eduSourceCanada

RÉSEAU CANADIEN DE BANQUES
D'OBJETS D'APPRENTISSAGE

CANADIAN NETWORK
OF LEARNING OBJECT REPOSITORIES

eduSource Suite of Tools
Use Cases Specifications

Version <0.6>

Contributors	
LICEF/COGIGRAPH/TELUQ	Gilbert Paquette, Anis Masmoudi, Gérard Lévy
ATHABASCA	Terry Anderson, Chris Hubrick
NBDEN/NRC	Stephen Downes, Gilbert Babin
LEARNING COMMONS/UC	Mike Matson
SFU	Marek Hatala, Griff Richards

7-Jun-03

eduSourceCanada

RÉSEAU CANADIEN DE BANQUES
D'OBJETS D'APPRENTISSAGE

CANADIAN NETWORK
OF LEARNING OBJECT REPOSITORIES

Revision History

Date	Version	Description	Author
20-Mar-03	<0.1>	First draft	Gilbert Paquette
25-Mar-03	<0.2>	Draft with new use cases	Gilbert Paquette
28-Mar-03	<0.2.2>	Draft revised + complete first use case diagram group	Gilbert Paquette
07-Avr-03	<0.2.4>	Draft revised + use cases for each service	Anis Masmoudi
17-Avr-03	<0.3>	Use cases revised + Draft sequence diagrams Models	Anis Masmoudi
10-May-03	<0.3.5>	Draft use cases diagram group 2, 4 and 5 added	Gilbert Paquette
28-May-03	<0.4>	Final version of Business and functionalities use cases Models	Gilbert Paquette and Anis Masmoudi
2-Jun-03	<0.5>	Revised version and complete set of use cases (Phase 1)	Gilbert Paquette
7-Jun-03	<0.6>	Validated version of sections 1,2,3,4	Gilbert Paquette and Anis Masmoudi

Table of Contents

Table of Figures

Use Cases Specifications

1. Introduction

1.1 Purpose

This document provides a use case view of a suite of open source tools and services for learning, training and knowledge management labeled **“Repository-in-the-box**”, providing a set of freely accessible components to support the implementation, management and use of learning object repositories.

The present document is aimed as a bridge between users, designers, and the software development team that will build the intended suite of tools and components. It will also provide a coordination support to the software development plan managers and team leaders. It will be built in parallel with the development and evaluation of the system, its different versions providing a history of the system’s development.

1.2 Scope

This Use Cases Specification document provides a set of conceptual specifications for the complete infrastructure of the Repository-in-the box. These specifications are expressed in the UML graphical language and produced using the Rational Unified Process (RUP). They are based on a previously defined Vision document and will help orient the Software Architecture and the Software Development Plan of the eduSource project.

The complete set of eduSource architecture documents include.

The eduSource Suite of Tools - Vision Document

The eduSource Suite of Tools - Use Cases Specifications

The eduSource Suite of Tools - Software Architecture

The eduSource Suite of Tools - Glossary

1.3 Definitions, Acronyms, and Abbreviations

DOI – Digital Object Identifier
DRI – Digital Repositories Interoperability
DRM – Digital Rights Management
EML - Educational Modeling Language
GUID – Globally Unique Identifier
HTTP – HyperText Transfer Protocol
LCMS – Learning Content Management System
LMS – Learning Management System
LD – Learning Design
LO – Learning Object
LOM – Learning Object Model
LOR – Learning Objects Repository
OAI – Open Archive Initiative
ODRL - Open Digital Rights Language
PMH - Protocol for Metadata Harvesting
SCORM – Shareable Courseware Object Reference Model
SOAP – Simple Object Access Protocol
UML – Unified Modeling Language
XML – eXtensive Markup Language
Z39.50 – (Protocol for library information retrieval)

1.4 References

- ADL SCORM, see: ADL http://www.adlnet.org
- Cancore Metadata Guidelines, version 1.1, Sue Ficsher, Norm Friesen, Anthony Roberts, Athabaska University, 2002.
- CLEO, content aggregation: http://www.lsal.cmu.edu/lsal/expertise/projects/cleo/report20010701/working/aggregation.html
- EduSource Communication Language (ECL) – Working document 0.2, Jean-François Arcand
- EduSource Vision, Version 3, December 12, 2002, contributors: Stephen Downes, Toni Roberts, Rory McGreal, Norm Friesen, John King, Terry Anderson, Michael Magee, Mike Mattson, Gilbert Paquette, Griff Richards
- EduSource: The Distributed Network – Functional Specifications for eduSource- Version 2, Stephen Downes, December 8, 2002].
- IEEE LTSC LOM (Learning Object Metadata), http://ltsc.ieee.org
- IETF http://www.ietf.org, relevant specifications: URI, ftp, news, smtp, http
- IMS Content Packaging Specification, see: IMS http://www.imsglobal.org
- IMS Digital Repository Interoperability Specification, see: IMS http://www.imsglobal.org
- IMS Learning Design Specification, see: IMS http://www.imsglobal.org
- IMS/LOM Meta-Data, see: IMS http://www.imsglobal.org
- Open Archives Initiative Protocol for Metadata Harvesting, www.openarchives.org/OAI/openarchiesprotocol.html
- OKI Architecture Overview, Open Knowledge Initiative, http://web.mit.edu/oki/product/whtpapers/arch_overview.html
- Reference Architecture for EML Design-time (RAED) Tooling, Eric Kuijfhout, The Valkenburg Group, Open University of the Netherlands.
- Unified Modeling Language: http://www.omg.org/technology/documents/formal/uml.htm
- W3C http://www.w3c.org (consortium for Web related interoperability specifications) relevant specifications: HTML, XHTML, XML 1.0, XML schema, XML namespaces, XSLT.

1.5 Overview of this document

This Use Cases Specification document groups UML diagrams and texts that provide a use-case view of requirements for the eduSource suite of tools, hereafter called “the system”.

- Section 2 present the objectives and contextual elements of the system. It is a summary of elements contained in the eduSource Vision document.

- Section 3 presents a set of use-case diagrams each containing several use cases interacting with the different users of the system. Use cases are grouped in a use-case diagram according to the roles of the main actors. The most important are described using sequential diagrams and some are further described in narrative form.

- Section 4 presents another user view organized around conceptual packages. Each package provides a way to classify some of the deliverables in the system. It also provides a conceptual basis for the Software Architecture document that will describe the decomposition of the system into subsystems, classes and class utilities.

- Section 5 presents some user-end view of the functions of some software components, based on the use cases descriptions in section 3 and organized around functional packages in section 4. These functional specifications lead to a proposed user interface.

2. Vision Goals and Constraints

EduSource Canada has adopted objectives in the form of design principles intended to govern the development of the architecture of a distributed learning object repository network. This section describes the software objectives and constraints deriving from these principles that have some significant impact on the use cases specifications.

1. *Support a true network model* as opposed to a silo model with separate repositories fed by publishers often on the basis of separate licensing agreements that increases the cost and restricts the choice of learning materials for all users and especially for small users.
2. *Use standards and protocols that are royalty free* to ensure that there is no a priori overhead cost incurred by agencies wishing to offer services compatible with eduSource.
3. *Implement and support emerging specifications* such as Cancore, whenever practical, and support the use of defined and controlled vocabularies so as to increase interoperability and functionality provided to end users searching and retrieving learning objects.
4. *Enable, don't require*, that is, try to achieve a higher level of consensus among core participants where possible without imposing it as a condition for entry among all participants.
5. *Infrastructure Layer and Service Layer*. The set of software tools comprising the infrastructure layer (described in the present document) will be developed and distributed as royalty-free open source Over and above the infrastructure layer, components with increased functionality may be developed as free and open applications, or they may embody commercial and proprietary components.
6. *Distributed Architecture.* EduSource infrastructure and services are to be designed as a set of related components, each of which fulfills a specific function in the network as a whole. Any given software tool provided by eduSource may be replicated and offered as an independent service to provide robustness and ensure that no single service provider or software developer may exercise control over the network.
7. *An Open Marketplace.* eduSource will support the registration and indexing of various providers, this registration will be free and optional. eduSource will accommodate free content distribution, co-op or shared content distribution, and commercial fee-based content distribution.
8. *Multiple metadata description* of a given learning resource will ensure that potential users of learning resources can obtain and input multiple descriptions of that material.
9. eduSource will be seen as an *implementation of and an extension of the semantic web*, using sector-specific ontologies, into its own design, to support the widest reach possible and reduce the duplication of effort between developers working in specific domains and educators working in the same domain.
10. *Open Rights Management*. Where possible, the acquisition of rights and the exchange of funds will be automated. Multiple digital rights models will be provided for free materials, cooperative sharing and commercial offering on a per-view or subscription based model. No single rights agency will govern all transactions. A given provider of learning materials will work with one of many brokers who sell to multiple purchasers, and a given user may one of many agents who conduct transactions with multiple vendors.

3. Use-Case Models

This section present use cases diagrams that represent some significant, central functionality of the final system, together with a large architectural coverage—they exercise many architectural elements or they stress or illustrate a specific, delicate point of the architecture.

The first three subsections will give a general view of the system (the eduSource suite of tools) from a user point of view.

Each of the following subsections will present one use case diagram grouping several use cases, together with the descriptions of their use cases in narrative form and/or followed by a sequence diagram. We have classified the use cases in five categories:

- Elearning Systems Interoperation
- Digital Repository Development
- Search and Select Learning Objects in Multiple Repositories
- Learning Design and Object Aggregation
- Digital Rights Management

3.1 EduSource General Use Cases View

This first diagram presents the main actors that manipulate the eduSource suite of tools.

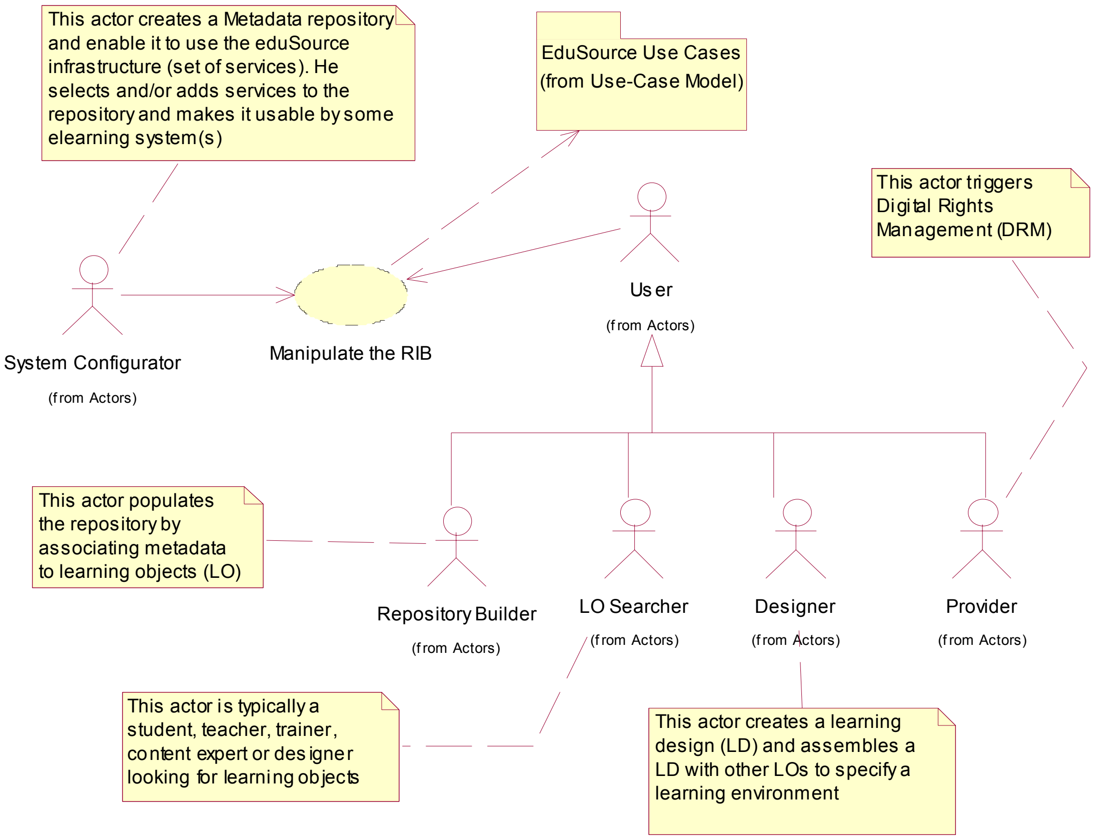

This actor creates a Metadata repository and enable it to use the eduSource infrastructure (set of services). He selects and/or adds services to the repository and makes it usable by some elearning system(s)
EduSource Use Cases
(from Use-Case Model)
This actor triggers Digital Rights Management (DRM)
User
(from Actors)
System Configurator
(from Actors)
Manipulate the RIB
This actor populates the repository by associating metadata to learning objects (LO)
Repository Builder
(from Actors)
LO Searcher
(from Actors)
Designer
(from Actors)
Provider
(from Actors)
This actor is typically a student, teacher, trainer, content expert or designer looking for learning objects
This actor creates a learning design (LD) and assembles a LD with other LOs to specify a learning environment

Figure 1 - General use case model

3.2 EduSource General scenario

The following sequence diagram presents an interaction view of the main actors with the system. These actors usually intervene in the following way even though other interactions patterns will generally occur. The System Configurator enables users to use the system by setting up a set of eduSource services. Then a Repository Builder can start constructing a learning object repository; a Designer can aggregate learning objects (LO) with a learning design and package it for the other users; a Publisher uses a DRM system; and finally a LO Searcher (teacher, trainer, learner, content expert, manager, etc.) searches multiple metadata repositories, displays objects (resources) and selects some for his/her own purposes.

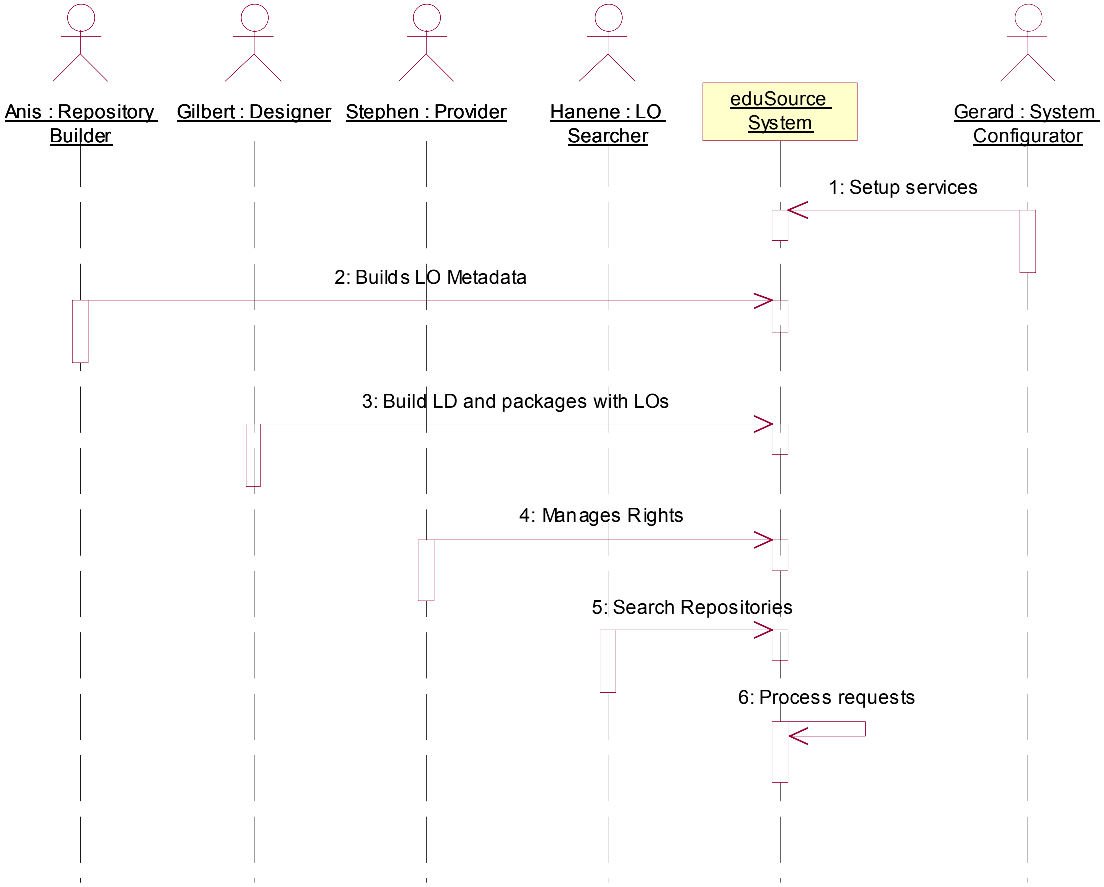


Figure 2 - EduSource general sequence diagram

3.3 EduSource Actors

This third general diagram shows the interaction between the main actors and secondary actors that will appear in the more detailed diagrams to be presented in the following sections.

We see here a user that wants to access Metadata Repositories through the use of Metadata Repository Services, possibly with the help an eLearning system (any LMS, LCMS, platform or computer agent). This user is either a Repository Builder, a LO Searcher, a LO Publisher or a Designer.

The Repository Builder needs the help of a System Configurator, using an eduSource Web services directory, to start building and maintaining a repository. The Repository Builder can also use a Search Result Display (SRD) to find objects to integrate in the repository. The SRD, and also the LO Searcher, both obtain their LOM (s) by using one or more methods, personified by actors called Harvester, Federator, or Distributor. The designer uses a Content Packager to aggregate learning designs with learning objects: the CP and the Launcher both interact with LO (resource) Repositories directly. Finally, the Publisher uses a Provider Broker and the LO searcher uses a Purchaser Broker, both communicating with an External Payment System, to protect IP while responding to requests from human Publishers or purchasers or learning objects.

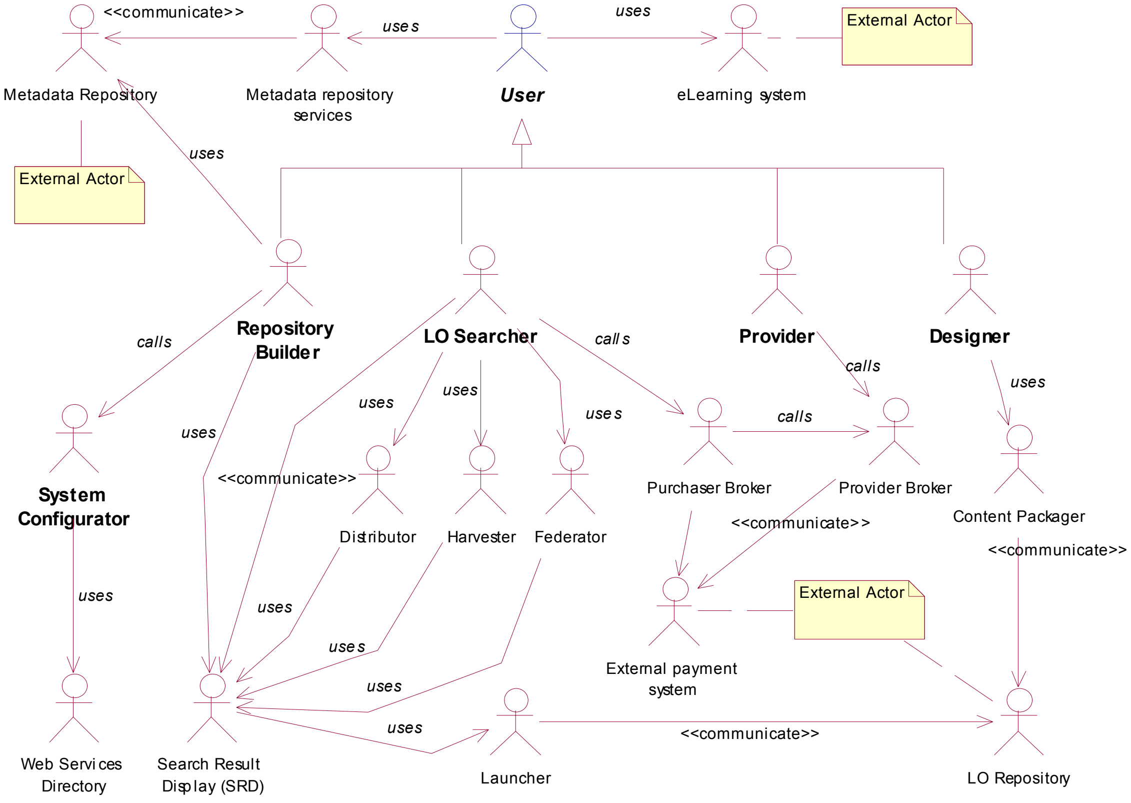


Figure 3 - EduSource actors relations diagram

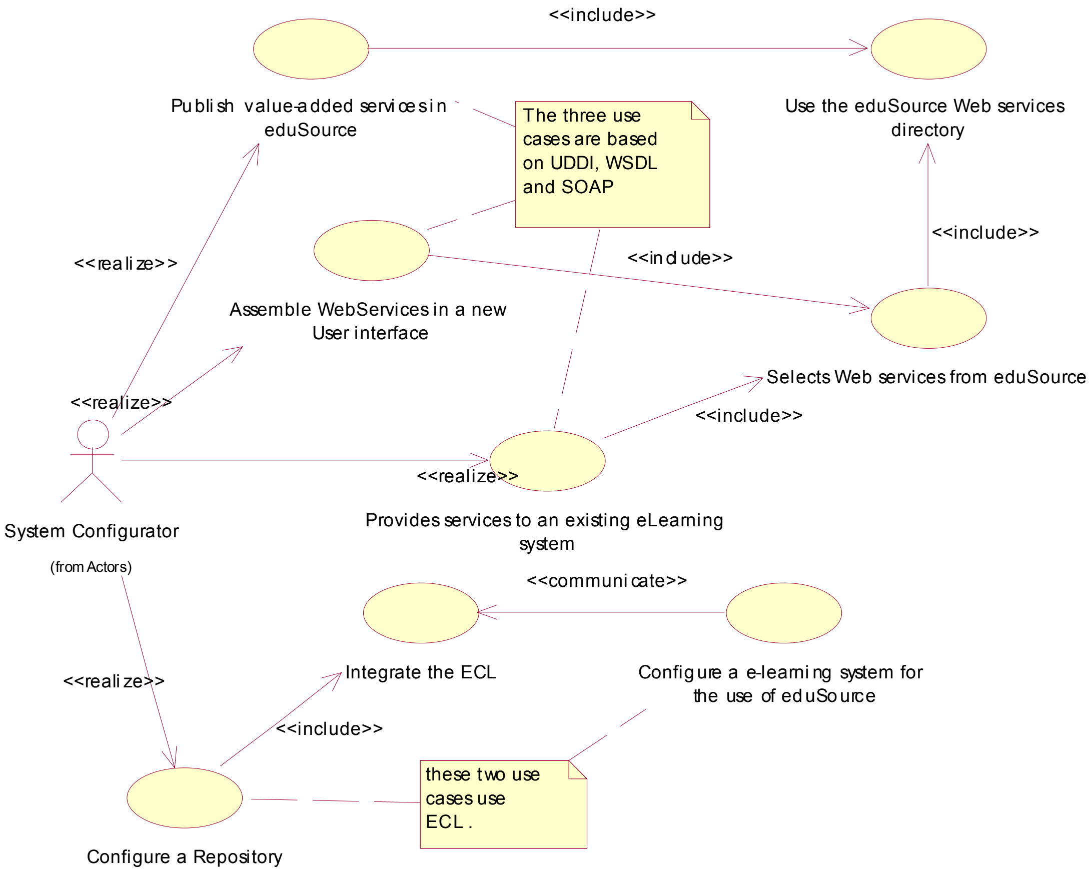
3.4 Configuration and administration

This sub-section presents essential operations by which a *System Configurator or an eduSource Administrator (usually a person)* can make elearning systems (platforms, LMSs, LCMs, agents, etc.) inteoperate with the eduSource infrastructure services, and facilitate access to multiple Metadata Repositories and LO Repositories.

3.4.1 Configuration/Administration Use Cases

The following use case diagram shows five main use cases. The first three concern the interaction of the System Configurator with a Web services directory, either to add new value-added services to the eduSource infrastructure or to make such services available to other users by assembling them in a user interface or integrating them in an existing eLearning system.

The two other use cases concern the configuration of repositories and eLearning systems to facilitate communication with the eduSource infrastructure. This is achieved by integrating an *EduSource Communication Language* (ECL) module. ECL is a metaprotocol using SOAP messaging for enabling repositories and eLearning systems to communicate together and with eduSource services together through the most common communication protocols. All the eduSource tools and services will be ECL enabled.



<<include>>
Publish value-added services in eduSource
The three use cases are based on UDDI, WSDL and SOAP
Use the eduSource Web services directory
<<realize>>
<<include>>
<<include>>
Assemble WebServices in a new User interface
Selects Web services from eduSource
<<realize>>
<<include>>
<<realize>>
System Configurator
(from Actors)
Provides services to an existing eLearning system
<<communicate>>
<<realize>>
Integrate the ECL
Configure a e-learning system for the use of eduSource
<<include>>
these two use cases use ECL .
Configure a Repository

Figure 4 - Configuration use cases diagram

3.4.2 Configuration Sequence Diagram

This diagram illustrates a sequencing of the main interactions of a *System Configurator* with a Web services server through which the System Configurator will enable interoperability between a eLearning system and a user such as the Repository Builder. Such an actor can ask for eduSource services either to the eLearning system, or directly to the System Configurator if he/she doesn't use a eLearning system.

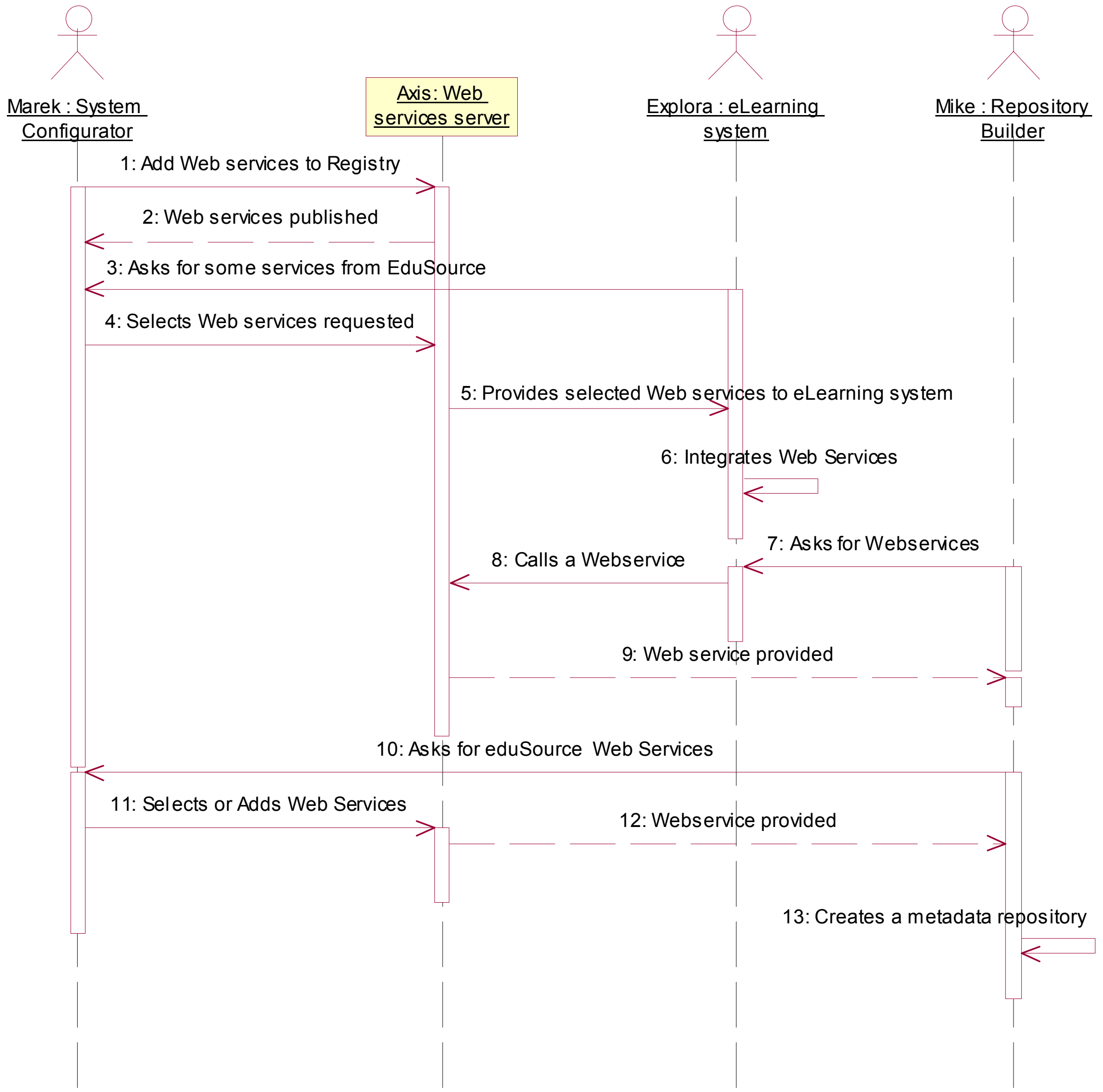

Marek : System Configurator
Axis: Web services server
Explora : eLearning system
Mike : Repository Builder
1: Add Web services to Registry
2: Web services published
3: Asks for some services from EduSource
4: Selects Web services requested
5: Provides selected Web services to eLearning system
6: Integrates Web Services
7: Asks for Webservices
8: Calls a Webservice
9: Web service provided
10: Asks for eduSource Web Services
11: Selects or Adds Web Services
12: Webservice provided
13: Creates a metadata repository

Figure 5 - Configuration sequence diagram

3.5 Metadata Repository Development and Maintenance

This sub-section describes essential operations by which a *Repository Builder*, generally a person, builds a metadata repository, makes it available on the Web to other users, manages and maintains its components.

3.5.1 Build and Maintain Use Cases

Basically, the *Repository Builder* can create or suppress a repository, manage metadata records including publishing new LOM records, modifying or suppressing such records.

Before this can be done, a LO must be found on the Web, on a LAN, or through a search in other Metadata repositories. This last case involves a LO searcher that will be described in section 3.5 of this document and it leads to modifying existing metadata records or adding a different LO description to the metadata repository. On the other hand, the Repository Builder car use usual techniques to find an object on the Web or on a LAN that is not already described by metadata or is not obtainable through metadata search methods. This case can lead to adding a new record to the metadata repository. In both cases, once a LO has been found and selected, a metatagging tool must be used to enter metadata in the repository.

The Repository Builder also manages the access rights to users, either to the whole repository, to groups of metadata records, to individual records or to learning objects referenced by the metadata. This use case must be distinguished from the Publisher (DRM) use cases that will be described in section 3.8. In fact, here, the Repository Builder takes care of the response of the Repository to requests made by a user. This user might be a purchaser having obtained rights through the DRM system, or a Publisher that has set conditions to use some learning objects. When a metadata record contains restrictions, the Repository Builder must obtain a credential made available by the Publisher to a purchaser, to enable the use of the corresponding object by the purchaser.

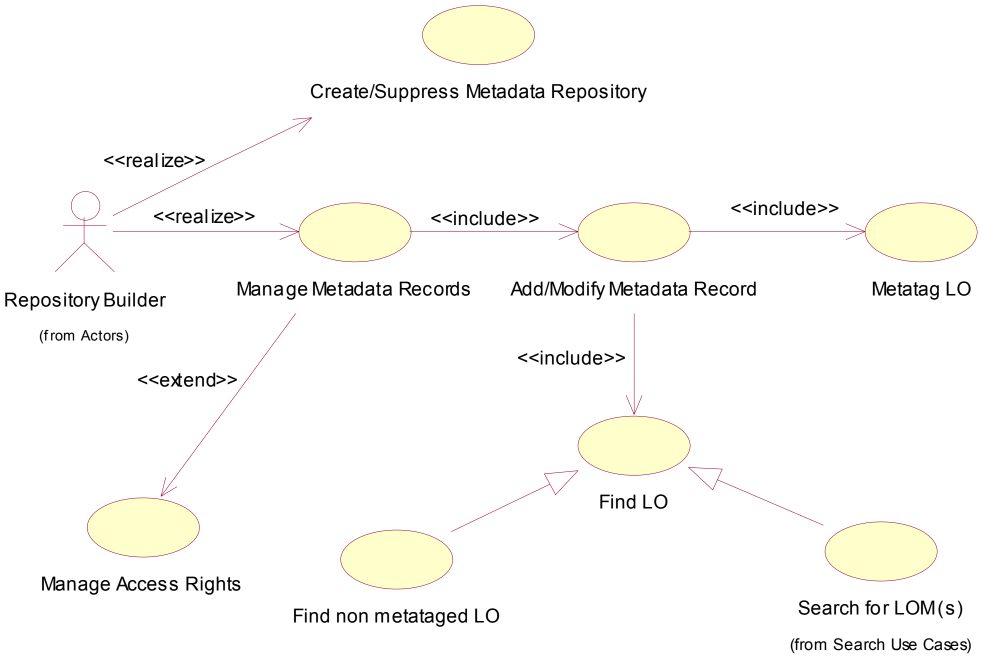


Figure 6 - Build use cases diagram

3.5.2 Build and Maintain Sequence Diagram

The following sequence diagram shows the main interactions of a Repository Builder with a Metadata Repository. In the process, this actor interacts with LOM (s) resulting from a search in possibly multiple repositories, or with raw learning objects obtained through the Web or through browse and search on a Local Area Network (LAN).

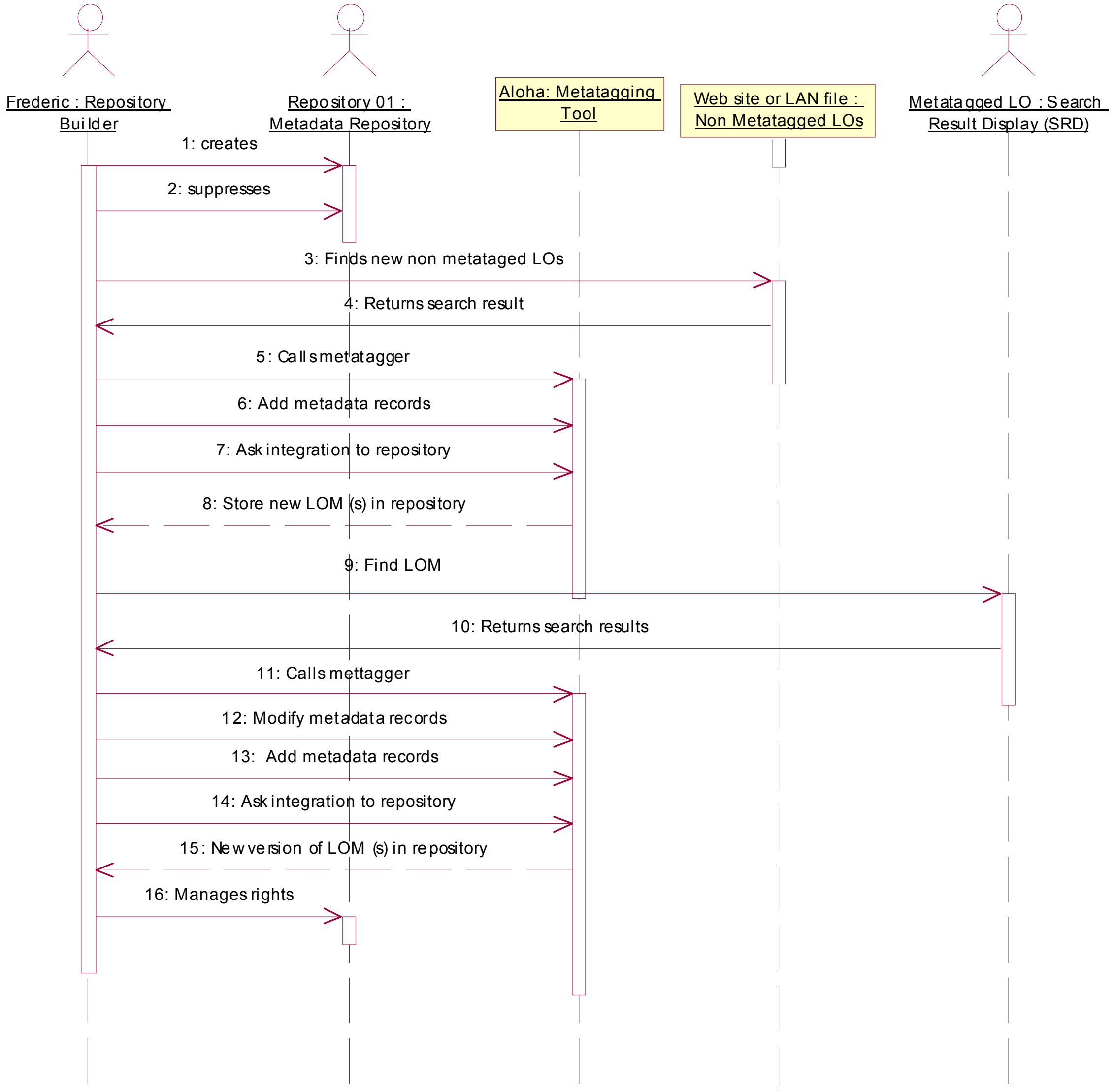

Frederic : Repository Builder
Repository 01 : Metadata Repository
Aloha: Metatagging Tool
Web site or LAN file : Non Metatagged LOs
Metatagged LO : Search Result Display (SRD)
1: creates
2: suppresses
3: Finds new non metataged LOs
4: Returns search result
5: Calls metatagger
6: Add metadata records
7: Ask integration to repository
8: Store new LOM (s) in repository
9: Find LOM
10: Returns search results
11: Calls mettagger
12: Modify metadata records
13: Add metadata records
14: Ask integration to repository
15: New version of LOM (s) in repository
16: Manages rights

Figure 7 - Build repository sequence diagram

3.6 Search and Select Learning Objects in Multiple Repositories

This sub-section describes essential operations by which a LO Searcher (a person or a software agent) can select a search method, query a network of learning object repositories (LORs) using that method, obtain and display a set Learning Object Metadata (LOM), see the corresponding objects, and finally select one or more objects whose addresses are stored as hyperlinks in a Web page.

3.6.1 Search and Select Use Cases

Most often the LO Searcher will be a person (learner, trainer, designer, manager), but it may very well be software running as part of an LCMS or an agent launching a scheduled Intelligent Search Agent on behalf a human user. The other actors in the following diagram are software agents or tools provided as services to the LO Searcher.

The diagram presents 6 use cases. The first one, Display Metadata & Objects, captures the LO Searcher's goal to create a Web page containing links to a set of selected learning objects; this page is generated by a Search Result Display actor using a Display Metadata and Objects tool. This use case is extended by a generic search service use case that can trigger one, two or three search methods. These methods are shown on the diagram as sub-cases of the Search for LOM (s) use case. Actors called the Federator, the Harvester and the Distributor are responsible for these three specialized search use cases:

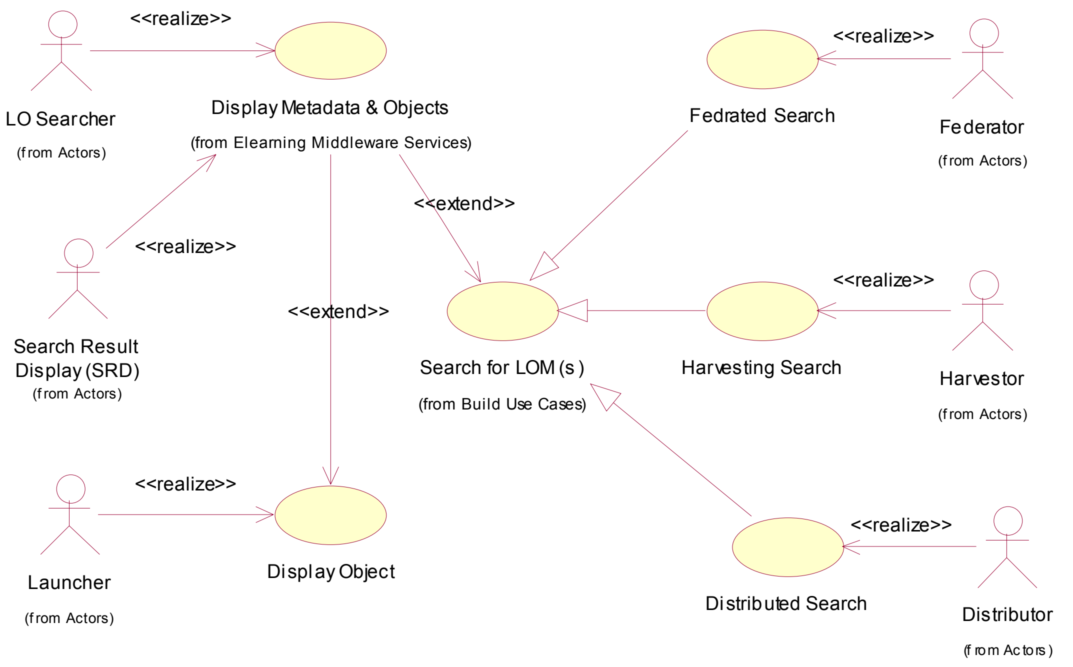


Figure 8 - Search and display uses cases

3.6.2 General Display Sequence Diagram

The following sequence diagram present the general search and display operations. The LO searcher uses a Search Result Display tool to start a search. The tool present a choice of search methods and corresponding query interfaces. The LO search selects certain search methods and enter the query parameters

Whatever search methods are used, a resulting list of learning object metadata (LOMs) records is presented to the LO Searcher who can ask to see any of them. An actor called the Launcher is responsible to contact a server for that object and display the object.

The LO searcher can then retain the object and its metadata record in a list for further use.

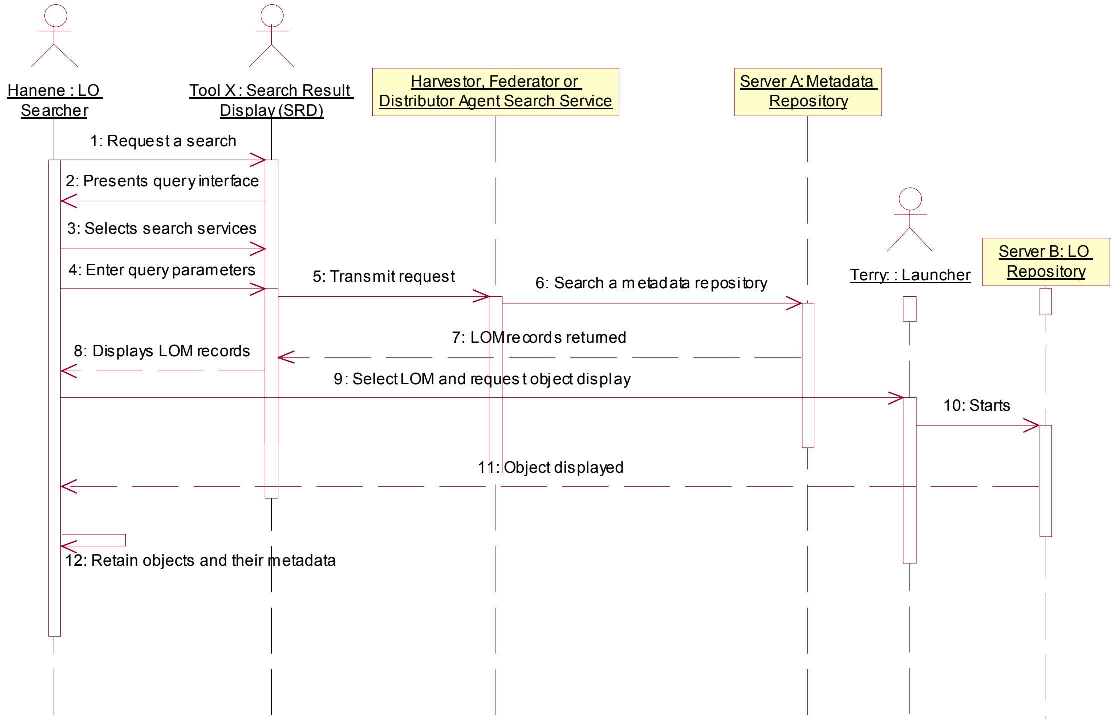

Hanene : LO Searcher
Tool X : Search Result Display (SRD)
Harvestor, Federator or Distributor Agent Search Service
Server A: Metadata Repository
Terry: : Launcher
Server B: LO Repository
1: Request a search
2: Presents query interface
3: Selects search services
4: Enter query parameters
5: Transmit request
6: Search a metadata repository
7: LOM records returned
8: Displays LOM records
9: Select LOM and request object display
10: Starts
11: Object displayed
12: Retain objects and their metadata

Figure 9 – General Search and Display sequence diagram

We now present a detailed narrative of this use case

Description This use case presents a high-level process through which a LO Searcher uses the eduSource software services to find, view and select one or more learning objects, their address being retained in a Web page.

Goal in Context To provide users (designers, learners, trainers, managers, etc) with diversified ways to find and aggregate a set of related learning objects

Preconditions The LO Searcher has access freely (no DRM component involved in this case) to one or more learning object repositories on the Internet.

Success End Condition The LO Searcher has retained a (possibly empty) set of learning objects and the system has displayed hyperlinks to these objects in a Web page.

Failed End Condition No Web page is produced because either search has failed or the display has failed.

Primary Actor **LO Searcher**: a person (or a software application) wanting to search a network of metadata repositories. Most often the LO Searcher will be a person, but it may very well be software running as part of an LCMS or an agent launching a scheduled Intelligent Search Agent on behalf a human user.

Secondary Actors

Search Result Display (SRD): an eduSource tool
One or more search service that can be a federator, a harvester or a distributed search.
Launcher: an actor that can display a learning object or a content package in a viewable format

Trigger

A LO Searcher calls upon the SRD service

FLOW OF EVENTS	**Step**	**Action**
	1	The LO Searcher finds the address of a SRD service in a registry
	2	The SRD tools presents a choice of search methods to the LO searcher.
	3	The LO Searcher selects a search service or a combination of search services
	4	The SRD presents query interfaces corresponding to the selected search services
	5	The LO Searcher enters the query and validates
	6	The SRD calls the selected search services (SSs)
	7	The SSs search the Metadata Repository network and return lists of Metadata Records to the SRD
	8	The SRD displays the list of Metadata Record names and addresses to the LO Searcher
	9	The LO Searcher asks the SRD to display the Metadata of a selected object
	10	The LO Searcher asks the Launcher to display selected objects
	11	The LO Searcher asks the SRD to group the retained objects and their Metadata Records hyperlinks in a Web page
EXTENSIONS	**Step**	**Branching Action**
	7a	IF a selected SS is a federator, call upon the Federation use case
	7b	IF a selected SS is a harvester, call upon the Harvesting use case
	7c	IF a selected SS is a distributor, call upon the Distribute Search use case
	10a	IF the address of the selected object in not a URL, call upon an appropriate object launching use-case (ex: content packaging), otherwise, display the Web page at the URL.
VARIATIONS	**Step**	**Branching Action**
	1a	IF the LO Searcher has already selected a search method without calling the SRD, Go to step 8 using the appropriate SS and launch the SRD for step 9-12.

COMMENTS

Authors: Gilbert Paquette

To be validated by the Functional Architecture Group

3.6.3 Federated Search Sequence diagram

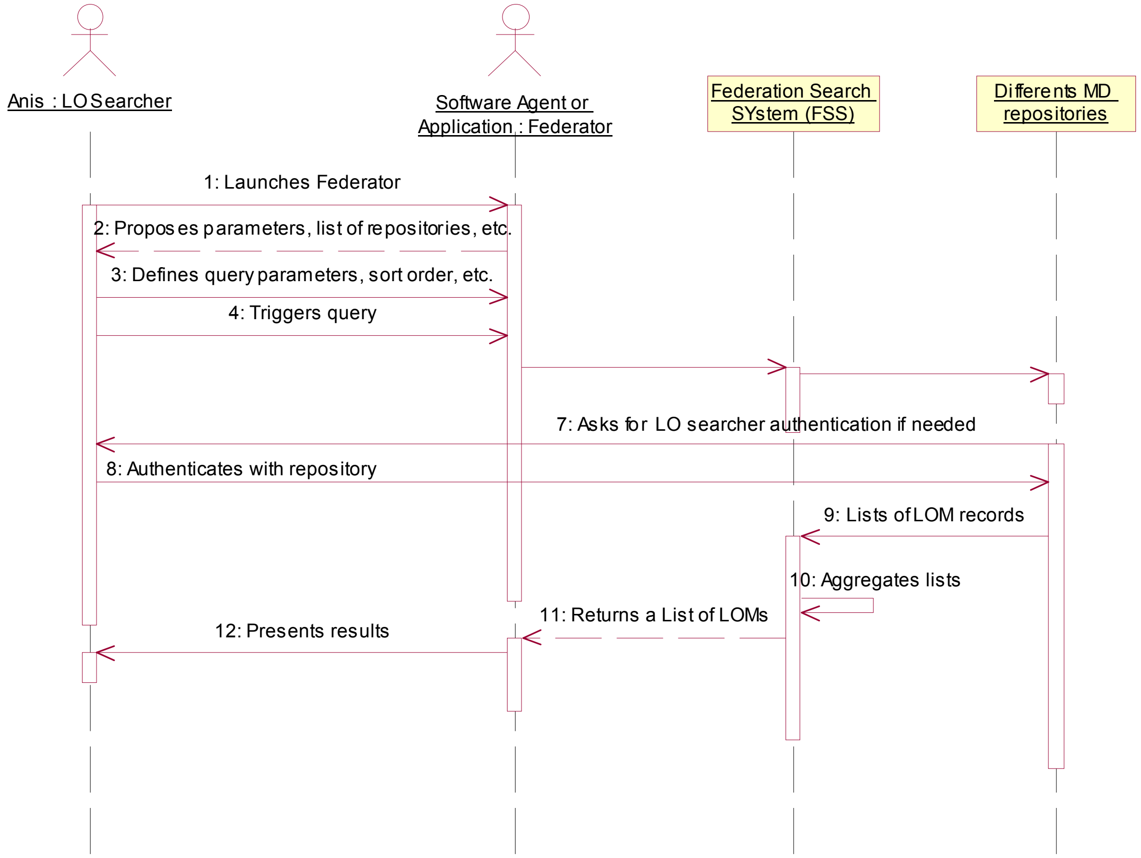

Anis : LO Searcher
Software Agent or Application : Federator
Federation Search SYstem (FSS)
Differents MD repositories
1: Launches Federator
2: Proposes parameters, list of repositories, etc.
3: Defines query parameters, sort order, etc.
4: Triggers query
7: Asks for LO searcher authentication if needed
8: Authenticates with repository
9: Lists of LOM records
10: Aggregates lists
11: Returns a List of LOMs
12: Presents results

Figure 10 - Federated search sequence Diagram

We now present a detailed narrative of this use case.

Description	This use case details the process through which a Federator interacts within eduSource to search a network of Metadata repositories and to locate a set of LOMs.
Goal in Context	The goal is to create a service for a LO Searcher (teacher, student ,course developer or software agent) to search for a learning object across multiple repositories, providing an aggregated result list of metadata records in return.
Preconditions	The LO Searcher must have access to the Internet.
Success End Condition	The Federator possesses a (possibly empty) result list of learning objects from a variety of repositories which meet a search criterion provided by the LO Searcher.
Failed End Condition	The Federator has not gained possession of any result list, the search aborts or returns an incomplete list before searching all selected sites
Primary Actor	**Federator**: a software agent responsible for providing the interaction between the LO Searcher and the Federated Search System (FSS). The Search Agent may be extremely minimal - it might just be a web browser - or it could be an Artificially Intelligent Agent

responsible for searching and harvesting metadata applicable to a students' course load or other general educational requirements defined by the LMS. It is the responsibility of the Search Agent to work within the framework of the LMS when interacting with a human user while generating queries or presenting results.

Secondary Actors

LO Searcher: a person (or a software application) wanting to search a network of metadata repositories. Most often the LO Searcher will be a person, but it may very well be software running as part of an LMS or an agent launching a scheduled Intelligent Search Agent on behalf a human user.

Federated Search System (FSS): a software interface that can access a set of metadata repositories, search each one according to their conditions and returns a set of Metadata Records.

Trigger

The LO Searcher, directly or using the Search Result Display tool asks the Federator to query a network of metadata repositories.

FLOW OF EVENTS

Step **Action**

1 A LO Searcher launches the Federator, directly or through the SRD.

2 The Federator selects search parameters appropriate for this LO Searcher: This includes proposing a list of eduSource repositories available for searching given the DRM rights and classification of the LO Searcher. Parameters include methods for general keyword searching, as well as facilities for substring and full/partial-match phrase searching in general or specific metadata fields or groups of fields (ie, people). Optional facilities would include those for Natural Language Processing, or conducting searches and "joins" across the actual learning object content and it's metadata..

3 The LO Searcher selects MD repositories, defines query parameters such as search priorities et presentation order.

4 The LO Searcher triggers the query

5 The Federator submits the query to the Federated Search System (FSS). This can be done using Xform search query

6 The FSS processes the form submission generating a separate query for each repository. The format of these queries will depend upon what protocols the FSS instance supports. They will most likely be XQueries according to the IMS DRI spec, but could also be in the form of z39.50, or repository specific www-form-url-encoded keyword searches. The FSS can even launch it's own Search Agent for querying other Federated Search Systems.

7 Depending on the LOM requested, some repositories can ask the LO searcher to authenticate with the repository.

8 The LO searcher authenticates as requested

9 The repositories provide lists of LOM records to the FSS.

10 The FSS aggregates the results - collating multiple results sharing the same learning object global unique identifiers (GUID's), sorting them according to LO Searcher preferences etc.

11 The FSS returns the aggregated result list, in XML Format to the Federator

		performing the query.
	12	The Federator makes these results available to the LO Searcher directly or to the SRD (depending on the origin of the call).
EXTENSIONS	**Step**	**Branching Action**
	5a, 12a	If the LO Searcher is a person calling directly the Federator, instead of returning XML code, the Federator returns a Web page using CSS and XSLT. (If the LO Searcher has called a search through the SRD, the SRD takes care of the presentation of results).
VARIATIONS	**Step**	**Branching Action**
	8a	If the LO searcher decides to not authenticate with all or some of the MD repositories. The corresponding LOM will not be returned to the FSS at step 9. But action will proceed.
	10 a	If the FSS returns an empty list to the Search Agent, it may, at the LO Searchers option, adjust the parameters of the search to broaden the scope. The converse is also true in that the Agent, the FSS, or a repository may further restrict the scope of the query to narrow the result list - perhaps by removing common or unnecessary keywords from the query, etc. The Federator may also restrict results by specifying time or resource limits in the query parameters to the FSS.
COMMENTS	Authors: Chris Hubick and Terry Anderson	
	XForms is suggested for this protocol; this would most likely require (or strongly benefit from) defined XForm control names and types.	
	In non-Xquery searches, standards for form control names/types and other interactions will be needed.	
	Should the FSS result list use Xschema?	
	Should agent control, display of returned metadata with expandable fields such as Cancore general, lifecycle etc. or should this be handled by the FSS, based upon parameters passed from the Agent with the search request?	

3.6.4 Harvesting Search Sequence diagram

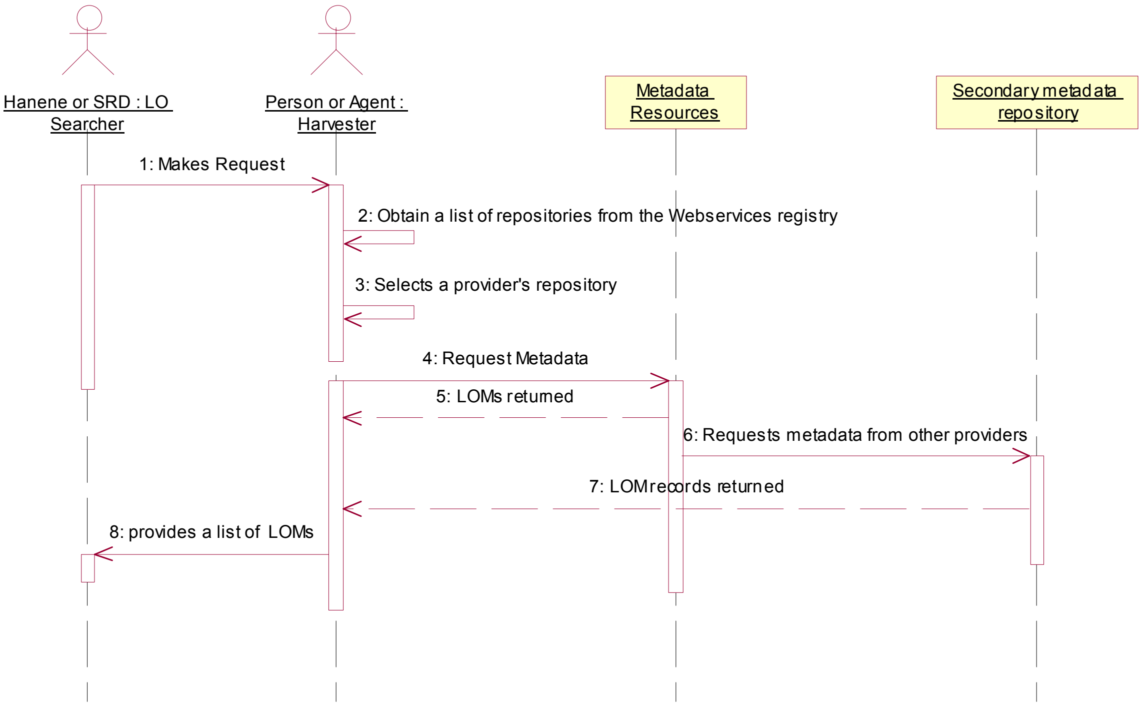

Hanene or SRD : LO Searcher
Person or Agent : Harvester
Metadata Resources
Secondary metadata repository
1: Makes Request
2: Obtain a list of repositories from the Webservices registry
3: Selects a provider's repository
4: Request Metadata
5: LOMs returned
6: Requests metadata from other providers
7: LOM records returned
8: provides a list of LOMs

Figure 11 - Harvesting search sequence diagram

We now present a detailed narrative of this use case.

Description	A Harvester is operated by a service provider as a means of collecting metadata from repositories, here defined as network accessible servers that can process the 6 OAI-PMH requests (GetRecord, Identify, ListIdentifiers, ListMetadatFormats, ListRecords, ListSets). An acceptable alternative to OAI-PMH is to use D-Space repositories.
Goal in Context	To provide LO-Searchers with a way to use a Harvester as a search service and enable resource Publishers (commercial and non-commercial) with a simple way to make their resources available for use through the eduSource network.
Preconditions	The LO Searcher has access to a Harvester on his/her workstation. At least one learning resource Publisher has created at least one metadata repository and a corresponding resource repository available to Harvesters in a Registry that list all such repositories available for harvesting.
Success End Condition	The LO Searcher has located a resource through a Harvester from at least one learning resource Publisher.
Failed End Condition	No search results can be obtained from the harvesting system.
Primary Actor	**Harvester**: a client application that issues OAI-PMH compliant requests and makes metadata available to LO Searchers.

Secondary Actors

Resource Publisher a person (or a software agent) creating repositories for the harvesting system.

Registry service: a service making available a list of metadata and resource repositories.

Trigger

The LO searcher makes a request to a Harvester on his client station, directly or though a SRD (Search Results Display) tool.

FLOW OF EVENTS

Step	**Action**
1	LO searcher or SRD makes a request to a Harvester
2	Harvester requests list of repositories from the registry service
3	Harvester selects a Publisher's repository
4	Harvester requests all metadata for each resource listed in the list of metadata resources.
5	Harvester returns LOM
6	The harvester requests metadata from other Publishers having repositories listed in the Registry.
7	Harvester returns more LOMs
8	The harvester provides a list of metadata records matching values set originally by the LO searcher or the SRD, harvest from the selected Publisher or from third party repositories..

EXTENSIONS

Step	**Branching Action**
4a	Harvester may request a subset of all metadata, for example, recent metadata or metadata satisfying search conditions, if this capacity is provided by the Publisher's metadata repository.
6a	The harvester requests metadata from other Publishers using a Federator or a Distributor (see other use cases)

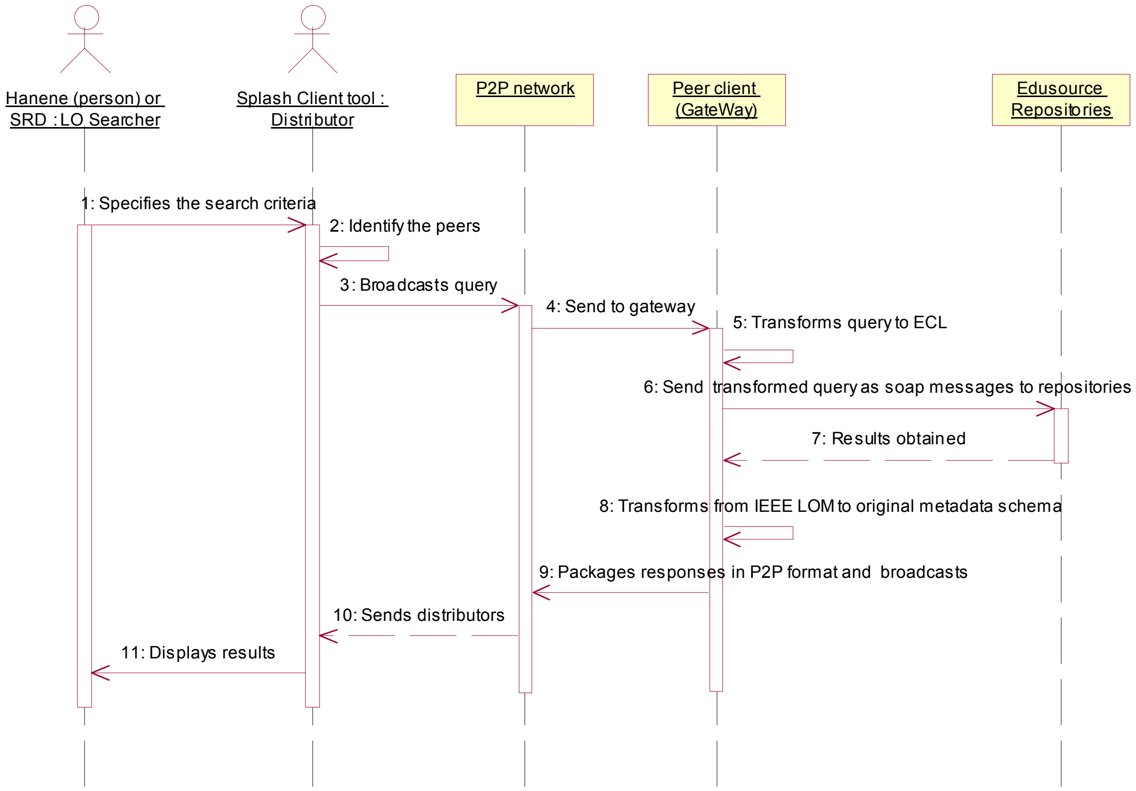
VARIATIONS

Step	**Branching Action**
7a.	Harvester may provide full or partial metadata as a response to parameters of search request (for example a DC or CANCORE set of metadata).

COMMENTS

Authors: Stephen Downes

3.6.5 Peer-to-Peer Distributed Search Sequence diagram



Hanene (person) or SRD : LO Searcher
Splash Client tool : Distributor
P2P network
Peer client (GateWay)
Edusource Repositories
1: Specifies the search criteria
2: Identify the peers
3: Broadcasts query
4: Send to gateway
5: Transforms query to ECL
6: Send transformed query as soap messages to repositories
7: Results obtained
8: Transforms from IEEE LOM to original metadata schema
9: Packages responses in P2P format and broadcasts
10: Sends distributors
11: Displays results

Figure 12 - Distributed P2P search sequence diagram

We now present a detailed narrative of this use case.

Description	The user (LO searcher) issues a search request to P2P network, directly or though the Search Result Display tool. The query is transformed and sent to the repository (-ies) that can be reached through the eduSource gateway. The results from the repository are received by the gateway and after transformation are propagated via P2P network to the user.
Goal in Context	The LO searcher receives metadata results from eduSource reachable repositories on peers and LO Publisher's servers.
Preconditions	The peer is connected to P2P network The gateway between P2P network and eduSource is configured and running Some eduSource repositories are open for search requests
Success End Condition	The metadata records from the eduSource reachable repositories are received by the peer and displayed to the user
Failed End Condition	No results reach the LO searcher, even when preconditions are met.

Primary Actor **Distributor**: a PtoP distributed search agent on the client

Secondary Actors **LO Searcher**: a person (or a software application) wanting to search a network of metadata repositories. Most often the LO Searcher will be a person, but it may very well be software running as part of an LMS or an agent launching a scheduled Intelligent Search Agent on behalf a human user.

Trigger The LO Searcher or the SRD asks for a search to a P2P client

FLOW OF EVENTS

Step	**Action**
1	The LO Searcher or the SRD specifies the search criteria to the Distributor (SPLASH or other peer client connected to the P2P network).
2	The distributor identifies the peers to which it will broadcast the query (based on the peer selection mechanism which is part of the P2P network design) and broadcasts the query.
3	The query is broadcasted through the P2P network.
4	The query reaches the peer performing a function of gateway between eduSource and the P2P system. The gateway performs the following operations: — Transforms the query from the P2P transport protocol to ECL protocol using XQuery specification (temporarily Xpath will replace XQuery) — If required, maps the metadata schema used within the P2P network to IEEE LOM (this step is omitted if the P2P system uses CanCore as is the case with SPLASH)
5	Sends a SOAP message containing the transformed query to the eduSource repositories and awaits for the responses.
7	Results are returned to the gateway
8	After response from the eduSource repositories are received, the gateway transforms the results from the IEEE-LOM to the original request format
9	The gateway transform the results in P2P format
10	The results are propagated back via the P2P network when they eventually reach the originating peer (the Distributor).
11	The Distributor displays results to the LO searcher directly or through the SRD

EXTENSIONS

Step	**Branching Action**
4a	If the request does not reach the gateway, the query request is not sent to the edusource repositories.
4b	The connection to (some of) eduSource repositories is down. Only the results from the available repositories are processed in step 5.
5a	The response is not received within the timeout specified in the original request. Only the results from the repositories received within the timeout are processed.
6a	The response does not reach the peer within the query time out. The results are discarded on their way back to the originating peer

8a The user’s browser cannot connect to the object repository or location is broken. The object is not displayed.

VARIATIONS **Step** **Branching Action**

2a If the LO searcher selects a parameter specifying directly some target repositories, the Distributor identifies the gateway peers for the specified repositories and proceeds to step 3.

COMMENTS Authors: Marek Hatala

Remains to be validated by the Functional Architecture Group

3.6.6 View Objects Sequence diagram

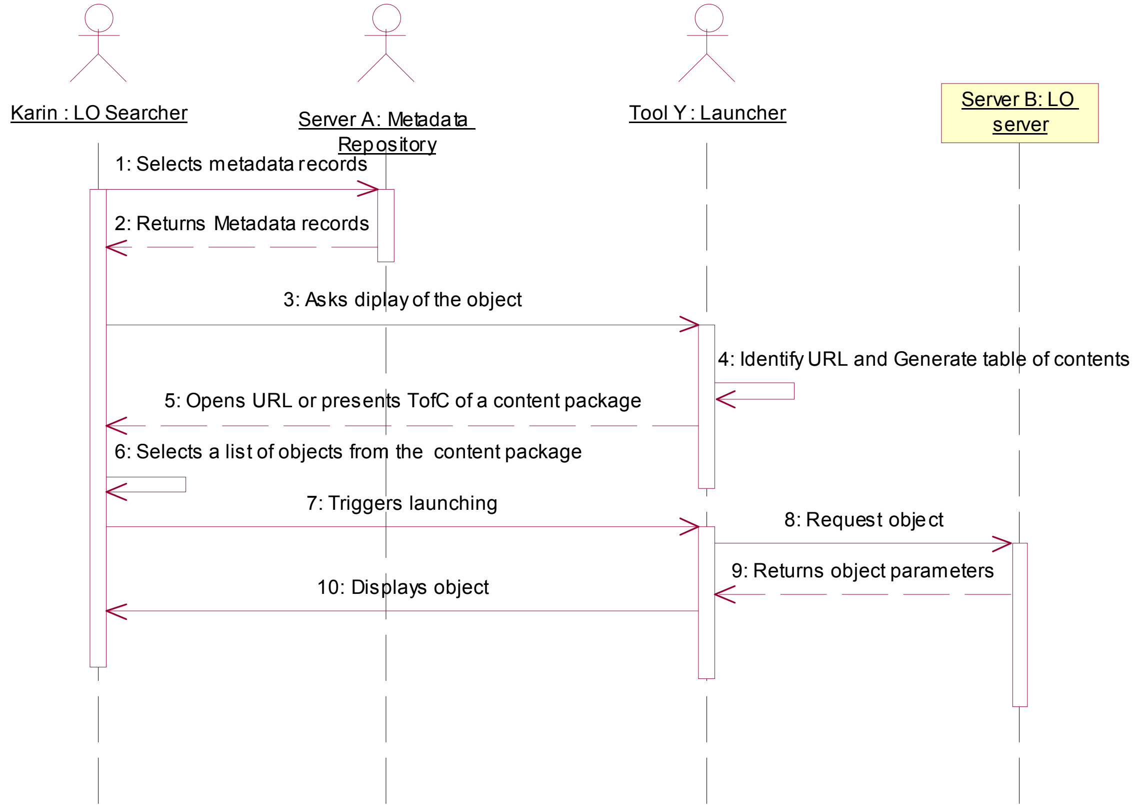

Karin : LO Searcher
Server A: Metadata Repository
Tool Y : Launcher
Server B: LO server
1: Selects metadata records
2: Returns Metadata records
3: Asks diplay of the object
4: Identify URL and Generate table of contents
5: Opens URL or presents TofC of a content package
6: Selects a list of objects from the content package
7: Triggers launching
8: Request object
9: Returns object parameters
10: Displays object

Figure 13 - View an object or a content package sequence diagram

We now present a detailed narrative of this use case.

Description This use case details the process through which a Launcher interacts with a network of Metadata and Learning Object repositories and displays the items in a content package, or simple un-packaged objects.

Goal in The goal is to create a service for a LO Searcher (teacher, student ,course developer or

Context software agent) to view the objects/resources referenced by a metadata record.

Preconditions The LO Searcher must have access to some LORs, both physically and logically through attaining appropriate rights according to the DRM infrastructure.

Success End Condition The Launcher has displayed items in a content package, or a single object, corresponding to the LO Searcher's request.

Failed End Condition The Launcher has failed to display items or objects required by a LO Searcher that is authorized to access objects in one or more LORs

Primary Actor **Launcher:** an agent that can decompose an IMS-LD or SCORM package and display any of its items or launch an un-packaged object/resource

Secondary Actors **LO Searcher**: a person (or a software application) wanting to search a network of metadata repositories.

Trigger A LO Searcher asks to see the objects in a content package, directly or through a SRD

FLOW OF EVENTS

Step	**Action**
1	The LO Searcher or the SRD, selects Metadata records
2	The MD repository return corresponding LOMs
3	LO searchers ask for display of the objects
4	The Launcher generates a table of contents for the content package
5	The Launcher presents the table of contents to the LO Searcher
6	The LO Searcher selects an item from the table of contents
7	The Launcher request an object/resource for the item from an appropriate server (LO repository) in the network
8	A server returns the media parameters for the asset to the Launcher
9	The Launcher displays the asset to the LO Searcher
10	The Launcher stops

EXTENSIONS

Step	**Branching Action**
7a	If there are more items in the table of contents, the Launcher presents the other items, asks the LO Searcher to select one and goes to step 5. If none are selected the Launcher stops and waits for another call

VARIATIONS

Step	**Branching Action**
1a, 7a	If the LO Searcher has selected an object that is not an IMS/SCORM Content Package but has a URL, the Launcher displays the Web page at this URL
1a, 7a	If the LO Searcher has selected an object that is not a Content Package and the Metadata Record does not have a URL, the Launcher calls upon a specialized launcher.

COMMENTS Authors: D'arcy Norman and Mike Mattson

Remains to be validated by the Functional Architecture Group

Specialized launchers can be made available as Web services from the Explor@ resource manager.and other elearning systems or agents

3.7 Learning Design Aggregation

This sub-section describes essential operations by which a *Designer* Actor, usually a person such as an instructional designer, a professor, a trainer, a learning object creator, or a software aggregator agent) can use the eduSource infrastructure to create, build and test a unit of learning by aggregating learning objects/resources contained in LO repositories with a learning design.

3.7.1 Learning Design Aggregation Use Cases

A unit of learning is an abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc. It can be modeled for example as an IMS Content Package where the organization part is replaced by an IMS Learning Design description.

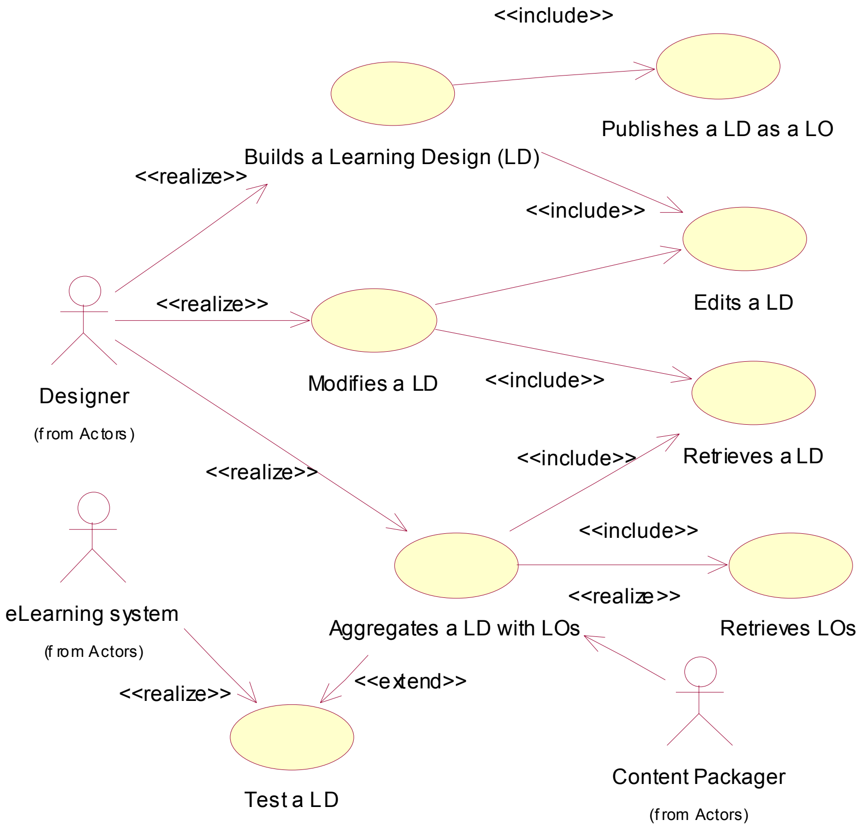


Figure 14 – Learning design sequence diagram

A Learning Design (LD) is composed of an optional title, learning objectives and prerequisites, and metadata referencing the unit of learning as in the IMS content packaging specification. It must have a components and a method part. When activating a unit of learning, the method element is central. It has to be located within the unit of learning set of XML files. This unique element and its sub-elements control the behavior of the unit of learning as a whole, coordinating the activities of the players in the various roles and their use of resources (learning objects). The Method, Plays, Acts and Role-parts are all nested within each other.

As shown on the preceeding use case diagram, the designer controls four main use cases.

- The Designer builds a Learning Design. This includes editing the method (or instructional scenario) with the help of an *LD Editor*. He then publishes a LD XML file and is metadata in the same way as other learning objects.
- The Designer sometimes modifies an existing LD using the LD Editor. For this, he must be able to retrieve a LD from a metadata repository. This is done using the Search and Display use case, presented earlier
- To be able to validate his design, the Designer has to aggregate the LD with learning objects retrieved from metadata reposoitories. This is done using a *Content Packager* actor.
- Finally, the Designer will test the result using a LD enabled e-learning system. This system is outside the scope of the eduSource project, but it needs to be represented as an abstract software actor called here the *eLearning System.*

3.7.2 *Learning Design Aggregation Sequence Diagram*

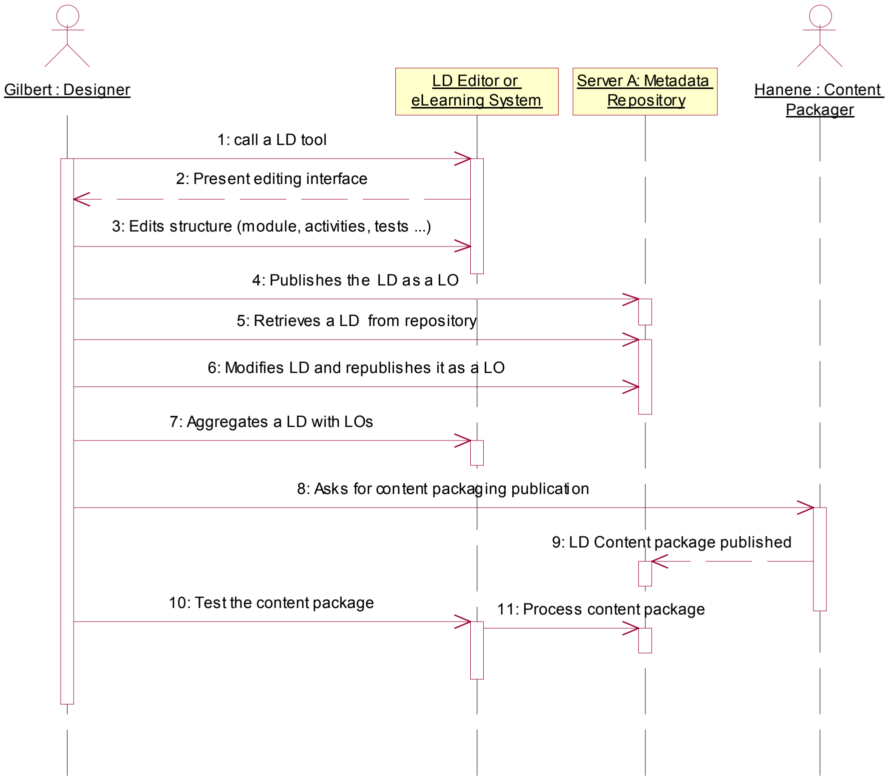

Gilbert : Designer
LD Editor or eLearning System
Server A: Metadata Repository
Hanene : Content Packager
1: call a LD tool
2: Present editing interface
3: Edits structure (module, activities, tests ...)
4: Publishes the LD as a LO
5: Retrieves a LD from repository
6: Modifies LD and republishes it as a LO
7: Aggregates a LD with LOs
8: Asks for content packaging publication
9: LD Content package published
10: Test the content package
11: Process content package

Figure 15 – Learning design sequence diagram

3.8 Digital Rights Management

This sub-section describes essential operations by which a LO Publisher can select a Digital Rights Management (DRM) method, enter conditions for users to access the Metadata records and the objects and to process transactions if needed. It also describes the needed operations from the point of view of the user.

3.8.1 Digital Rights Management Use Cases

The Publisher is the main actor. He calls the Provider Broker, a software agent representing a person that wants to publish a new Learning Object to a metadata repository or to modify rights metadata of an existing LO.

The Provider Broker first presents a set of rights metadata models to the Publisher. Each model includes secondary metadata that specify conditions, for example a certain form of payment, that must be fulfilled in order to gain access to the object. The Publisher selects a model and fill out specific conditions that are associated by the Provide Broker to the LO to be integrated by the Repository Builder presented in section 3.5.

A Purchases Broker, a software agent acting on behalf of a person that want to buy access to an object, obtains rights metadata and asks the purchaser (a LO searcher) for payments prescribed in the rights metadata. It sends any required payment to an External Payment System. This system informs the Publisher that sends a credential (a key) to the LO searcher.

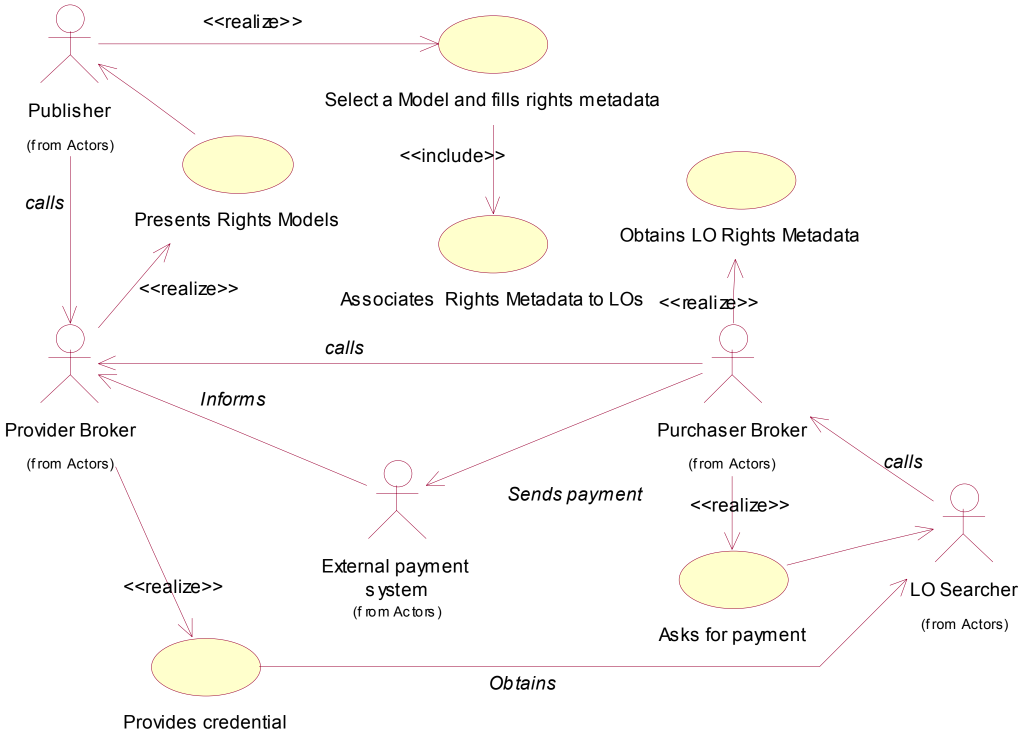
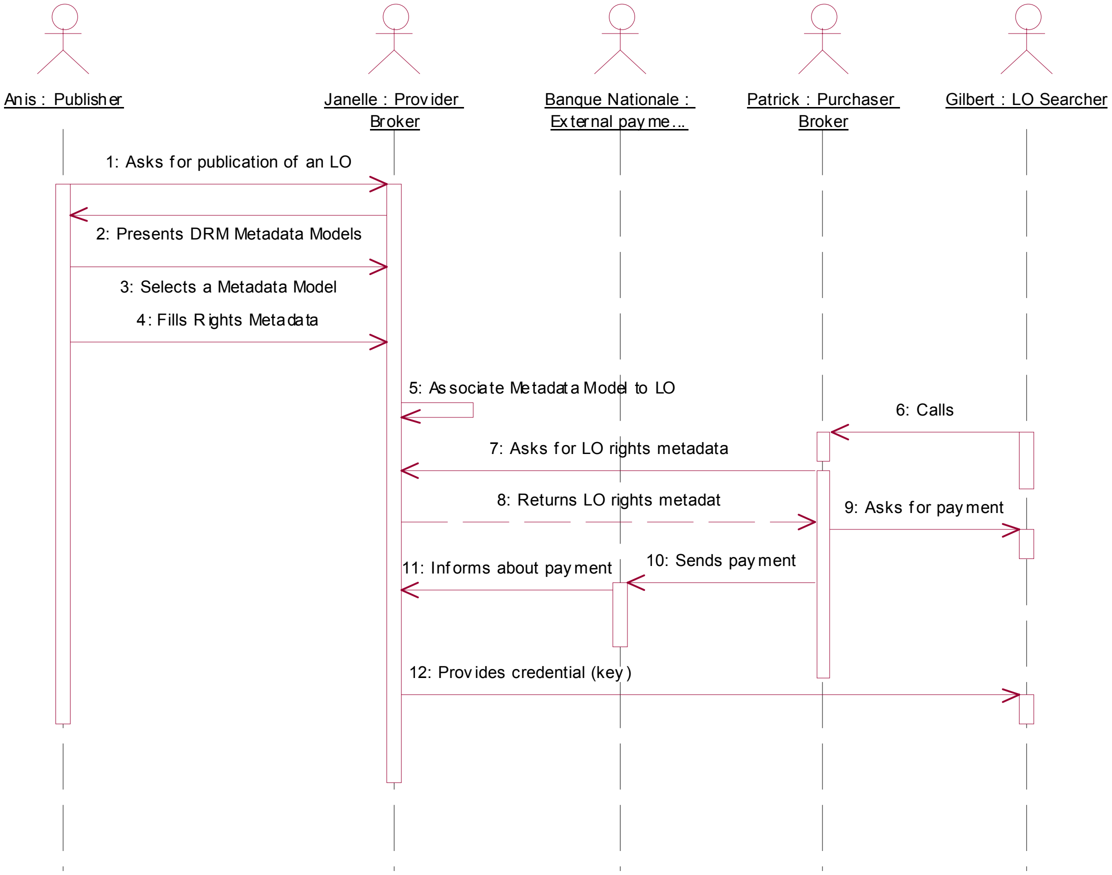


Figure 16 - Digital rights management use cases

3.8.2 Digital Rights Management Sequence Diagram



Anis : Publisher
Janelle : Provider Broker
Banque Nationale : External payme...
Patrick : Purchaser Broker
Gilbert : LO Searcher
1: Asks for publication of an LO
2: Presents DRM Metadata Models
3: Selects a Metadata Model
4: Fills Rights Metadata
5: Associate Metadata Model to LO
6: Calls
7: Asks for LO rights metadata
8: Returns LO rights metadat
9: Asks for payment
10: Sends payment
11: Informs about payment
12: Provides credential (key)

Figure 17 - Digital rights management sequence diagram

4. Conceptual View of the Main Packages

This section describes the decomposition of the system into five conceptual packages grouping the uses cases presented in section 3.

4.1 Systems's Overview

The UML diagram on figure 18 offers a general functional view of the suite of eduSource software tools and services. In the Center five packages of tools and services compose the infrastructure of the eduSource suite of tools and will be described in the following sections

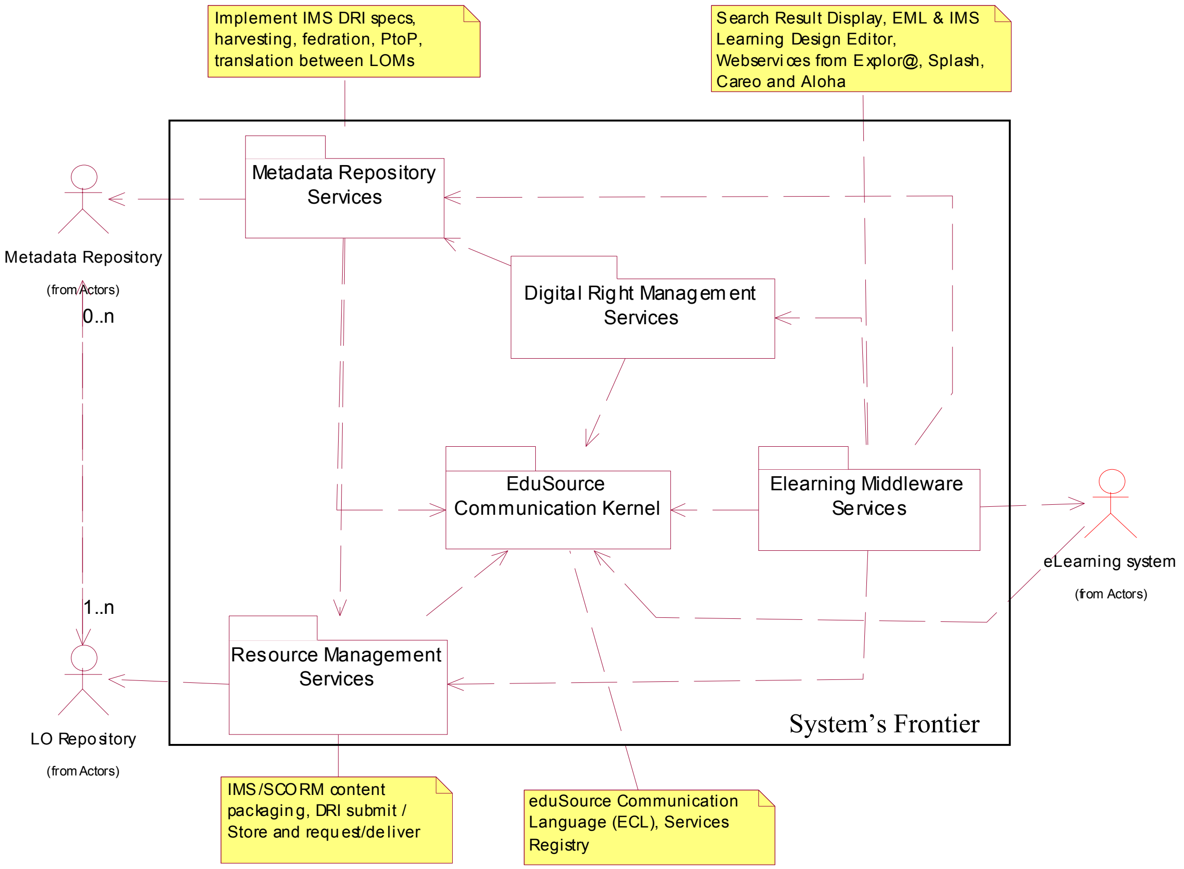
Three sets of components shown on the figure are outside the frontiers of the eduSource system but use it or are used by it. They delimit the frontiers of the system. On the right side are existing or future e-learning systems, LMS, LCMS, agents or tools that can contribute to build and/or use repositories. On the left side, there are two classes of repositories, one grouping metadata repositories and the other grouping digitized resources.These two classes are separated for the sake of clarity, but it is possible that a repository groups both metadata records and the learning objects they describe.

Each *Metadata Repository* groups a set of metadata files referencing educational resources (learning objects). Here we use a very broad definition of a learning object as proposed both in the IEEE LOM and the IMS Design Learning information model. This definition includes the following categories of resources:

- Documents and educational materials, such as multimedia, web pages, texts, software, data records, etc, that hold information and knowledge
- Tools and applications that support the processing of information and knowledge.
- Services sustained by persons such as subject matter experts, trainers, technical assistants, managers.
- Events (or learning opportunities) such as courses, seminars, learning activities, conferences, and discussion group meetings.

Each *Digitized Resource Repository* holds a set of digitized resources: documents and materials, tools and applications. It can also hold digitized information in support to services or events. For example, a course can be described by an Learning Design model that aggregates services from a presenter, delivered on the Web or in a classroom at certain dates, plus learning and assistance scenarios and a set of diversified course materials.

There is a many to many correspondence between metadata repositories and digitized resource (assets) repositories. This is a way to implement a full network approach as opposed to a silo approach. It enables (not require) a metadata repository to reference resources in more than one resource repository and, conversely, it also enables a resource repository to be referenced by more than one metadata repository.



Implement IMS DRI specs, harvesting, fedration, PtoP, translation between LOMs
Search Result Display, EML & IMS Learning Design Editor, Webservices from Explor@, Splash, Careo and Aloha
Metadata Repository
(from Actors)
0..n
1..n
LO Repository
(from Actors)
Metadata Repository Services
Digital Right Management Services
EduSource Communication Kernel
Elearning Middleware Services
eLearning system
(from Actors)
Resource Management Services
System's Frontier
IMS/SCORM content packaging, DRI submit / Store and request/deliver
eduSource Communication Language (ECL), Services Registry

Figure 18 - eduSource general use cases packages

The system is based on the five use cases packages at the center. They hold the suite of software components that are to be developed, most often through Web services. These services are all referenced in one or more eduSource Web services registries available from the Web. Any service can be called by any eduSource-compliant e-learning system or agent, represented on right side of the figure.

There is no unique central piece in the system and components can be duplicated for redundancy and robustness. Registries can be one or many and services can be offered in more than one version. Of course, the extent of this redundancy will be limited by the 18 months time-frame of the project, but the architecture of the eduSource system will embed these principles right from the start, providing for future evolution.

4.2 Communication Kernel

The Communication Kernel includes the eduSource Communication Language (ECL)[1], a meta-protocol facilitating interactions between tools, services and communication protocols, in particular OAI and

[1] The eduSource Communication Language (ECL) – Working document 0.2, Jean-François Arcand

Z39.50. It also contains the eduSource Services Registry (ESER) that references all the services in the infrastructure, from which a eduSource user can select the services that he wishes to use.

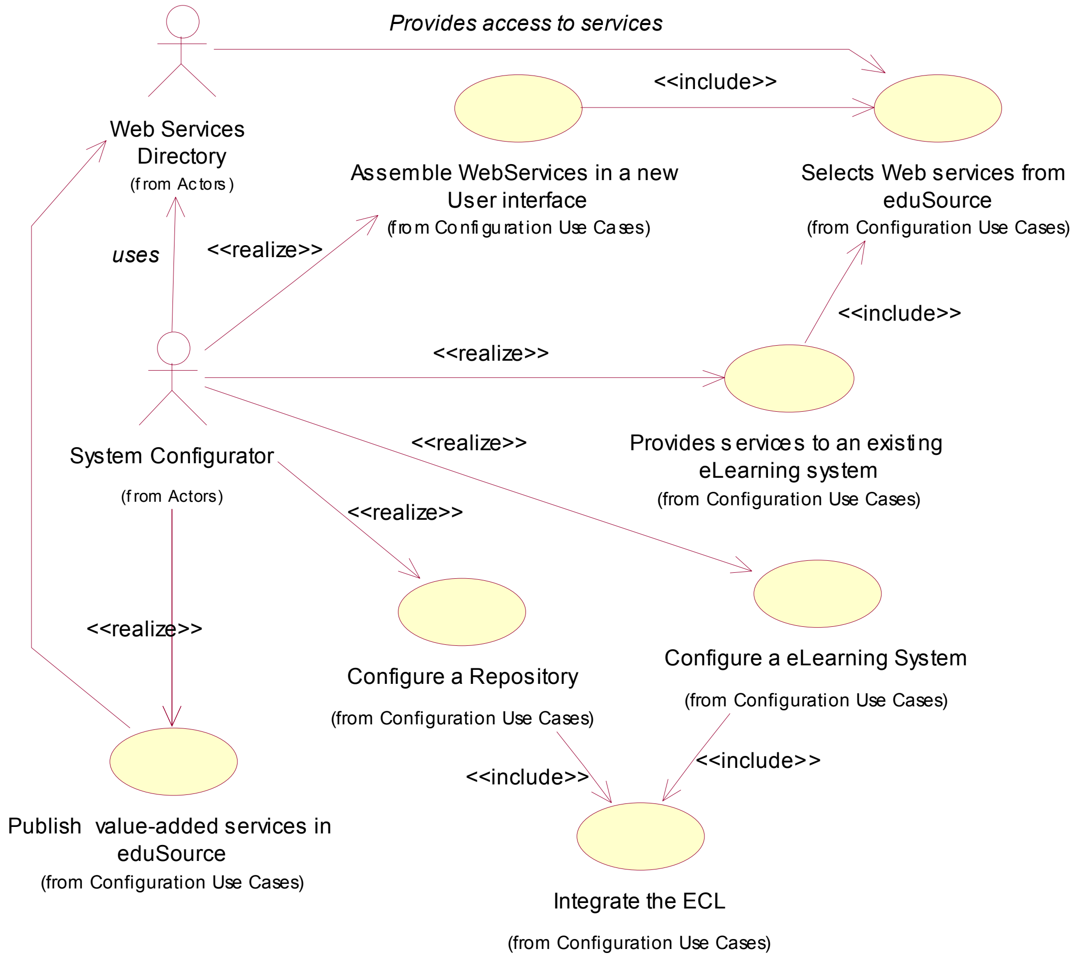

Provides access to services
<<include>>
Web Services Directory
(from Actors)
Assemble WebServices in a new User interface
(from Configuration Use Cases)
Selects Web services from eduSource
(from Configuration Use Cases)
uses
<<realize>>
<<include>>
<<realize>>
System Configurator
(from Actors)
<<realize>>
Provides services to an existing eLearning system
(from Configuration Use Cases)
<<realize>>
<<realize>>
Configure a Repository
(from Configuration Use Cases)
Configure a eLearning System
(from Configuration Use Cases)
<<include>>
<<include>>
Publish value-added services in eduSource
(from Configuration Use Cases)
Integrate the ECL
(from Configuration Use Cases)

Figure 19 - Use cases in the Communication Kernel

This package groups uses cases around a System Configurator. This actor uses Web services directories to publish value-added services or to select services to be integrated to an existing eLearning system or a new user interface. The System Configurator also integrated the ECL to a repository or to an eLearning system to facilitate the interoperation of communication and metadata protocols.

4.3 ELearning Middleware Services

The E-Learning Middleware Services groups all the functionalities enabling users to interact with elearning systems or agents whose providers agree to be referenced in eduSource. Figure 20 shows that services giving access to functionalities in Explor@, Splash, Careo, Aloha, or any other systems can be publishes in the eduSource Webservices registry. It also shows that these services can be integrated in an eduSource enabled eLearning system.

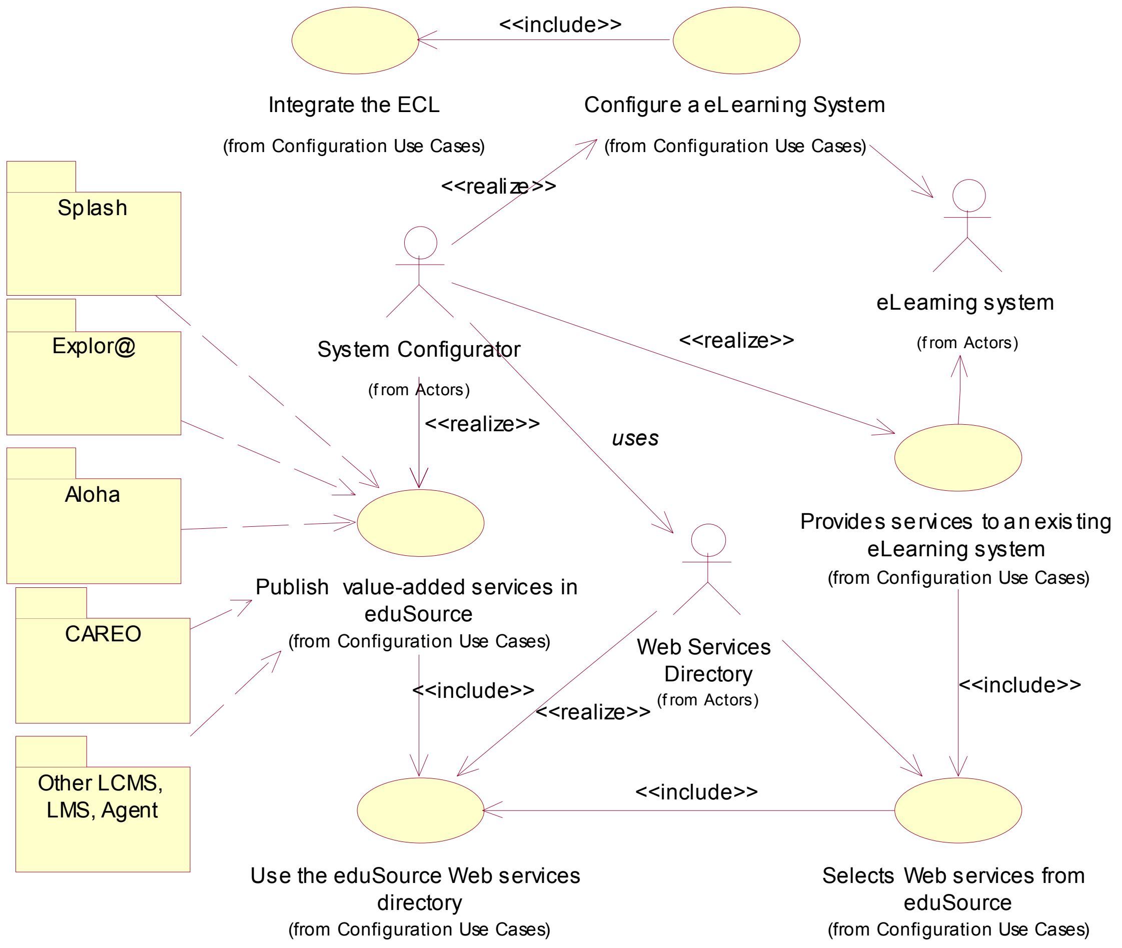

<<include>>
Integrate the ECL
(from Configuration Use Cases)
Configure a eLearning System
(from Configuration Use Cases)
<<realize>>
Splash
Explor@
Aloha
CAREO
Other LCMS, LMS, Agent
System Configurator
(from Actors)
eLearning system
(from Actors)
<<realize>>
<<realize>>
uses
Publish value-added services in eduSource
(from Configuration Use Cases)
Provides services to an existing eLearning system
(from Configuration Use Cases)
Web Services Directory
(from Actors)
<<include>>
<<realize>>
<<include>>
<<include>>
Use the eduSource Web services directory
(from Configuration Use Cases)
Selects Web services from eduSource
(from Configuration Use Cases)

Figure 20 - Use cases in the eLearning Middleware services

4.4 Metadata Repository Services

Metadata Repository Services is a package that implements the most essential functionalities to fully exploit a set of (partly redundant) metadata repositories. In particular, it will implement some of the IMS Digital Repository Interoperability specs for searching, harvesting, federating such as gather/expose and search/expose, and also peer-to-peer distributed search. It will also include translation services between metadata standards, specifications or profiles such as DC, LOM, CANCORE and MARC, and also between natural languages, starting with French-English translations of metadata.

Figure 21 shows the main Metadata repository services that enable eduSource users to build Metadata repositories and to search such repositories using one or more search methods.

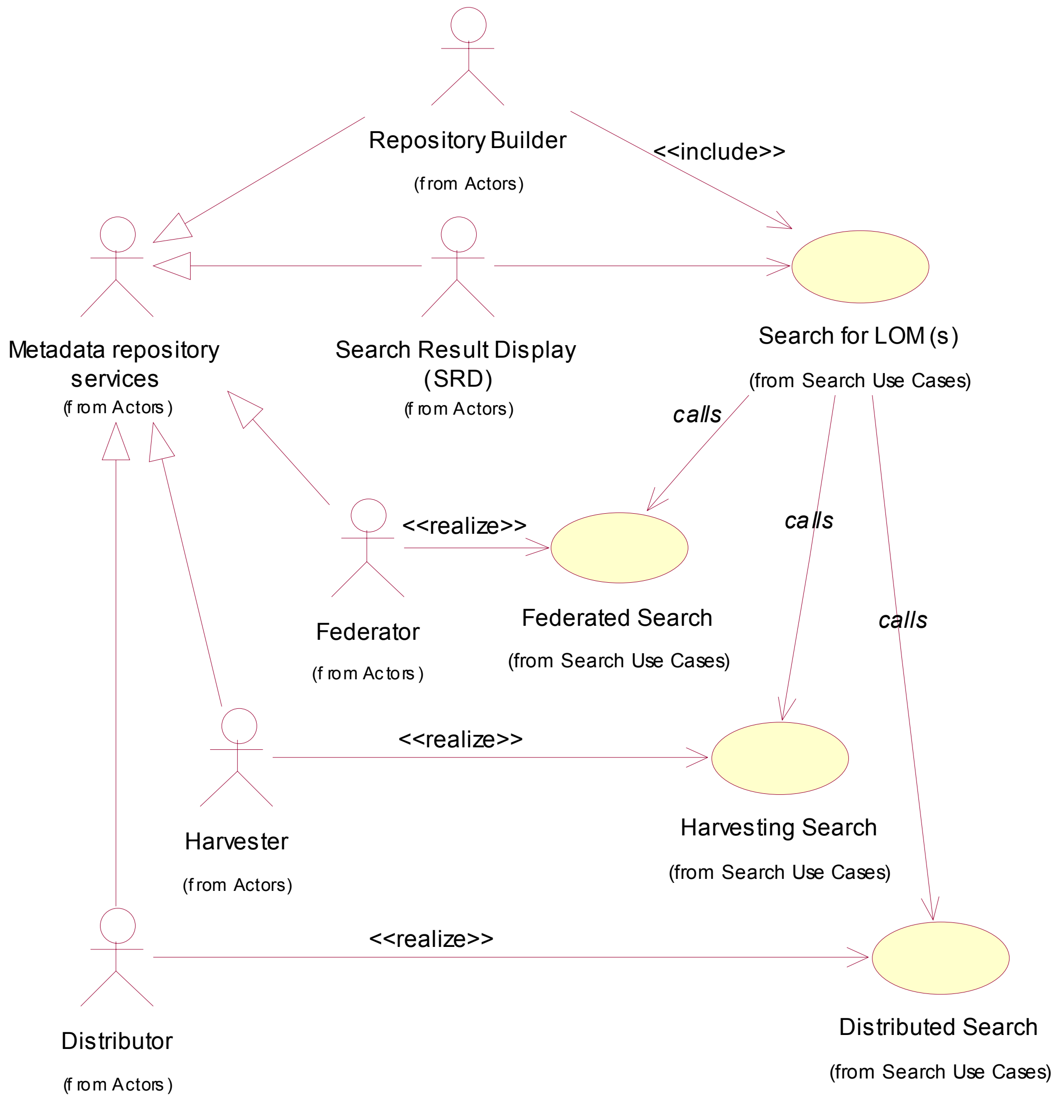


Figure 21 - Use cases in the Metadata Repository services

4.5 Resource Management Services

Resource Management Services is a package that takes care of operations needed to launch, aggregate, package, transport the actual resources or learning objects required by any other service or system.

Figure 22 centers these services on an actor called here a designer. Here, the designer retrieves or build a Learning Design and aggregates it with other learning object in the form of an IMS-LD content package or a SCORM package. To enable these operations, other resource management services such as IMS- DRI submit/store and request/deliver will have to be implemented.

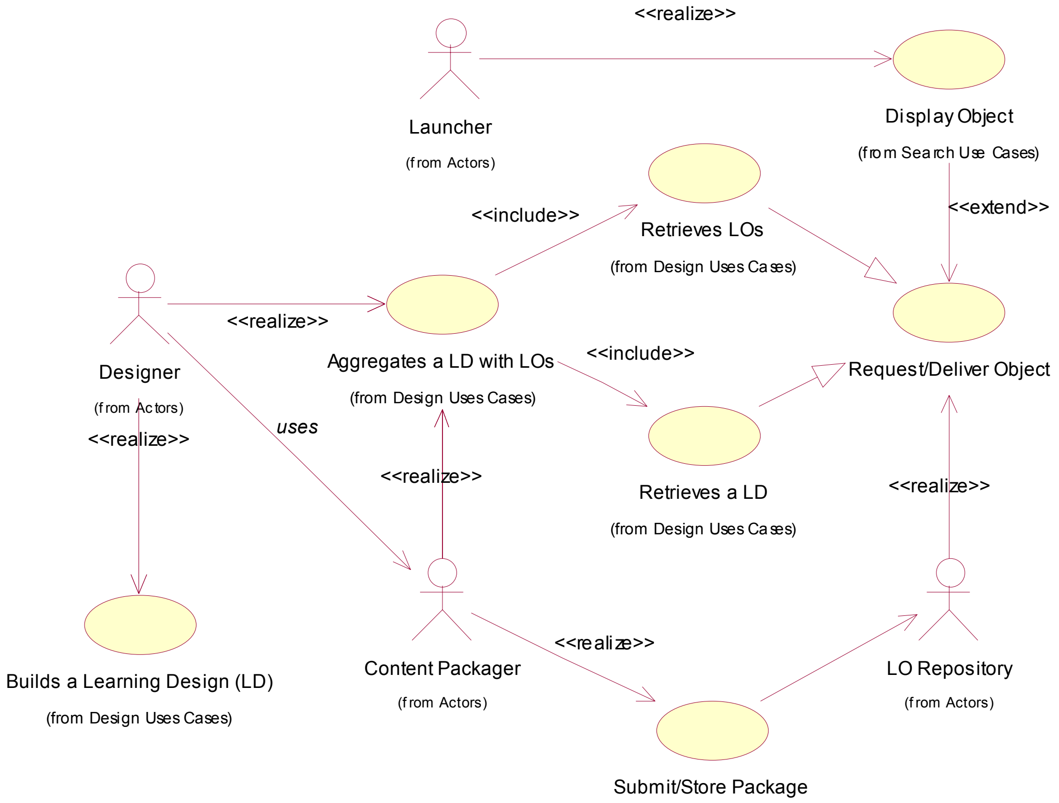


Figure 22 – Use cases in the Digitized Resources Repository services

4.6 Digital Rights Management Services

Digital Rights Management Services is a package grouping all the components for the management of interactions on digital rights and intellectual property between providers and users of resources and services. As shown on figure 22, it is centered on the interaction between a Publisher, a Provider Broker and a Purchaser Broker.

The Provider Broker enables a learning object Publisher to select a DRM model and produce the associated DRM metadata. The Purchaser Broker provides user identification, payment transactions and authorization to deliver the learning object. It will provide a simple encryption mechanism to secure transactions and adapt the LOM metadata for digital rights managements.

.

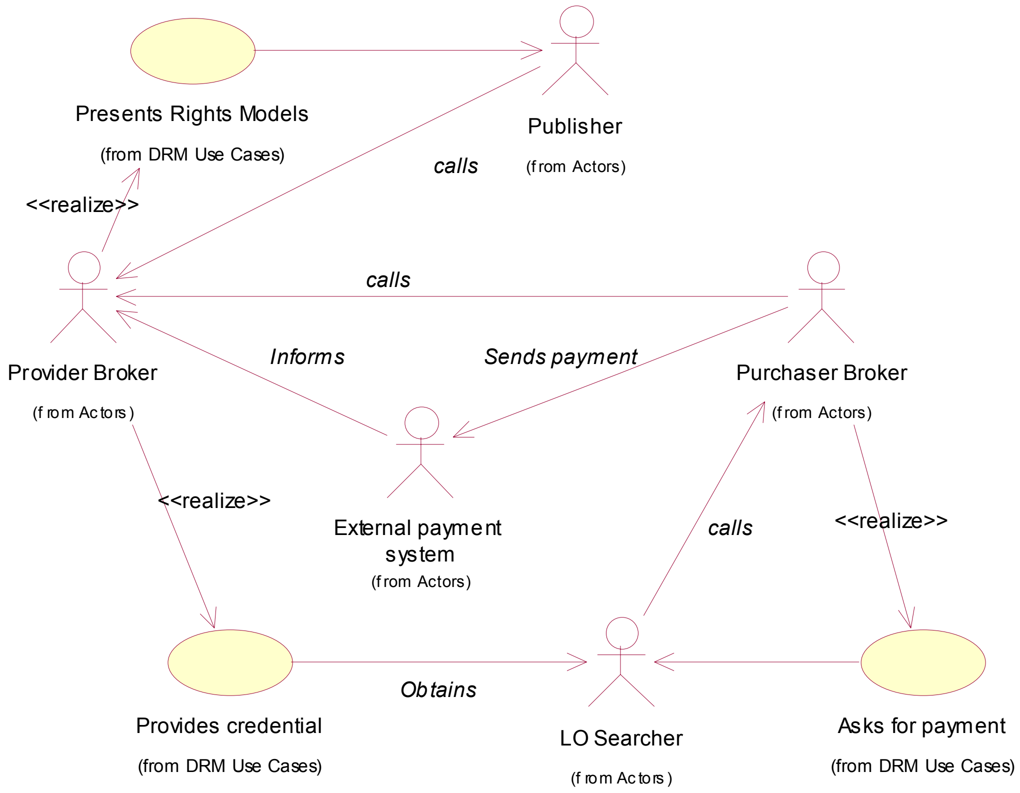


Figure 23 – Use cases in the DRM package

5. User Interface View

In this section, we provide general functional specification of the interface needed by the main actors in the system described in the preceding sections, that is the System Configurator, the Metadata Repository Builder, the Learning Objects Searcher, the Designer and the Provider.

We describe each interface from a user's point of view, present the main functions of the interface and sketch a view of a possible instantiation of the interface.

NB – This section will be completed when a first version of the Software Architecture document will be agreed upon.

5.1 The eduSource Configuration Interface

We center here on the System Configurator's roles.

5.1.1 Description

TBD

5.1.2 Main Functions

TBD

5.1.3 System Configurator's Interface

TBD

5.2 A Metadata Repository Builder Interface

We now center on the Metadata Repository Builder's roles.

5.2.1 Description

TBD

5.2.2 Main Functions

TBD

5.2.3 Repository Builder's Interface

TBD

5.3 A Search Result Display (SRD) Interface

We now center on the Learning Object Searcher's roles..

5.3.1 Description

TBD

5.3.2 Main Functions

TBD

5.3.3 LO Searcher's User interface

TBD

5.4 A Design and Package Interface

We now center on the Designer's roles.

5.4.1 Description

TBD

5.4.2 Main Functions

TBD

5.4.3 Designer's User interface

TBD

5.5 A Digital Rights Management Interface

We finally center on Provider's roles.

5.5.1 Description

TBD

5.5.2 Main Functions

TBD

5.5.3 Provider's User interface

TBD