ECP 2007 EDU 417008

ASPECT

Design of Data Model and Architecture for a Registry of Learning Object Repositories and Application Profiles

Deliverable number	*D2.2*
Dissemination level	*Public*
Delivery date	*9 March 2009*
Status	*Final*
Author(s)	*David Massart (EUN),* *Neil Smith (VMG),* *Rob Tice (VMG)*

e*Content*plus

This project is funded under the *e*Content*plus* programme[1],
a multiannual Community programme to make digital content in Europe more accessible, usable and exploitable.

[1] OJ L 79, 24.3.2005, p. 1.

0 General Introduction

A registry is a catalog service that provides up-to-date information about a type of resources. Different types of registries can facilitate interoperability between learning object repositories.

Registries of learning object repositories (i.e., registries that hold the parameters necessary to apply protocols such as OAI-PMH, SQI, SPI, SRU/SRW) can:

- Either be consulted by humans (such as developers who want to provide access to the repository, or end users who want to decide whether the repository is relevant to them)
- Or be accessed by software processes (that can, for example, use the data to forward a federated search to a repository or to initiate its harvesting).

Similarly, registries of application profiles (i.e., registries that describe how learning objects in a repository are described) can be used:

- Either by people (such as developers who, for example, may need to map between metadata elements in different repositories)
- Or by software processes (that, for example, may want to restrict federated searches to those repositories that include the metadata elements to which the queries refer).

This deliverable proposes data models for these two types of registries. Section 1 introduces a data model for a repository of learning object repositories that was produced in collaboration between partners of the ASPECT project, members of the IMS LODE Workgroup and members of the GLOBE alliance, whereas Section 2 introduces a data model for a registry of application profiles. In addition, Section 3 describes the data model of the registry required by the digital rights management model proposed in the ASPECT Deliverable D3.1. Section 4 proposes a set of standard Application Programming Interfaces (APIs) for managing these registries. Finally, Section 5 describes how these repositories can be combined with other ASPECT services in a service-oriented architecture.

1 Learning Object Repository Registry (LORRy)

1.1 Introduction

A learning object is a digital object that can be used for teaching or learning. Learning objects range from simple learning assets, for example an image, a text or a video that can be used to illustrate a lesson, to very complex learning resources including, for example a complex multi-media course that follows a precise pedagogical approach and may include assessment. Metadata is used to described learning objects in a machine-readable way. It makes learning objects easier to find and to access.

Learning object repositories are specialized software systems used to manage collections of learning objects and/or their metadata. They often support one or more protocols that allow them to expose their metadata in order to make their content more globally accessible. Examples of such protocols include the Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH) [1], the Simple Query Interface (SQI) [2], Search/Retrieval via URL[1] (SRU), the Simple Publishing Interface (SPI) [3]

Thanks to these protocols, it is possible to federate learning resource repositories in order to offer seamless access to their content. The Learning Resource Exchange (LRE) [4] is an example of such a federation.

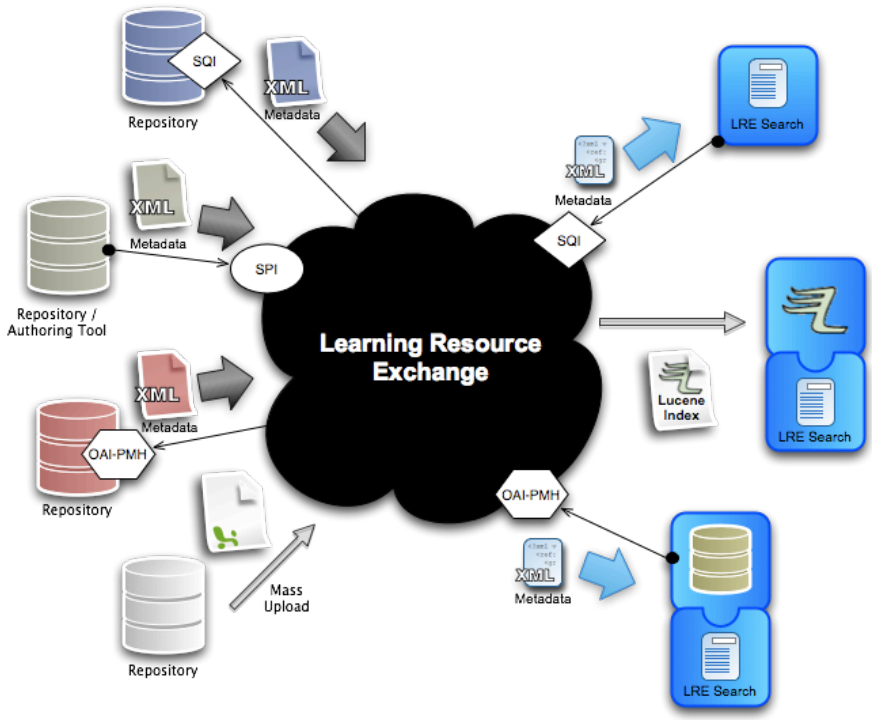

Figure 1.1: The LRE as an infrastructure for exchanging learning content in a federated way.

[1] http://www.loc.gov/standards/sru/

As described in Figure 1.1, the LRE tries to support as many ways as possible to collect and expose learning object metadata. Virtual Learning Environments and educational portals that want to obtain access to the LRE collections of learning resources can:

- Search the LRE using SQI;
- Regularly import a complete index (e.g., Apache Lucene) of the LRE and search it locally; or
- Regularly harvest LRE metadata using OAI-PMH

The LRE also supports the following ways to collect metadata:

- It can search SQI-compliant repositories;
- It can harvest metadata using OAI-PMH;
- It supports SPI that can be used by authorized repositories to push metadata into the LRE;
- Finally, it offers to partners who have learning objects but no metadata repository, the possibility to mass upload the descriptions of their objects using an Excel spreadsheet.

At present, there is no way for a federation such as the LRE to automatically discover new repositories. Currently, the locations of the LRE repositories have to be entered manually and the knowledge of the protocols supported by these repositories for exposing resources is implicit.

This document proposes a data model for describing learning object repositories in an active registry. As suggested in Figure 1.2, such a registry can be queried by the LRE or other federations for finding repositories hosting relevant learning objects and for facilitating connections to these repositories in an automatic way.

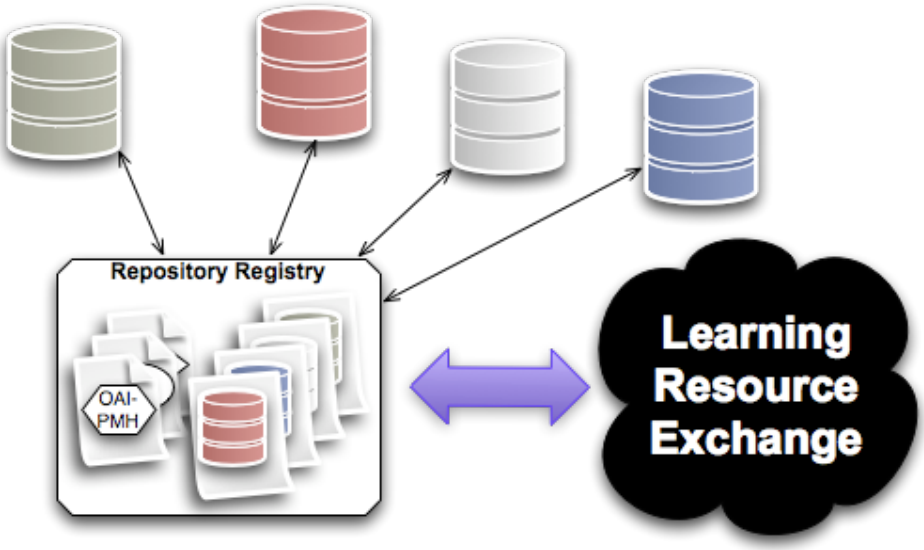

Figure 1.2: An active registry for discovering and connecting to learning object repositories in an automatic way.

The proposed data model is the result of a collaborative work between the ASPECT project, the IMS LODE Working Group[1] and the Global Learning Objects Brokered Exchange

[1] See http://www.imsglobal.org/lode.html.

(GLOBE) alliance[1]. It is based on current work on the ISO 2146 standard: "Registry Services for Libraries and related organisations" [4] that is developed by ISO TC46 SC4 WG7 and proposes a framework for building registry services for libraries and related organisations.

1.2 Information Model Overview

The class diagram in Figure 1.3 provides an overview of the Repository Registry information model. Actually, it does not model repositories themselves but collections and how to get access to them.

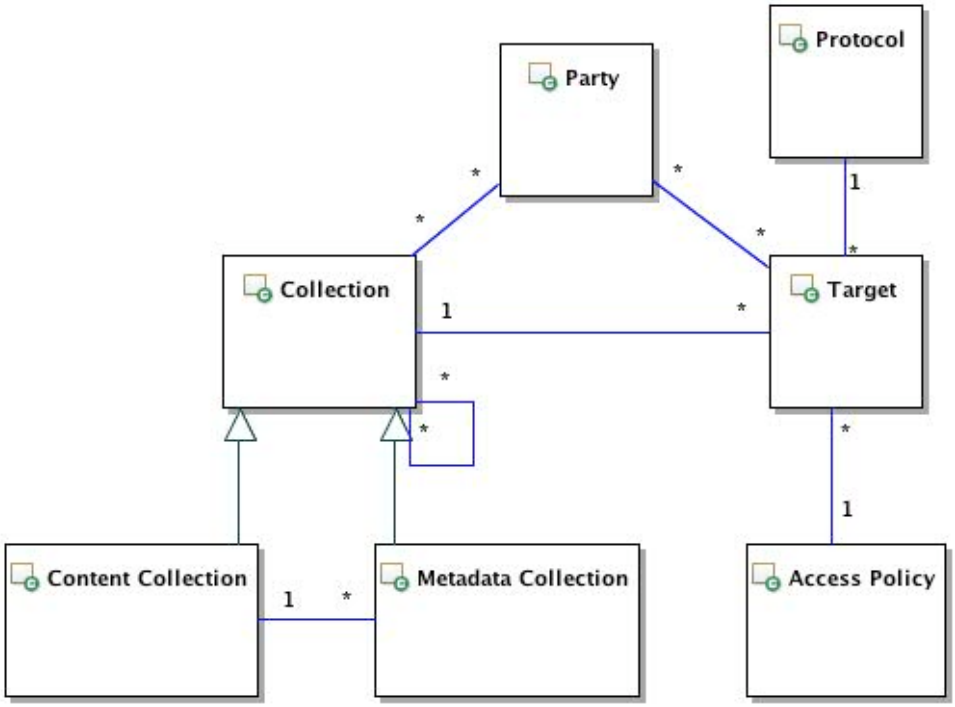

Figure 1.3: Collections, protocols, and targets. (Class diagram derived from USQ UML for ISO-2146 and LODE Registry Model 2009-02-10 – CC-BY-SA)

Collections can be either collections of content or collections of Metadata describing this content. Collections can be related to other collections in different ways: A collection can be part of another collection (is-part-of/has-part) or can be derived from another collection (is-version-of/has-version).

Collections can be accessed using one or more targets. Each target implements a given protocol and follows a certain access policy.

Finally, Parties (e.g., persons or organisations) may relate to collections and targets in different ways. A party can be the owner or the collector of a collection. A party can also play the role of technical contact, policy contact, owner or provider of a target.

1.3 Data Types

There are five data types in the LOM information model and they are:

[1] See http://globe-info.org/.

CharacterString: text can be entered in the attribute directly.

LangString: the text must identify its language and there can be one or more character strings in the element.

Identifier: this is a generic type. Profiles of the specification must provide an identifier type such as, for example, the type of identifier defined in LOM and consisting of a catalog and an entry [7].

VocabularyTerm: the element contains source and value where source is a reference to publicly sourced and maintained value set and value is a value from that set.

1.4 Collection

The collection description presented here is based on the "Dublin Core Collections Application Profile" [6] that has been completed with element from ISO 2146 [4] and IEEE LOM [7] in order to meet requirements that are learning-specific as expressed in [5].

Collections are described using different categories of attributes. Some attributes describe the collection as a whole (e.g., the identifier of the collection, its description, or its average annual increase) and others describe the properties of its components[1] (e.g., the language of the items in the collections or the rights attached to these items). In order to describe the latter, we introduce a special data type that we refer to as "Property". A property type has three components: a source, a value, and a strength where:

- Source is a reference to a publicly sourced and maintained value set
- Value is a value from that set
- Strength is a quantifier that can take one of the following three values: "some", "most", and "all" to indicate that the property applies to some, most, or all the item of the collection. The default strength is "some" meaning that, if no strength is mentioned, the property applies to at least one of the items of the collection.

1.4.1 Content Collection

The following attributes describe content collections as a whole:

- Identifier: An identifier of the content collection. (Identifier)
- Description: A description of the collection. (LangString)
- Collection Rights: A description of the rights that are applied to the collection as a whole. (LangString)
- Keywords: A list of keywords for describing the collection as a whole. (LangString)
- Average Annual Increase: The average annual increase in the number of learning obects in the collection. Note that this can be a negative number. (Integer)
- Accrual Periodicity: An attribute relying on a controlled vocabulary from DCCAP [6] to describe how often new learning objects are added to (or removed from) the collection. (VocabularyTerm)

The following attributes describe properties of the learning content (i.e., learning objects) in the collection:

[1] I.e., collection items.

- Quality Procedure: The quality procedure(s) applied to the items in the collection. (Property)
- Subject: The subject(s) covered by the learning objects in the collection (Property)
- Language: The language(s) of the learning objects in the collection (Property)
- Type: The learning resource type(s) -- as in LOM 5.2 -- of the learning objects in the collection. For example: “assessment”. (Property)
- Representation: The representation(s) in which the learning objects in the collection are available. For example: “QTIv2.1”. (Property)
- Format: The technical format(s) of the learning objects in the collection. For example: “application/swf”. (Property)
- Intended User Role: The intended role(s) of the user of the learning objects in the collection. (Property)
- Context: The context(s) of use of the learning objects in the collection. (Property)
- Typical Age Range: The age range(s) of the typical users of the learning objects in the collection. (Property)
- Item Rights: The rights that apply to the learning objects in the collection. (Property)
- Cost: Do the learning objects in the collection have a cost? (Property)

1.4.2 Metadata Collection

The following attributes are used to describe a metadata collection as a whole:

- Identifier: An identifier of the metadata collection. (Identifier)
- Description: A description of the metadata collection. (LangString)
- Collection Rights: A description of the rights that applied to the metadata collection as a whole. (LangString)
- Average Annual Increase: The average annual increase of the number of metadata instances in the collection. Note that this can be a negative number. (Integer)
- Accrual Periodicity: An attribute relying on a controlled vocabulary from DCCAP [6] to describe how often metadata instances are added to (or removed from) the collection. (VocabularyTerm)

The following attributes describe properties of the metadata instances that compose the collection:

- Quality Procedure: The quality procedure(s) applied to the metadata in the collection. (Property)
- Language: The language(s) of the metadata in the collection (Property)
- Format: The format(s) of the metadata in the collection. For example: “IEEE LOM XML binding”. Ideally, the metadata format should be identified by the namespace of its binding as referenced in the Application Profile Registry (cf. Section 2). (Property)
- Item Rights: The rights that apply to the metadata in the collection. (Property)

1.4.3 Party

Parties (i.e., persons or organisations) are described with two attributes:

- Identifier: A unique identifier of the party. (Identifier)
- A VCARD description of the entity

1.4.4 Protocol

Protocols are described with the following attributes:

- Identifier: A unique identifier of the protocol. (Identifier)
- Name: The name of the protocol. For example, “Simple Query Interface”. (CharacterString)
- Version: The version of the protocol. For example, “1.0”. (CharacterString)
- Protocol Description Binding Name Space: The namespace of an XSD used to describe a particular implementation of the protocol. (URI)
- Protocol Description Binding Location: The location of this XSD. (URL)

1.4.5 Target

Targets, i.e., protocol end-points, are described with the following attributes:

- Target Identifier: A unique identifier of the target. (Identifier)
- Protocol Identifier: The Identifier of the protocol supported by the target. (Identifier)
- Location: The location of the target. (URL)
- Protocol Implementation Description: A description of the properties of specific implementation of the protocol supported by the target. (This should be a valid instance of the XSD referred to in the “Protocol Description Binding Location” attribute of the corresponding protocol.

Protocols and targets

Protocols, such as the Simple Query Interface, can be implemented in different ways. For example, an SQI target can be used synchronously or asynchronously, can support different query languages and result formats, can accept anonymous session or not. The XSD referred to in the “Protocol Description Binding Location” for SQI should permit the expression of all these characteristics of the protocol and it is an XML instance of this XSD that should be used in “Protocol Implementation Description”.

1.4.6 Access Policy

Access policies are described with two attributes:

- Identifier: A unique identifier of the policy. (Identifier)
- A description of the access policy. (LangString)

1.5 Profiling

This specification is generic and needs to be profiled before it can be used. Profiling the specification mostly consists of:

- Selecting an Identifier Type
- Selecting mandatory attributes (a priori, all of them are optional).
- Selecting controlled vocabularies for the different property attributes.
- Agreeing on XSD bindings for the different protocol descriptions

1.6 Binding

An XML binding for the registry information model is currently in preparation as a collaborative work between ASPECT, IMS LODE and partners of the GLOBE alliance.

2 Application Profile Registry (APR)

2.1 Overview

The application registry for the ASPECT project is designed to build on the work undertaken in the creation and provision of an application profile registry for learning on behalf of the CEN/ISSS Workshop on Learning Technologies (WS/LT) (http://registry.k-int.com/apr). As the registry and model evolve, it is also expected that it will reference work undertaken in the development of the JISC IE Metadata Schema Registry (http://www.ukoln.ac.uk/projects/iemsr/) which is due to be completed in March 2009.

The APR is the registry component which provides the detailed information about the metadata profiles used by various components within the information landscape. This allows both human users and machines to gather information to help them locate the resources they require from a repository independently of how the resources have been organised and classified within a particular repository. It is ultimately designed to support the ASPECT registry infrastructure in order to facilitate seamless querying and cross walking between repositories of learning objects, although this goal would still involve significant mapping work between vocabularies.

In addition, it can also be used as a tool to support the evolution of new application profiles by facilitating the easy querying and browsing of existing profiles and base standards, thus promoting re-use and interoperability.

Application profile descriptions which reside within this registry are referenced from within the Learning Object Repository registry (see section 1.4.2) using persistent URL's. This application profile registry will itself reference vocabularies using persistent URL's as described in the Aspect Vocabulary Bank for Education (see deliverable 2.3.1).

2.2 Data model Overview

The APR data model (which includes descriptions of relevant base standards as well as profiles based on these standards) is not intended to be a complete replication of the detailed data model of either the base standard or application profile. Its purpose can be defined as being designed:

- to provide a reference to the original standards with additional descriptive metadata elements to aid location and discrimination
- to provide a reference to application profiles of base standards with additional descriptive metadata to aid location and discrimination
- provide a simple description of elements and types within a base standard so that they can be easily compared with an application profile.
- provide a simple description of elements and types within an application profile with reference to the elements in the base standard which they refine
- to indicate how elements of the original base standard have been changed and refined within each individual application profile

As there are many profiles and data models which reuse elements and types at different points in their hierarchy, a simple approach is proposed which represents the model as a basic

network of elements and attempts to simplify the model for querying, comparison and presentation. It is envisaged that the model will evolve and be refined during the project. In the case where controlled vocabularies are used in descriptive metadata it is expected that these vocabularies will be hosted in the VBE.

For purposes of easy comparison between profiles the reference to any controlled vocabulary used or changed will resolve to the VBE.

As part of the project, we will also investigate the possibility of auto generating the relevant data model instances from supplied xml schema bindings.

2.3 Data model for descriptive data

This data model is designed to describe the base standard or application profile

2.3.1 Title

The multilingual title of the base standard or application profile (mandatory, LangString)

2.3.2 Description

The multilingual description of the base standard or application profile (mandatory, LangString)

2.3.3 Location of original

A URL referencing the information site describing the data model of the application profile or base standard (mandatory)

2.3.4 Target domain

An entry from a controlled vocabulary representing a community of use (optional, VocabularyTerm)

2.3.5 Language

The ISO language of the standard or profile (optional, repeatable, CharacterString)

2.3.6 Places

Spatial data describing the countries/locations in which it is used (optional, repeatable, CharacterString)

2.3.7 Subjects

Controlled Subject data describing any specific subject area of relevance (optional, repeatable, VocabularyTerm)

2.3.8 Publication date

Date on which the standard or application profile was published (optional, date)

2.3.9 Status

Status of the application profile from a controlled vocabulary (e.g. draft) (mandatory, VocabularyTerm)

2.3.10 Related to

Either an internal or external link to a related standard or profile (not used to indicate which base standard a profile is derived from) (optional)

2.3.11 Derived from

An internal link to a base standard in the registry which a particular application profile is derived from (mandatory)

2.3.12 Profile type

Entry from a controlled vocabulary summarising whether the profile is purely a restriction of the base standard or if there are any extensions (mandatory, VocabularyTerm)

2.3.13 Notes

General notes on the base standard or profile (optional, repeatable)

2.3.14 Bindings

Information about the location and type of any bindings available (mandatory, repeatable)

- Location – the location of a binding for the base type or profile (can be a URL or internal link)
- The type of the binding (from a controlled list, VocabularyTerm)
- Note about the binding

2.4 Data model for element comparisons

This data model is designed to represent the structure of a base standard or application profile for comparison of profiles with each other and the base standard from which they are derived

2.4.1 Name Reference

An object which identifies an object by name and namespace in an application profile or base type

- Name – the name of a referenced entity
- Namespace URI – the namespace of a referenced entity

2.4.2 Root Element

A name reference to a single element declaration which represents the root entry point for the base model or application profile (e.g. LOM)

2.4.3 Element Declaration

An object which describes an element in a data model, comprising of a name reference for its name and a name reference for its type

- Name reference describing the name of the element
- Name reference describing the data type of the element

2.4.4 Element reference

An object which describes the multiplicity of an element declaration within data model type definition

- Element declaration
- Min occurs
- Max occurs
- Smallest permitted maximum

2.4.5 Attribute reference

An object which describes an attribute of an element within an application profile or base standard.

- Name reference describing the attribute name
- Name reference describing the data type of the element
- Source vocabulary (used in the case of a vocabulary which should be used). This element is linked through the the VBE in order to resolve controlled values and indicate differences in choice of vocabularies within application profiles.
- Required (Boolean indicating if the attribute is mandatory)

2.4.6 Type Definition

A simplified representation of a data model data type which can be used for purposes of comparison between standards and application profiles.

- Type (simple or complex)
- List of element references (used in the case of complex type - ordered)
- Source vocabulary (used in the case of a simple type which is from a controlled source). This element is linked through the VBE in order to resolve controlled values and indicate differences in choice of vocabularies within application profiles.
- List of attribute references (ordered)
- Max length
- Min length

3 Learning Object Broker Registry Service (LOBR)

In a federation such as the LRE, there are three main categories of distribution models[1]: open educational content, license-based access, and credit-based access. In this document, a provider is a LRE member who provides learning objects. A requestor is a LRE member who wants to access learning objects. Rights apply to these learning objects. They can be seen as the result of a transaction between a requestor and a provider. "Access" could mean play, print, transfer, or any action that applies to a learning object.

A. Open Educational Content
Open educational content is defined as learning content that resides in the public domain or is released under an intellectual property license that authorizes its free use. Although, the use of open educational content is free, some usage restrictions may apply. Open educational content is identified by the use of open license types such as Creative Commons[2]. The distribution of open content may thus require the acknowledgment of an applicable license prior to actual learning object access.

B. License-based Access
The second popular distribution model is license-based access. In this model, buying a learning object is similar to buying software. An individual user, a group of users, or all end users of a LRE member can access a learning object or a group of learning objects when the license is granted. A license agreement between a requestor and a provider must be obtained before the requestor sends an access request.

C. Credit-based Access
In this model, end-users "purchase" access to learning objects by spending LRE credits. Basically, users have a certain amount of LRE credits that they may choose to spend to gain access to or use of LRE content. This situation is similar to that where an end-user would go to a shop to buy some goods with real money. A requestor presents money to a provider that checks that the amount is correct before providing the requested goods.

3.1 Learning Object Broker

A key challenge the LRE has to face consists of specifying a common service for content protection that bridges different distribution models, different authorization services, and technical implementations. The proposed broker provides a unified way to mediate access control independently from actual distribution models.

As shown in Figure 3.1, the broker relies on a LRE Registry Service (or registry) to collect information about available providers. It is the responsibility of individual providers to ensure that the registry contains the appropriate data. This includes the list of supported distribution models together with service endpoints where their local authorization service can be reached.

[1] A more complete description of the ASPECT approach to content access controlled can be found in the ASPECT Deliverable 3.1.

[2] http://creativecommons.org/

In the LRE, a transaction starts when a requestor contacts the broker to request access to a provider's learning object. The request specifies the learning object identifier and the provider identifier, as well as a context that contains all the information needed by a provider to make an authorization decision. The broker checks the request validity, queries the registry to retrieve the endpoint corresponding to the requested provider's distribution model, and forwards the request to that endpoint. Finally, the broker receives the result, in the form of a handle that is then returned to the requestor. The handle contains all the necessary information for the requestor to access the learning object or to determine whether access to the learning object is granted or not.

The context is different in each distribution model. It is transparent for open educational content, meaning that these learning objects are freely accessible. In the case of the license-based access model, the learning object context might be a pair that contains the user's or the requestor's identifier and an access code. It is also possible that the broker processes a very complex context to grant access to a requestor based on a license agreement. When a requestor wants to obtain access to a learning object distributed via the credit-based access distribution model, the context can be a user identifier and the available number of credits. If needed, this simple concept can be extended to accommodate new distribution models.

A handle can also be either simple (e.g., a learning object URI) or complex (e.g., it can point to an encrypted learning object and include a cryptographic key that allows the learning object to be read). An optional expiration date provided by the authorization service is included in the learning object handle, to support handle caching. Here also, the handle could be made as complex as required by the distribution model, by including, for example, cryptographic keys.

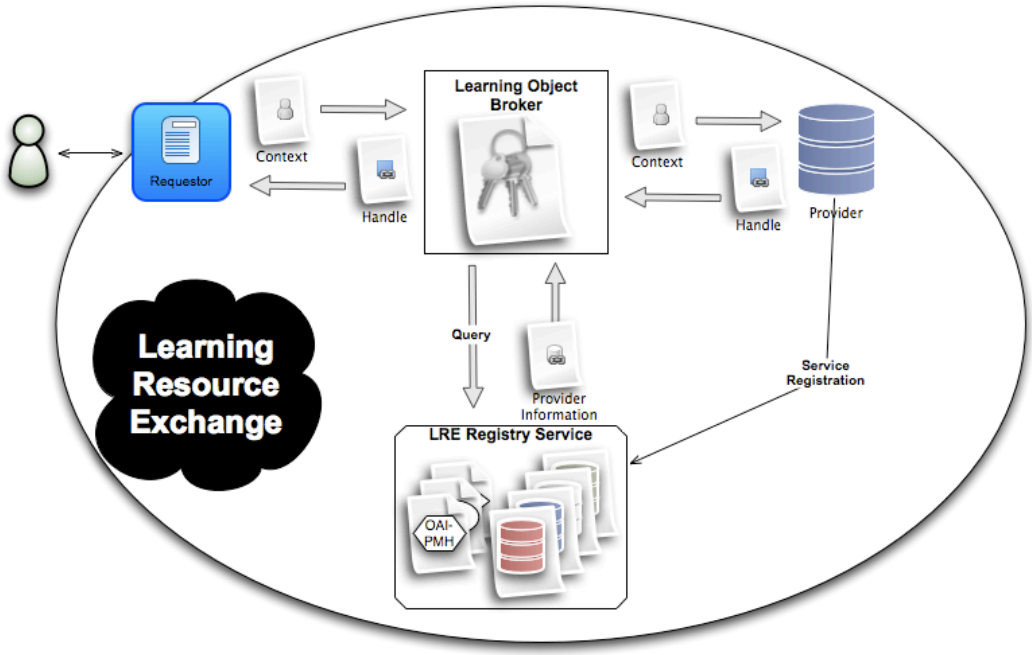

Figure 3.1. Access Control Mediation.

3.2 Information Model of the Learning Object Broker Registry Service

The Learning Object Broker Registry Service (or registry) maintains information about the content providers that are offering their content through the federation. Note that this information is time-dependent, in that the structure of the Federation varies across time, with providers joining and others leaving. The registry aims at maintaining the most accurate information at any point in time.

To mediate access to the resources made available throughout the federation, the registry maintains a list of providers, together with the distribution models they support and the actual location of their authorization service.

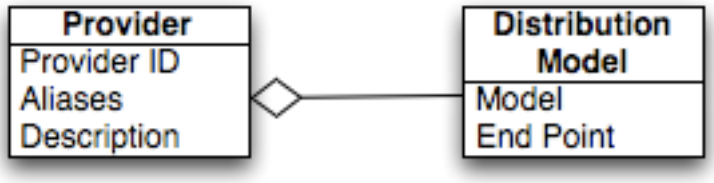

Figure 3.2. Broker Registry Information Model.

The registry information model is described on the class diagram in Figure 3.2. It defines the two classes:

1. Class “Provider” models content providers. Information stored at this level is:
   - Provider ID: To uniquely identify providers in the Federation.
   - Aliases: Aliases are alternate names by which a provider can be referred to.
   - Description: Textual descriptions of providers included for explanatory purposes.

2. Class “Distribution Model” stores information about instances of a distribution model for a particular provider. It stores:
   - Model: The identifier of the distribution model, which is selected from a vocabulary of existing models.
   - End Point: The exact location of the provider's authorization service where requests for accessing content offered via this model can be accessed. Technically speaking, this is the URL of a Web Service endpoint.

The Learning Object Broker Registry Service offers two types of services:

1. Registration and deregistration: These services are used by providers when joining and leaving the federation.
2. Query: This service is used by the broker to query a provider's information. Typically, this occurs when a content consumer (an end-user) requests access for some content that is covered by one of the supported distribution models. As described above, the broker receives the request and forwards it to the proper location on the provider's side, that is determined by querying the registry.

The interfaces necessary to support these services are discussed in Part 4 of this document.

4 Registry Interfaces

Using a registry consists of:

- Managing its content (i.e., registering items, keeping their descriptions up-to-date, and removing them),
- Querying its content, and, possibly,
- Synchronizing its content with other registries.

These functionalities are generic in the sense that they apply to any registry regardless the kind of descriptions that it manages[1]. Actually, these functionalities are similar to those supported by repositories. Therefore, it is proposed to use generic repository specifications to support them. This permits the use of such a repository registry to register repository registries themselves.

4.1 Managing Registry Content with SPI

The Simple Publishing Interface (SPI – [3]) is a specification for publishing content and/or metadata in a repository. It allows for publishing, updating, and deleting XML descriptions, which is exactly the functionalities required for managing registries' content.

4.2 Querying Registries with SQI and SRU

Simple Query Interface (SQI – [2]) and Search and Retrieval via URL (SRU[2]) are two search specifications. Provided that they are combined with the appropriate query language, nothing prevents one from using them for querying a registry.

4.3 Synchronizing Registries with OAI-PMH

The rationale for synchronizing registries might not seem obvious. Actually, it is a requirement of the GLOBE Alliance for Learning Object Repository Registries. The idea is that, in the GLOBE alliance, different groups of repositories are registered in different registries. For example, the repositories of the European Ministries of Education would be registered in the EUN registry while Australian repositories could be registered in a registry hosted by EdNA (the Australian GLOBE partner). Similarly, different federations would rely on different registries. For example, the Learning Resource Exchange would select participating repositories using the European Schoolnet registry. Synchronizing registries would allow, for example, the LRE federation to include Australian collections whose descriptions are primarily managed in the EdNa registry and copied into the EUN one.

[1] This is only true to a certain extent since, for example, the expression power of a query language used to query a registry need to be adapted to the informational content of this registry.

[2] http://www.loc.gov/standards/sru/

The Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH – [1]) is a protocol for mirroring XML documents and provides the functionalities required to synchronize registries of repository descriptions.

5 Architecture Overview[1]

This section indicates how the three registries described in this document relate to the other services developed by the ASPECT project.

These services include:

- A handle server (HS) that provides unique and persistent identifiers.
- A vocabulary bank (VB) that manages controlled vocabularies and cross-walks between these vocabularies.
- An automatic metadata translator (AMTl) that translates metadata in one language into another language.
- An automatic content format transcoder (ACFT) that turns content in a format, for example SCORM 2004, into content in another format, for example IMS Common Cartridge version 1.0.
- An automatic metadata transformer (AMTf) that transforms metadata in a format, for example the LRE LOM Metadata application profile, into another format e.g., an application profile of Dublin Core.
- Metadata repositories (MR)
- Learning object repositories (LOR)

The relationships between these services are described on Figure 5.1. Each service is represented by a hexagon. The arrows show the services used by each component of the architecture. For example, the arrows from the three registries (LOBR, APR, and LORRy) to the vocabulary bank (VB) shows that these 3 repositories use the service of the vocabulary bank to manage the controlled vocabularies used by their descriptions.

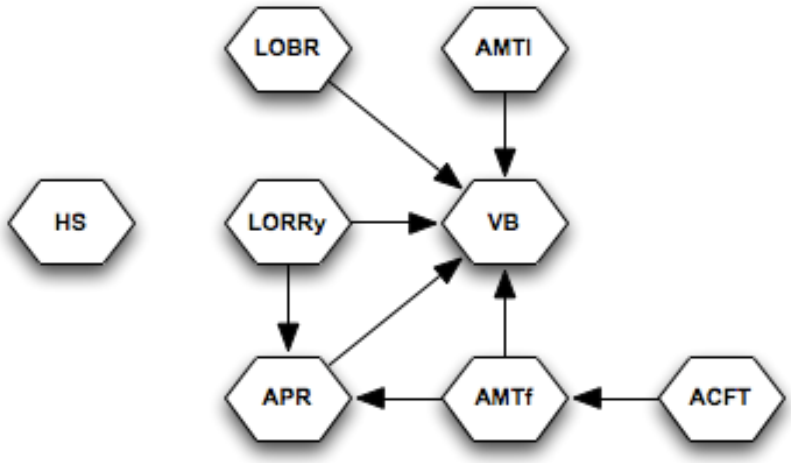

Figure 5.1. Dependencies between the main ASPECT services.

[1] This architecture is meant to evolve. It represents the view of the ASPECT WP2 at the time of writing, i.e., February 2009. For an up-to-date description of the ASPECT architecture, please consult the ASPECT deliverable D2.4 "Wiki with Material from Repository to Support Training and Dissemination" at http://ariadne.cs.kuleuven.be/aspect-wiki/index.php/D2.4.

When turning a content format into another, the automatic content format transcoder (ACFT) relies on the automatic metadata transformer (AMTf) to transforms the metadata from and to the metadata formats associated with the content formats involved. These metadata formats are identified in the application profile registry (APR). When it needs to turn a term belonging to a controlled vocabulary associated to a metadata application profile into a term belonging to a controlled vocabulary belonging to another metadata application profile, the automatic metadata transformer (AMTf) relies on the cross-walks of the vocabulary bank (VB). As mentioned in Section 1, the learning object repository registry (LORRy) uses the application profile registry (APR) to manage the descriptions of the application profile supported by the metadata collections that it describes. Finally, the automatic metadata translator (AMTl) relies on the vocabulary bank (VB) to translate the terms of the controlled vocabularies.

6 References

[1] C. Lagoze, H. Van de Sompel, M. Nelson, and S. Warner, *The open archives initiative protocol for metadata harvesting version 2.0*, Document Version 2004/10/12T15:31:00Z. Also available as http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm, 2002.

[2] B. Simon, D. Massart, F. Van Assche, S. Ternier, and E. Duval, *A simple query interface specification for learning repositories*, CEN Workshop Agreement (CWA 15454). Also available from ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/WS-LT/CWA15454-00-2005-Nov.pdf, November 2005.

[3] S. Ternier, D. Massart, F. Van Assche, N. Smith, B. Simon, and E. Duval, *A simple publishing interface for learning object repositories*, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008 (Vienna, Austria), AACE, June 2008, pp. 1840–1845.

[4] ISO TC 46/SC 4/WG 7, *Information and documentation – Registry services for libraries and related organizations*. Draft version, May 2008.

[5] F. Van Assche and D. Massart, *Interoperability of federations of learning object repositories*, Proceedings of the Second International Workshop on Search and Exchange of e-le@rning Materials (SE@M'08), vol. 385, CEUR Workshop Proceedings, September 2008.

[6] Dublin Core Collection Description Task Group, *Dublin core collections application profile*, also available as http://dublincore.org/groups/collections/collection-application-profile/, March 2007.

[7] IEEE Standards Department, *IEEE 1484.12.1-2002, Learning Object Metadata Standard*, also available as http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, July 2002.