

# A Distributed Content Repositories Model and a Decisions Support System for Learning Objects in Latin-America

Carlos Cobos Erwin Meza Martha Mendoza Jorge Moreno  
Norman Muñoz \*

## Abstract

This paper describes a content object and metadata repositories model, which provides an alternative to search and to access repositories developed in Colombia and Latin America. Along with the repositories model, an initial design of a decision support system for repositories is presented. In order to preserve the autonomy of each repository, a common web services interface is considered. This interface can be implemented on different software platforms or operating systems, and it also can be integrated within existing solutions. The repositories that accomplish with the services interface are based in IMS DRI 1.0 and SCORM 1.2, and they will have an internal catalogue that manages the content objects, their associated metadata and other catalogues for the development of distributed operations. By the other hand, the decision support system is a tool that provides the information required to evaluate and to control the quality of both the services and the resources offered by the repositories, among other parameters. The development of the decision support system will require technologies such as Data Warehouses and On Line Analytic Processing. At a technical level, this document describes a relational database model which allows manipulating the learning object metadata stored in a content repository, no matters if the database engine supports XML or not. According to the relational model, the tables, their functional description and their relationships are presented. Finally, the dimensional models of the decision support system and the web interface offered by the current repository implementation are described.

**Keywords:** *IMS DRI, Content Repositories, SCORM, Data Warehouse, OLAP.*

## 1 Introduction

Nowadays the Sharable Content Object Reference Model (SCORM) [1], integrates the IEEE Learning Object Metadata (LOM) [2] meta-data schema for describing learning objects (Sharable Content Objects, SCOs), a Run-Time Environment and an Application Programming Interface (API), in order to provide the means to share learning objects across many courses offered by one or more Learning Management Systems (LMS). Based on this development, the creation of digital repositories has taken place, with the aim of providing some services for the storage, organization and management, and also searching and accessing learning objects [3]. However, in spite of the existence of LMS and repositories that make use of SCORM as reference model, there is not a globally accepted standard which allows the required interoperability to perform searching and accessing of learning objects and their meta-data among different repositories. An

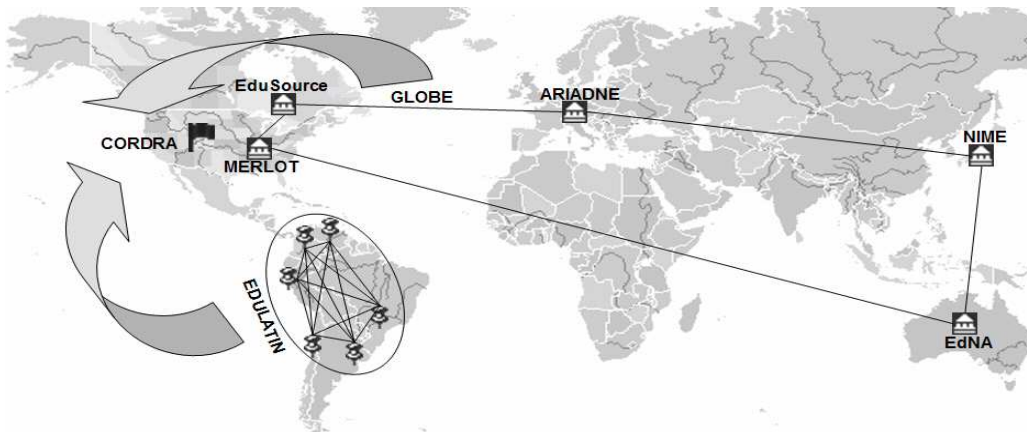
---

\* University of Cauca, Systems Department, Calle 5 # 4-7, Popayán, Colombia {ccobos, emezav, mmendoza, jjmoreno, nmunoz}@unicauca.edu.co

approach to solve this problem consists in developing an architecture specification for learning object repositories such as the Digital Repositories Interoperability (IMS DRI) [4], that proposes a collection of services which a repository must provide, or CORDRA™ (Content Object Repository Discovery and Registration/Resolution Architecture) [5] that provides a model to search and access learning objects based on a federated repositories schema which performs a registry into a central system. Merlot [6], EdNA [7], ARIADNE 0, EduSource Canada 0, NIME 0 are some existing developments that allow searching, accessing and recovering learning objects and their meta-data from content and / or metadata repositories.

The current developments will let that in the near future there will be a big amount of learning objects spread across many latin-american repositories that comply with some metadata specifications such as IEEE LOM o Dublin Core. In this sense, our first purpose is to develop a model to locate and use learning objects in many LMS around of different countries, initially in Latin America and later at global level. Fig. 1 shows in first place, how Merlot, EduSource Canada, Ariadne, NIME y EdNA are working together on the GLOBE (Global Learning Object Brokered Exchange) project, and in second place, it illustrates how a collection of latin-american repositories (EduLatin) can be integrated into a repository federation such as CORDRA.

With the principle of avoiding the required effort at the time to develop content repositories and their metadata, this work will provide an initial repository implementation which complies with the interoperability specifications described in the proposed model. This repository implementation can be adapted to specific requirements in each community. By using this implementation, each country or entity will be able to focus in their own process of building of learning objects and courses, as they can use the functionalities provided by the repositories to manage and distribute their resources. By the other hand, the model will also be able to work together with existing digital repositories, through a public services interface that extends their functionality in order to offer search and access services to learning objects and stored metadata in any repository inside the Latin-American Repositories Network (EduLatin). EduLatin is a Web Services based Network that follows the IEEE LOM metadata standard, and in the near future it will allow the integration of the Latin-American repositories network with both GLOBE (distributed repositories registry) and CORDRA (centralized registry).



**Fig. 1. Integration of EduLatin into the global repositories network**

At this point, creating and validating a model for interoperable repositories is the main purpose of the SCORM Public-Access Repository (SPAR) project at the Information Technologies Group or GTI (Spanish Abbreviations) of the University of Cauca (Colombia), this project is derived from the Virtual University of Cauca project. The SPAR project is intended to improve some of the

functionality implemented at the Virtual University of Cauca, and allows that each LMS can focus only on its own responsibilities. Based on the study of the Specification for Digital Repositories Interoperability developed by IMS and the CORDRA model, created by Carnegie Mellon University 0, there are some key aspects of the proposed model:

- The increasing needs due to provide a service to search and access learning objects and metadata through Internet, no matters the details of storage implementation and repository management.
- A network of repositories with a standard functionality, where learning objects and metadata can be shared across many repositories for searching and accessing services from any network repositories node.
- The generated structure will have to support to different metadata schemes that can be represented as XML documents (XML Binding), such as LOM or Dublin Core 0, besides of considering other specifications that are part of the metadata schemes but that they are not represented as XML, such as VCARD specification 0.

After defining the core functionalities of the model for interoperable repositories, the SPAR project also considered implementing a strategy to measure some of the aspects related to the repositories, the resources and the services they provide. In this direction, a decision support system is being developed, which will allow evaluating the quality of the contents and metadata, the efficiency of the web services interface, and the user preferences.

Next sections will present a brief description about the proposed repositories model and its more important components, some details of the current implementation, and also some models of the decision support system. Finally some conclusions about the developed work will be presented.

## 2 Digital repositories interoperability model

In order to develop the interoperability model, some of the online education standards, reference models and specifications were studied. Initiatives such as the Advanced Distributed Learning (ADL) (<http://adlnet.org/>), the SCORM [1], the IMS Global Learning Consortium Inc. (<http://www.imspj.org/>), the Aviation Industry CBT (Computer-Based Training) Committee (AICC), the Alliance of Remote Instructional Authoring & Distribution Networks for Europe (ARIADNE in <http://www.ariadne-eu.org/>) and finally the Technology Standards Committee of the IEEE Computer Society [2]. The following key aspects were considered in this work, based on these specifications:

- A descriptive collection of attributes concerning resources and media (assets), sharable content objects and courses, known as learning object metadata (LOM). These metadata working together with some guidelines for building SCOs, allow increasing the possibility of sharing content resources among many institutions with different LMS like WebCT, BlackBoard, Moodle and others.
- A specification to build an interoperable digital repository, courses and media capable of working and sharing information with many LMS. This specification proposes the following core of operations [4]:
  - **Search/Find:** The capability to search/find a learning object from the repository. Visualization ability can be included.
  - **Request:** To request for a learning object that has been located.
  - **Retrieve:** A located learning object can be retrieved.
  - **Submit:** Sending and storing a learning object to a repository. In other words, adding a learning object to a specific repository.
  - **Store:** Saving a learning object to a specific repository with a Global Unique Identifier in order to assure its localization and manipulation.

- **Gather (push/pull):** Obtaining metadata information concerning learning objects located in another federated repositories.
- **Publish:** Providing metadata information about learning objects from a repository to others repositories and systems.
- The proposed specification has been taking some aspects related to the Learning Systems Architecture Laboratory work at the Carnegie Mellon University. At this time, this group is defining a Federated Repositories Model for the CORDRA project [5].

Now, the proposed model at the SPAR project is based on a repositories architecture where each repository collaborates with others (regardless of its physical location) in order to provide some services to search and access learning objects and their metadata (Fig. 2). The main feature of this model consists of the non-dependent logic repository implementation and the ability for repositories interaction from any network node through a well defined services interface. To achieve these features, XML Web Services were proposed as a developing platform to support information communication/transmission operations among repositories. Because XML Web Services are based on HTTP Protocol, a new protocol or middleware implementation is not required, this is especially appropriate to ensure the integration of current repositories by developing an additional services layer to hide the logic implementation on each repository. The Fig. 2 describes the main components of the proposed model.

The purpose of the LMS is to provide the necessary resources to develop significant learning experiences, so, the LMS uses the digital repositories and it allows them to manage learning objects and their metadata. A LMS must implement some searching, visualization and evaluation intermediate services for learning objects between clients and repositories. The clients request for contents, the repositories manage them and the LMS access them (the Fig. 2 shows the first and second steps: firstly the user requests a learning object, secondly the LMS requests the learning object from the repository).

The digital repositories can be defined as collection of resources that can be accessed by a network without knowledge about its internal structure. The stored resources are composed by learning objects on a folders structure directly accessible from Internet and their metadata is stored into XML files or into a Database Management System (DBMS). At the SPAR Project a DBMS with native XML support to store the metadata schema was considered, though, not all DBMS supports native XML, so, the final choice was an auto-reflexive tables in a relational model (see Fig. 5). This would allow the implementation of a repository no matter the systems or platforms. In addition, this model deals with the storage of hierarchic metadata like LOM, or another data representations like attribute-value which are not XML. This advantage of the proposed relational model allows the storage of the complete specifications and the inclusion of non-XML specifications at the same metadata schema, such as VCARD 1.0 (attribute-value) which is part of LOM (XML format).

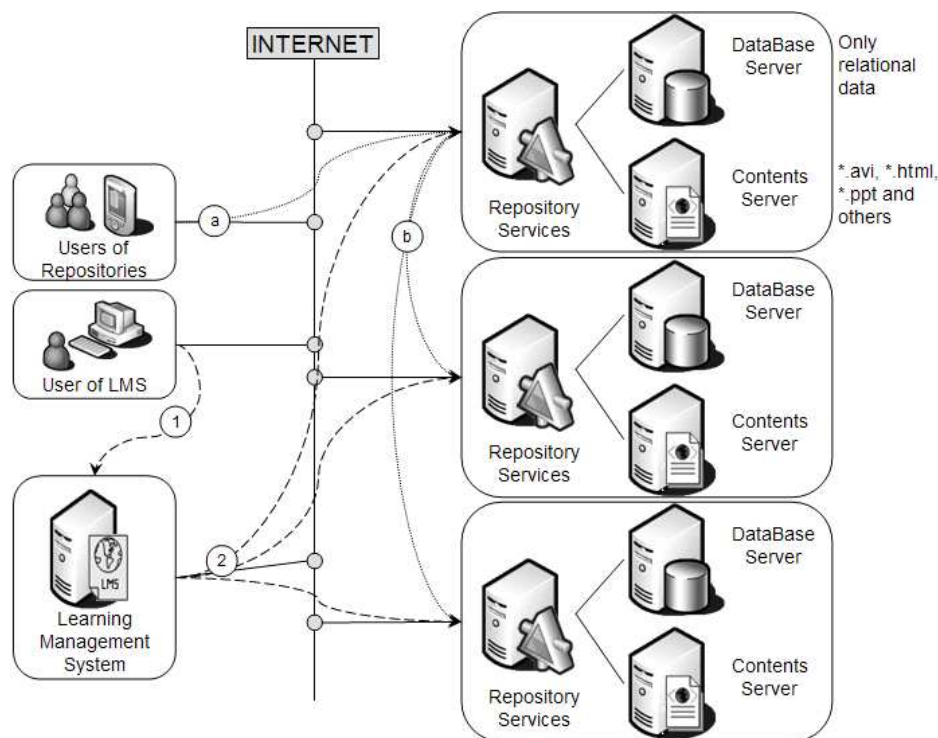


Fig. 2. A Distributed Repositories Model

The Fig. 3 shows the internal components of a EduLatin Network repository. Each repository must have the following internal components:

- **A database model which is responsible for object learning metadata storage (learning objects metadata and own data repository).** This model can be implemented in any relational database engine, no matter if it doesn't support XML. Currently, the model has been tested to store SCORM 1.2, LOM 1.0 and VCARD 1.0 and some tests are running to store the Dublin Core metadata schema.
- **A Distribution Master Catalogue.** It allows “twin” repositories registration across the network and it contains the required information to execute the distributed operations, such as searching for learning objects across the repositories network. It is expected that in the near future this catalogue can be used to perform data replication on demand. The catalogue along with the distribution services logic enables the communication between a repository and the network on a distributed database environment.
- **A collection of basic repository services.** These services fulfill IMS DRI, and they implement operations such as: Search/Find, Request, Retrieve, Submit, Store, Gather (push/pull not fully implemented) and Publish.

An important feature of this model is that the source of the learning objects is not significant, neither the characteristics of the LMS which are requesting them. This feature allows storing the learning objects or the metadata in separate repositories spread across the network, even if those objects or metadata belong to the same course or content aggregation. To implement this feature, the repositories and the LMS should meet the requirements of a public interface, which permits to establish the connection to the repository network and request, recover and present the learning

objects to the users, regardless of the source repositories. The repository network will be responsible of ensuring the discovery and delivery of the learning objects and the metadata.

In addition, becoming an active member of the repository network requires the registration of the new repository into the local distribution catalogue of any of the repositories already registered on the network. This repository must report the addition to the remaining members of the network, in order to replicate the registry on their local catalogues. To avoid the duplicity of learning objects and repositories, a Global Unique Identifier scheme is used to identify each repository and learning object. This approach allows the recovery of the learning objects no matter their physical location or the implementation of the storing repository. In the same way, when a new learning object is registered and submitted into a repository, a replication of its information is performed among all the members of the network. To achieve this goal, the repositories must implement two required services: “Gather”, which allows the recovery of learning objects metadata from another repositories, and “Publish”, which provides the metadata of the learning object to all other repositories.

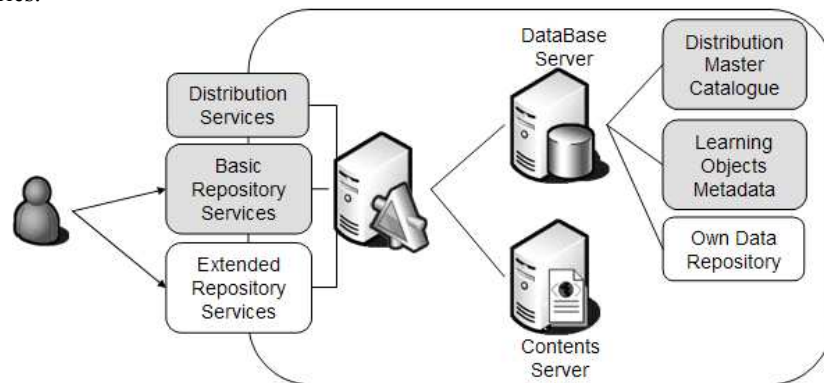


Fig. 3. Internal View of a Repository

### 3 Web Services Architecture

A study about the core services defined at the IMS DRI specification has been required by the SPAR project. In that specification a set of repository requirements must be accomplished for integrating the repositories into EduLatin. These requirements were raised in terms of the services that must be implemented by the repositories, and that allow them to interact with the learning management systems and with other repositories, at a web services level. The most important requirements and their relation with IMS DRI can be observed in Fig. 4 and they are described later in this document.

#### 3.1 Request and Response Structures

To achieve a high interoperability level between many digital repositories implementations, simple SOAP messages were chosen to make requests and responses. These messages use data types defined on the public interface WSDL document on each repository, allowing them to hide their functionality and internal resources and metadata database structure to the users. The proposed services are kind of stateless request/response but the model will incorporate more services soon.

#### 3.2 User Authentication

One of the most important aspects of the interoperability repositories model (partially defined at the IMS DRI specification), is the repository capability to recognize its users no matters if they are

LMS, Web Services Based Applications or another repositories. To achieve this goal, a user key based security policy was defined similar to the Google Search Service 0, in which each single user who uses services from a network repositories must be authenticated by means of a key. This key is obtained through a simple registration process and must be passed as a parameter to each service call, due to all model operations are stateless.

### **3.3 Repository Information**

This service allows obtaining metadata schema information supported by each EduLatin Network repository and can be used later to structure all resources search/insertion and metadata administration requests. The user must specify his/her/its unique key, then, the repository gives back a response with a list of reference models and its supported components and the version of metadata schemas.

### **3.4 Resource Search and Location**

This service is based on the IMS DRI Search/Expose function, allowing search/locate any resource from the EduLatin Network Repository. To perform a search, the user must establish a connection to a specific repository. Then, he/she/it specifies a search pattern, his/her/its unique key and the resource type (Assets, SCOs and content aggregations). With those parameters, the repository that receives the request, executes the search at its own local database, at the same time, it automatically sends the request to all repositories where it is registered. Once obtained, the results are sent back to the user in form of a XML document at the body of the SOAP message. Inside the returned information there is a Global Unique Identifier for each located object, its name, location and type depending of each specification. In addition, with these results, the user can execute others requests, such as obtain metadata information (gather/expose) or requests the resource for saving/visualization (request/deliver).

### **3.5 New resource addition**

This service fulfills the established IMS DRI requirements for the Submit/Store function allowing the addition of new resources to an EduLatin Network's repository. The user can use this service by means of a submit request, which includes his/her/it unique key, the desired resource type to store, an URL address where the resource is located (satisfying the IMS Content Packing Specification) and same basic information about the new resource. The provided URL address is used by the repository to obtain the desired resource, verify its contents and save it to the structure of directories. Once the transfer of the new resource is completed, the metadata schema is determined by the repository and all the basic information provided by the request is added to that schema. Finally a success/fail message is returned by the repository to the user.

### **3.6 Addition/Update of Metadata**

This service allows adding or updating the resource metadata inside a repository. The user must specify his/her/it unique key, the Global Unique Identifier of the resource and the new resource metadata in XML format, then, the repository will update the old metadata information (after a previous metadata validation) with the new one.

### **3.7 Metadata Publication**

This service allows obtaining the metadata registry of any resource inside at any repository and can be used for the LMS or other repositories. In next future, this service could be extended to "Push" Gather and "Pull" Gather functions, in order to share the metadata between the repositories that will integrate the EduLatin network.

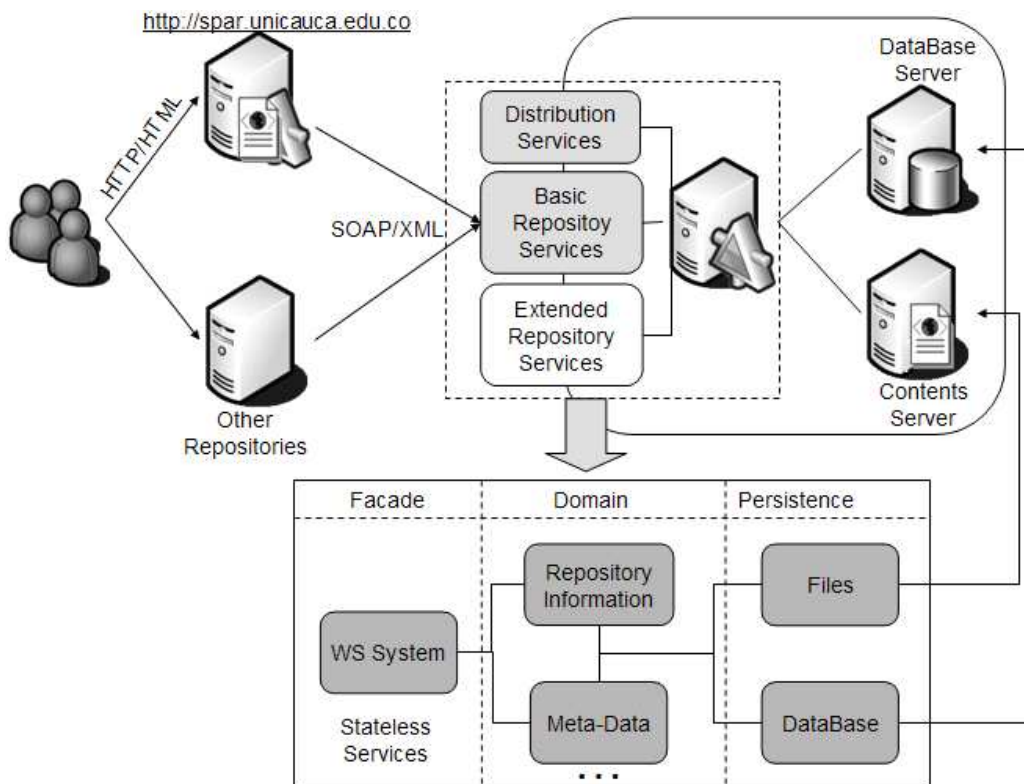


Fig. 4. XML Web Services Architecture and interaction with other systems

#### 4 Physical database model for metadata management

A review about metadata schema management and XML documents (stored at text fields in a database or plain text, and databases with native XML storage), was performed 00. A set of software technologies was found, for example: Ariadne, ImseVisme, LOM editor, EUN, ADL SCORM, Reggie, EM2, Reload Editor and Metadata Tool 0000. These technologies provide a diversity of interfaces from standalone client applications to Web Applications fully integrated with existing repositories and with functionality to import/export metadata in XML format.

The plain text storage was discarded, due to performance requirements of the repository. By the other hand, XML native databases (with Xquery and Xpath) won't be considered, because not all RDBMS provide this feature and we require that the proposed model can be instantiated over all possible RDBMS. Also some specifications (like VCARD 1.0 inside LOM) are not represented as XML, so Xquery and Xpath could not be used to perform a search in these data. The following are some basic requirements of the final solution:

- To store any kind of Standard metadata specification for SCOs (not only SCORM 1.2 and LOM 1.0).
- The client application (Windows, Web, Pocket Windows, and others) must be able to dynamically create metadata forms (to get data) based on schema attributes of a learning object selected by the user.
- A Global Unique Identifier for each learning Object must be stored no matter the repository.



- To perform the storage of the metadata the system must fulfill the SCORM and LOM schema specification (or any other metadata schema selected), and it should not establish any type or restriction about the multiplicity of the metadata elements to fill, or the handling of those elements whose cannot be mapped to XML, such as personal and organization virtual cards (VCARD). This requirement emerged after the analysis of the RELOAD editor, which allows adding only one instance of each XML element (one title, one description, and so on).

The Metadata tool developed by Learning Systems Architecture Lab at Carnegie Mellon University was considered after the mentioned review above. A relational database model was conceived for different metadata schemas (already tested with SCORM 1.2, LOM 1.0 and VCARD 1.0) and the values of the learning objects that are stored in the repository. This model stores the hierarchy of the schemas and XML files by means of self-reflexive tables. An important consideration is related to the use of database with XML Native Support will not guarantee a better performance but will be easier. Therefore, a deeper analysis will be made to compare native XML database and database with capability of XML storage at text fields 0.

The Fig. 5 shows the database model (tables and its relationships) which allows to store the SCORM 1.2, LOM 1.0 and VCARD 1.0 metadata schema, basic information about content objects and their metadata, information about the system users, operation and access statistics and finally some query key fields to each specification. Later on Table 1 each table of the model is described.

With this physical database model, generic data capture/importation/modification programs can be implemented, with diverse interfaces for Windows, Web, Pocket Windows and Batch, regardless of the XML schema data and logic specification used. The operations that can be performed on the model are similar to XPath, but are implemented in form of recursive operations with trees. For example, a query about the first level child elements of <LOM> should return general, lifecycle, classification, and so on. Also, it is possible to operate with any stored data for a specific Learning Object. Detailed information about this model and the implemented operations can be requested via e-mail to any of the authors.

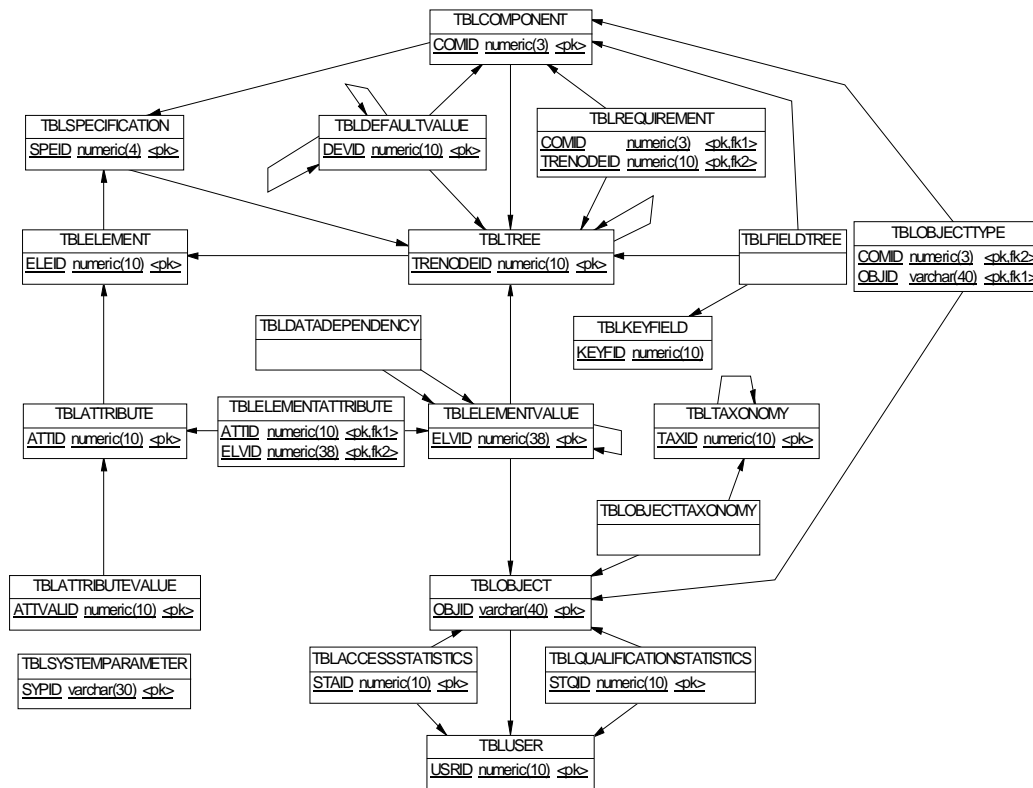


Fig. 5. Physical model for storing learning object metadata

Table 1. Description of the tables for storing several specifications

TABLE NAME	DESCRIPTION
TBLSPECIFICATION	This table is used to store records from different specifications; it has different fields that store the specification's version and how it will be bound to XML or another markup language. For example: <ELEMENT>VALUE</ELEMENT>.
TBLCOMPONENT	This table is used to store records that contain the different components of the specification SCORM 1.2 and other specifications. This table has different fields to define every component with its name, specification which belongs to and tree's root associate. For example, SCO, Asset and Content Aggregation are valid records for SCORM 1.2 specification.
TBLELEMENT	This table is used to store records that contain the different elements (XML tags or names) of SCORM 1.2 and other specifications. This table has different fields that allow mapping every element to XML. For example, GENERAL, TITLE, DESCRIPTION.
TBLATTRIBUTE	This table is used to store records that contain the different attributes associated to a specification's element. For example, the XML:LANG attribute for LANGSTRING in SCORM.
TBLATTRIBUTEVALUE	This table is used to store records that contain the different values for an attribute associated to an element that belongs to a specification. For example, ES (Spanish) for the XML:LANG attribute in SCORM.
TBLTREE	This table is used to store records that allow building the hierarchical structure of SCORM 1.2 and other specifications. This table permits to

	build a tree-like structure for each specification. For example, the hierarchical relation between GENERAL element and IDENTIFIER, TITLE y CATALOG ENTRY elements in SCORM.
TBLREQUIREMENT	Stores information about mandatory elements (of specification hierarchy) for each component. For instance: the CATALOG ENTRY element is mandatory for SCOs and CONTENT AGGREGATION, but it is optional for ASSETS.
TBLDEFAULTVALUE	Stores the records with values about the SCORM 1.2 vocabularies. It has in first place, a reflexive relation which allows knowing the possible default values after to select a default value of a parent element, for example, the CREATOR and VALIDATOR values in the ROLE element in SCORM. In second place, it has another reflexive relation to define the dependency of value of two fields, for example: the SCORM TECHNICAL element has a NAME (4.4.2) element; its value depends of the selected value for the TYPE element.
TBLOBJECT	This table is used to store records that contain the different learning objects that have been stored in a repository. It has different fields that allow finding a certain object with its characteristics. This table stores the Global Unique Identifier (GUID), the URL where was stored in the file system, visualization initial file and the Object's MD5 hash (simple strategy to avoid the registry of repeated resources).
TBLELEMENTVALUE	Stores the value (data) registry about the each learning object metadata. It has a self-reflexive relation which allows provide information about stored values at the specification hierarchy, this information charges the storing of data, but it is necessary to accomplish all multiplicities and metadata store.
TBLELEMENTATTRIBUTE	Stores specific values of elements attributes.
TBLDATADEPENDENCY	This table is used to store the value's dependency that has a data respect another.
TBLSYSTEMPARAMETER	Stores basic system parameters, such as the current disk location where the learning objects are stored in the server. This feature allows a better balanced storage of server discs file system or the content servers.
TBLOBJECTTYPE	It allows registering the component type (on a specification) for each object. This feature will allow the automatic forms and tree generation for data entry.
TBLUSER	Stores information about system users, for example: login, a password (authenticated with MD5), name and last name, an email address and a Personal Webpage URL.
TBLACCESSSTATISTICS	Stores access and operation statistics for all learning objects that users executes including the anonymous/public user.
TBLQUALIFICATIONSTATISTICS	Stores registries concerning qualitative(suggestions or comments) and quantitative(a scale from zero to 100) evaluation about learning objects, in order to establish a top most requested ranking for learning objects depending of users requests.
TBLKEYFIELD, TBLFIELDTREE	These tables stores and relates the key fields used by the system on different specifications. For example the TITLE field will be mandatory at the queries but not all specifications call it the same way. It does not mean these fields not will be used at the queries but there are some transversal elements to all specifications especially for fast search on distributed repositories.
TBLTAXONOMY	This table is used to store the classification of a learning object. This table has a reflexive relation which allows knowing the classification's path
TBLOBJECTTAXONOMY	This table is used to store the different objects with his classification's path.

## 5 A Decision Support System for SPAR

In order to incorporate a strategy for measuring and analyzing the quality of the resources and the efficiency of the services interface provided by SPAR, along with the preferences of the users, a decision support system was conceived. The design of this system has required studying of some methodologies and technologies such as Data Warehouses and Online Analytical Processing (OLAP).

Once finished, this study allowed the SPAR Project Team to select the Dimensional Modelling proposed by Ralph Kimball 0, as the methodology used to define the Data Warehouse System Architecture. The Data Warehouse System will be implemented on Microsoft SQL Server 2000, which offers the required services for building the Data Warehouse and to performing different kinds of analysis of the information gathered by the SPAR application.

The first step for designing the decision support system was to define the major areas of interest inside the repository, which was defined as content demand, content quality and user preferences identification. Once defined, a dimensional model for each area was developed. These models are explained below.

### 5.1 Content Demand Model

This model is intended to measure some topics related to the queries and submissions of new resources into the SPAR repository. The information provided by this model can be used to promote the creation of resources for specific areas of knowledge, and also improving the availability of the existing resources. It is also possible to determine the periods of time in which the repository is being used more frequently, and which users are interacting with the repository at those times (See Fig. 6).

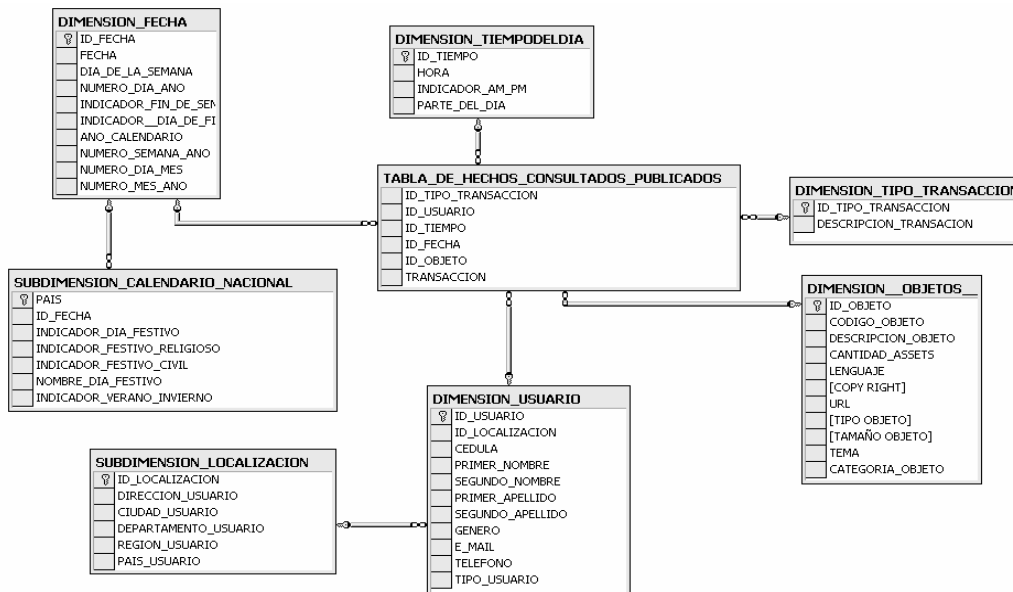


Fig. 6. Content Demand Dimensional Model

In this model there is a Factless table (Center table in Fig. 6) that stores each event (transaction) performed into the repository, it can be either a new resource submission or a query. The following dimensions are considered into the content demand model:

- Date: Allows identifying special features of a country's calendar, such as holidays, seasons, and so on.
- Learning object: It contains the information related to each learning object (Asset, SCO or Content Aggregation) currently stored into the repository.
- User: This dimension stores some descriptive attributes for each user (either publisher or query user). There was a location sub-dimension associated to the user dimension, in order to control its size and the update rate.
- Daytime: It contains fine-grained attributes that identify each day (eg. morning, afternoon, night).
- Type of transaction: This dimension allows identifying the kind of transaction performed by a specific user (either query or submission).

With the information provided by these dimensions, it is possible to store the (Factless) events, and also tracking the query or publication that each user performs into the repository.

## 5.2 Content Quality Model

This model will be used to evaluate the quality of the resources in terms of reliability and update. In order to gather the information from the users regarding the quality of the resources, a simple survey must be filled while the users are visualizing each resource. With this information, decisions about strategies for improving the quality of the resources can be made. Fig. 7 shows content quality model defined at the present day.

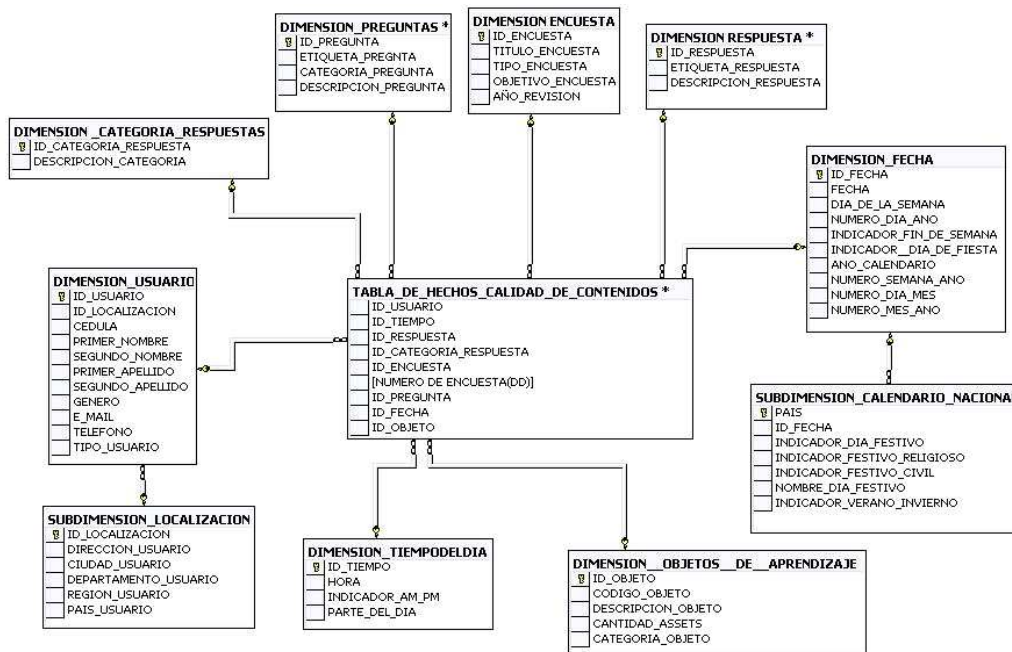


Fig. 7. Content Quality Dimensional Model

The model show in Fig. 7 contains a fact table in which each record represents the answers to the survey obtained from each user. The new dimensions considered for this model are:

- Survey: This dimension has some attributes that define the different kinds of surveys, and it includes information such as survey objectives, titles, and so on.
- Answers: It contains the attributes that define the answers for the surveys, which will be evaluated and categorized, in order to determine the quality of the resources stored into the SPAR repository.
- Answer Category: This dimension stores the different attributes that allow classifying the different kinds of answers obtained from the users.

### 5.3 User Preferences Model

This model is intended to determine the individual interests and preferences from users. In order to capture this information, a user tracking system was conceived to record each event and action taken by the users inside the web interface. With this information, the system can decide what kind of resources the users are more interested in, and how to adapt the interface of the web application in order to ease the user interaction and navigation. Fig. 8 shows the current User Preferences Dimensional Model.

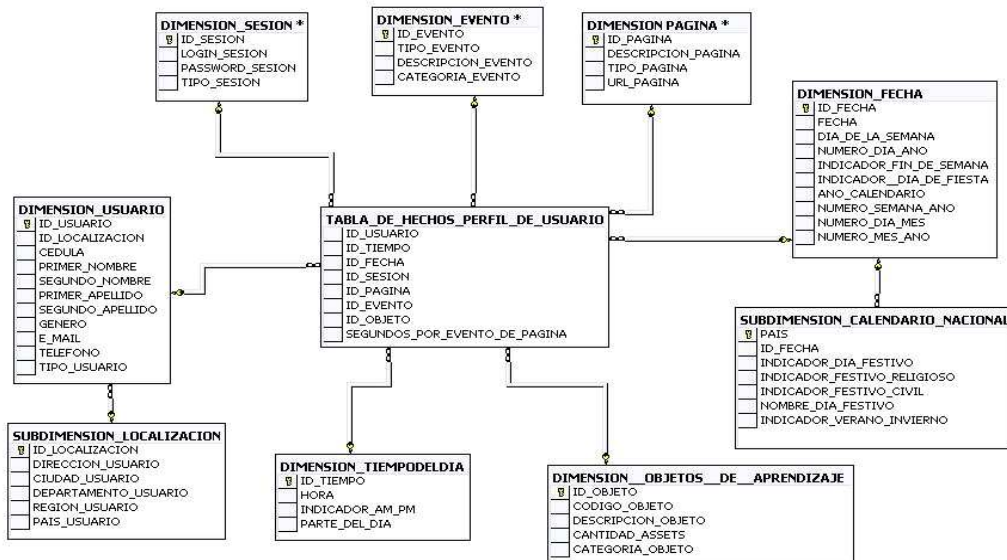


Fig. 8. User Preferences Dimensional Model

Into this model, each single event performed by the user is registered. These events can be as simple as a page click, or as complex as a submission of a new resource or a query to the repository. The new dimensions for this model are:

- Event: It stores the type of events that are tracked by the system. Attributes such as type of event and description are stored.
- Page: It describes the different attributes that identify each page of the SPAR Web interface. For this dimension the page description, page type and URL are stored.
- Session: This dimension is used to store the different attributes associated with a user session. A User session is the set of events triggered by the user from the login event to the logout event.

## 6 SCORM Public-Access Repository implementation

The first stable version of SCORM Public-Access Repository (SPAR 1.0) was released in February 2006. This version includes an initial implementation of the previously described Web Services Interface and Physical Model. It also includes a Web Interface, which allows the human users to have direct access to the services provided by SPAR.

The start page of the Web Interface (see Fig. 9) has a common header, which displays the name of the application (SCORM Public-Access Repository, in English, and “Repositorio de acceso público basado en SCORM”, in Spanish), and also a general menu that allows the users to access the services offered by the application, namely: Login, Switch to language (currently English and Spanish are available), Upload Resource, Modify my resources, Reference external resource and Download the SPAR current release, installation guide and general documentation. These and other options are briefly described below.

Bienvenido a SPAR [Entrar](#)

**Repositorio de acceso público basado en SCORM**  
 Busque, Almacene y Comparta Objetos de Aprendizaje

[Inicio](#)
[Idioma](#)
[Subir recurso](#)
[Mis recursos](#)
[Buscar recursos](#)
[Referenciar recurso](#)
[Descargas](#)

<p><b>Ingresar al sistema</b></p> <p>Nombre: <input type="text"/></p> <p>Clave: <input type="text"/></p> <p style="text-align: right;"> <input type="button" value="Registrar"/> <input type="button" value="Aceptar"/> </p> <p style="text-align: center;"><a href="#">¿Olvidó su clave?</a></p> <p><b>Los mejor calificados</b></p> <p><a href="#">DoS y DDoS</a> <a href="#">mas</a>  <span style="font-size: small;">★★★★★</span></p> <p><b>Los más accedidos</b></p> <p><a href="#">DoS y DDoS(9)</a>  <a href="#">mas</a></p> <p><a href="#">Historia de la Informatica 500 A.C. hasta 1949 D.C.(5)</a>  <a href="#">mas</a></p>	<p><b>Clasificación</b></p> <table style="width: 100%; border: none;"> <tr> <td style="width: 33%;"><a href="#">Artes (0)</a></td> <td style="width: 33%;"><a href="#">Negocios (0)</a></td> <td style="width: 33%;"><a href="#">Educación (0)</a></td> </tr> <tr> <td><a href="#">Humanidades (0)</a></td> <td><a href="#">Matemáticas y Estadísticas (0)</a></td> <td><a href="#">Ciencia y Tecnología (3)</a></td> </tr> <tr> <td><a href="#">Ciencias sociales (0)</a></td> <td></td> <td></td> </tr> </table> <p><b>2 Recursos encontrados</b></p> <p><b>DoS y DDoS</b>          Presentación que habla acerca de los diferentes tipos de ataques de denegación de servicios realizados actualmente. Tambien presenta algunas de las herramientas mas utilizadas en este tipo de ataques          es          Asset          22/02/2006 18:21:26  <span style="font-size: small;">★★★★★</span>  <a href="#">Descargar recurso mas</a></p> <p><b>Historia de la Informatica 500 A.C. hasta 1949 D.C.</b>          Presentacion interactiva de la historia de la informatica desde 500 a.c hasta 1949          es          Sco          24/02/2006 18:21:15  <span style="font-size: small;">★★★★★</span>  <a href="#">Descargar recurso mas</a></p>	<a href="#">Artes (0)</a>	<a href="#">Negocios (0)</a>	<a href="#">Educación (0)</a>	<a href="#">Humanidades (0)</a>	<a href="#">Matemáticas y Estadísticas (0)</a>	<a href="#">Ciencia y Tecnología (3)</a>	<a href="#">Ciencias sociales (0)</a>		
<a href="#">Artes (0)</a>	<a href="#">Negocios (0)</a>	<a href="#">Educación (0)</a>								
<a href="#">Humanidades (0)</a>	<a href="#">Matemáticas y Estadísticas (0)</a>	<a href="#">Ciencia y Tecnología (3)</a>								
<a href="#">Ciencias sociales (0)</a>										

**Fig. 9. Initial SPAR 1.0 Web Interface**

After loading the SPAR initial page, the user can browse the objects stored into the repository, through a list of categories (See upper part of Fig. 9). These categories are defined by the repository administrator, but they can be adapted to reflect the needs of the users. Once the user has selected one category, only the resources that are classified into that category are shown page by page. For each record, the resource title, description, type (according to a reference model), date in which the resource was into the repository, and qualification are shown. The user can click on the resource title to visualize it, and it is also possible to download the resource and to see more information about each one of them.

In order to access the core functionality of the repository, all users must register into the system. Registration by a simple process, in which the user fills a form with his/her name, e-mail, intended username and password, amongst other data. When a user fills the registration form and submits the data back to the repository, a record is created and the new user is authenticated by

first time and logged into the system. For next logins, the user can type his/her username and password into the text fields presented by the repository start page.

Once the identity of the user has been verified, he/she can search for resources, upload a new resource, modify resources and metadata, download resources, and/or change the own user profile (password and general information).

The **Upload Resource** menu option allows the user to upload a new resource into the repository. When this option is selected, a simple web form is displayed, in which the user must fill the new resource name and description, the type of resource (according to the reference models available on the repository), the initial classification into the categories established by the repository, the primary language of the resource and its URL. After filling this data, the user posts the form back to the application, and the resource is retrieved from the URL and stored into the repository, according to a specific disk storage strategy. Once the resource has been stored, the user must fill its metadata record, in order to allow that future searches can locate the new resource.

The metadata editing interface allows the user to add, modify and delete metadata elements according to the reference model selected for a specific content object. Fig. 10 shows the metadata editing interface for a SCORM 1.2 content object.

Bienvenido a SPAR [trunksdbz](#) [Salir](#)

**Repositorio de acceso público basado en SCORM**  
 Busque, Almacene y Comparta Objetos de Aprendizaje

[Inicio](#) [Idioma](#) [Subir recurso](#) [Mis recursos](#) [Buscar recursos](#) [Referenciar recurso](#) [Descargas](#)

Modificar Metadata

	Elemento	Descripción	Insertados	Maximo
<input type="button" value="Insertar"/>	Información Educativa	Esta categoría describe las características claves educacionales o pedagógicas de este recurso	0	1
<input type="button" value="Insertar"/>	Relación	Esta categoría define la relación entre este recurso y otros recursos	1	100
<input type="button" value="Insertar"/>	Anotación	Esta categoría provee comentarios sobre el uso educacional de este recurso, quien creo esta anotación y cuando, cuando multiples anotaciones son necesitadas multiples instancias de esta categoría	0	30

**Descripción**

Raiz de la especificación SCORM 1.2

**Fig. 10. Metadata editing interface for a SCORM 1.2 Content Object**

The **My Resources** menu option can be used to obtain a list of all resources owned by the user. The result list has the same format that the general resource list and it also has edit resource and metadata options.

The **Search Resources** option is an exact-phrase query utility that can be used by the repository users to locate content resources. This option will query into the metadata records of the resources and it will return a list of the matching resources.

The **Reference Resource** menu option allows the users to insert a reference to an external resource and it is intended to provide a simple mechanism of referencing resources such as web



pages or resources stored into other repositories. As opposite to uploading a resource, by using this option the resource is not stored into the repository, instead only the URL of the resource is stored.

Finally, the **Download** menu option provides access to materials related to SPAR project, such as installation and setup guide, database scripts and application source. The former can be downloaded without any restriction, while the latter requires the users to read and to accept the Microsoft Shared Source License.

## 7 Project state and future work

At the time of this writing, we expect to officially launch the first stable and operational version of the repository and the web system to access it. This event will involve a test process involving a data load by real users of the system, in order to retrieve detailed information about application performance, reliability and its usability, among others.

The gathered information will be used to maintain and to improve the system and release it incrementally later in several versions of SPAR. In the near future, two versions are scheduled: the first version, SPAR 1.1, corresponds to a non-substantial improvement and correction of bugs detected in March-May 2006. The second version, SPAR 2.0, will include the development of the Decision Support System based on Data Warehouse and OLAP with distributed databases. The improvements and required new functionalities identified at the present day are:

- Interface and usability improvements (navigation, color and font schemes).
- To create a new specification, so SPAR can provide support multiple standards and specifications.
- Multilanguage support for the application (English full support).
- Advanced searches, by using logical operators and subset handling.
- Uploading metadata from an XML document (currently under development)
- Metadata validation to detect if the required metadata for a resource is complete or not.
- Redefine the web services interface in order to allow the users uploading a resource from third-party systems, such as Learning Management Systems or other repository implementations.
- To develop a query interface available for Pocket PC mobile devices.

There are some features not defined yet, but they will be studied at the SPAR project and other projects derived from it. These features are:

- It is necessary to analyze the optimization mechanisms to retrieve learning objects from different network repositories; in this sense a metadata and learning objects cache implementation has been proposed, based on asynchronous replication by demand. With this mechanism, the repositories with a bigger demand will use local storage to save learning objects and metadata (from other repositories).
- By the other hand, the learning object's identification at global level in the repositories network has not been defined yet; initially the project proposes a Global Unique Identifier composed by a unique identifier inside each repository added to a namespace identifier of the repository.
- Duplicity control of learning objects at the repositories network has been proposed by the use of Hashing algorithms like MD5, which allows obtain a learning objects "finger print" able to be requested from any repository.

Finally, the Project Team expects to make an implementation with XML native Support and compare its performance with the model described in this paper.

## 8 Conclusions

The proposed model based on a Distributed Repositories network complies with the IMS DRI, SCORM and LOM standards, and it supports other metadata schemas like Dublin Core. The model, and specially the EduLatin network, has the ability to integrate itself with other worldwide initiatives like GLOBE or CORDRA through the confederate repositories schema. The integration process requires the development of some registry services at the main CORDRA servers, but using the developed web services architecture and the standard communication support (SOAP/XML), it should be an easier work.

Implementation of each repository of the model can be realized under any language, platform or database engine. The final implementation developed at the SPAR project, the source code and the data models will be released to the community (for study and use) for free. The web services model allows to others (enterprises, universities and communities) to develop applications to import/export metadata and resources from the repositories, this means that <http://spar.unicauca.edu.co> will not be the unique repository interface.

The relational model above has a more complex design and use compares to a DBMS with native XML support, instead it can be deployed in any DBMS and is very useful for generating LOM/SCORM editors in standalone client applications with the selected standard or specification (or many at same time).

Global Unique Identifier (GUID) in TBLOBJECT table and specification's stored data are synchronized across stored data into TBLKEYFIELD and TBLFIELDTREE tables, getting information's integrity and making easier next move tasks (Gather/Publish/Import/Export) of Learning Objects between repositories. This mechanism is used for URL information synchronization.

In the near future, the implementation of the decision support system will allow to refine and to optimize the logic, the web services interface and the quality of the resources stored into the SPAR repository. The information from this system could be used to take some decisions concerning on-line education organizations.

## 9 Acknowledgements

The author of this paper wishes to express thanks to the sponsors of the SPAR Project, University of Cauca (Colombia) and Microsoft Research at the "2004 Microsoft Research eScience and SciData RFP Awards". Also, the author wishes to thank to University of Cauca Systems Engineering Students (Jesús Muñoz, Jorge Giraldo, Diego Bayona and Alexander Calvache), who have been working and supporting the development of the SPAR project.

## References

- [1] Advanced Distributed Learning. Sharable Content Object Reference Model (SCORM™) Version 1.4. <http://adlnet.org/>.
- [2] Technology Standards Committee of the IEEE Computer Society. Learning Technology Systems Architecture and Learning Object Metadata. <http://ltsc.ieee.org/>.
- [3] Looms, T., Christensen, C. Advanced Distributed Learning. Emerging and Enabling Technologies for the Design of Learning Object Repositories Report. Version 1.0. November 26, 2002.

- [4] IMS Digital Repositories Interoperability Specification Version 1.0. <http://www.imsglobal.org/digitalrepositories>.
- [5] Rehak, D., Dodds P. and Lannom L. A Model and Infrastructure for Federated Learning Content Repositories. Proceedings, Interoperability of Web-Based Educational Systems Workshop at the 14th International World Wide Web Conference (WWW2005) May 10, 2005, Chiba, Japan. <http://www.lsal.cmu.edu/lsal/expertise/projects/cordra>.
- [6] MERLOT. Multimedia Educational Resource for Learning and Online Teaching (Visited 2/12/04). URL <http://www.merlot.org>.
- [7] EdNA. Educational Network Australia (Visited 2/3/05). URL <http://www.edna.edu.au/edna/page1.html>.
- [8] ARIADNE. Ariadne foundation for the European knowledge Pool (Visited 2/3/05). URL <http://www.ariadne-eu.org/index.html>.
- [9] EduSource Canada. Canadian Network of Learning Repositories (Visited 2/3/05). URL [http://www.edusource.ca/english/home\\_eng.html](http://www.edusource.ca/english/home_eng.html).
- [10] NIME. National Institute of Multimedia Education (Visited 2/3/05). URL <http://www.nime.ac.jp/index-e.html>.
- [11] Learning Systems Architecture Lab. Carnegie Mellon University (Visited 2/3/05). URL <http://www.lsal.cmu.edu>.
- [12] Dublin Core Metadata Initiative (Visited 2/3/05). URL <http://dublincore.org/>.
- [13] Vcard Specification 1.0 (Visited 3/5/05). URL <http://www.imc.org/pdi/>.
- [14] Dr. Thelma Looms, Clark Christensen. Advanced Distributed Learning. Emerging and Enabling Technologies for the Design of Learning Object Repositories Report. Version 1.0. November 26, 2002.
- [15] Qu, C., and Nejd, W. (2004). Integrating XQuery-enabled SCORM XML Metadata Repositories into an RDF-based E-Learning P2P Network. Educational Technology & Society, 7 (2), 51-60.
- [16] Sampson, D., Papaioannou, V. and Karadimitriou, P. EM<sup>2</sup>: An Environment for Editing and Management of Educational Metadata. Educational Technology & Society 5 (4) 2002. ISSN 1436-4522. [http://ifets.ieee.org/periodical/vol\\_4\\_2002/sampson\\_papaioannou.html](http://ifets.ieee.org/periodical/vol_4_2002/sampson_papaioannou.html).
- [17] Metadata Tool. Learning Systems Architecture Lab. Carnegie Mellon University (Visited 2/2/05). URL

[http://www.lsal.cmu.edu/lsal/expertise/projects/compio/compio2001finalreport/scorm\\_tools/metadata\\_tool.html](http://www.lsal.cmu.edu/lsal/expertise/projects/compio/compio2001finalreport/scorm_tools/metadata_tool.html)

- [18] Zhen Yin, Zhengfang Xu and Abdulmoteleb El Saddik. Study of metadata for advanced multimedia learning objects. CCGEI 2003 Proceedings.
- [19] Learning Technology. Volume 5 Issue 1. Publication of IEEE Computer Society Learning Technology Task Force (LTTF), January 2003. ISSN 1438-0625 (Visited 2/3/05). URL [http://lttf.ieee.org/learn\\_tech/issues/january2003/#9](http://lttf.ieee.org/learn_tech/issues/january2003/#9)
- [20] Anand Vivek Srivastava. Comparison and Benchmarking of Native XML Databases. November, 2004. CS497 Report. Report Id: 420. Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur.
- [21] Google Web APIs Reference (Visited 1/8/05). URL <http://www.google.com/apis/reference.html>
- [22] SCORM Public-Access Repository. URL <http://spar.unicauca.edu.co>
- [23] Kimball, Ralph et Al. The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses. Wiley and Sons, Inc. 1998.