMENT mandatory (#PCDATA)>
```

The XML schema

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE metadata SYSTEM "metadata.dtd">
<metadata>
<category>

<attribute>
<number>1</number>
<name>General</name>
<explanation>General descriptive information on the learning object as a whole.</explanation>
<size>1</size>
<order>N/A</order>
</attribute>

<attribute>
<number>1.1</number>
<name>Identifier</name>
<explanation>A globally unique label for identifying the learning object.</explanation>
<size>1</size>
<order>N/A</order>
<datatype>Reserved.</datatype>
<example>Not Used.</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
```

```
</attribute>

<attribute>
<number>1.2</number>
<name>Title</name>
<explanation>Name of the learning object.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels> All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.3</number>
<name>CatalogEntry</name>
<explanation>Defines an entry within a catalog assigned to this learning object.</explanation>
<size>Smallest permitted max: 10 items.</size>
<order>No.</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>1.3.1</number>
<name>Catalog</name>
<explanation>The name of the catalog (i.e. the listing identification system).</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>Characterstring (smallest permitted max: 1000 char).</datatype>
<example>ISBN, ARIADNE</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>1.3.2</number>
<name>Entry</name>
<explanation>Actual string value of the entry within the catalog defined in 1.3.1.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>1.4</number>
<name>Language</name>
<explanation>The primary human language(s) used within this L.O.</explanation>
<size>Smallest permitted max: 10 items.</size>
<order>No</order>
```

<valuespace>LanguageID =Langcode (´-´Subcode)*,ISO 639, ISO 3166</valuespace>
<datatype>Characterstring (smallest permitted max: 100 char)</datatype>
<example>"en"; "en-GB"; "de"; "fr-CA"; "it"</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.5</number>
<name>Description</name>
<explanation>A textual description of the content of this L.O.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 2000 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes for Course, Package and Program</mandatory>
</attribute>

<attribute>
<number>1.6</number>
<name>Keywords</name>
<explanation>Keywords or phrases describing this L.O.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>Langstring (1000 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.8</number>
<name>Structure</name>
<explanation>Underlying organisational structure of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Collection, Mixed, Linear, Hierarchical, Networked, Branched, Parcelled, Atomic</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Material, Package, Program</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>1.10</number>
<name>Provider</name>
<explanation>Information on the provider or organiser of the L.O.</explanation>
<size>N</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>1.10.1</number>
<name>Institution</name>
<explanation>The name and other information on the institution that organises or provides this L.O.</explanation>
<size>N</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>University of X</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.10.2</number>
<name>Person</name>
<explanation>Information on the persons related to the L.O.</explanation>
<size>N</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example></example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>1.10.2.1</number>
<name>Role</name>
<explanation>The roles of the persons involved in providing the L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Administrator, Advisor, Assistant, Contact person, Examiner, Lecturer, Teacher, Tutor</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.10.2.2</number>
<name>Information</name>
<explanation>Information on the person related to the L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.11</number>
<name>CUBER Identifier</name>
<explanation>A label for identifying the L.O. Valid only within CUBER.</explanation>
<size>1</size>

<order>N/A</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>Characterstring (min-max: 100 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.12</number>
<name>CUBER Aggregation</name>
<explanation>The functional granularity of the L.O.s included in CUBER.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>0=material, 1=course, 2=package, 3=program</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.13</number>
<name>Date</name>
<explanation>The time span or important dates of the L.O. </explanation>
<size>N</size>
<order>-</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>1.13.1</number>
<name>Begin</name>
<explanation>The begin date of the L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Date</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.13.2</number>
<name>End</name>
<explanation>The end date of the L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Date</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>1.13.3</number>
<name>Kind</name>
<explanation>The nature of contribution or action required with regard to the dates announced.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Enrolment, Exam period, Publishing time, Study period, Other</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>ALl</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>1.14</number>
<name>Title in English</name>
<explanation>Name of the L.O. in English.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>Java programming</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>2</number>
<name>LifeCycle</name>
<explanation>This category describes the history and current state of this L.O. and its contributors.</explanation>
<size>1</size>
<order>N/A</order>
</attribute>

<attribute>
<number>2.1</number>
<name>Version</name>
<explanation>The edition of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:50 char)</datatype>
<example>3.0, 3.1, 1.2alpha</example>
<aggregationlevels>Material, Course</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>2.2</number>
<name>Status</name>
<explanation>The state or condition of this L.O. </explanation>
<size>1</size>

<order>N/A</order>
<valuespace>Draft, Final, Revised, Archived, Unavailable</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>2.3</number>
<name>Contribute</name>
<explanation>This element describes the people and organisations that have affected the state of this L.O.</explanation>
<size>Smallest permitted maximum 30 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>2.3.1</number>
<name>Role</name>
<explanation>Kind of contribution.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Author, Editor, Publisher, Content provider, Graphical designer, Instructional designer, Initiator, Terminator, Technical implementer, Educational validator, Technical validator, Script writer, Unknown</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Material</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>2.3.2</number>
<name>Entity</name>
<explanation>Information on people and organisations contributing to this L.O.</explanation>
<size>Smallest permitted maximum 40 items</size>
<order>Yes</order>
<valuespace>V-card</valuespace>
<datatype>Characterstring (smallest permitted max: 1000 chars)</datatype>
<example>-</example>
<aggregationlevels>Material</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>2.4</number>
<name>Recurrence</name>
<explanation>This element indicates whether the L.O. is unique or repeated periodically.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Repeated, One time study element, Every 3 or 6 months, Every year, Last occasion</valuespace>
<datatype>Vocabulary; see Ontology</datatype>

<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>3</number>
<name>MetaMetadata</name>
<explanation>This category describes the M.D. record itself.</explanation>
<size>1</size>
<order>N/A</order>
</attribute>

<attribute>
<number>3.1</number>
<name>Identifier</name>
<explanation>A globally unique label that identifies this M.D. record.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Reserved</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>3.2</number>
<name>CatalogEntry</name>
<explanation>This element describes an entry within a catalog given to the M.D. instance.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>3.2.1</number>
<name>Catalog</name>
<explanation>The name of the catalog (i.e. listing identification system).</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>Characterstring (smallest permitted max: 1000 char)</datatype>
<example>Ariadne</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>

<number>3.2.2</number>
<name>Entry</name>
<explanation>Actual string value of the entry in the catalog.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>(en,KUL532)</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>3.3</number>
<name>Contribute</name>
<explanation>Describes the people and organisations that have affected to the state of this M.D.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>Yes</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>3.3.1</number>
<name>Role</name>
<explanation>Kind of contribution.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Creator, Validator</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>3.3.2</number>
<name>Entity</name>
<explanation>Information on the people and organisations contributing this M.D. instance.</explanation>
<size>Smallest permitted max: 10 itmes</size>
<order>Yes</order>
<valuespace>V-card</valuespace>
<datatype>CharacterString (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>3.3.3</number>
<name>Date</name>
<explanation>The date of the contribution.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>

<datatype>Date</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>3.4</number>
<name>Metadata Scheme</name>
<explanation>The name and version of the authoritative specification used to create this M.D. instance.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>No</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>Characterstring (smallest permitted max: 30 char)</datatype>
<example>LOM-1.0</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>3.5</number>
<name>Language</name>
<explanation>Language of this M.D. instance. Default value for all the Langstring values in this M.D. instance.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>LanguageID= Langcode (´-´Subcode)*, ISO 639, ISO 3166</valuespace>
<datatype>Characterstring (smallest permitted max: 100 char)</datatype>
<example>"en", Default in CUBER is English.</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>4</number>
<name>Technical</name>
<explanation>This category described the technical requirements and characteristics of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
</attribute>

<attribute>
<number>4.1</number>
<name>Format</name>
<explanation>Technical data type(s) of this L.O.</explanation>
<size>Smallest permitted max: 40 items</size>
<order>No</order>
<valuespace>MIME types based on IANA registration (see RFC2048) or "non-digital"; see Vocabulary & Ontology</valuespace>
<datatype>Characterstring (smallest permitted max: 500 char)</datatype>
<example>Video/mpg, Text/html, Application/x-toolbook, Book</example>
<aggregationlevels>Material</aggregationlevels>

<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>4.3</number>
<name>Location</name>
<explanation>A string to access this L.O. Physical location of the L.O. Exact location or method to locate.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>Yes</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>Characterstring (smallest permitted max: 1000 char)</datatype>
<example>Http://host/id</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>4.4</number>
<name>Requirements</name>
<explanation>Describes the technical capabilities required in order to use this L.O.</explanation>
<size>Smallest permitted max: 40 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>4.4.1</number>
<name>Type</name>
<explanation>The technology required to use this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Operating system, Browser</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Material, Course</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>4.4.2</number>
<name>Name</name>
<explanation>Name of the technology required to use this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>If Type =Operating system, then: PC-DOS, MS-Windows, MacOS, Unix, Multi-OS, None; If Type=Browser, then: Any, Netscape, Internet Explorer, Opera</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Material, Course</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>4.4.3</number>

<name>Minimum Version</name>
<explanation>Lowest possible version of the required technology to use this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>CharacterString (smallest permitted max: 30 char)</datatype>
<example>-</example>
<aggregationlevels>Material, Course</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>4.8</number>
<name>Material Size</name>
<explanation>Size of a digital or a non-digital L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>150 pages, 11700 words, 50 kB, 1H 15Min</example>
<aggregationlevels>Material</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>4.9</number>
<name>Description</name>
<explanation>Further description on the technical characteristics of the L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>Guidelines and commands for using Unix.</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>5</number>
<name>Educational</name>
<explanation>This category describes the key educational and pedagogical characteristics of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
</attribute>

<attribute>
<number>5.2</number>
<name>Learning Resource Type</name>
<explanation>Specific kind of L.O.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>Yes</order>

<valuespace>Exercise, Simulation, Questionnaire, Diagram, Figure, Graph, Index, Slide, Table, Narrative text, Exam, Experiment, ProblemStatement, SelfAssessment</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Material</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.6</number>
<name>Context</name>
<explanation>The principal environment within which the use of this L.O. is intended to take place.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>No</order>
<valuespace>DL0 General studies, DL1 Basic, Bac., DL2 Intermediate, Bachelor, DL3 Advanced, Master, DL4 Post-graduate, L/D DL5 Vocational, further education</valuespace>
<datatype>Vocabulary; See Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes for Course, Package and Program</mandatory>
</attribute>

<attribute>
<number>5.9</number>
<name>Typical Learning Time</name>
<explanation>Typical time it takes to work through this L.O. (e.g. hours, days, weeks, months)</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Time, Date</datatype>
<example>Used in CUBER only when ECTS not available.</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.12</number>
<name>Teaching Activity</name>
<explanation>Description of principal teaching activities used for this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>5.12.1</number>
<name>Teaching Method</name>
<explanation>This sub-element describes the principal teaching method used for this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Face-to-face, Distance (www-based), Distance (independent), Mixed face-to face and distance, Undefined</valuespace>
<datatype>Vocabulary; see Ontology</datatype>

<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>5.12.2</number>
<name>Dependence on time </name>
<explanation>This sub-element describes the dependence on time of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Given schedule, Negotiable schedule, No time-restrictions, Undefined</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>5.12.3</number>
<name>Dependence on place </name>
<explanation> This sub-element describes the dependence on place of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Given place, Negotiable place, No place-restrictions, Undefined</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>5.13</number>
<name>ECTS Credits</name>
<explanation>This element describes the ECTS credits of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Characterstring (smallest permitted max: 30 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes for Course</mandatory>
</attribute>

<attribute>
<number>5.14</number>
<name>Dedication</name>
<explanation>This sub-element describes how intensively the learner must work.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Part-time, Full-time, Mixed (part&full), No time limits, Undefined</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.15</number>
<name>Evaluation</name>

<explanation>This element describes the principal method(s) and amount of evaluation as for this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>5.15.1</number>
<name>Assessment</name>
<explanation>This sub-element describes the assessment related to this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Formal assessment, Informal assessment, Final assessment, Continuous assesment, Several assessment, No assesment, Undefined</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.15.2</number>
<name>Method</name>
<explanation>The principal method(s) of assessment for this L.O.</explanation>
<size>N</size>
<order>No</order>
<valuespace>Exam with attendance, Electric exam in distance, Exercises, Assignment, Participation, Presentation, Essay, Seminar paper, Portfolio, Undefined</valuespace>
<datatype>Vocabulary; see Ontology</datatype>
<example>-</example>
<aggregationlevels>Course, Package</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.15.3</number>
<name>Number</name>
<explanation>The number of tasks or exams that form the basis for evaluation for this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>ISO 646, but only digits ´0´...´9´</valuespace>
<datatype>Characterstring (smallest permitted max: 30 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.16</number>
<name>Enrolment</name>
<explanation>This element holds information on the enrolment, e.g. method of enrolment.</explanation>
<size>1</size>
<order>N/A</order>

<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.17</number>
<name>Study Guidance</name>
<explanation>This element describes the guidance or tutoring provided for the learner.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.18</number>
<name>Pre-requisites</name>
<explanation>This element describes the required skills in order to take this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>Good computer literacy; Fluent German.</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>5.19</number>
<name>Degree</name>
<explanation>The official degree related to this study element. </explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max: 1000 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes for program</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>6</number>
<name>Rights</name>
<explanation>This category describes the intellectual property rights and the conditions of use for this L.O.</explanation>
<size>1</size>
<order>N/A</order>
</attribute>

<attribute>
<number>6.2</number>
<name>Copyright</name>
<explanation>Copyright and other restrictions on use of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Yes, No</valuespace>
<datatype>Vocabulary</datatype>
<example>-</example>
<aggregationlevels>Material</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>6.3</number>
<name>Description</name>
<explanation>Comments on conditions of use of this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>-</example>
<aggregationlevels>Material</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>6.4</number>
<name>Cost in EURO</name>
<explanation>The amount of payment in EURO.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Characterstring (smallest permitted max: 100 char)</datatype>
<example>100 EURO</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>6.5</number>
<name>Financing</name>
<explanation>The financing possibilities or grants available for the learner.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>7</number>
<name>Relation</name>
<explanation>This category describes the relationships between this L.O. and other L.O.s</explanation>
<size>Smallest permitted max: 100 items</size>
<order>No</order>
</attribute>

<attribute>
<number>7.1</number>
<name>Kind</name>
<explanation>Nature of relationship between this L.O. and the target L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>HasPart, Requires</valuespace>
<datatype>Vocabulary, see Ontology</datatype>
<example>NOTE2: HasPart and Requires are used in CUBER.
Optional, Additional, Compulsory and Exchangeable can be used too but only in free text.
</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>Yes for package and program</mandatory>
</attribute>

<attribute>
<number>7.2</number>
<name>Resource</name>
<explanation>The target L.O. that this relationship references.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>7.2.1</number>
<name>Identifier</name>
<explanation>Unique identifier of the target L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Reserved</datatype>
<example>Not used.</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>7.2.3</number>
<name>CatalogEntry</name>
<explanation>Defines an entry within a catalog assigned to this L.O.</explanation>
<size>Smallest permitted max: 10 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>

<mandatory>Yes for package and program</mandatory>
</attribute>

<attribute>
<number>7.3</number>
<name>Dependencies</name>
<explanation>Description of dependencies between the study elements.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>-</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>8</number>
<name>Annotation</name>
<explanation>This category provides comments on the educational use of this L.O.</explanation>
<size>Smallest permitted max: 30 items</size>
<order>No</order>
</attribute>

<attribute>
<number>8.1</number>
<name>Person</name>
<explanation>The person who created this annotation.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>V-card</valuespace>
<datatype>Characterstring (smallest permitted max: 1000 char)</datatype>
<example>Information from the V-card will be defined separately.</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>8.2</number>
<name>Date</name>
<explanation>Date this annotation was created. </explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Date</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

<attribute>
<number>8.3</number>
<name>Description</name>
<explanation>The content of this annotation.</explanation>

<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>-</example>
<aggregationlevels>All</aggregationlevels>
<mandatory>No</mandatory>
</attribute>

</category>

<category>

<attribute>
<number>9</number>
<name>Classification</name>
<explanation>This category describes where this L.O. falls within a particular classification system.</explanation>
<size>Smallest permitted max: 40 items</size>
<order>No</order>
</attribute>

<attribute>
<number>9.1</number>
<name>Purpose</name>
<explanation>The purpose of classifying this L.O.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Discipline, Subject, Idea, Prerequisite, Educational objective, Accessibility restrictions, Educational level, Skill level, Security level</valuespace>
<datatype>Vocabulary, see Ontology</datatype>
<example>Discipline and
Subject will be used in CUBER.</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>9.2</number>
<name>TaxonPath</name>
<explanation>This element describes a taxonomic path in a specific classification system.</explanation>
<size>Smallest permitted max: 15 items</size>
<order>No</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>9.2.1</number>
<name>Source</name>
<explanation>The name of the classification system.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>

<datatype>Langstring (smallest permitted max:1000 char)</datatype>
<example>(en, ACM), (en, ARIAD-NE)</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>9.2.2</number>
<name>Taxon</name>
<explanation>This element describes a particular term within taxonomy.</explanation>
<size>Smallest permitted max: 15 items</size>
<order>Yes</order>
<valuespace>-</valuespace>
<datatype>-</datatype>
<example>-</example>
<aggregationlevels>-</aggregationlevels>
<mandatory>-</mandatory>
</attribute>

<attribute>
<number>9.2.2.1</number>
<name>Id</name>
<explanation>The identifier of the Taxon.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>Repertoire of ISO/IEC 10646-1</valuespace>
<datatype>Characterstring (smallest permitted max: 100 char)</datatype>
<example>320, 4.3.2, BF180</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

<attribute>
<number>9.2.2.2</number>
<name>Entry</name>
<explanation>The textual label of the Taxon.</explanation>
<size>1</size>
<order>N/A</order>
<valuespace>-</valuespace>
<datatype>Langstring (smallest permitted max:500 char)</datatype>
<example>(en, Medical sciences)</example>
<aggregationlevels>Course, Package, Program</aggregationlevels>
<mandatory>Yes</mandatory>
</attribute>

</category>

</metadata>

Appendix 6 The Data Exchange

Data exchange between CUBER and other systems using LOM

1. The information contained by the following metadata elements will be used by the receiving system and CUBER as such:

- 1.1 General.Identifier
- 1.2 General.Title
- 1.3 General.CatalogEntry (including sub-elements)
- 1.4 General.Language
- 1.5 General.Description
- 1.6 General.Keywords
- 1.8 General.Structure
- 2.1 LifeCycle.Version
- 2.2 LifeCycle.Status
- 2.3.1 LifeCycle.Contribute.Role
- 2.3.2 LifeCycle.Contribute.Entity
- Category 3 Meta-Metadata entirely as defined in LOM
- 4.1 Technical.Format
- 4.3 Technical.Location
- 4.4.1 Technical.Requirements.Type
- 4.4.2 Technical.Requirements.Name
- 4.4.3 Technical.Requirements.Minimum_Version
- 5.2 Educational.Learning_Resource_Type
- 5.6 Educational.Contexts
- 5.9 Educational.Typical_Learning_Time
- 6.2 Rights.Copyright
- 6.3 Rights.Description
- 7.1 Relation.Kind
- 7.2.1 Relation.Resource.Identifier
- 7.2.3 Relation.Resource.CatalogEntry

- Category 8 Annotation entirely as defined in LOM
- 9.1 Classification.Purpose
- 9.2 Classification.Taxon_Path (including sub-elements)

2. a. The information contained by the following CUBER metadata elements will not be used by the receiving system:

- 1.10 General.CUBER_Identifier
- 2.4 LifeCycle.Recurrence
- 2.5.2 LifeCycle.Date.End
- 2.5.2 LifeCycle.Date.Kind
- 4.8 Technical.Material_Size
- 4.9 Technical.Decription
- 5.12 Educational.Teaching_Activity (including the sub-elements)
- 5.13 Educational.ECTS_Credits
- 5.14 Educational.Dedication
- 5.15 Educational .Evaluation (including the sub-elements)
- 5.16 Educational.Enrolment
- 5.17 Educational.Study_Guidance
- 5.18 Educational.Pre-requisites
- 5.19 Educational.Degree
- 6.4 Rights.Cost_in_EURO
- 6.5 Rights.Financing
- 7.3 Relation.Dependencies

2. b.The information contained by the following LOM elements will not be used by the CUBER system:

- 1.7 General.Coverage
- 4.2 Technical.Size
- 4.4.4 Technical.Maximum_Version
- 4.5 Technical.Installation_Remarks
- 4.6 Technical.Other_Platform_Requirements

- 4.7 Technical.Duration
- 5.1 Educational.Interactivity_Type
- 5.3 Educational.Interactivity_Level
- 5.4 Educational.Semantic_Density
- 5.5 Educational.Intended_End_User_Role
- 5.7 Educational.Typical_Age_Range
- 5.8 Educational.Difficulty
- 5.10 Educational.Description
- 5.11 Educational.Language
- 6.1 Rights.Cost
- 7.2.2 Relation.Resource.Description
- 9.3 Classification.Description
- 9.4 Classification.Keywords

3. The information contained by the following CUBER metadata elements will be mapped to the LOM standard when importing or exporting data:

- 1.11 General.CUBER_Aggregation will be mapped to the 1.9 General.Aggregation_Level.
- 2.5.1 LifeCycle.Date.Begin will be mapped to the 2.3.3 LifeCycle.Contribute.Date

ALTERNATIVE SOLUTIONS FOR MAPPING SPECIFIC METADATA ELEMENTS:

- 4.8 Technical.Material_Size and 4.2 Technical.Size may be partially mapped. The metadata element 4.2 contains the digital size of the Learning Object, and this digital size of the 4.2 may be used in the CUBER system in the extension element 4.8. However, the element 4.8 may contain the size of the material in any form, which makes it difficult or even impossible to use information contained in the 4.8 in the LOM 4.2.

- LOM 6.1 Rights.Cost and CUBER extension 6.4 Rights.Cost_in_EURO may be partially mapped in the following way: if the value of 6.4 is "0" (zero), then the value of 6.1 is "no"; if the value of 6.4 is 1 or more EURO (1<EURO), then the value of 6.1 is "yes".