A Framework and an Architecture for Supporting Interoperability between Digital Libraries and eLearning Applications

Polyxeni Arapi, Nektarios Moumoutzis, Manolis Mylonakis, Stavros Christodoulakis
Laboratory of Distributed Multimedia Information Systems and Applications/
Technical University of Crete (MUSIC/TUC)
{xenia, nektar, manolis, stavros}@ced.tuc.gr

Abstract

One of the most important applications of Digital Libraries (DL) is learning. However, DLs and their standards have been developed independently on eLearning applications and their standards, raising interoperability issues between digital libraries and eLearning applications. In order to enable the development of eLearning applications that easily exploit DL contents it is crucial to bridge the interoperability gap between digital libraries and eLearning applications. For this purpose, a generic interoperability framework has been developed that could also be applied to other types of applications which are built on top of digital libraries, although this paper focuses on the detailed requirements of the eLearning applications. In this context, a framework and an algorithm for supporting personalization in elearning applications has been developed that performs automatic, on-demand, creation of personalized learning experiences using reusable (audiovisual) learning objects, taking into account the learner profiles and a set of abstract training scenarios (pedagogical templates). From a technical point of view, all the framework components have been organized into a service-oriented Architecture that Supports Interoperability between Digital Libraries and ELearning Applications (ASIDE). A prototype of the ASIDE Framework has been implemented.

Categories and Subject Descriptors

H.3 [Information Storage and Retrieval]: H.3.3 Information Search and Retrieval; H.3.7 [Digital Libraries]: Standards; K.3.1 [Computer Uses in Education]: Distance Learning; D.2.11 Software Architectures; D.2.12 Interoperability

General Terms

Interoperability, Personalization

Keywords

Digital libraries, eLearning, interoperability architecture, learning design, ontologies, personalization algorithm, METS, SCORM, LOM, MPEG-7

1 Introduction

Digital Libraries are an important source for the provision of eLearning resources (McLean, 2004). However, digital library metadata standards and eLearning metadata standards have been developing independently, which has as result the existence of interoperability problems between digital libraries and eLearning applications. This is a complex and multi-level problem which is often encountered between digital libraries and several types of applications that run on top of digital libraries. It can be seen as coming from the existence of a stack of conceptual layers where each one is built on top of the previous one (left part of Figure 1): There are different data representations, objects, concepts, domains, contexts and metacontexts in the layer stack that should be efficiently managed in a standardized way. Metadata models are languages that are used to represent the knowledge in a particular application area. Each metadata model is shown as a vertical bar on this stack to cover a specific region that represents the parts that the model tries to capture and describe in a standard way. If we place different metadata models besides this stack, we may identify gaps and intersection regions so that it becomes apparent where the interoperability problems among these models occur. Interoperability problems exist also in the overlapping areas. But in these areas solving the problem of interoperability is easier and can be solved with standard methods (e.g. by means of mappings). The major problems arise in the areas with no overlaps between the two metadata standards. The right part of Figure 1 shows such a picture in the case of MPEG7 (MPEG7 2001) and SCORM (SCORM 2004), the major metadata standards in the audiovisual and eLearning domains respectively. It is apparent from this graphical presentation that MPEG7 and SCORM are not completely overlapping meaning that we need additional models to provide interoperability mechanisms between them.

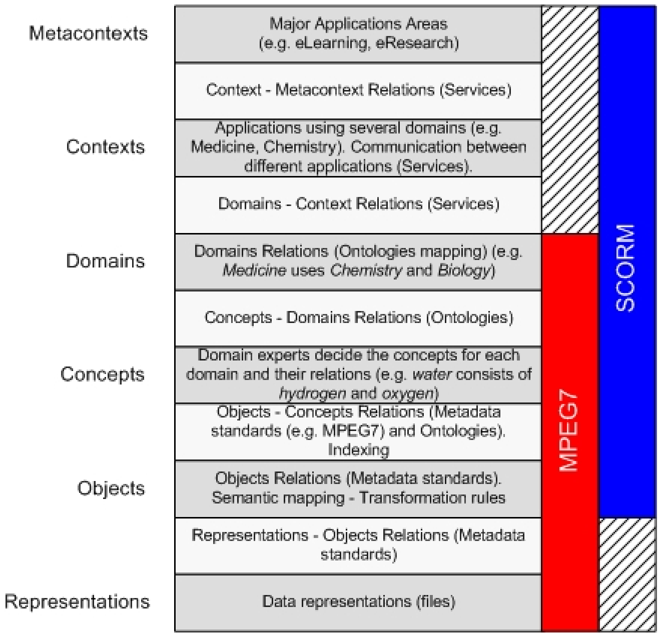

Figure 1 The multilevel problem of interoperability

For example, SCORM contains an educational part that cannot be mapped, directly or indirectly, completely or partially, to MPEG7 elements. That is because MPEG7 does not include information about possible educational use of audiovisual (A/V) objects because it is not an application-specific context metadata standard. However, educational information is very important in the case that MPEG7 (and generally an A/V digital library) is used for educational purposes. On the other hand, MPEG7 offers a comprehensive set of audiovisual Description Tools to guide the creation of audiovisual content descriptions, which will form the basis for applications that provide the needed effective and efficient access to audiovisual content, which can not be represented in SCORM.

Modifying the above standards (e.g. mixing parts of them) is not acceptable, since they have been developed to satisfy the needs of different communities. Overcoming the above shortcomings is crucial in order to develop an integrated model that will allow for the unified description of audiovisual eLearning material i.e. unified metadata descriptions of audiovisual objects and their parts from an educational perspective. Neither MPEG7 nor SCORM could be used as they are to satisfy this critical need because of the shortcomings presented above. To overcome them and fill in the gaps between SCORM and MPEG7 we have to use a higher level metadata model that is able to encapsulate both SCORM and MPEG7 in the context of a digital library. This model should essentially be a wrapper that will allow for the use of MPEG7 metadata of existing audiovisual objects and parts of them together with the necessary LOM metadata (IEEE LOM 2002) that are used in SCORM to specify the educational characteristics of these objects and their parts.

The above considerations lead to a concrete framework and architecture that address the identified interoperability problems and offer a generic framework for the automatic creation of personalized learning experiences using reusable audiovisual learning objects.

In the next sections we will firstly propose a methodology for supporting multiple-contexts views of digital objects and its application in the case of A/V learning objects, without loss of important information (educational or A/V) (Section 2), and thereafter a generic architecture that overcomes the interoperability problem between eLearning applications and digital libraries will be presented (Section 3). The implementation of this architecture offers a generic framework for the automatic creation of personalized learning experiences using reusable A/V learning objects which is also presented. A review of the related literature is presented afterwards (Section 4) and the paper ends with some concluding remarks and future work.

2 Supporting multiple-contexts views of digital objects with METS

As already mentioned the problem of interoperability between digital libraries and applications is a complex and multi-level one. Although in this task we focus on the interoperability problem between digital libraries and eLearning applications, the methodology proposed here for the description of digital objects that reside in a digital library, can be applied to other contexts (e.g. eScience) as well.

A digital object can be described in many ways and delivered to many applications (upper part of Figure 2). Usually, digital objects have a source metadata description that is appropriately transformed to a target metadata description when this object should be delivered to an application. However, performing just a transformation between the source metadata scheme and the target metadata scheme is not always applicable (Arapi, Moumoutzis and Christodoulakis 2006). As shown in Figure 1, standards do not always completely overlap. In the non-overlapping areas the interoperability problem cannot be simply solved using transformations.

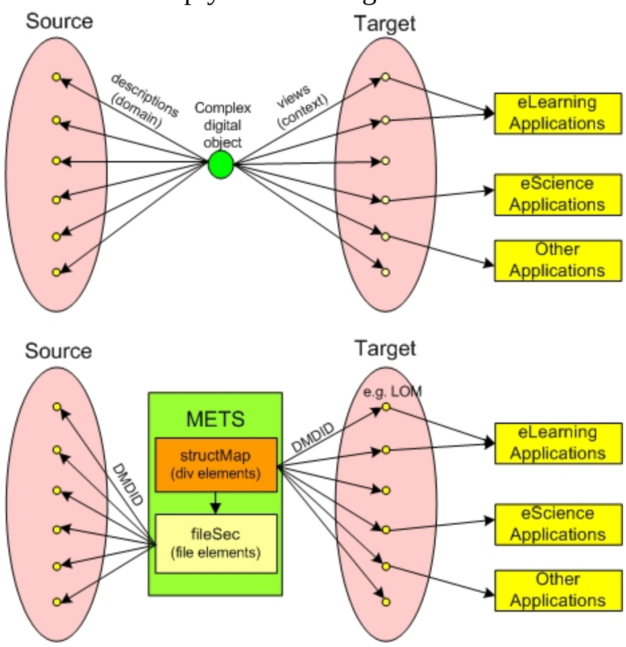

Figure 2 Supporting multiple-contexts views of a digital object using METS

For example, an audiovisual digital object that resides in a digital library and is described with MPEG7 can be used in eLearning or eScience applications. However, the pure MPEG7 description does not say anything about the educational use (e.g. learning objectives) of the digital object nor contains any information useful for eScience applications. Performing just a transformation between the source metadata scheme and the target metadata scheme does not solve the problem. For example, a Learning Management System that searches for appropriate learning objects in a digital library should have access to descriptive metadata regarding the educational use of a digital object to select objects in this context.

So, we need a way to incorporate in a digital object description both source metadata (domain) and target metadata (context). We should have multiple descriptions (source metadata (domain), target metadata (context) - pairs) for a digital object showing possible views of the object. Context and domain information should reside in different levels, where context information is above domain information.

A flexible model that satisfies the above needs is METS (METS 2005). METS is the first widely-accepted standard designed specifically for digital library metadata. METS is intended primarily as a flexible, but tightly structured, container for all metadata necessary to describe, navigate and maintain a digital object, primarily the three types defined by the Digital Library Federation. For each digital object, three types of metadata are possible:

- **Descriptive metadata**: information relating to the intellectual contents of the object, akin to much of the content of a standard catalogue record: this enables the user of a digital library to find the object and assess its relevance.
- **Administrative metadata**: information necessary for the manager of the electronic collection to administer the object, including information on intellectual property rights and technical information on the object and the files that comprise it.
- **Structural metadata**: information on how the individual components that make up the object relate to each other, including the order in which they should be presented to the user: for example, how the still image files that comprise a digitized version of a print volume should be ordered.

Each type of metadata is described in a separate section, which is linked to its counterparts by a comprehensive system of internal identifiers. The metadata itself may be held physically within the METS file, or may be held in external files and referenced from within the METS document: it may follow any preferred scheme, although a number of these are recommended specifically for use within METS.

Using METS we can create different views of a digital object pointing to both source metadata description and target metadata description (context) in different levels (Arapi, Moumoutzis and Christodoulakis 2006). The methodology is illustrated in the lower part of Figure 2. Using the DMDID attribute of the <div> elements of the structMap section where the structure of the digital object is described we can point to an appropriate metadata scheme creating a context (view) of this object and its parts. For example, we can use LOM metadata (IEEE LOM 2002) to describe the educational characteristics of each the object and its parts, so that being able for this object to be searched and retrieved (whole or parts of them) by eLearning applications (educational context) (Figure 3). In parallel, using the DMDID attribute of the <file> elements of the fileSec section, where all files comprising this digital object are listed, we can point to a source metadata scheme that describes the lower level features or the semantics of this object (e.g. using MPEG7). This is useful when applications want to further filter the resulted objects according to its multimedia characteristics. For example, consider an intelligent system (as the Personalized Learning Experience Assembler (PALEA) component of the interoperability architecture described later in this document) that assembles and provides personalized learning experiences to learners using audiovisual content that is stored in a digital library. In the retrieval and selection process, information regarding the semantics or low features of audiovisual content can be taken into account beyond its educational characteristics.

Here, we combine METS, MPEG7 and LOM to give to the audiovisual objects and their parts educational characteristics constructing this way audiovisual learning objects. Moreover, the same MPEG7 document can be referenced by many METS documents that include LOM descriptions, having this way multiple views (contexts) for the same audiovisual objects. This is illustrated in Figure 3.

Domain and Context is separated in two levels. The DMDID attribute of the <file> element is used to reference the MPEG7 metadata (domain metadata) describing the audiovisual object referenced by FLocat element. In an upper level we put the Context Metadata (in our case educational metadata) using the DMDID attribute of the div element to reference LOM metadata. This way the video decomposition to segments is described through the METS document (as a complex object) and there is no need to be described in a MPEG7 document using for example the TemporalDecomposition element. An MPEG7 description is used for each segment to give semantics (Domain information) to this segment.

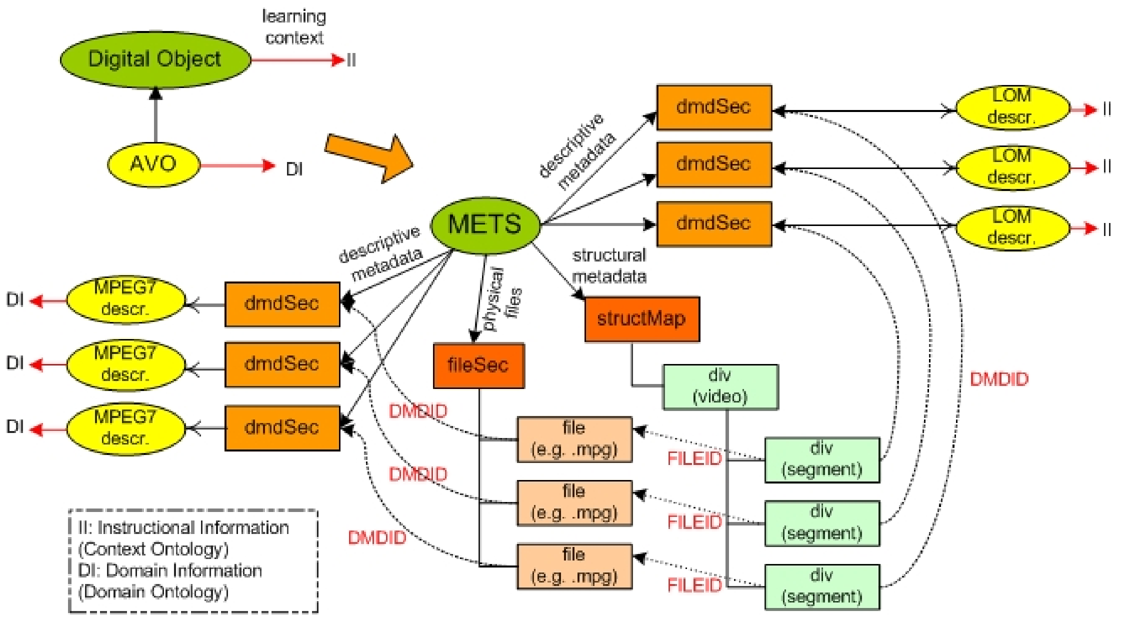

Figure 3 Combining METS, LOM and MPEG7 to build audiovisual learning objects

After presenting a framework for the representation and description of digital objects that reside in a digital library in order to support multiple-context views so that these objects can be retrieved from different applications (in this case eLearning applications), in the next section we will present an Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications (ASIDE).

3 The ASIDE architecture: An Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications

The previous section presented a framework for the representation and description of digital objects that reside in a digital library in order to support multiple-context views so that these objects can be retrieved from different applications (in this case eLearning applications). This section presents an Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications (ASIDE).

The architecture addresses the identified interoperability problems in a layered manner where eLearning (and other) applications are built on top of digital libraries and utilize their content. The ASIDE architecture (Arapi, Moumoutzis and Christodoulakis 2006) offers a generic framework for the automatic creation of personalized learning experiences using reusable A/V learning objects. It is service-oriented and conforms to the IMS Digital Repositories Interoperability (IMS DRI) Specification (IMS DRI 2003). The IMS DRI specification provides recommendations for the interoperation of the most common repository functions enabling diverse components to communicate with one another: search/expose, submit/store, gather/expose and request/deliver. These functions should be implementable across services to enable them to present a common interface. IMS DRI splits services into three categories:

- Access services (resource utilizers): Services with which the end user interacts (e.g. LMS/LCMS, portal)
- Provision services (repositories): Services that make content available, and
- Intermediaries: Services that reside between the above two (e.g. aggregators, brokers)

The DRI specification acknowledges a wide range of content formats and is applicable internationally to both learning object repositories, as well as to other traditional content sources, such as libraries and museum collections.

Figure 4 illustrates the architecture components, which are the following:

- The **Digital Library**, where digital objects are described using METS+LOM (eLearning context), and MPEG7 (A/V descriptions) thus building interoperable A/V learning objects, which can be transformed to SCORM and delivered to eLearning applications (METS/SCORM transformation component). Some important elements used in the LOM descriptions are: educational objectives expressed as verb (Bloom's Taxonomy (Bloom and Krathwohl 1965))+subject (term from a Domain Ontology) using the classification part of LOM, context, typicalAgeRange and difficulty. Regarding the MPEG7 descriptions, the methodology described in (Tsinaraki, Polydoros and Christodoulakis 2004) is used for extending MPEG7 with domain-specific knowledge descriptions expressed in OWL (OWL 2004)(domain ontologies).

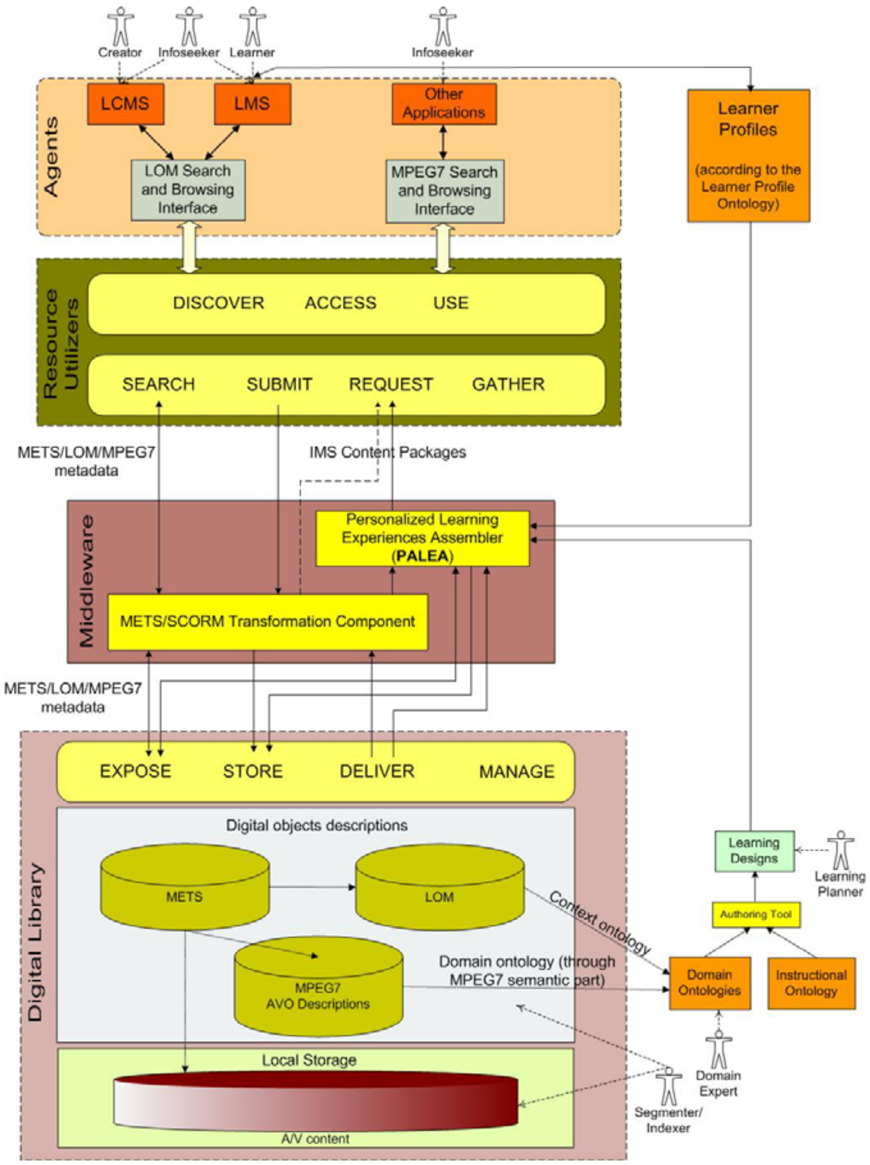

Figure 4 The interoperability architecture

- **Ontologies**
  - **Domain Ontologies** that provide vocabularies about concepts within a domain and their relationships.
  - **Instructional Ontology** that provides a model for the construction of abstract training scenarios. These are pedagogical approaches (instructional strategies/didactical templates), which can be applied to the construction of learning experiences.
- **Learning Designs** are abstract training scenarios in a certain instructional domain built according to the model given in the instructional ontology.
- The **Middleware** consists of the following parts:

- The **METS/SCORM transformation component**, which is responsible for the transformation of the METS descriptions pointing to LOM and MPEG7 descriptions to SCORM Content Packages (SCORM 2004). This includes not only simple transformation from METS XML file to SCORM manifest file, but also the construction of the whole SCORM package (PIF). Moreover, the type of the files is taken into account and, if needed, intermediate html pages are constructed with links to these files (e.g. in case of video files).
- The **Personalized Learning Experiences Assembler (PALEA)**, which, taking into account the knowledge provided by the Learning Designs (abstract training scenarios) and the Learner Profiles described later, constructs the personalized learning experiences and delivers them in the form of IMS Content Packages. Before transforming the resulted learning experience to a SCORM package, it is stored as METS+LOM+MPEG7 description in the digital library according to the interoperability framework, being ready and available in an interoperable way for later requests. The dashed arrow in the left side of PALEA indicates that using this component is optional, and that digital library services can be directly accessed (e.g. a teacher wants to find appropriate learning objects to construct manually a learning experience).

- **Applications** (Software Agents in terms of IMS DRI, like Learning Content Management Systems, Learning Management Systems etc.) that discover, access and use the content of the A/V content of the digital library through appropriate services (resource utilizers). The generated personalized A/V learning experiences are delivered to the applications in the form of SCORM packages. Any SCORM-compliant system can recognize and "play" those packages.
- The **Learner Profiles** constructed using the vocabulary given in the Learner Profile Ontology, which represents a Learner Information Model (LIM) for the creation of learner profiles. Elements from IEEE PAPI (IEEE PAPI 2002) and IMS LIP (IMS LIP 2005) specifications have been also used in this model. Some important elements of this model are: learner goals, previous knowledge, educational level and learning style.

3.1 The instructional ontology

We present here a model for the construction of abstract training scenarios which is represented in an ontology coded in OWL (Figure 5) that has the important characteristic that learning objects are not bound in the training scenarios on design time, as in current eLearning standards and specifications (e.g. IMS Learning Design (IMS LD 2003) and SCORM). Whereas, pedagogy is separated and independent from content achieving this way reusability of learning designs or parts of them that can be used from the systems for the construction of "real" personalized learning experiences, where appropriate learning objects according to the learner profile are bound to the learning experience at run-time. This is possible, since the model gives the opportunity to specify in each Activity the learning objects' requirements, instead of binding the learning objects themselves, as IMS LD and SCORM impose. This ontology borrows some elements and ideas from IMS LD and LOM.

A *Training* is a collection of abstract training scenarios regarding one domain. The same subject can be taught in several ways (*TrainingMethods*) depending on the *LearningStyle* and the *EducationalLevel* of the Learner. There are several categorizations of Learning Styles (Konsolaki, Kapidakis and Arapi 2006) and Educational Levels, thus these elements are flexible so that being able to point to values of different taxonomies. A *TrainingMethod* consists of a hierarchy of *ActivityStructures* built from *Activities* (elements taken from IMS LD). Existing *ActivitiesStructures* or paths of *ActivitiesStructures* can be reused in many Learning Designs. Each *Training*, *ActivityStructure* and *Activity* has a *LearningObjective*. Learning Objectives are treated here in a more formal way (as in SeLeNe project (Keenoy, Levene and Peterson 2004)), than pure text descriptions. Thus, each *LearningObjective* has: (a) a *learning_objective_verb*, taken from a subset of the outcome-illustrating verbs which characterise each type of learning objectives specified by a committee of college and university examiners in 1956 (known as "Bloom's

Taxonomy (Bloom and Krathwohl 1965)). This subset has been selected for the description of Learning Objectives by the SeLeNe project. (b) a *learning_objective_topic* that indicates the topic that the learning objective is about, referenced as an entry in the RDF binding of a subject taxonomy or ontology (context ontology e.g. ACM Computing Taxonomy (ACM 1998)), and (c) *learning_objective_annotation* that indicates additional textual description of the learning objective; for example, to specify areas within the topic at a greater level of detail than is catered for by the subject taxonomy (or ontology). The *LearningObjectType* class is used to describe the desired Learning Object characteristics (requirements) without binding specific objects with *Activities* on design time. Via the *related_with* property we can further restrict the preferred learning objects connecting them with *DomainConcepts* or individuals from a domain ontology.

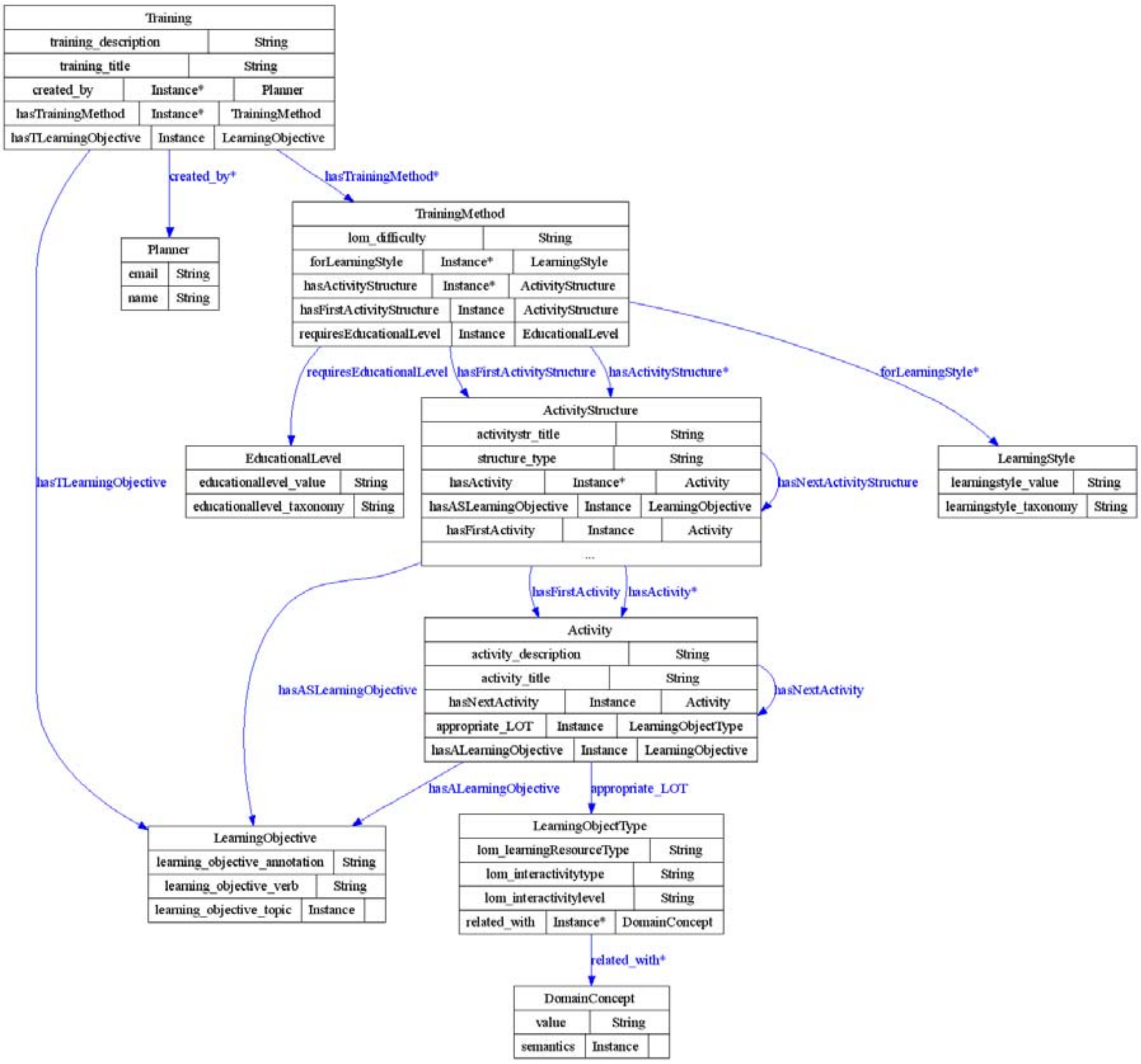

Figure 5 The instructional ontology

3.1 The METS/SCORM transformation component

The METS/SCORM transformation component is responsible for the transformation of the METS descriptions pointing to LOM and MPEG7 descriptions to SCORM Content Packages (SCORM 2004). This includes not only simple transformation from the METS XML file to the SCORM manifest file, but also the construction of the whole SCORM package (PIF). Moreover, the type of the files is taken into account and, if needed, intermediate html pages are constructed with links to these files (e.g. in case of video files). The mapping between METS and SCORM is presented in the following table:

Table 1 Mapping between METS and SCORM

METS	SCORM IMS Manifest
structMap	organizations/organization
structMap/@ID	organizations/@default
structMap/@ID	organizations/organization/@identifier
structMap/div/@LABEL	organization/title
structMap/div/@ID	organization/item/@identifier
div/@LABEL	organization/item/title
div/fptr/@FILEID	item/@identifierref
fileSec	Resources
fileGrp	resources/resource
fileGrp /@ID	resources/resource/@identifier
file/FLocat/@xlink:href	resources/resource/@href
fileGrp/file	resources/resource/dependency
fileGrp/file/@ID	resources/resource/dependency/@identifierref
fileGrp/file/@ID	resources/resource/@identifier
fileGrp/file/FLocat/href	resources/resource/file/@href
If dmdSec/mdWrap/[@MDTYPE=LOM] dmdSec/mdWrap/xmlData	<adlcp:location>lomfiles/FG1.xml</adlcp:locat-ion> Creates an xml document with the LOM metadata for each resource.

3.1 The Learner Profiles

There are two major attempts to standardize a learner profile, the IEEE Personal and Private Information (PAPI) (IEEE PAPI, 2002) and the IMS Learner Information Package (LIP) (IMS LIP 2005). According to Dolog and Nejdl (Dolog and Nejdl 2003) these standards have been developed from different points of view. The PAPI standard reflects ideas from intelligent tutoring systems where the performance information is considered as the most important information about a learner. PAPI also stresses the importance of inter-personal relationships. On the other hand the LIP standard is based on the classical notion of a CV and inter-personal relationships are not considered. One way forward is therefore to draw on both standards as a means of developing an adequate learner model. Moreover, other elements should be added in order to cover important aspects of a Learner, as for example the learning style.

Since our intention here is to support personalization there is a need to focus on the learner's goals and preferences. Of course, other elements could be also included in a learner profile but in this case they are not important. We organized the most important elements of a Learner Profile in a Learner Profile Ontology which is presented in (Christodoulakis, Arapi, Mylonakis, Moumoutzis, Patel, Kapidakis, Vagiati, Konsolaki 2006). These are the LearnerGoals and Preferences. A LearnerGoal is expressed in terms of LearningObjectives according to the format that is followed in the instructional ontology. A LearnerGoal has a status property which is a float number from 0.0 to 1.0 indicating the satisfaction level of the goal (e.g. a value of 0.0 shows that this goal has been not at all satisfied, while a value of 1.0 shows that this goal has been fully satisfied). From this information we can extract the previous knowledge of the Learner. The Learner can also define a priority for each LearnerGoal. The Learner can have several types of Preferences: EducationalLevel and LearningStyle matching with the corresponding elements of the instructional ontology, Language, LearningProvider (the author or organization making available the learning objects), LearningPlanner (the person that develops learning designs) and Technical preferences.

3.2 The Personalized Learning Experiences Assembler (PALEA)

The Personalized Learning Experiences Assembler (PALEA) algorithm takes into account the knowledge provided by the Learning Designs (abstract training scenarios) and the Learner Profiles and constructs personalized learning experiences that are delivered next to eLearning applications in the form of SCORM Content Packages. Specifically, the goal of the algorithm is to find an appropriate learning design (built according to the instructional ontology) that will be used thereafter to construct a learning experience adapted to the Learner's needs. At the beginning, the algorithm tries to find an appropriate Training and an appropriate Training Method for this Training, according to the Learner's learning objectives, educational level, preferred difficulty, learning style and preferred planner (optional). The structure of this Training Method is further refined, by removing from it Activity Structures and Activities with learning objectives that have been satisfied by the Learner (the Learner can define a threshold value TH, so that Learning Objectives with satisfaction value>TH are considered as satisfied). Finally, appropriate learning objects are retrieved from the digital library and bound to each node (Activity) of this structure constructing the learning experience. Here, the Learning Object Type describing the characteristics of appropriate learning objects for each Activity are taken into account along with other learner's preferences (e.g. content provider, technical preferences). The resulted learning experience is transformed to SCORM and delivered to the Learner.

Due to its size, the complete algorithm is presented in detail in the Annex of this paper.

4 Related Work

Efforts trying to integrate or use in cooperation eLearning standards and A/V standards include the Video Asset Description (VAD) Project (Bush et al. 2004), the MultImedia Learning Object Server (Amato et al. 2004) and the Virtual Entrepreneurship Lab (VEL) (Klamma, Jarke and Wulf 2002). Most of these approaches (Amato et al 2004; Klamma, Jarke, and Wulf 2002) use mappings between standards (e.g. MPEG7 and LOM) or propose adding MPEG7 elements to SCORM elements (Bush et al. 2004). As already discussed, using mappings is not enough to solve the interoperability problem between digital libraries and eLearning applications. Extending SCORM is again not acceptable as a general interoperability solution. It results in a model that is not standard and cannot be interoperable with standard SCORM compliant applications and systems. The framework proposed here is more general and does not depend on the strict use of MPEG7 and LOM.

The interoperability architecture proposed here conforms to the IMS Digital Repositories Interoperability (IMS DRI) Specification. Web service implementations based on IMS DRI include the EduSource Canadian Network of Learning Objects Repositories (EduSource Canada), the Learning Objects Network (LON) and the Campus Alberta Repository of Educational Materials (CAREO). Our approach differs in that it provides an interoperable framework of educational and application specific metadata so that eLearning applications can easily use and reuse digital library objects in multiple contexts. Moreover, intelligent construction of personalized learning experiences is supported, so that courseware creation and reuse of educational resources (including learning designs) may be automated.

As presented in (Christodoulakis, Arapi, Mylonakis, Moumoutzis, Patel, Kapidakis, Vagiati, Konsolaki 2006) there is currently much work in progress in the area of personalisation and adaptation in eLearning systems, a considerable amount of this relates to the use of ontologies and semantic web technologies (Razmerita, Angehrn and Maedche 2003; Dolog, Hemze, Nejdl and Sintek 2004; Denaux, Dimitrova and Aroyo 2004; Baldoni et al. 2004). Several important projects are also related with this area: The ELENA Project (Dolog, Hemze, Nejdl and Sintek 2004) has built a dynamic learner profile, which includes a learning history, learner specific information and learning goals and is based on both the LIP (IMS LIP 2005) and PAPI (IEEE PAPI 2002). The SeLeNe Project (Keenoy, Levene and Peterson 2004) has developed a User Profile for SeLeNe users, which combines elements from existing learner profile schemes and adds extra elements where these schemes are insufficiently expressive to adequately support SeLeNe's personalisation requirements (Keenoy, Levene and Peterson 2004). We adopted in this paper the SeLeNe model for the representation of Learning Objectives. The Learning Systems Architecture Lab (LSAL) at Carnegie Mellon (http://www.lsal.cmu.edu/) is also investigating customised learning using a web services approach. Their customised learning prototype is used to present the right learning materials to the learner, on demand, using a model of role and competency-based content customisation. Our model for the representation of Learner Profiles is not far from the ideas presented in the above models. It uses some elements from IMS LIP and PAPI, but it focuses on learner's goals and preferences and it is harmonized with the personalization framework presented in this paper. However, our personalization framework, in contrast with the research presented above, takes into account another important parameter, the pedagogy, which is separate and independent from content. This achieves reusability of learning designs, or parts of them, that can be exploited in systems for the construction of "real" personalized learning experiences, where appropriate learning objects according to the learner profile are bound to the learning experience at run-time. We have developed a personalization algorithm that implements this functionality.

5 Conclusions and Future Work

We have presented a framework for the representation and description of digital objects that reside in a digital library in order to support multiple-context views so that these objects can be retrieved from different applications (in this case eLearning applications). We have also presented ASIDE, an architecture that supports the integration of eLearning applications on top of digital libraries. The architecture supports interoperability between digital libraries and eLearning applications so that eLearning applications can easily use and reuse digital library objects in multiple contexts. The architecture is generic and it provides a framework for integrating in eLearning applications material selection from digital libraries, and personalization of the material selected. We have outlined the various aspects of the implementation of the ASIDE architecture.

Special emphasis has been placed in the definition and implementation of a software engineering framework along with an involved algorithm for supporting personalization in the ASIDE architecture. The algorithm performs automatic creation of personalized learning experiences using reusable (audiovisual) learning objects, taking into account the learner profiles and a set of abstract training scenarios (pedagogical templates). This work provides the basis of a generic architectural framework for integrating diverse application classes (like eLearning and eScience applications) on top of digital libraries so that digital library objects are also reused across application classes that have been built on top of digital libraries.

We intend to parameterize the personalization algorithm in order to support alternative ways for the construction of learning experiences and to evaluate the algorithm performance in terms of its execution time and the quality of the generated learning experiences. The quality of the learning experiences can be evaluated either empirically (from learners/educationalists), or in comparison with manually constructed learning experiences by experts that correspond to particular training methods which are considered as optimal.

6 Acknowledgements

The work presented in this paper is partially funded in the scope of the DELOS II Network of Excellence in Digital Libraries (IST – Project Record Number 507618) JPA2 subproject, named "Interoperability of eLearning applications with digital libraries".

7 References

ACM. 1998. ACM Computing Classification System. Available at: http://www.acm.org/class/1998/

Arapi P., Moumoutzis N. and Christodoulakis S. 2006. ASIDE: An Architecture for Supporting Interoperability between Digital Libraries and ELearning Applications. *Proceedings of the 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006), Kerkrade (The Netherlands), 5-7 July 2006.* Washington: IEEE Computer Society. 257-261

Baldoni M., Baroglio C., Patti V. and Torasso L. 2004. Reasoning about learning object metadata for adapting SCORM courseware. *Proceedings of International Workshop on Engineering the Adaptive Web (EAW'04), Eindhoven, The Netherlands, August 2004.* Technische Universiteit Eindhoven. 4-13

Bloom B.S. and Krathwohl D.R. 1965. Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain. Longman, New York

Bush M., Melby A., Anderson T., Browne J. and Hansen M. 2004. Customized Video Playback: Standards for Content Modeling and Personalization. Educational Technology, 44, 2004. 5-13.

CAREO, http://careo.netera.ca/

Christodoulakis S., Arapi P., Mylonakis M., Moumoutzis N., Patel M., Kapidakis S., Vagiati V., Konsolaki H. 2006. Demonstrator of mapping between the eLearning and AV content description standards, DELOS Deliverable 5.4.2, 25th July 2006.

Denaux R., Dimitrova D. and Aroyo L. 2004. Interactive Ontology-Based User Modeling for Personalized Learning Content Management. *Proceedings of Workshop on Applications of Semantic Web technologies for E-Learning (SW-EL) held in conjunction with Adaptive Hypermedia 2004, Eindhoven (The Netherlands), 23-26 August 2004*.

Dolog P., Hemze N., Nejdl W. and Sintek M. 2004. Personalization in Distributed e-Learning Environments. *Proceedings of the 13th international World Wide Web conference (WWW 2004), 17-22 May 2004, New York (USA).* New York: ACM Press. 170-179

Dolog P. and Nejdl W. 2003. Challenges and Benefits of the Semantic Web for User Modelling. *Proceedings of AH2003: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, 12th International WWW Conference, Budapest (Hungary), 20-24 May 2003*.

Edusource Canada, http://www.edusource.ca

IEEE LOM 2002. IEEE 1484.12.1-2002 Learning Object Metadata Standard. Available at: http://ltsc.ieee.org/wg12/

IEEE PAPI 2002. Personal and Private Information (PAPI), Draft 8 Specification. Available at: http://edutool.com/papi

IMS DRI 2003. IMS Digital Repositories specification V1.0. Available at: http://www.imsglobal.org/digitalrepositories/

IMS LD 2003. IMS Learning Design specification V1.0. Available at: http://www.imsglobal.org/learningdesign/

IMS LIP 2005. IMS Learner Information Package Specification V1.0.1 Available at: http://www.imsglobal.org/profiles/

Keenoy K., Levene M. and Peterson D. 2004. SeLeNe –Self E-Learning Networks (IST-2001-39045); WP4 Deliverable 4.2, Personalisation and Trails in Self e-Learning Networks, Jan 2004

Klamma R., Jarke M. and Wulf V. 2002. Das Virtual Entrepreneurship Lab (VEL): Eine MPEG7 basierte E-Learning Plattform für potentielle Gründer. Informatik 2002, Dortmund (Germany). 359-363.

Konsolaki H., Kapidakis S. and Arapi P. 2006. DELOS WP5 -Task 5.4, Report on Learning Styles. Internal Deliverable. v0.2, 8th January 2006

Learning Objects Network (LON), http://www.learningobjectsnetwork.com/

McLean N. 2004. The Ecology of Repository Services: A Cosmic View. Keynote Address. 8th European Digital Library Conference ECDL 2004. UK

METS 2005. Metadata Encoding and Transmission Standard (METS) Official Website. Available at: http://www.loc.gov/standards/mets/

MPEG7 2001. ISO/IEC JTC 1/SC 29/WG 11/N3966, Text of 15938-5 FCD Information Technology – Multimedia Content Description Interface – Part 5 Multimedia Description Schemes. MPEG Group, Singapore 2001

OWL 2004. Web Ontology Language (OWL). Available at: http://www.w3.org/TR/owl-ref/

SCORM 2004. Sharable Content Object Reference Model (SCORM) 2004 2nd edition. Available at: http://www.adlnet.org

Razmerita L., Angehrn A. and Maedche A. 2003. Ontology-based User Modeling for Knowledge Management *Systems. Proceedings of the 9th International Conference on User Modeling (UM 2003), Johnstown, PA (USA), 22-26 June 2003*. Lecture Notes in Computer Science 2702. Springer. 213-217.

Tsinaraki C., Polydoros P. and Christodoulakis S. 2004. Integration of OWL ontologies in MPEG7 and TVAnytime compliant Semantic Indexing. Proceedings of the 16th International Conference on Advanced Information Systems Engineering (CAiSE 2004), Riga (Latvia), 7-11 June 2004. Lecture Notes in Computer Science 3084. Springer. 398-413.

ANNEX - The Personalized Learning Experiences Assembler (PALEA) algorithm

createLearningExperiencefromTrainings(L, EL, D, LS, PL, OLR, TH)
Input
Let *L* be the set of Learner's Learning Objectives.
Let *EL* be the Learner's Educational Level.
Let *D* be the Learner's preferred Difficulty.
Let *LS* be the Learner's Learning Style.
Let *PL* be the Learner's preferred Planner (optional).
Let *OLR* be some other Learner preferences (content provider, technical etc.).
Let *TH* be a threshold value. Learning Objectives with satisfaction value>*TH* are considered as satisfied.
Output
A Learning Experience *le* or *null*.

```
{
    tr = findTraining(L, EL, D, LS, PL)
    if tr is null return null
    tm = findTrainingMethod(tr)
    if tm is null return null
    le = createLearningExperince(tm, TH)
    return le
}
```

The algorithm in detail:

findTraining(L, EL, D, LS, PL)
Input
Let *L* be the set of Learner's Learning Objectives.
Let *EL* be the Learner's Educational Level.
Let *D* be the Learner's preferred Difficulty.
Let *LS* be the Learner's Learning Style.
Let *PL* be the Learner's preferred Planner (optional)
Output
A training *t* or *null*

```
{
    L_init = L
    do {
        find the learning objective l in L with minimum satisfaction value and remove it from L
        let T be the set of trainings having l as their learning objective.
        if |T|=1 return the unique training t in T
        if |T|>1
            if |L|=1 return findTrainingUsingSingleLearningObjective(T, EL, D, LS)
            else return findTrainingUsingMultipleLearningObjectives(T, L, EL, D, LS)
    } while |L|>0 AND |T|=0
    /* If no training has been found in the previous steps... */
    L = L_init
    do {
        find the learning objective l in L with minimum satisfaction value and remove it from L
        let T be the set of trainings having learning objective topic equal to l topic.
        if |T|=1 return the unique training t in T
        if |T|>1
            if |L|=1 return findTrainingUsingSingleLearningObjective(T, EL, D, LS)
            else return findTrainingUsingMultipleLearningObjectives(T, L, EL, D, LS)
    } while |L|>0 AND |T|=0
```

/* If no training has been found in the previous steps... */
L = L_{init}
do {
find the learning objective *l* in *L* with minimum satisfaction value and remove it from *L*
let *T* be the set of trainings containing activity structures or activities having *l* as their learning objective.
if |*T*|=1 return the unique training *t* in *T*
if |*T*|>1
if |*L*|=1 return *findTrainingUsingSingleLearningObjective(T, EL, D, LS)*
else return *findTrainingUsingMultipleLearningObjectives(T, L, EL, D, LS)*
} **while** |*L*|>0 AND |*T*|=0
/* If no training has been found in the previous steps… */
L = L_{init}
do {
find the learning objective *l* in *L* with minimum satisfaction value and remove it from *L*
let *T* be the set of trainings containing activity structures or activities having learning objective topic equal to *l* topic.
if |*T*|=1 return the unique training *t* in *T*
if |*T*|>1 return *findTrainingUsingSingleLearningObjective(T, EL, D, LS)*
} **while** |*L*|>0 AND |*T*|=0
/* If no training has been found, return null */
return *null*
}

findTrainingUsingMultipleLearningObjectives(T, L, EL, D, LS, PL)
Input
Let *T* be a set of trainings.
Let *L* be the set of Learner's Learning Objectives.
Let *EL* be the Learner's Educational Level.
Let *D* be the Learner's preferred Difficulty.
Let *LS* be the Learner's Learning Style.
Let *PL* be the Learner's preferred Planner (optional)
Output
A training *t* or null
{
Keep in *T* only those trainings that have the most activity structures and activities satisfying the learning objectives in *L*.
if |*T*|=1 return the unique Training *t* in *T*
if |*T*|>1 /*draw*/
return *findTrainingUsingSingleLearningObjective(T, EL, D, LS, PL)*
/* |*T*|=0 is impossible. In the worst case when no AS or A exist that satisfy some learning objectives in L, all trainings in T will be returned (draw). */
}

findTrainingUsingSingleLearningObjective(T, EL, D, LS, PL)
Input
Let *T* be a set of trainings.
Let *EL* be the Learner's Educational Level.
Let *D* be the Learner's preferred Difficulty.
Let *LS* be the Learner's Learning Style.
Let *PL* be the Learner's preferred Planner (optional)
Output
A training *t* or null
{
T_{init}= *T*

```
    Keep in T only trainings with training methods having educational level el=EL and learning
    style ls=LS and difficulty d=D.
    if |T|=1 return the unique training t in T
    else if  |T|>1{
        if (PL<>" ")
            return findTraining_with_pref_Planner(T, PL)
        else
            return the training t from T that the learner selects
    else {  /* |T|=0 */
        T = T_init
        Keep in T only trainings with training methods having educational level el=EL and
        difficulty d=D.
        if |T|=1 return the unique training t in T
        else if  |T|>1
            if (PL<>" ")
                return findTraining_with_pref_Planner(T, PL)
            else
                return the training t from T that the learner selects
        else {  /* |T|=0 */
            T = T_init
            Keep in T only trainings with training methods having educational level el=EL and
            difficulty d<D and closer to D.
            if |T|=1 return the unique training t in T
            else if |T|>1
                return findTraining_with_TrainingMethods_with_LS_equal_to(T, LS)
            else{   /* |T|=0 */
                T = T_init
                Keep in T only trainings with training methods having educational level el=EL and
                difficulty d>D and closer to D.
                if |T|=1 return the unique training t in T
                else if |T|>1
                    return findTraining_with_TrainingMethods_with_LS_equal_to(T, LS)
                else{  /* |T|=0 */
                    Keep in T only trainings with training methods having educational level el<EL
                    and closer to EL and difficulty d the greatest.
                    if |T|=1 return the unique training t in T
                    else if |T|>1
                        return findTraining_with_TrainingMethods_with_LS_equal_to(T, LS)
                    else{  /* |T|=0 */
                        Keep in T only trainings with training methods having educational level
                        el>EL and closer to EL and difficulty d the lowest.
                        if |T|=1 return the unique training t in T
                        else if |T|>1
                            return findTraining_with_TrainingMethods_with_LS_equal_to(T,  LS)
                        else  /* |T|=0, no training could be found because there is no adequate training method in the candidate trainings */
                            return null
                    }
                }
            }
        }
    }
}
```

findTraining_with_TrainingMethods_with_LS_equal_to(T, LS, PL)
Input
Let *T* be a set of trainings.
Let *LS* be the Learner’s Learning Style.
Let *PL* the Learner’s preferred Planner (optional)
Output
A set of trainings *T’*
{
$T_{old} = T$
Keep in *T* only trainings with training methods having learning style *ls=LS*.
if |*T*|=1 return *T*
else if |*T*|>1{
if (PL<>” ”)
return *findTraining_with_pref_Planner(T, PL)*
else
return the training *t* from *T* that the learner selects
else /* |*T*|=0 */
if (PL<>” ”)
return *findTraining_with_pref_Planner*(T_{old}, *PL)*
else
return the training *t* from T_{old} that the learner selects
}

findTraining_with_pref_Planner(T, PL)
Input
Let *T* be a set of trainings.
Let *PL* the Learner’s preferred Planner (optional)
Output
A training *t*
{
$T_{old} = T$
Keep in *T* only trainings having planner=*PL*
if |*T*|=1 return the unique training *t* in *T*
else if |*T*|>1{
return the training *t* from *T* that the learner selects
}
else { /* |*T*|=0 */
return the training *t* from T_{old} that the learner selects
}
}

findTrainingMethod(T)
Input
Let *T* be a training
Output
A training method *tm* from the training *T* or *null*
{
Let *TM* be the set of training methods of training *T*
if |*TM*|=1 return the unique training method *tm* in *TM*.
else if |*TM*|>1{
$TM_{old} = TM$
Keep in *TM* only training methods having educational level *el=EL* and learning style *ls=LS* and difficulty *d=D*.
if |*TM*|=1 return the unique training method *tm* in *TM*.
else if |*TM*|=0{

```
TM = TM_old
Keep in TM only training methods having educational level el=EL and difficulty d=D.
if |TM|=1 return the unique training method tm in TM.
else if  |TM|>1 return the unique training method tm from TM that Learner selects.
else {  /* |TM|=0 */
       TM = TM_old
       Keep in TM only training methods having educational level el=EL and difficulty
       d<D and closer to D.
       if |TM|=1 return the unique training method tm in TM.
       else if |TM|>1{
           TM' = TM
           Keep in TM only training methods having learning style ls=LS.
           if |TM|=1 return the unique training method in TM.
           else if |TM|>1 return the unique training method tm from TM that Learner
           selects.
           else   /* |TM|=0 */
                 return the unique training method tm from TM' that Learner  selects.
      }
      else {  /* |TM|=0 */
           TM = TM_old
           Keep in TM only training methods having educational level el=EL and
           difficulty d>D and closer to D.
           if |TM|=1 return the unique training method tm in TM.
           else if |TM|>1{
                 TM' = TM
                 Keep in TM only training methods having learning style ls=LS.
                 if |TM|=1 return the unique training method tm in TM.
                 else if |TM|>1 return the unique training method tm from TM that Learner
                 selects.
                 else   /* |TM|=0 */
                       return the unique training method tm from TM' that Learner  selects.
                 }
           else {  /* |TM|=0 */
                 TM = TM_old
                 Keep in TM only training methods having educational level el<EL and
                 closer to EL and difficulty d the greatest.
                 if  |TM|=1return the unique training method tm in TM.
                 else if |TM|>1{
                       TM' = TM
                       Keep in TM only training methods having learning style ls=LS.
                       if |TM|=1 return the unique training method tm in TM.
                       else if |TM|>1 return the unique training method tm from TM that
                       Learner  selects.
                       else   /* |TM|=0 */
                             return the unique training method tm from TM' that Learner selects.

                 }
                 else {  /* |TM|=0 */
                       TM = TM_old
                       Keep in TM only training methods having educational level el<EL and
                       closer to EL and difficulty d the lowest.
                       if |TM|=1 return the unique training method tm in TM.
                       else if |TM|>1{
                             TM' = TM
```

Keep in *TM* only training methods having learning style *ls=LS*.
if |*TM*|=1 return the unique training method *tm* in *TM*.
else if |*TM*|>1 return the unique training method *tm* from *TM* that Learner selects.
else /* |*TM*|=0 */
return the unique training method *tm* from *TM'* that Learner selects.
}
/* |*TM*|=0 impossible */
}
}
}
}
}
}
else /* |*TM*|=0 */
return *null*
}

createLearningExperience(TM,TH)
Input
Let *TM* be a Training Method
Let TH be a threshold value. Learning Objectives with satisfaction value>TH are considered as satisfied.
Output
A learning experience *le*
{
Remove from the tree of activity structures and corresponding activities rooted at the training method *TM* those activity structures and activities having learning objectives with satisfaction value greater than threshold *TH*. If an activity structure is removed, then all its children activities are also removed. Let *T* be the resulting tree.
Let *AS* and *A* be the sets of activity structures and activities in the tree *T* respectively.
/* Create the learning experience by finding appropriate learning objects from the digital library that will be thereafter bound to each node of the learning experience structure. */
create a root courseware node *c*
for each activity structure *as* in *AS* {
create a courseware node *cas* corresponding to the activity structure *as*
for each activity *a* in *as* {
let *l* be the learning objective of activity *a*
let *LO* be the set of learning objects having *l* as their learning objective
if |*LO*|>1 LO_{final} = *filterLOs(LO, A)*
else if |*LO*|=1 LO_{final}=*LO*
else { /* |*LO*|=0 */
let *lt* be the learning objective topic of *l*
let *LO* be the set of learning objects having *lt* as the learning objective topic of their learning objective
if |*LO*|>1 LO_{final} = *filterLOs(LO, A)*
else if |*LO*|=1 LO_{final}=*LO*
else /* |*LO*|=0 */
LO_{final} = {blank page}
}
create a courseware node *ca* corresponding to activity *a* consisting of the learning objects in LO_{final}
append *ca* as a new child of *cas*

```
        }
        append cas as a new child of c
    }
    return the courseware structure rooted at c
}
```

filterLOs(LO, A, OLR)

Input

```
    Let LO be a set of candidate learning objects for an activity A
    Let OLR be some other Learner preferences (content provider, technical etc.)
```

Output

```
    A set of learning objects
{
    LO_old=LO
    Let LOT be the learning objects type (learning objects requirements) associated with activity A.
    if LOT exists {
        Keep in LO only those learning objects satisfying LOT.
        if |LO|=1 return LO
        else if |LO|>1{
            LO_old=LO
            Keep in LO only those learning objects satisfying some requirements of the Learner
            Profile (content provider, technical etc.).
            if |LO|=1return LO
            else if |LO|>1 return LO
            else /* |LO|=0 */
                return LO_old
        }
        else { /* |LO|=0 */
            LO =LO_old
            Keep in LO only those learning objects satisfying some other Learner requirements
            OLR expressed in the Learner Profile (content provider, technical etc.).
            if |LO|=1return LO
            else if |LO|>1{
                LO_old=LO
                Keep in LO only those learning objects satisfying some requirements of the Learner
                Profile (content provider, technical etc.).
                if |LO|=1 return LO
                else if |LO|>1 return LO
                else /* |LO|=0 */
                    return LO_old
            }
            else  /* |LO|=0 */
                return LO_old
    }
    else {
        Keep in LO only those learning objects satisfying some requirements of the Learner Profile
        (content provider, technical etc.).
        if |LO|=1return LO
        else if |LO|>1 return LO
        else /* |LO|=0 */
            return LO_old
    }
}
```