Annales UMCS Informatica AI XIII, 2 (2013) 17–27
DOI: 10.2478/v10065-012-0037-8

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Content Repository in Object Oriented data model

Dariusz Dobrowolski[1*], Michał Chromiak[1†]

[1] *Institute of Computer Science, Maria Curie-Skłodowska University, Akademicka 9, 20-033 Lublin, Poland*

Abstract – The need for creating content repository stores for e-learning systems grows as the number of available materials increases. Moreover, along with the number of courses, the problem of describing them in a unified form appears. While there are standards used for strict classification of e-learning content, the store model still seems to be based on preservative relational databases approach. In this paper we introduce an idea to represent the e-learning content management information in the well organized object-oriented form based on a prospective object-oriented database.

1 Introduction. E-learning principles and infrastructure

The problem of creating a well designed e-learning course should meet all the requirements These requirements must be considered in a wide spectrum from the ones defined by the standards and educational platforms to those that have been defined by the end user expectations. This leads to numerous challenges that have to be taken along the way to a well created e-learning course.
The creation process is very time consuming and expensive. It includes numerous multimedia and graphic contents but sometimes its content is mostly insufficient and incomplete. On the other hand, there are courses including very well designed and interesting content but not provided with the multimedia part that makes it less attractive for the end user. The idea of facilitating ready to use courses and changing the existing ones came up when the multiple server archives with the courses became more extensive.

At present most of the courses and files enclosed within are described with metadata i.e. the short information containing data about e.g. name, language, version etc.

*dariusz.dobrowolski@umcs.lublin.pl
†mchromiak@umcs.pl

This scheme is gathered as the LOM (Learning Object Metadata) specification. This enables building uniform interface for accessing the data about e-learning courses and all of the resources gathered in libraries. The information about available courses could then be found in the same way as the books or magazines in the library are.

1.1 LOM

The IEEE LOM standard conceptual data schema lists all the meta-data elements in the tabular format. Fig. 1 presents a graphical illustration of the elements in the data schema, which shows how the elements are divided into nine top level categories: General, Life Cycle, Meta-Metadata, Technical, Educational, Rights, Relation, Annotation, and Classification. Each of these branches comprises several elements, some of which are leaves, others are sub-branches which lead to leaves. Not shown in Figure 1 are the leaf elements associated with the element types such as "LangString", "Vocabulary", "DateTime" and "Duration". The reader is referred to the IEEE LOM conceptual data schema for a full account of the types and value spaces of each element.

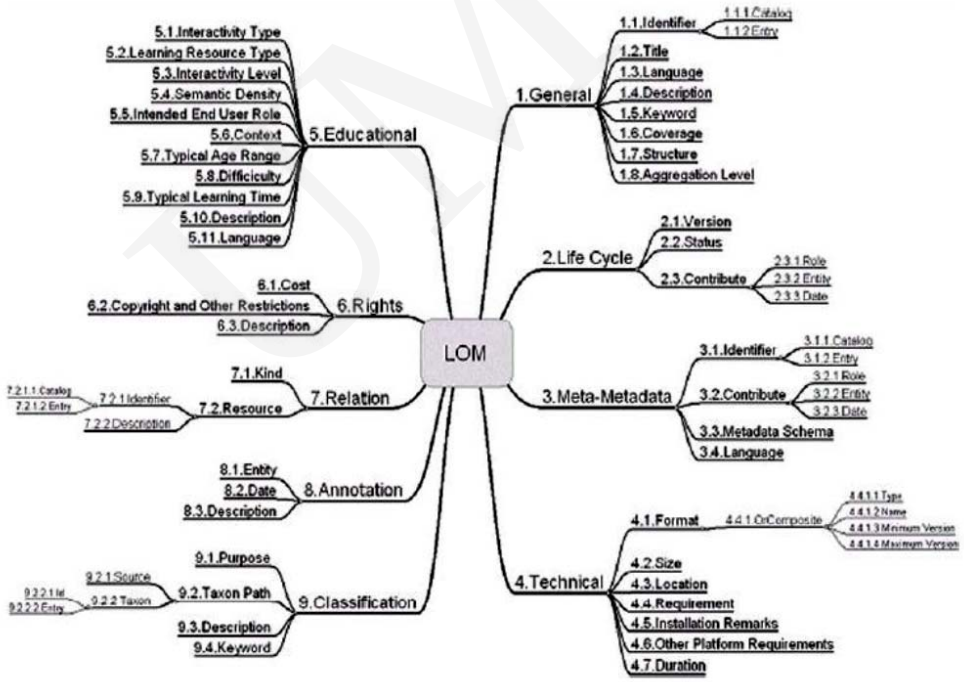

Fig. 1. The Learning Object Metadata scheme.

1.2 SCORM

Sharable Content Object Reference Model (SCORM) is a collection of standards and specifications for web-based e-learning. It defines communications between the client

side content and the host system called the run-time environment, which is commonly supported by a learning management system. The standard uses XML to display data. It is therefore highly structured. SCORM is a common standard format for e-learning content whilst other specifications allow for the transporting of learning contents or categorizing metadata (LOM).

The Sharable Content Object Reference Model (SCORM) was first developed by the U.S. Department of Defense (DOD) to address training development and delivery inefficiencies across its service branches. E- learning content was being developed on different platforms, using different standards and specifications, and delivered on different, incompatible systems. To address these costly inefficiencies, the DOD knit together the best emerging e-learning specifications with those developed in the prior decade by the Aviation Industry CBT Committee (AICC).

The result is a field-tested common reference model published by the Advanced Distributed Learning (ADL) Initiative, a collaborative effort between government, industry, and academia sponsored by the Office of the Secretary of Defense. The SCORM standard is focused on enabling the plug-and-play interoperability, accessibility, and reusability of Web-based learning content, with the ultimate goal of ensuring ubiquitous access to the highest quality education and training, tailored to individual needs, and delivered cost-effectively anywhere and anytime. Based on accepted technology standards including XML and JavaScript, SCORM is fast-becoming the de facto e-learning technology standard widely embraced and supported today by world-leading corporations, universities, system providers, and content vendors.

2 Motivation for the store model

As the discussed problem introduces the need to utilize databases as the most prospective approach seems to be introduced by dint of application of the objectivity. However, the approach presented mostly by the ODMG standard [**1**] for object databases or database-related Java technologies [**2, 3, 4**], despite significant role of object-oriented solutions in the remaining areas of software development, have not been succeessful yet. A different approach, in the form of SBA, shown in [**5**] has introduced existing object-oriented mechanisms (classes, encapsulation, inheritance, polymorphism, objects) for database programming. Moreover by applying the SBA, some new mechanisms have been introduced like dynamic object roles [**6, 7**] or interfaces on the database views [**8, 9**].

The second argument for utilizing the SBA is a powerful query language extended to a programming language i.e. SBQL (Stack Based Query Language). As the most important feature of the ODRA(Object Database for Rapid Application development) prototype SBA implementation, SBQL alone makes it possible to create fully fledged database-oriented applications. In the case of such solution the development of database application tasks with just one, very high level language, can greatly improve programmers' efficiency and software stability along the development life cycle.

```
<?xml version="1.0" encoding="utf-8"?>
<!--This is an ATutor SCORM 1.2 Content Package document-->
<!--Created from the ATutor Content Package Generator - http://www.atutor.ca-->
<manifest xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2" xmlns:imsmd="http://www.imsglobal.org/xsd
xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 http://www.imsglobal.org/xsd/imscp_v1p1p4.xsd
version = "CP 1.1.4">
    <metadata>
        <schema>ADL SCORM</schema>
        <schemaversion>1.2</schemaversion>
        <imsmd:lom>
          <imsmd:general>
            <imsmd:title>
              <imsmd:langstring xml:lang="en">Podstawy Inżynierii Oprogramowania [PIO - ZAOCZNI]</imsmd:la
            </imsmd:title>
            <imsmd:description>
              <imsmd:langstring xml:lang="en">Inżynieria oprogramowania - dziedzina inżynierii systemów za
            </imsmd:description>
          </imsmd:general>
          <imsmd:lifecycle>
            <imsmd:contribute>
              <imsmd:role>
                <imsmd:source>
                  <imsmd:langstring xml:lang="x-none">LOMv1.0</imsmd:langstring>
                </imsmd:source>
                <imsmd:value>
                  <imsmd:langstring xml:lang="x-none">Author</imsmd:langstring>
                </imsmd:value>
```

Fig. 2. Example of a metadata file conforming SCORM for e-learing course content description.

By dint of this solution, integration of e-learning resources stored in the object-oriented store model, enables the possibility of utilizing all of the SBA/SBQL best features and extend the performance and maintenance potential of such appliance.

2.1 Object- orientedness

Object is an abstract entity representing or describing some idea existing in a real world. Object is distinguishable from other objects with its unique name and distinct limits. For the purpose of this paper the object then is a data structure stored in memory area of a computer.

In contrast to the relational model, the object model does not assume a need for determining an attribute (or a set of attributes) that identifies the object in an unambiguous way (so called "primary key"). Instead the object has its own identity i.e. it exists irrespective of its own state or other objects. This sums up to an idea of unambiguous external object ID – OID. OID is not relevant regarding the content it describes in the working information system.The user of administrator never uses OID directly, instead the symbolic name[1] of an object is being the subject of the object manipulation.

[1]The name is exchanged to the OID along the binding process.

Every object has a name that is a handle to this object. The name of an object does not have to be unique (we can create multiple objects named *Course*). Classic object models e.g. ODMG model, name such issue a collection of objects, where the name is the name of entire collection not just the single element. In the discussed model we will assume that each and every object has its name e.g. *Course*, but there will be possibility to group entire collection as one object named *Courses*.

Every object has state represented as a combination of its every component mainly values of its all attributes and references to other objects. The state of an object can change in time. In our approach to present the e-learning content we will discuss below some of the attributes that can be utilized to make the standards describing each course fit the store model.

- Atomic attribute: such as the course symbolic name. It includes exactly one value, which is indivisible from the point of view of a user
- Complex attribute: such as the course author address. It includes many atomic values. It has the hierarchic structure where each branch of the hierarchy has its name (e.g. city, street, flat number)
- Pointer attribute: contains as a value the value of the OID of referenced object
- Multimedia attribute: is an atomic value of an extensive size, representing most often some multimedia data e.g. audio, video, graphics etc
- Repetitive attribute: it includes a variable in time number of values. Those values can be of atomic, complex or pointer type (e.g. list of issues addressed by e-learning course)
- Optional attribute: in a particular instance of an object it can have a value but it is not mandatory
- Derivative attribute: value that derives from other attributes
- Class attribute: value that is common to a set of objects belonging to the same class

This list is not complete. However, the rule of object relativity mentions that every object can be composed of an unlimited number of sub-objects. This way every attribute is an object. Moreover, each attribute has its type. Therefore the combination of attribute types is the type of an object.

2.2 Store Model – AS0

To introduce the e-learning course information to the object-oriented manner store, working with the SBA prototype implementation there is a need for adopting at least the simplest store model i.e. AS0 [**5**]. In contrast to the relational model, the object model requires to use far more concepts. There is also different understanding for many terms. Therefore, it is hard to introduce a model that can be simple and at the same time applicable for all cases equally. The prototype includes the whole hierarchical family of store models, each responsible for extending the possibilities of the predecessor

but all based on the same semantic base. For the purpose of this paper we mention only the most basic but sufficient for this appliance model – AS0.

AS0 can cover arbitrarily connected hierarchical data structures. However, it does not include the aspects of the class, inheritance or interface. It was originally designed to express the semantics of relational query languages. What is the essential part of it, is the possibility of representing semi-structural data in general and the XML data structures in particular.

Regarding this store model we will assume the object relativity rule and related to it the rule for the inner identification. The first one has already been mentioned in the previous section. The second rule states that every object that can be a component of different objects has to include its own unique inner identifier. Let us explain some basic terms

- Inner identifier of an object. It is given automatically by the system and cannot be used in the semantics of the outer manipulation of the objects. Its purpose is to identify objects stored in memory.
- Outer object name. In contrast to the inner identifier this name is created by the system designer, administrator or a programmer. It is linked to the conceptual model of the application working with the AS0 based store. Moreover, it involves the use of informal semantics for the outer processes e.g. the name can be *Author or Contributing Institution*. The outer object name does not have to be unique.
- Atomic value. It is a kind of object value that is indivisible from the point of view of the creator hence not including any parts.

Identifiers are marked as i , the names as n and the atomic values with the letter $\ni$. In AS0 every object contains a unique inner identifier, an outer name and the value that can be atomic, pointer or complex. We will present the objects by the following definition. The object is a triple:

- $< i, n, v >$ - when the object is going to be atomic
- $< i_1, n, i_2 >$ - when the object is going to be a pointer or a reference object. This object is identified thanks to i_1 where i_2 is the pointer value of the object being a reference to another object.
- $< i, n, T >$ where T is a set of any type of objects. This object we will call a complex object. This rule is recursive, therefore enabling building objects with unlimited complexity and a number of hierarchy levels.

In AS0 the data store is defined as pair $< S, R >$ where S is a set of objects and R is a set of object ids also named the staring identifiers.

The R set sets the starting points for the data store i.e. those objects that can be a starting point for the navigation in the entire set of objects. Most often those objects would be just simply the ones that are in the main level of the object hierarchy i.e. those that are not included as part of the other objects.

There are a few rules regarding the data store that has to be complied:

- each and every object, sub-object, etc. in data store has its unique identifier

- if there is a pointer object <i1, n, i2>, then the pointed object i2 has to exist.
- each and every identifier from the R set is an identifier of an object located in store.

3 Content Repository

Introducing a well designed e-learning content, fulfilling all requirements defined by educational platforms and standards as well as end user expectations indeed has been a major problem since the e-learning appeared. This process almost always includes many graphical and multimedia elements as well as a written content, however, the proportions between the number of valuable content and the multimedia are often inappropriate for fast and effective learning. All this makes the creation of a valuable e-learning course a very complex process. In this situation sharing ready to use courses between parties has become desirable. There have been many systems developed for this purpose based on p2p or jixta, moreover some mechanisms have been devised for reediting the content.

3.1 What is repository

Let us define repository as a store of objects or other data. Most frequently it is used to describe a store containing data of different types (documents, classical data, multimedia data, software data, scripts, rules or views) and delivering visual access interface. In practice, this definition fits the needs of sharing the repository educational content and in particular, SCORM that moves the responsibility for sharing objects from the dedicated tools to the LMS platform.

In e-learning there are three main levels of interest:

- content editing level
- communication and exchange level
- facilitate level

The content editing level can be easily presented (Fig. 3):

According to the mentioned standard, the LMS platform should enable the presentation of the whole content created with various tools to the end user. Moreover, the content created by a particular author should be accepted by any platform created by providers. It forces the platform and tools providers to conform to the standard constraints. In the case of the platform providers, the standard is obeyed, however, in the case of the proprietary tools unfortunately, the standard directives are often omitted. This effectively disables transfer of objects between platforms. The functions implemented are as depicted in Fig. 4.

3.2 Object model based repository

The scheme working for the e-learning systems can, however, be evolved to object model itself by adopting a prototype object database model of AS0.

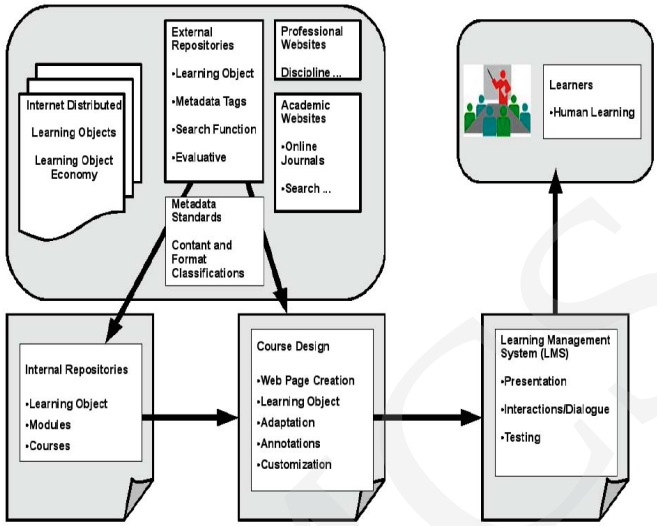

Fig. 3. E-learning course editing model.

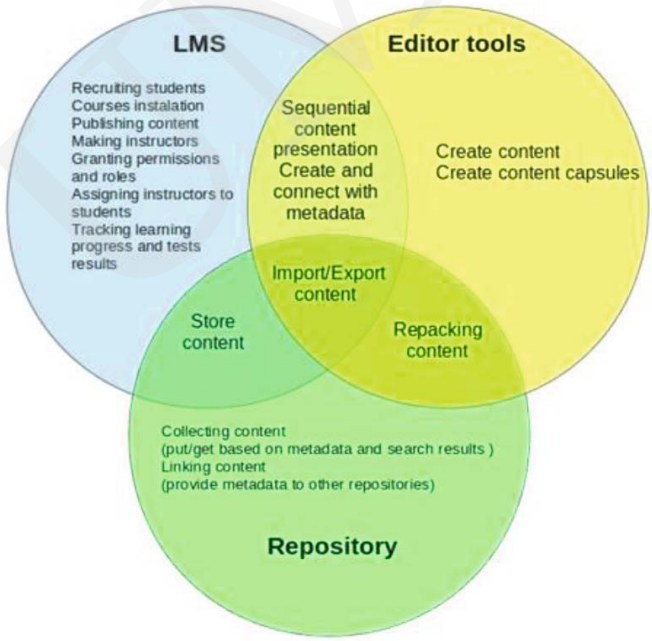

Fig. 4. Each product functions.

As discussed in the previous sections, the data according to the standards can be moved to an XML scheme. It is possible for both, import or export of the e-learning course content and description.

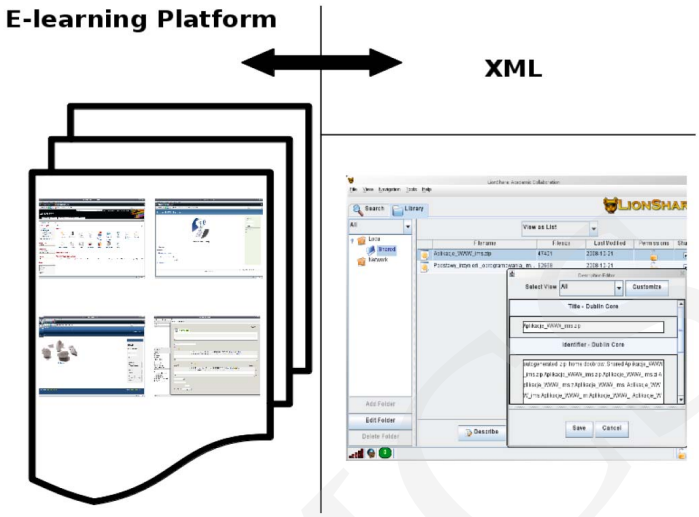

Fig. 5. Export and import of a standard conformed schemes.

We propose to adopt AS0 in a manner that conform to the e-learning standards and moreover, allows the adoption of an object oriented database model. The transformation of the structured data provided by the e-learning import/export mechanism has been modified to the AS0 model as depicted in Fig. 6.

```
<i1, metadata,{ <i2, schema, "ADL SCORM">,
    <i3, schemaversion, "1.2">,
    <i4, imsmd_lom, {
        <i5, imsmd_general, {
            <i6, imsmd_title, {
                <i7, imsmd_langstring, "Podstawy Inżynierii Oprogramowania (PIO - ZAOCZNI)">
              }>
            <i7, imsmd_description, {
                <i8, imsmd_langstring, "Inżynieria OProgramowania - dziedzina inżynierii systemów (...)">
              }>
          }>
        <i9, imsmd_lifecycle, {
            <i10, imsmd_contribute, {
                <i11, imsmd_role, {
                    <i12, imsmd_source, {
                        <i13, imsmd_langstring, "LOMv1.0">,
                      }>,
                    <i14, imsmd_value, {
                        <i15, imsmd_langstring, "Author">
                      }>
(...)
                  }>
              }>
          }>
      }>,
  } >
```

Fig. 6. The e-learning course data modified to the AS0 model.

The whole metadata has been treated as a complex object. The AS0 model allows the creation of subobjects, pointer objects and atomic object which is enough to conform the standards present in e-learning.

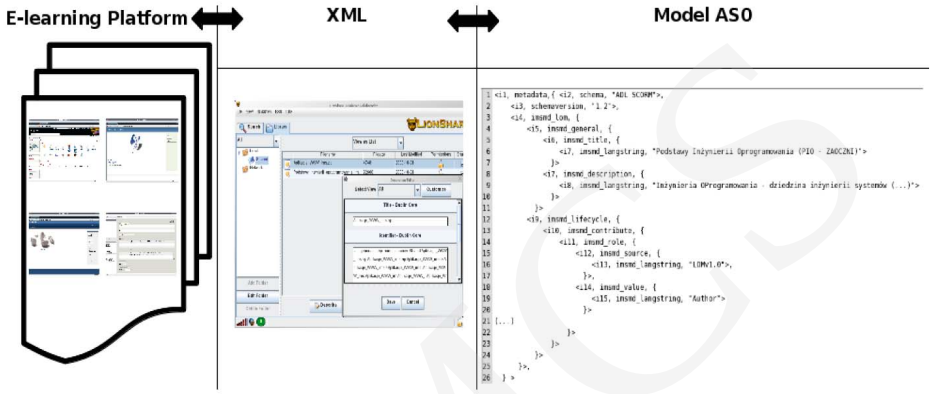

Fig. 7. E-learning data presented and utilized in the AS0 model.

At this stage we can further utilize the data not only form the level of object database store but what is more, we can use it as a source of data that can be utilized by front end client applications that are aware only of the database itself. Therefore the e-learning data can be modified, copied and presented in the arbitrary and custom way relevant only to the database user connection.

4 Conclusions

The goal achieved by the research was to enable the prototype object oriented database AS0 model to perform persistent data mediation between them. Introducing object orientedness to the ground of e-learning exchange platform brings it to the whole new level of metamodel. This gives us all the possibilities of an object oriented approach and causes that a database engine leads to data storing as persistency, durability, high availability, reliability with transactions. Creating folders with the information about available e-learning courses based on general knowledge from the available repositories seems extremely interesting – there is the ability to create new courses with the use of existing ones adopted to individual learners' needs. Using the reliable and tested catalogue patterns makes navigation through the high volume e-learning course information data stores amazingly easy. It would be enough to point to an author,a title or the requested course topic to efficiently obtain the wanted materials and even ready to use course components (text, graphics, multimedia etc.). Apart from that there are the issues related to copyrights regarding the reuse of the requested content, however, this is out of scope in the paper.

References

[1] Cattell R. G. G., Barry D. K., The Object Data Standard: ODMG 3.0. Morgan

[2] Cook W. R., Rosenberger C., Native Queries for Persistent Objects A Design White Paper (2006); http://www.db4o.com/about/productinformation/whitepapers/Native%20Queries%20Whitepaper.pdf

[3] Lentner M., Subieta K., ODRA: A Next Generation Object-Oriented Environment for Rapid Database Application Development Advances in Databases and Information Systems, 11th East European Conference, ADBIS 2007, September 29-October 3, 2007, Proceedings., LNCS 4690, Springer, ISBN 978-3-540-75184-7 (2007): 130.

[4] Hibernate - Relational Persistence for Java and .NET. http://www.hibernate.org/ (2006).

[5] Subieta K., Theory and Construction of Object-Oriented Query Languages. PJIIT - Publishing House, ISBN 83-89244-28-4 (2004), 522 pages (in Polish).

[6] Albano A., Bergamini R., Ghelli G, Orsini R., An Object Data Model with Roles. Proc. VLDB Conf. (1993): 39.

[7] Jodlowski A., Habela P., Plodzien J., Subieta K., Objects and Roles in the Stack-Based Approach. Proc. DEXA Conf., Springer LNCS 2453 (2002).

[8] KozankiewiczH., Updateable Object Views. PhD Thesis (2005); http://www.ipipan.waw.pl/~subieta/ -> Finished PhD-s -> Hanna Kozankiewicz

[9] Kozankiewicz H., Leszczylowski J., Subieta K., Updateable XML Views. Proc. of ADBIS'03, Springer LNCS 2798 (2003): 385.

[10] IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata Version 1.3 Final Specification.