“Diseño de repositorios digitales interoperables”

Autor: Antonio Sarasa Cabezuelo
Directora: María Antonia Huertas

Índice del trabajo

1. Introducción
2. Estado del arte
   2.1. Metainformación asociada al objeto de aprendizaje.
      2.1.1.IEEE Learning Object Metadata(LOM).
      2.1.2.Otras especificaciones.
   2.2. Búsqueda federada
      2.2.1.Simple Query Interface(SQI)
      2.2.2.Otras especificaciones.
   2.3. Publicación de objetos de aprendizaje.
      2.3.1.Simple Publishing Interface(SPI)
      2.3.2.Otras especificaciones.
   2.4. Harvesting de metadatos
      2.4.1.Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH).
      2.4.2.Otras especificaciones
   2.5. Lenguajes de consulta
      2.5.1.ProLearn Query Language(PLQL).
      2.5.2.Otras especificaciones.
   2.6. Modelos globales
      2.6.1.IMS Digital Repositories Interoperability(IMS DRI)
      2.6.2.Otras especificaciones.
3. Proyectos de implementación de repositorios digitales interoperables.
   3.1. Proyecto Elena.
   3.2. Proyecto Edusource.
   3.3. Proyecto Agrega.
   3.4. Otras fuentes de información.
4. La especificación “IMS Learning Object Discovery and Exchange”(IMS LODE)
5. Desarrollo de un caso de uso de IMS LODE para el contexto de un repositorio digital de material de lógica matemática.
6. Conclusiones.
   Bibliografía.
   Apendices
   Apendice A: Casos de uso
   Apendice B: IMS LODE Registry.
   Apendice C: IMS LODE CQL

1. Introducción.

Actualmente los recursos digitales de las instituciones se han empezado a guardar en grandes almacenes digitales que han recibido el nombre genérico de repositorios digitales[Dripps]. Estos repositorios ofrecen diferentes facilidades a los usuarios que quieren hacer uso de los mismos, principalmente centradas en la búsqueda y recuperación de los contenidos digitales [Polsani-2003]. Aunque estos repositorios podrían crearse de manera particular por cada institución, la realidad es que la tendencia preponderante en internet es la de colaborar y compartir. Así un escenario real de uso [Littlejohn-2003], consiste en la posibilidad de buscar y recuperar de diferentes repositorios que formen una federación, para lo cual es necesario haber normalizado una serie de características o métodos de implementación de servicios, formatos, metainformación,…, de manera que se haga factible la interoperabilidad entre los mismos.

El objetivo de este trabajo es realizar una revisión sobre el estado de la normalización de los diferentes aspectos que hacen factible la interoperabilidad entre repositorios de objetos de aprendizaje. Bajo estos supuestos, el presente trabajo se artícula de la siguiente manera. En la sección 2 se realiza un estado del arte sobre las principales especificaciones que se han definido para estandarizar determinados aspectos que influyen en la interoerabilidad entre repositorios de objetos de aprendizaje. En el capítulo 3, se hace una breve comparativa de repositorios desde el punto de vista de la interoperabilidad que ofrecen. A continuación en el capítulo 4, se presenta la especificación IMS LODE(IMS Learning Object Discovery and Exchange), la cual intenta dar respuesta de una manera única a todos los aspectos que influyen en la interoperabilidad entre repositorios, tomando como base las anteriores especificaciones. El capítulo 5, describe la aplicación de IMS LODE a un caso de estudio real contextualizado en la creación de un repositorio de objetos de aprendizaje de material de lógica matemática. Por último el capítulo 6, presenta un conjunto de conclusiones, y líneas de trabajo futuro.

2. Estado del arte

2.1. Metainformación asociada al objetode aprendizaje

Un servicio básico de un repositorio digital es la recuperación de objetos de aprendizaje de acuerdo a unos criterios de búsqueda. Para que estos criterios de búsqueda puedan ser usados es necesario poder compararlos con la metainformación disponible de los objetos de aprendizaje almacenados en el repositorio[Duval-2002]. En esta sección se describe la principal especificación de metadatos educativos.

2.1.1. IEEE Learning Object Metadata(LOM).

Es una especificación realizada por el grupo de trabajo número 12 del IEEE Learning Technology Standard Committee[LOM], en la que se definen un conjunto de metadatos que permitan describir recursos educativos digitales(learning objects) con la finalidad de facilitar la recuperación de los mismos. Los metadatos se estructuran jerárquicamente en base a nueve categorías principales:

- ***Categoría 1.General***: agrupa la información general que describe un objeto de aprendizaje en su conjunto.
- ***Categoría 2.Ciclo de vida***: describe la historia y estado actual de un objeto de aprendizaje, así como aquellas entidades que han intervenido en su creación y evaluación
- ***Categoría 3.Meta-metadatos***: describe el propio registro de metadatos. Describe como puede ser identificada una instancia de metadatos, quién la creó, cómo, cuándo y con qué referencias.
- ***Categoría 4.Técnica***: describe los requisitos y características técnicas del objeto de aprendizaje.
- ***Categoría 5.Uso Educativo***: describe las características educativas y pedagógicas fundamentales del objeto de aprendizaje. Concretamente, es la información didáctica esencial para aquellos agentes involucrados en una experiencia educativa de calidad. Algunos de estos agentes son estudiantes, profesores, tutores y administradores.
- ***Categoría 6.Derechos***: describe los derechos de propiedad intelectual y las condiciones de uso aplicables al objeto de aprendizaje.
- **Categoría 7.Relación:** describe las relaciones existentes, si las hubiese, entre un objeto de aprendizaje y otros. Para definir relaciones múltiples deben utilizarse varias instancias de esta categoría. Si existen varios objetos de aprendizaje con los cuales está relacionado, cada uno de ellos tendrá una instancia propia de esta categoría.
- **Categoría 8.Anotación:** proporciona comentarios sobre la utilización pedagógica del objeto de aprendizaje, e información sobre quién creó el comentario y cuando fue creado. Esta categoría permite a los educadores compartir sus valoraciones sobre el objeto de aprendizaje, recomendaciones para su utilización, etc.
- **Categoría 9.Clasificación:** describe dónde se sitúa el objeto de aprendizaje dentro de un sistema de clasificación concreto. Para definir múltiples clasificaciones, deben utilizarse múltiples instancias de esta categoría.

Cada una de las categorías está formada por una estructura jerárquica de metadatos que pueden ser de dos tipos: agregados o simples. Los metadatos agregados se caracterizan porque están formados a su vez por otros metadatos,

que podrían ser a su vez agregados o simples o de ambos tipos. Los metadatos simples son aquellos que realmente contienen información, son las hojas de la estructura jerárquica. La información la constituyen valores asignados al metadato correspondiente. Los valores pueden ser valores procedentes de vocabularios controlados o con un formato determinado o bien valores de texto libre. Además de la información que albergan los metadatos hay dos características más acerca de los metadatos, que es su obligatoriedad de uso y su cardinalidad. La cardinalidad o tamaño representa el número máximo de instancias que pueden utilizarse de un metadato. En este sentido algunos metadatos permiten que se use más de una instancia del mismo metadato para mejor describir al objeto de aprendizaje. Respecto a la obligatoriedad, la especificación determina que todos los metadatos son optativos, y son las instituciones que hagan uso de la especificación, las que deben establecer las restricciones sobre qué metadatos deben ser obligatorios y cuáles no. En la Figura 1 se puede ver una panorámica general de la especificación.

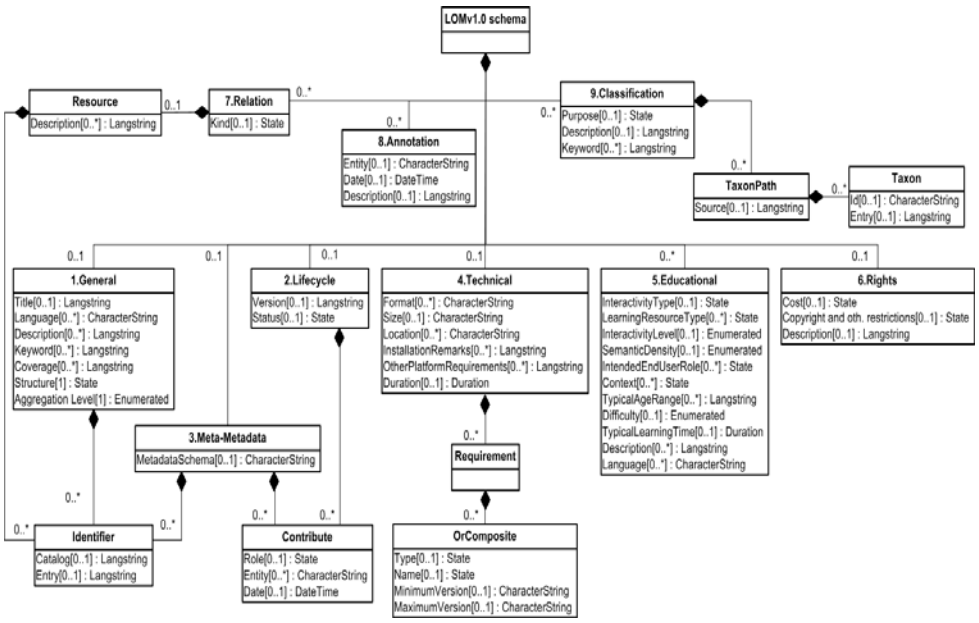

Figura 1.Representación esquemática de la jerarquía de metadatos de LOM

Los metadatos descritos en LOM tienen una orientación generalista que puede ser insuficiente para cubrir las necesidades particulares de metainformación para los recursos de un colectivo determinado. Para estas situaciones, la especificación proporciona un mecanismo de adaptación que se denomina perfil de aplicación. Un perfil de aplicación [2] es una adaptación de la especificación, la cual se implementa mediante una extensión del mismo conforme a [9]:

- Debe mantener los tipos de datos y espacios de valores de los elementos del esquema base de LOM.
- No puede definir nuevos tipos de datos ni espacios de valores para los elementos agregados de LOM.
- Se puede realizar una extensión que mantendría la conformidad de LOM(existen dos niveles de conformidad con respecto a LOM, conformidad estricta cuando solo se usa los elementos especificados en LOM, y conformidad cuando se usa algún tipo de extensión de LOM):

1. Extensión de elementos. Se trata de nuevos elementos, que deben definirse en un nuevo espacio de nombres, y pueden añadirse a cualquier elemento de LOM agregado (elemento compuesto o formado por otros elementos).
2. Extensión de atributos. Se pueden extender los atributos de elementos de LOM, siempre que estos se asocien a espacios de nombres distintos a los existentes.
3. Extensión de los vocabularios. Se pueden extender los vocabularios controlados en dos sentidos: a) extendiendo los vocabularios ya existentes, en cuyo caso hay que especificar la nueva fuente de los valores añadidos y b)nuevos vocabularios para nuevos elementos definidos en el perfil de aplicación.

Para facilitar el uso de la especificación, se ha definido un binding del modelo de metadatos de LOM a un lenguaje XML, de manera que los metadatos se corresponden con un lenguaje de etiquetas, y los metadatos asociados a un recurso se pueden representar como un documento XML. Esta representación tiene como ventaja mantener se forma separa la representación sintáctica de los metadatos de la semántica o significado de los mismos. En la Figura 2 se puede ver un fragmento de un documento XML con metadatos LOM.

```
<lom:lom>
  <lom:general>
    <lom:identifier>Fig00089</lom:identifier>
    <lom:title>
      <lom:langstring>Función de densidad de probabilidad Normal</lom:langstring>
    </lom:title>
    <lom:catalogentry>
      <lom:catalog>imágenes</lom:catalog>
      <lom:entry>
        <lom:langstring>Fig00089</lom:langstring>
      </lom:entry>
    </lom:catalogentry>
    <lom:language>es</lom:language>
    <lom:description>
      <lom:langstring>Gráfico de la función de densidad de probabilidad  de una normal.</lom:langstring>
    </lom:description>
    <lom:keyword>
      <lom:langstring>probabilidad</lom:langstring>
    </lom:keyword>
    <lom:keyword>
      <lom:langstring>normal</lom:langstring>
    </lom:keyword>
    <lom:coverage>
      <lom:langstring>1823</lom:langstring>
    </lom:coverage>
    <lom:structure>
      <lom:source>
        <lom:langstring>LOMv1.0</lom:langstring>
      </lom:source>
      <lom:value>
        <lom:langstring xml:lang="en">atomic</lom:langstring>
      </lom:value>
    </lom:structure>
    <lom:aggregationlevel>
      <lom:source>
        <lom:langstring>LOMv1.0</lom:langstring>
      </lom:source>
      <lom:value> <lom:langstring>1</lom:langstring>  </lom:value>
    </lom:aggregationlevel>
  </lom:general>
```

Figura 2.Fragmento de una instancia de metadatos LOM en formato XML

Desde el punto de vista del diseño, LOM cumple los siguientes principios:

- Modularidad, esto es, los metadatos, los vocabularios usados, así como los bloques de construcción pueden ser combinados según las necesidades, permitiendo la interoperabilidad tanto desde un punto de vista sintáctico como semántico.
- Extensibilidad, esto es, el esquema de metadatos debe permitir extensiones para captar nuevas necesidades de un entorno de aplicación determinado.
- Refinamiento, esto es, cada dominio de aplicación puede utilizar el esquema de metadatos a diferentes niveles de detalle, de acuerdo a sus necesidades.
- Multilingüismo, esto es, el esquema de metadatos debe respetar la realidad lingüística y la diversidad cultural de los posibles entornos de aplicación.

2.1.2. Otras especificaciones.

La principal especificación alternativa a LOM es el Dublin Core(DC) [Dublin Core], eleborada por el DCMI(Dublin Core Metadata Initiative), en la que se define un conjunto de metadatos de estructura plana, y que tiene como objetivo facilitar el descubrimiento de los recursos digitales a partir de la metainformación. Consta de 15 metadatos o categorías semánticas con nombres descriptivos que pretenden transmitir un significado semántico a los mismos. Pueden agruparse según el tipo de información que almacenan en tres tipos:

- Metadatos que describen el contenido del recurso:
  En este grupo se encuentran los siguientes metadatos:
  - Título: Es el nombre dado a un recurso.
  - Claves: Son los tópicos del recurso. Típicamente expresará las claves o frases que describen el título o el contenido del recurso. Se fomentará el uso de vocabularios controlados y de sistemas de clasificación formales.
  - Descripción: Es una descripción textual del recurso. Puede ser un resumen en el caso de un documento o una descripción del contenido en el caso de un documento visual.
  - Fuente: Es una secuencia de caracteres usados para identificar unívocamente un trabajo a partir del cual proviene el recurso actual.
  - Lengua: Es la lengua/s del contenido intelectual del recurso.
  - Relación: Es un identificador de un segundo recurso y su relación con el recurso actual. Este elemento permite enlazar los recursos relacionados y las descripciones de los recursos.
  - Cobertura: Es la característica de cobertura espacial y/o temporal del contenido intelectual del recurso. La cobertura espacial se refiere a una región física, uso de coordenadas o nombres de lugares extraidos de una lista controlada. La cobertura temporal se refiere al contenido del recurso, no a cuándo fue creado o puesto accesible.
- Metadatos que describen las características del recurso en cuanto a propiedad intelectual:
  - Autor o Creador: Es la persona o organización responsable de la creación del contenido intelectual del recurso.

- Editor: Es la entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual.
- Otros Colaboradores: Es una persona u organización que haya tenido una contribución intelectual significativa, pero que esta sea secundaria en comparación con las de las personas u organizaciones especificadas en el metadato creador.
- Derechos: Es una referencia en la que se describe información sobre derechos de autor del recurso con la finalidad de poder llevar a cabo la gestión de los derechos, y los términos y condiciones de acceso a un recurso.

▪ Metadatos referidos a la instanciación del recurso:

- Fecha: Es la fecha en la cual el recurso se puso a disposición del usuario en su forma actual. Esta fecha no se tiene que confundir con la descrita en el metadato cobertura, que hace referencia a una fecha relacionada con el contenido del recurso.
- Tipo del Recurso: Es la categoría del recurso.
- Formato: Es el formato de datos de un recurso, usado para identificar el software y, posiblemente, el hardware que se necesitaría para mostrar el recurso.
- Identificador del Recurso: Es una secuencia de caracteres utilizados para identificar unívocamente un recurso.

La especificación promueve que los valores que tomen los metadatos procedan de vocabularios controlados de forma que se favorezca la interoperabilidad. Todos los metadatos son optativos, pueden repetirse y pueden aparecen en cualquier orden. Dado el carácter tan genérico de los metadatos, puede que para determinados tipos de recursos sería interesante introducir una mayor especificidad en los metadatos. En este sentido Dublin Core permite emplear calificadores que aumenten la especificidad y precisión de los metadatos. Para no disminuir la compatibilidad con otras aplicaciones que usen Dublin Core sólo deben escogerse elementos de un conjunto de calificadores normado por la recomendación sobre calificadores DCMI (Dublin Core Qualifiers) donde se establecen 2 tipos de calificadores:

- Refinación de elementos: Estos calificadores hacen que el significado de un elemento sea más específico. Un elemento refinado comparte el significado del elemento no calificado, pero con un alcance más restrictivo. Las definiciones de términos para refinamiento de elementos deben estar públicamente disponibles.

- Esquema de codificación (scheme): Estos calificadores identifican esquemas que ayudan en la interpretación del valor de un elemento. Estos esquemas incluyen vocabularios controlados y notaciones formales o reglas de parseo. Un valor expresado usando un esquema de codificación, será entonces un símbolo (token) escogido de un vocabulario controlado (por ejemplo, un término de un sistema de clasificación) o una cadena (string) con formato de acuerdo con una notación formal . La descripción definitiva de un esquema de codificación para calificadores debe estar claramente identificada y disponible para uso público.

Para facilitar el uso de la especificación, se ha definido un binding del modelo de metadatos de Dublin Core a un lenguaje XML, de manera que los metadatos se

corresponden con un lenguaje de etiquetas, y los metadatos asociados a un recurso se pueden representar como un documento XML. En la Figura 3 se puede ver un fragmento de un documento XML con metadatos Dublin Core.

```
<metadata
  xmlns="http://example.org/myapp/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://example.org/myapp/ http://example.org/myapp/schema.xsd"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/">
  <dc:title>
    UKOLN
  </dc:title>
  <dcterms:alternative>
    UK Office for Library and Information Networking
  </dcterms:alternative>
  <dc:subject>
    national centre, network information support, library ,community, awareness, research,
information
    services,public library networking, bibliographic management, distributed library systems,
metadata,
    resource discovery, conferences,lectures, workshops
  </dc:subject>
  <dc:subject xsi:type="dcterms:DDC"> 062 </dc:subject>
  <dc:subject xsi:type="dcterms:UDC"> 061(410) </dc:subject>
  <dc:description>
    UKOLN is a national focus of expertise in digital information management. It provides policy,
research and
    awareness services to the UK library, information and cultural heritage communities.UKOLN is
based at
    the University of Bath.
  </dc:description>
  <dc:description xml:lang="fr">
    UKOLN est un centre national d'expertise dans la gestion de l'information digitale.
  </dc:description>
  <dc:publisher>
    UKOLN, University of Bath
  </dc:publisher>
  <dcterms:isPartOf xsi:type="dcterms:URI">
    http://www.bath.ac.uk/
  </dcterms:isPartOf>
  <dc:identifier xsi:type="dcterms:URI">
    http://www.ukoln.ac.uk/
  </dc:identifier>
  <dcterms:modified xsi:type="dcterms:W3CDTF"> 2001-07-18 </dcterms:modified>
  <dc:format xsi:type="dcterms:IMT"> text/html </dc:format>
  <dcterms:extent> 14 Kbytes </dcterms:extent>
</metadata>
```

Figura 3.Fragmento de una instancia de metadatos Dublin Core en formato XML

En la especificación de IEEE LOM se establece una correspondencia entre sus metadatos y los metadatos contemplados por Dublin Core, la cual se muestra en la figura 4.

DC.Identifier	General.Identificador.Entrada
DC.Title	General.Título
DC.Language	General.Idioma
DC.Description	General.Descripción
DC.Subject	General.Palabra Clave o Clasificación con Clasificación.Propósito igual a "disciplina" o "idea".
DC.Coverage	General.Ámbito
DC.Type	Uso Educativo.Tipo de Recurso Educativo
DC.Date	Ciclo de Vida.Contribución.Fecha cuando Ciclo de Vida.Contribución.Tipo tiene como valor "editor".
DC.Creator	Ciclo de Vida.Contribución.Entidad cuando Ciclo de Vida.Contribución.Tipo tiene como valor "autor".
DC.OtherContributor	Ciclo de Vida.Contribución.Entidad con el tipo de contribución especificada en Ciclo de Vida.Contribución.Tipo.
DC.Publisher	Ciclo de Vida.Contribución.Entidad cuando Ciclo de Vida.Contribución.Tipo tiene como valor "editor".
DC.Format	Técnica.Formato
DC.Rights	Derechos.Descripción
DC.Relation	Relación.Recurso.Descripción
DC.Source	Relación.Recurso cuando el valor de Relación.Tipo es "IsBasedOn".

Figura 4.Tabla de equivalencias entre Dublin Core y LOM

Actualmente ambas especificaciones están ampliamente extendidas. Sin embargo IEEE LOM es más usado para describir metainformación de recursos educativos digitales, y Dublin Core es usado de una forma más genérica para cualquier tipo de recurso. En el contexto de los repositorios digitales, se repite la situación, así si el repositorio tiene una marcada finalidad educativa, se prefiere IEEE LOM a Dublin Core, en cambio en repositorios de propósito más general se prefiere Dublin Core. También en los protocolos de recolección de metadatos de la red como OAI-PMH, se usa Dublin Core frente a IEEE LOM. Aunque es posible encontrar usos combinados de ambas especificaciones.

2.2. Búsqueda federada

La búsqueda federada sobre un conjunto de repositorios de objetos de aprendizaje, ofrece un gran valor al usuario final dado que le permite buscar sobre una vasta cantidad de objetos de aprendizaje. La arquitectura de una búsqueda federada consiste en[Broisin-2005]:

- Un buscador federado que ofrece una API para buscar, a través de la cual se envían consultas a la red de repositorios y se recopilan los resultados.
- Un registro que contiene todos las direcciones de los repositorios sobre las que el buscador puede realizar búsquedas. El buscador puede dinámicamente cargar en el registro información acerca de nuevos repositorios de objetos de aprendizaje.
- Los almacenes de metadatos en los cuales el buscador federado puede realizar búsquedas.
- Un cliente de búsqueda que es capaz de buscar en la federación a través del buscador federado.

En la figura siguiente se muestra un esquema de búsqueda federada usando los protocolos SQI[Simon et al., 2005] y OAI-PMH [Van de Sompel-2004]

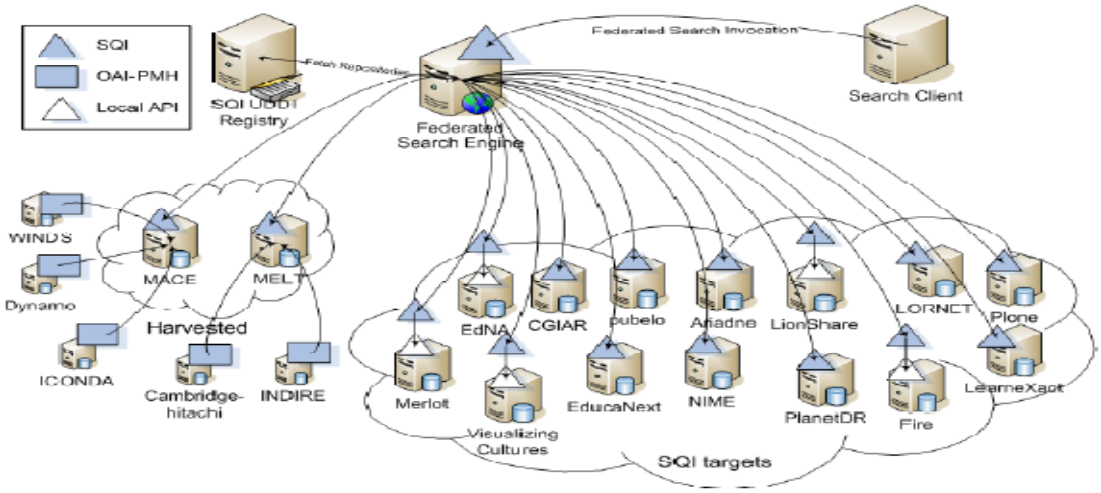

Figura 5.Búsqueda federada usando SQI y OAI-PM.

En esta sección se describe la principal especificación para implementar una búsqueda federada.

2.2.1. Simple Query Interface(SQI)

Es un protocolo creado por el CEN ISSS Workshop agreement (CWA) 15454:2005 [Simon et al., 2005] que proporciona interoperabilidad entre repositorios de objetos de aprendizaje y aplicaciones de búsqueda y recuperación de objetos de aprendizaje, estando diseñado para soportar una gran variedad de tecnologías de búsqueda. Para poder realizar una consulta con SQI a un repositorio de objetos de aprendizaje, en primer lugar la fuente de la consulta debe establecer una conexión con el destino via una sesión. Dentro de una sesión, los dos sistemas pueden comunicarse el uno con el otro. Las sesiones pueden ser persistentes, se mantienen indefinidamente, o no persistentes, se mantienen un periodo de tiempo limitado. En SQI, se permiten sesiones no persistentes, pudiendo ser destruidas manualmente o bien automáticamente si en un periodo de tiempo no se produce comunicación. Una vez que una sesión ha sido establecida, la interface

de consultas en el destino espera que sea enviada la petición de búsqueda. Los principales métodos para gestionar las sesiones son:

a) Create Session

Permite crear una sesión asociada a un usuario, para lo cual requiere del usuario y del password como parámetros, y retorna un identificador de sesión si el usuario es autenticado. Al usar este método se pueden producir algunos errores:

- "WRONG_CREDENTIALS" si se envía un userID o un password invalidos.
- "METHOD_FAILURE" es lanzado en caso de hay algo que no funciona correctamente en el uso del método.

Nombre Método	createSession
Tipo Retornado	String
Parámetros	String Parameters String userID String password
Fallos	WRONG_CREDENTIALS METHOD_FAILURE

Tabla 1.Método “Create Session”

b) Create Anonymous Session

Permite crear una sesión no asociada a un usuario, y puede ser usado en un sistema que permite a todo las personas crear consultas. Al usar este método se pueden producir algunos errores:

- "METHOD_FAILURE" es lanzado en caso de hay algo que no funciona correctamente en el uso del método.

Nombre Método	createAnonymousSession
Tipo Retornado	String
Parámetros	String Parameters String userID String password
Fallos	METHOD_FAILURE

Tabla 2.Método “Create Anonymous Session”

c) Destroy Session

Permite a la fuente que inició la sesión finalizar la sesión, para lo que es necesario eliminar el identificador de sesión sessionID. Al usar este método se pueden producir algunos errores:

- “NO_SUCH_SESSION" si la sesión a eliminar no existe.
- "METHOD_FAILURE" es lanzado en caso de hay algo que no funciona correctamente en el uso del método.

Nombre Método	destroySession
Tipo Retornado	void
Parámetros	String sessionID
Fallos	NO_SUCH_SESSION METHOD_FAILURE

Tabla 3.Método “Destroy Session”

La interface de consultas del destino puede ser configurada. Las configuraciones se mantienen válidas durante la sesión o hasta que sean configuradas de otra

forma diferente, y en caso de no configurarse, entonces se usa el comportamiento por defecto. Los métodos para configurar la interface son:

a) Set Query Language

Permite a la fuente controlar la sintaxis usada en las instrucciones de la consulta por identificación del lenguaje de consulta. Los valores para el parámetro queryLanguageID con case-sensitivos. Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- QUERY_LANGUAGE_NOT_SUPPORTED si el lenguaje de consultas usado en la petición no está soportado por el destino.
- "METHOD_FAILURE" si la operación falla por otras razones.

Nombre Método	setQueryLanguage
Tipo Retornado	void
Parámetros	String targetSessionID String queryLanguageID
Fallos	NO_SUCH_SESSION QUERY_LANGUAGE_NOT_SUPPORTED METHOD_FAILURE

Tabla 4.Método "Set Query Language"

b) Set Maximum Number of Query Results.

Define el número máximo que una consulta puede producir, especificándose como un porcentaje. El parámetro maxQueryResults debe ser cero o más grande, de manera que si se configura a cero, se entiende que la fuente no límita el número de resultados devueltos.Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- INVALID_MAX_QUERY_RESULTS si un número invalido es proporcionado como valor del parámetro maxQueryResults. Esto puede suceder con números negativos o bien con valores que superan el límite de resultados que es capaz de devolver una fuente.
- "METHOD_FAILURE" si la operación falla por otras razones.

Nombre Método	setMaxQueryResults
Tipo Retornado	void
Parámetros	String targetSessionID Integer maxQueryResults
Fallos	NO_SUCH_SESSION INVALID_MAX_QUERY_RESULTS METHOD_FAILURE

Tabla 5.Método "Set Maximun Number of Query Results"

c) Set Maximum Duration

Permite a la fuente configurar un tiempo de espera en caso de que la consulta opera en una interface de consultas asíncronas. El parámetro que controla el tiempo de espera se expresa en milisegundos. Cuando un destino recibe una consulta asíncrona, puede retornar los resultados por un máximo de tiempo expresado en los milisegundos del parámetro maxDuration. Después de pasado este tiempo, una destino no puede usar el método queryRestultsListener. El valor del parámetro maxDuration debe ser cero o

más grande. Si toma el valor de cero, se entiende que la fuente delega la gestión del tiempo de espera en el destino. El valor por defecto es cero.Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- INVALID_MAX_DURATION si un número invalid es proporcionado para el parámetro maxDuration.
- "METHOD_FAILURE" si la operación falla por otras razones.

Nombre Método	setMaxDuration
Tipo Retornado	void
Parámetros	String targetSessionID Integer maxDuration
Fallos	NO_SUCH_SESSION INVALID_MAX_DURATION METHOD_FAILURE

Tabla 6.Método "Set Maximum Duration"

d) Set Results Format

Permite a la fuente controlar el formato de los resultados retornados por el destino. El formato es especificado en el parámetro resultsFormat en forma de URI. Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- RESULTS_FORMAT_NOT_SUPPORTED cuando el format proporcionado en le parámetro resultsFormat no es soportador por el destino.
- "METHOD_FAILURE" si la operación falla por otras razones.

Nombre Método	setResultsFormat
Tipo Retornado	void
Parámetros	String targetSessionID String resultsFormat
Fallos	NO_SUCH_SESSION RESULTS_FORMAT_NOT_SUPPORTED METHOD_FAILURE

Tabla 7.Método "Set Results Format"

Las consultas pueden ser enviadas usando dos tipos de métodos:

A- Métodos síncronos

Se trata de un conjunto de métodos que sólo pueden ser usados en modo de consulta síncrona. Cuando un destino sólo soporta consultas asincrónas, estos métodos deben indicarlo con un fallo de tipo QUERY_MODE_NOT_SUPPORTED. En modo síncrono, los resultados de la consulta son retornados directamente por el método synchronousQuery. La cantidad de resultados retornados por invocación este método es configurado con el método setResultSetSize. Por otro lado, el método getTotalResultsCount retorna el número total de ejemplares de metadatos que encajan con la búsqueda.

a) Set Results Set Size .

Define el máximo número de resultados, que por defecto son 25. Si toma el valor de cero, entonces se entiende que quiere recuperar todos los resultados.Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- INVALID_RESULTS_SET_SIZE si un número invalido es provisto para el parámetro resultsSetSize

- QUERY_MODE_NOT_SUPPORTED en caso de que el destino no soporte consultas síncronas.
- "METHOD_FAILURE" si la operación falla por otras razones.

Nombre Método	setResultsSetSize
Tipo Retornado	void
Parámetros	String targetSessionID Integer resultsSetSize
Fallos	NO_SUCH_SESSION INVALID_RESULTS_SET_SIZE QUERY_MODE_NOT_SUPPORTED METHOD_FAILURE

Tabla 8.Método "Set Results Set Size"

b) Synchronous Query

Envía una consulta al destino a través del parámetro queryStatement. Dentro de una sesión identificada por el targetSessionID, multiples consultas pueden ser enviadas simultáneamente por invocación repetida de este método. El método retorna un conjunto de instancias de metadatos acordes con la consulta realizada. El parámetro startResults identifica la instancia de comienzo del conjunto de resultados,cuyo valor puede variar entre 1 y el número total de resultados. El índice del conjunto de resultados comienza en 1, y el número de resultados devueltos está controlado por setResultsSetSize y no puede ser mayor que el valor de setMaxQueryResults.Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- QUERY_MODE_NOT_SUPPORTED en caso de que el destino no soporte consultas síncronas.
- INVALID_QUERY_STATEMENT si las instrucciones de la consulta no cumplen con la síntaxis del lenguaje de consultas.
- INVALID_START_RESULT si un número invalid es proporcionado por startResult.
- NO_MORE_RESULTS si startResult está configurado a cero y ningún resultado más es proporcionado.
- "METHOD_FAILURE" si la operación falla por otras razones.

Nombre Método	synchronousQuery
Tipo Retornado	String
Parámetros	String targetSessionID String queryStatement String startResult
Fallos	NO_SUCH_SESSION QUERY_MODE_NOT_SUPPORTED INVALID_QUERY_STATEMENT INVALID_START_RESULT NO_MORE_RESULTS METHOD_FAILURE

Tabla 9.Método "Synchronous Query"

c) Get Total Results Count

Retorna el número total de resultados existentes de una consulta. Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- QUERY_MODE_NOT_SUPPORTED en caso de que el destino no soporte consultas síncronas.
- INVALID_QUERY_STATEMENT si las instrucciones de la consulta no cumplen con la síntaxis del lenguaje de consultas.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	getTotalResultsCount
Tipo Retornado	Integer
Parámetros	String targetSessionID String queryStatement
Fallos	NO_SUCH_SESSION QUERY_MODE_NOT_SUPPORTED INVALID_QUERY_STATEMENT METHOD_FAILURE

Tabla 10.Método "Get Total Results Count"

B- Métodos asíncronos.

En caso de que la interface de consultas opere de manera asíncrona, el método queryResultsListener es llamado por el destino para dirigir los resultados de la consulta a la fuente.

a) Set Source Location.

Este método es llamado antes de una consulta sea enviada en modo asíncrono. El parámetro sourceLocation especifica la localización de fuente que espera los resultados, de manera que el destino pueda enviar los resultados. El sourceLocation es una cadena que puede resolver la localización de la queryResultsListener.Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- QUERY_MODE_NOT_SUPPORTED en caso de que el destino no soporte consultas asíncronas.
- INVALID_SOURCE_LOCATION si la localización no puede ser resuelta.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	setSourceLocation
Tipo Retornado	void
Parámetros	String targetSessionID String sourceLocation
Fallos	NO_SUCH_SESSION QUERY_MODE_NOT_SUPPORTED INVALID_SOURCE_LOCATION METHOD_FAILURE

Tabla 11.Método "Set Source Location"

b) Asynchronous Query

Permite a la fuente enviar una consulta al destino, mientras los resultados son retornados de manera asíncrona. La consulta es proporcionada a través del parámetro queryStatement. Un queryID proporcionado por la fuente es requerido con el fin de enlazar los resultados de la consulta con la consulta cuando son retornados. Usando queryIDs únicos es possible enviar un número arbitrario de consultas en cada sesión activa. La localización de la

fuente que espera los resultados es necesario y debe haber sido proporcionado antes de usar el método setSourceLocation. Debido a la naturaleza asíncrona del método, los resultados de la consulta podrían llegar desde consultas previas. La consulta es procesada y los resultados son enviados dentro de la franja de tiempo especificada en el método setMaxDuration. Al usar este método se pueden producir algunos errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID sea inválido.
- QUERY_MODE_NOT_SUPPORTED en caso de que el destino no soporte consultas asíncronas.
- NO_SOURCE_LOCATION en caso de que ninguna localización haya sido específicada ante de enviar la consulta en la sesión.
- INVALID_QUERY_STATEMENT si las instrucciones de la consulta no cumplen con la síntaxis del lenguaje de consultas.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	asynchronousQuery
Tipo Retornado	void
Parámetros	String targetSessionID String queryStatement String queryID
Fallos	NO_SUCH_SESSION QUERY_MODE_NOT_SUPPORTED NO_SOURCE_LOCATION INVALID_QUERY_STATEMENT METHOD_FAILURE

Tabla 11.Método "Asynchronous Query"

c) ***Query Results Listener.***

Es un método iniciado por el destino que tiene como objetivo dirigir los resultados a la fuente. El parámetro queryID es usado para enlazar los resultados de la consulta a consulta previamente enviada. El parámetro queryResults genera un conjunto de resultados consistentes en una lista de instancias de metadatos, los cuales están formateados de acuerdo al esquema específicado por el método setResultsFormat. Al usar este método se pueden producir algunos errores:

- INVALID_QUERY_RESULTS en cado de que el conjunto de resultados no puedan ser interpretados por la fuente.
- NO_SUCH_QUERY en caso de que el queryID sea invalid.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	queryResultsListener
Tipo Retornado	void
Parámetros	String queryID String queryResults
Fallos	INVALID_QUERY_RESULTS NO_SUCH_QUERY METHOD_FAILURE

Tabla 12.Método "Query Results Listener"

Observar que todos los métodos SQI están especificados de manera que hacen abstracción de la tecnología concreta usada para implementarlos.

2.2.2. Otras especificaciones

Existen otros protocolos que permiten implementar búsquedas federadas diferentes a SQI. A continuación se enumeran otros protocolos alternativos, y en la tabla se comparan:

- **SRU/W**[McCallum-2006]. La librería del Congreso mantiene dos protocolos de búsqueda. El Search/Retrieve via URL (SRU), que es un protocol estilo REST que codifica un método de búsqueda y parámetro como una URI y retorna una instancia XML como resultado de la búsqueda. Y el Search/Retrieve Web Service (SRW) que implementa el mismo protocol pero en una implementación SOAP.
- ***El EduSource Communication Layer (ECL)*** [Hatala et al-2004a] [Eap-2003] [Eap et al-2004] implementa el IMS DRI [IMS-2003]. Introduce 4 funciones para los repositorios digitales: consultar (Buscar/Exponer), publicar (Enviar/Almacenar), adquirir contenido (Solicitar/desplegar) y recolectar (recopilar/exponer).que propone publicar un mensaje para enviar objetos de aprendizaje o metadatos a un repositorio. Como respuesta el repositorio envía un petición de publicación. Por otro lado, las consultas deben ser expresadas en un formato XML común que puede ser mapeado a múltiples lenguajes de consulta. Se puede implementar tanto interaccion con estado o sin estado.
- ***El Open Knowledge Initiative (O.K.I)*** [Hatala et al-2004b] especifica la Open Service Interface Definitions (OSID) en la que se definen servicios tales como autenticación, autorización, repositorio, planificación, gestión de cursos. A diferencia del resto de iniciativas, las búsquedas no se basan en APIS, y usan interfaces que pueden tener diferentes implementaciones, haciendo la plataforma OSIDs dependiente. Es decir que para repositorio, OSIDs debe ser implementado en un específico lenguaje de programación. Se puede implementar tanto interaccion con estado o sin estado.

	SQI	SRW/SRU	ECL	OKI
Sincrono	Sí	Sí	Sí	Sí
Asíncrono	Sí	No	No	No
Soporte para multiples lenguajes de consulta	Sí	No	No	Sí
Soporte para múltiples esquemas de metadatos	Sí	Sí	No	Sí
Soporte para múltiples bindings	Sí, SOAP, REST, JMS, JAVA,…	Sí, pero límitado a SOAP, REST	No	Sí, JAVA y PHP
Modelo semántico abstracto	Sí	No	Sí, IMS DRI	Sí
Plataforma independiente	SOAP, sí JMS, no	Sí	Sí	No
Estado	Sí	Sí	Sí	--
Sin estado	Sí	Sí	Sí	--

Tabla 13.Tabla comparativa de SQI con otras especificaciones

2.3. Publicación de objetos de aprendizaje.

Un repositorio digital debería ofrecer un mecanismo transparente para poder publicar los objetos de aprendizaje junto a sus metadatos. Existen dos maneras de publicar, en modo por valor, en el que se envía el propio objeto de aprendizaje al repositorio, o bien en modo por referencia, en el que objeto de aprendizaje se encuentra almacenado en otro lugar, y se envía es una referencia a dicho almacen. Por otro lado, es posible encontrarse diferentes escenarios de publicación en función del tipo de almacenes que se encuentren en el repositorio digital[Manepalli et al., 2006]:

- Sólo es posible publicar metadatos.
- Sólo es posible publicar objetos de aprendizaje sin sus metadatos.
- Se puede publicar tanto objetos de aprendizaje como metadatos, debiendo conectar ambos.

En esta sección se describe brevemente el protocolo Simple Publishing Interface(SPI), uno de los principales protocolos usado para publicar objetos de aprendizaje, así como instancias de metadatos en un repositorio digital.

2.3.1. Simple Publishing Interface(SPI)

El Simple Publishing Interface (SPI) [Ternier and Duval, 2003]. es un protocolo que permite publicar objetos de aprendizaje y metadatos en un repositorio de objetos de aprendizaje. Para ello se define una API formada por un conjuntos de métodos independientes de la tecnología lo que facilita la interoperabilidad semántica de las implementaciones de los mismos.Todos los métodos que ofrece la API requieren la creación de una sesión en el destino en el que se va a publicar el objeto de aprendizaje. Así mismo la fuente que publica el objeto, siempre usa un parámetro targetSessionID para identificar la sesión. Los métodos pueden ser de dos tipos:

A-Orientados a la publicación de objetos de aprendizaje

d) Método set Data Format

Permite indicar el tipo de objetos que se va a almacenar en un repositorio. Dispone del paramétro setDataFormatID, que establece el formato de datos para objetos compuestos como los paquetes SCORM, AICC o IMS-QTI. Para objetos no compuesto, el parámetro indica el tipo de archivado y el formato de comprensión usado en el objeto. Cuando se usa éste método se pueden producir 3 tipos de fallos:

- NO_SUCH_SESSION si el TargetSessionID es invalido.
- DATA_FORMAT_NOT_SUPPORTED si el formato usado en la petición no es soportado.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	setDataFormat
Tipo Retornado	void
Parámetros	String targetSessionID String setdataFormatID
Fallos	NO_SUCH_SESSION DATA_FORMAT_NOT_SUPPORTED METHOD_FAILURE

Tabla 14 .Método setDataFormat

e) Método Set Source Location

Se usa antes de que un objeto de aprendizaje sea enviado en modo "por referencia". Dispone del parámetro sourceLocation que permite resolver la localización de la fuente del método notifyRetrievalStatus , y así enviar un mensaje de reconocimiento. Cuándo se usa este método se pueden producir 4 tipos de errores:

- NO_SUCH_SESSION en caso de que el targetSessionID sea inválido.
- SUBMISSION_MODE_NOT_SUPPORTED si no se soportan envíos en modo "por referencia"
- INVALID_SOURCE_LOCATION si la localización es incorrecta o no puede ser parseado.
- METHOD_FAILURE si la operación falla por otra razón.

Nombre Método	setSourceLocation
Tipo Retornado	void
Parámetros	String targetSessionID String sourceLocation
Fallos	NO_SUCH_SESSION SUBMISSION_MODE_NOT_SUPPORTED INVALID_SOURCE_LOCATION METHOD_FAILURE

Tabla 15 .Método setSourceLocation

f) Método Submit Resource.

Este método permite publicar un objeto de aprendizaje en el repositorio mediante dos modos diferentes, bien por referencia o bien por valor. El identificador que se asocia al objeto que se publica puede ser generado por la fuente o por el destino, existiendo así dos métodos diferentes por cada tipo de modo de publicación, uno en el que se especifica como parámetro el identificador y otro que no. Cuándo se usa este método se pueden producir algunos de los siguientes tipos de errores:

- NO_SUCH_SESSION en caso de que el TargetSessionID es inválido.
- METHOD_NOT_SUPPORTED si el método de publicación no es soportado por el destino.
- INSUFFICIENT_CREDENTIALS si el usuario no es autorizado a ejecutar este método.
- INVALID_RESOURCE_IDENTIFIER cuando el identificador provisto por la fuente ya existe o tiene una estructura no soportada.
- NO_SOURCE_LOCATION en caso de que ninguna localización para la fuente haya sido especificada antes de publicar el recurso.
- METHOD_FAILURE si la operación falla por alguna otra razón.

Nombre Método	submitResourceByValue	submitResourceByValue
Tipo Retornado	void	String
Parámetros	String targetSessionID binaryData data String Identifier	String targetSessionID binaryData data
Fallos	NO_SUCH_SESSION METHOD_NOT_SUPPORTED INSUFFICIENT_CREDENTIALS INVALID_RESOURCE_ IDENTIFIER METHOD_FAILURE	NO_SUCH_SESSION METHOD_NOT_SUPPORTED INSUFFICIENT_CREDENTIALS METHOD_FAILURE

Tabla 16 .Método submitResourceByValue

Nombre Método	submitResourceByReference	submitResourceByReference
Tipo Retornado	void	String
Parámetros	String targetSessionID String reference String identifier	String targetSessionID String reference
Fallos	NO_SUCH_SESSION NO_SOURCE_LOCATION METHOD_NOT_SUPPORTED INSUFFICIENT_CREDENTIALS INVALID_RESOURCE_IDENTIFIER METHOD_FAILURE	NO_SUCH_SESSION NO_SOURCE_LOCATION METHOD_NOT_SUPPORTED INSUFFICIENT_CREDENTIALS METHOD_FAILURE

Tabla 17.Método submitResourceByReference

g) Método Notify Retrieval Status.

Después de invocar un método submitResourceByReference, la fuente espera asíncronamente una notificación de transmisión completada. Así el destino usa el método notifyRetrievalStatus para informar a la fuente del estado de la transferencia. Después de completarse la transferencia, se debe enviar una notificación con estado "COMPLETO" a la fuente. Si el mensaje no puede ser entregado, entonces el destino debería borrar el recurso para el cual envió la notificación. Cuándo se usa este método se pueden producir 2 tipos de errores:

- NO_SUCH_IDENTIFIER si un identificador invalido es enviado.
- METHOD_FAILURE si la operación falla por cualquier otra razón.

Nombre Método	notifyRetrievalStatus
Tipo Retornado	void
Parámetros	String resourceIdentifier String status
Fallos	NO_SUCH_IDENTIFIER METHOD_FAILURE

Tabla 18.Método notifyRetrievalStatus

h) Método Delete Resource

Este método permite borrar un objeto de aprendizaje del destino. El borrado de un recurso no tiene un efecto cascada sobre las instancias de metadatos que están asociadas al recurso. Por tanto la fuente debe explicitar el borrado de los metadatos asociados.Cuándo se usa este método se pueden producir 7 tipos de errores:

- NO_SUCH_SESSION en caso de que el targetSessionID sea inválido.
- INVALID_RESOURCE_IDENTIFIER si el identificador tiene una estructura inválida.
- RESOURCE_DOES_NOT_EXIST si el objeto de aprendizaje asociado con el identificador no existe.
- INSUFFICIENT_CREDENTIALS si el usuario no autoriza el borrado del objeto de aprendizaje.
- DELETION_NOT_ALLOWED. El destino no permite borrar objetos de aprendizaje después de que ellos son publicados.
- METADATA_ASSOCIATED. El objeto de aprendizaje está asociado a instancias de metadatos. Disocia primero metadatos y objetos de aprendizaje.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	deleteResource
Tipo Retornado	void
Parámetros	String targetSessionID String resourceIdentifier
Fallos	NO_SUCH_SESSION INVALID_RESOURCE_IDENTIFIER RESOURCE_DOES_NOT_EXIST INSUFFICIENT_CREDENTIALS DELETION_NOT_ALLOWED METADATA_ASSOCIATED METHOD_FAILURE

Tabla 19 .Método deleteResource

B-Orientados a la publicación de metadatos

a) Método Set Metadata Schema

Permite a la fuente controlar el esquema de los metadatos que será publicado en el destino,y al destino le permite validar las instancias de metadatos. El parámetro metadataSchema debería ser una URI que identifique al esquema de metadatos. Cuando se usa éste método se pueden producir 3 tipos de fallos:

- NO_SUCH_SESSION si el TargetSessionID es invalido.
- SCHEMA_NOT_SUPPORTED cuando el esquema provisto por el parámetro metadataSchema no es soportado por el destino.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	setMetadataSchema
Tipo Retornado	void
Parámetros	String targetSessionID String metadataSchema
Fallos	NO_SUCH_SESSION SCHEMA_NOT_SUPPORTED METHOD_FAILURE

Tabla 20.Método setMetadataSchema

b) Método Submit Metadata

Permite enviar para publicar una instancia de metadatos. Tanto el destino como la fuente pueden generar un identificador para la instancia de metadatos. Después del envío, el destino debe almacenar este identificador con los metadatos enviados. Cuando se usa éste método se pueden producir 6 tipos de fallos:

- NO_SUCH_SESSION si el TargetSessionID es invalido.
- METHOD_NOT_SUPPORTED cuando el destino no soporta la creación de identificadores o viceversa
- INVALID_METADATA_IDENTIFIER cuando el identificador provisto por la fuente no es válido.
- INSUFFICIENT_CREDENTIALS si el usuario no está autorizado para publicar metadatos.
- VALIDATION_FAILURE cuando los metadatos enviados a publicar no validan contra el esquema de metadatos establecido.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	submitMetadataRecord	submitMetadataRecord
Tipo Retornado	void	String
Parámetros	String targetSessionID String metadataRecord String identifier	String targetSessionID String metadataRecord
Fallos	NO_SUCH_SESSION METHOD_NOT_SUPPORTED INVALID_METADATA_IDENTIFIER INSUFFICIENT_CREDENTIALS VALIDATION_FAILURE METHOD_FAILURE	NO_SUCH_SESSION METHOD_NOT_SUPPORTED INSUFFICIENT_CREDENTIALS VALIDATION_FAILURE METHOD_FAILURE

Tabla 21.Método submitMetadataRecord

c) Método Delete Metadata

Permite borrar instancias de metadatos del repositorio en el que se almacenan. Cuando se usa éste método se pueden producir 8 tipos de fallos:

- NO_SUCH_SESSION si el TargetSessionID es invalido.
- METHOD_NOT_SUPPORTED cuando el destino no soporta la creación de identificadores o viceversa
- INVALID_METADATA_IDENTIFIER cuando el identificador no tiene una estructura válida.
- METADATA_RECORD_DOES_NOT_EXIST si el identificador no existe.

- DELETION_NOT_ALLOWED . Este destino no permite borrar metadatos después de que ellos son publicados.
- RESOURCE_ASSOCIATED . La instancia de metadatos está asociada al objeto de aprendizaje. Hay que disociar ambas primero.
- INSUFFICIENT_CREDENTIALS si el usuario no está autorizado para borrar metadatos.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	deleteMetadataRecord
Tipo Retornado	Void
Parámetros	String targetSessionID String metadataIdentifier
Fallos	NO_SUCH_SESSION INVALID_METADATA_IDENTIFIER METADATA_RECORD_DOES_NOT_EXIST INSUFFICIENT_CREDENTIALS DELETION_NOT_ALLOWED RESOURCE_ASSOCIATED METHOD_FAILURE

Tabla 22.Método deleteMetadataRecord

C-Orientados a la publicación de metadatos

a) Método Associate

En un repositorio de objetos de aprendizaje puede contener tanto un almacen para los objetos y otro para los metadatos de manera independiente, de manera que si la instancia de metadatos se publica primero, entonces no puede referenciar al objeto de aprendizaje, hasta que este se publique. Así el método de asociación, permite añadir esta referencia. Sin embargo, el método no chequea la consistencia entre la instancia de metadatos y el objeto de aprendizaje al que se referencia. Cuando se usa éste método se pueden producir 6 tipos de fallos:

- NO_SUCH_SESSION si el TargetSessionID es invalido.
- INVALID_RESOURCE_IDENTIFIER si ningún recurso es identificado por el parámetro resourceIdentifier.
- INVALID_METADATA_IDENTIFIER si ninguna instancia de metadatos es identificada por el parámetro metadataIdentifier.
- RESOURCE_NOT_RETRIEVED en caso de recuperación por referencia, puede ocurrir que la recuperación de un recurso no se haya completado.
- INSUFFICIENT_CREDENTIALS si el usuario no está autorizado a ejecutar este método.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	associate
Tipo Retornado	void
Parámetros	String targetSessionID String resourceIdentifier String metadataIdentifier
Fallos	NO_SUCH_SESSION INVALID_RESOURCE_IDENTIFIER INVALID_METADATA_IDENTIFIER RESOURCE_NOT_RETRIEVED INSUFFICIENT_CREDENTIALS METHOD_FAILURE

Tabla 23.Método associate

b) Método Dissociate

Permite que una asociación pueda ser deshecha. Es necesario cuando una instancia de metadatos o un objeto de aprendizaje va a ser borrado.Cuando se usa éste método se pueden producir 6 tipos de fallos:

- NO_SUCH_SESSION si el TargetSessionID es invalido.
- INVALID_RESOURCE_IDENTIFIER si ningún recurso es identificado por el parámetro resourceIdentifier.
- INVALID_METADATA_IDENTIFIER si ninguna instancia de metadatos es identificada por el parámetro metadataIdentifier.
- INSUFFICIENT_CREDENTIALS si el usuario no está autorizado a ejecutar este método.
- NOT_ASSOCIATED cuando se intent disociar un recurso y una instancia de metadatos que no estabas asociadas.
- METHOD_FAILURE si la operación falla por otras razones.

Nombre Método	dissociate
Tipo Retornado	void
Parámetros	String targetSessionID String resourceIdentifier String metadataIdentifier
Fallos	NO_SUCH_SESSION INVALID_RESOURCE_IDENTIFIER INVALID_METADATA_IDENTIFIER INSUFFICIENT_CREDENTIALS NOT_ASSOCIATED METHOD_FAILURE

Tabla 24.Método dissociate

2.3.2. Otras especificaciones

Existen otros protocolos que permiten la publicación de objetos de aprendizajes, pero la principal diferencia con respecto a SPI, es que éste ha sido diseñado para soportar múltiples escenarios. A continuación se enumeran otros protocolos alternativos, y en la tabla se comparan:

- ***Fedora(Flexible Extensible Digital Object and Repository Architecture)*** [Stapleset al., 2003] [Payette and Lagoze, 1998], que es una arquitectura diseñada para librerías digitales, repositorios institucionales, y repositorios de objetos de aprendizaje.
- ***PENS(Package Exchange Notification Services)*** es un protocol que soporta un servicio de notificación para paquetes de contenido[PENS, 2005]. No permite publicación de instancias de metadatos ni soporta publicación en modo "por valor".El servicio avisa a un repositorio mediante una notificación, que puede recuperar un paquete desde una URL
- ***El SRU Record UPdate*** [McCallum, 2006] [Sanderson et al., 2005] permite crear, reemplazar y borrar registros de metadatos. No permite la publicación de recursos y por tanto solo puede implementarse sobre un almacen de metadatos.
- ***El EduSource Communication Layer (ECL)*** [Hatala et al., 2004a] [Eap, 2003] [Eap et al., 2004] implementa el IMS DRI [IMS, 2003] que propone publicar un mensaje para enviar objetos de aprendizaje o metadatos a un repositorio. Como respuesta el repositorio envía un petición de publicación.
- ***El Open Knowledge Initiative (O.K.I)*** [Hatala et al., 2004b] especifica la Open Service Interface Definitions (OSID) se pueden publicar recursos en un repositorio digital. El repositorio OSID define una JAVA Asset Interface que ofrece métodos para añadir y borrar registros tanto de metadatos como de recursos, así como asociarlos.

	SPI	**Fedora**	**PENS**	**SRU Record Update**	**ECL**	**OKI**
Proporciona Identificadores	Destino y Fuente	Destino	Fuente	Destino	Destino	No aplicable
Escenarios: **-Metadatos** **-Recursos** **-Ambos**	 Si Si Si	 Si Si Si	 No Si No	 Si No No	 Si Si Si	 Si Si Si
Por referencia	Si	No	Si	No	Si	No aplicable
Por Valor	Si	Si	No	No	No	No aplicable
Esquema de metadatos	Da igual	Mapeo a METS	No gestiona metadatos	Da igual	LOM	Mapeo a OKI registro
Modelo Abstracto	Si	No	Si	No	Si	Si

Tabla 25 .Tabla comparativa de SPI con otros protocolos de publicación

2.4. Harvesting de metadatos

El "metadata harvesting" es una técnica que permite hacer copias locales de registros de metadatos. El mejor protocol es el Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [Lagoze], que permite harvestear o bien una colección entera de metadatos o bien una selección.En esta sección se describe brevemente dicho protocolo.

2.4.1. Open Archives Initiative – Protocol for Metadata Harvesting (OAI-PMH).

OAI es una iniciativa para desarrollar y promover estándares de interoperabilidad para la difusión de contenidos en Internet. No está específicamente orientada a los contenidos educativos sino a cualquier contenido digital. OAI[Lagoze] ha sido creada bajo la supervisión de Digital Library Federation (DLF), Coalition for Networked Information (CNI) y La National Science Foundation (NSF). La idea básica que fomenta OAI es crear una forma simple y sencilla de intercambiar información (concretamente metadatos) entre repositorios heterogéneos que alberguen cualquier objeto que contenga metadatos asociados. OAI desarrolló el Protocolo PMH (Protocol for Metadata Harvesting) para permitir el intercambio de estos metadatos entre repositorios. Este protocolo define los mecanismos para recolectar los registros de los repositorios que contienen metadatos.

Los aspectos más relevantes que definen el protocolo OAI-PMH[OAI-PMH] son:

- Divide el mundo en proveedores de datos y proveedores de servicios. Los proveedores de datos albergan repositorios con los recursos que se quieren publicar y exponen los metadatos de dichos recursos para ser recuperados por los proveedores de servicios, por ejemplo los proveedores de datos pueden ser servidores de bases de datos. Los proveedores de servicios son entidades que recolectan metadatos de los proveedores de datos y los utilizan para dar servicios de alto nivel sobre dichos datos, por ejemplo buscadores. OAI utiliza estos conceptos para definir su modelo de cliente / servidor, donde los datos se corresponden con la parte del servidor y servicios

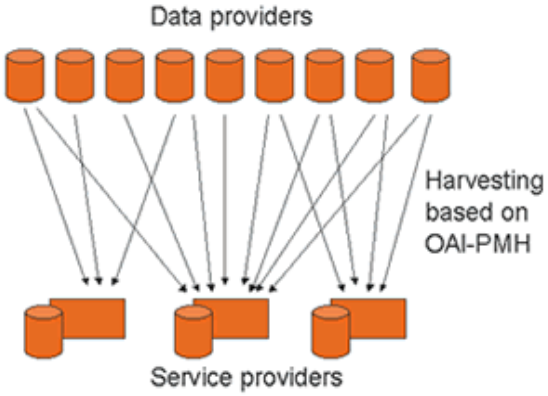

Figura 6.Proveedores de datos y proveedores de servicios

- Enfoque de recolección ("Harvesting") a nivel de metadatos. En interoperabilidad existen dos grandes enfoques bien diferenciados. El enfoque distribuido y el enfoque recolector. En el primero, se buscan y descubren los servicios remotos y la información. En el segundo, los datos/metadatos son transferidos desde la fuente remota al destino en el cuál se realizarán los servicios de búsqueda. OAI–PMH se basa en el enfoque recolector. Adopta

esta solución rechazando el modelo distribuido al asegurar que permite garantizar la interoperabilidad con el menos coste posible de implementación.

– El protocolo OAI–PMH proporciona una opción simple basada en el protocolo HTTP (las peticiones que se pueden realizar son operaciones GET y POST) y el formato XML. Los metadatos pueden estar en cualquier formato que sea acordado por la comunidad correspondiente, aunque toma como base el esquema proporcionado por Dublín Core. Es importante notar que OAI-PMH no proporciona una búsqueda a través de los metadatos, simplemente permitir tener todos los metadatos juntos en un solo lugar. Para proporcionar servicios sobre estos datos, este protocolo se debe combinar con otros mecanismos que los ofrezcan.

El esquema de funcionamiento del protocolo se puede resumir en la siguiente imagen:

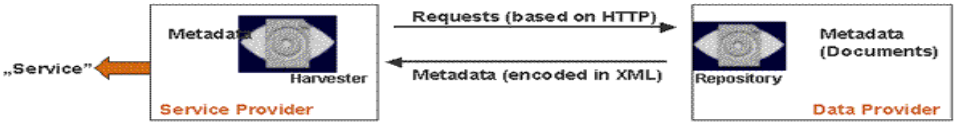

Figura 7.Funcionamiento del protocolo

El proveedor de servicios solicita al proveedor de datos una petición a través del protocolo HTTP. Ante la petición el proveedor de datos devuelve los metadatos en un fichero XML bien formado.Por otro lado, el proveedor de servicios lo que publica de cara al resto de sistemas son los distintos servicios que tiene implementados y a los que se puede acceder.

El protocolo OAI-PMH permite seis tipos de peticiones conocidas como “verbs”. Las respuestas que se reciben a estas peticiones son siempre ficheros XML. En la imagen se muestran las distintas peticiones que se pueden realizar desde un proveedor de servicios y las respuestas que se pueden obtener en función de estas peticiones.

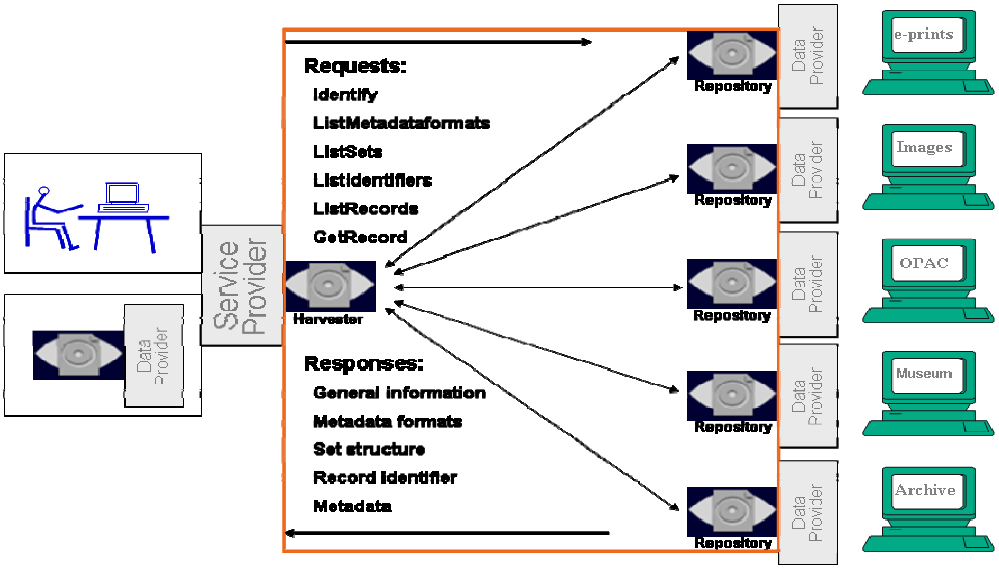

Figura 8.Posibles peticiones y respuestas del protocolo

Para poder ser un proveedor de datos se deben cumplir una serie de condiciones:

– Debe poseer los metadatos almacenados en sistemas de bases de datos

– Debe disponer de un servidor web
– Debe poseer una API de programación (php, perl, zope,…)

Y por otro lado debe tener una serie de componentes básicos:

– Un "Parser" de argumentos para validar las peticiones
– Un generador de errores (Respuestas XML)
– Debe tener acceso a BD para extraer los metadatos de acuerdo al formato de metadatos requerido
– Debe tener un generador XML para crear las respuestas

En la siguiente figura se puede ver el esquema de la arquitectura de un proveedor de datos.

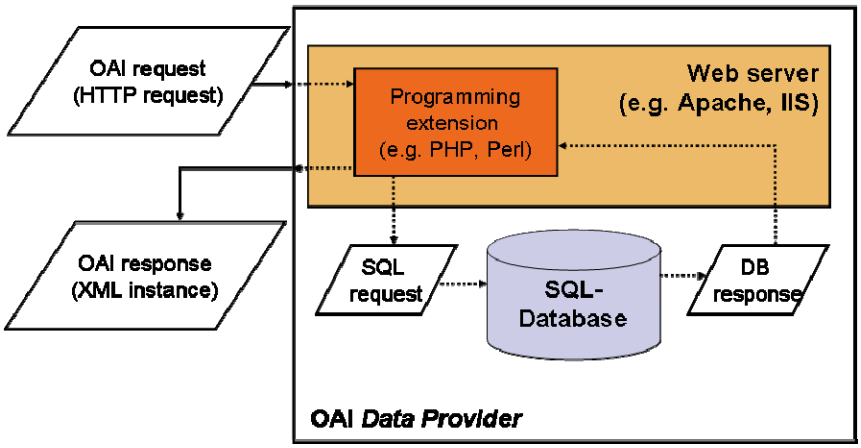

Figura 9.Arquitectura de un proveedor de datos

De igual forma para poder ser un proveedor de servicios, se deben cumplir una serie de condiciones:

– Debe tener un servidor conectado a internet
– Debe contar con un sistema de base de datos (relacional o XML)
– Debe disponer de un entorno de programación

Y debe tener una serie de componentes básicos:

– Debe poder seleccionar los repositorios de datos a recopilar
– Debe permitir una recopilación selectiva
– Debe poder planificar las recopilaciones
– Debe tener un control del flujo
– Debe disponer de un parser XML
– Debe poder detectar duplicidades
– Debe tener un módulo de "servicio al público"

En la siguiente imagen se puede ver el esquema de la arquitectura de un proveedor de servicios.

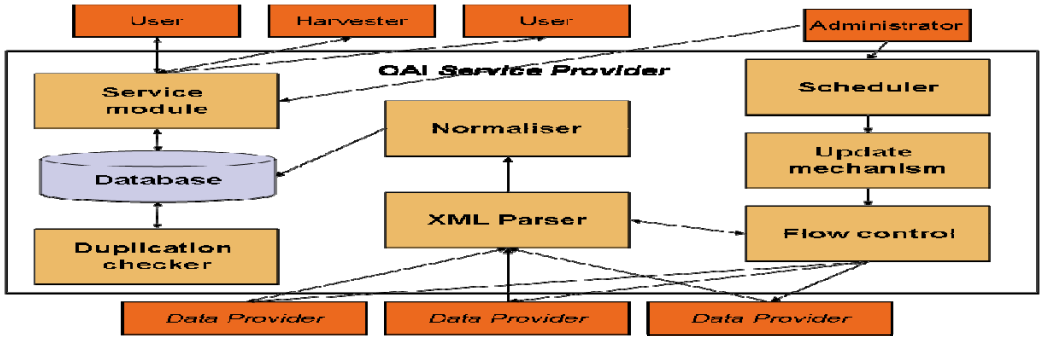

Figura 10.Arquitectura de un proveedor de servicios

2.4.2. Otras especificaciones.

La principal alternativa a OAI-PMH es OAI-ORE(*Object Reuse and Exchange)* [Lagoze-2007]. Esta especificación tiene como objetivo definir un estándar que permita identificar, describir e intercambiar agregaciones de recursos web. Para ello es necesario establecer:

- Cómo puede un proveedor de servicios codificar y desglosar descripciones de agregaciones.
- Cómo puede un usuario/máquina descubrir e interpretar descripciones de agregaciones.

La especificación representa el dominio de una agregación a través de un grafo, formado por nodos (recursos o propiedades) y arcos (relaciones). De manera que existe un nodo (llamado “mapa de recurso”) que describe al nodo que representa a la agregación en sí misma. Este nodo queda identificado con una *URI* (*URI-R*), que da acceso vía *Http*, a un fichero donde en un determinado esquema de metadatos, se proporcionan todos los recursos y relaciones existentes en el grafo. Observar que cada recurso agregado puede ser asimismo una agregación, y cada agregación puede tener recursos asociados con otro tipo de relaciones.

Existen tres formatos para serializar mapas de recursos: RDF/XML[RDF], RDFa[RDFa] y Atom [Atom]. De todas ellas, cabe destacar la opcion de Atom, basada en la noción de feed al cual se le asocia un número de entradas, muy similar a OAI-ORE. De esta forma, un feed se puede hacer corresponder con una agregación y, cada una de las entradas del feed, con cada uno de los recursos agregados.

Para localizar mapas de recursos, la especificación propone dos métodos:

- Orientados a la localización de un mapa de recurso
  Se basan en proporcionar la URI-R embebida en el recurso agregado, usando métodos ocultos al usuario (mediante el elemento link de Html o el Http link header), o visibles (mediante links en el cuerpo del Html).
- Orientados a la localización de muchos mapas de recursos
  Se basan en la recuperación en masa de mapas, uusando OAI-PMH, Sitemaps y redifusión de feeds (RSS o Atom).

2.5. Lenguajes de consulta

Los objetos de aprendizaje están descritos usualmente mediante metadatos, por lo que el acceso al mismo se facilitará si las consultas que se realizan tienen en cuenta los metadatos[Blinco et al., 2005]. Además muchos objetos de aprendizaje contienen material textual que puede ser indexado, pudiendo ser usados dichos índices para filtrar los encajen frente a las palabras claves provistas por el usuario.

2.5.1. ProLearn Query Language(PLQL).

PLQL[Ternier et al., 2008] es una propuesta de lenguaje estandar simple de consultas para recuperar objetos de aprendizaje de repositorios heterogéneos. Combina dos tipos de paradigmas de consultas, por una parte la búsqueda aproximada, y la búsqueda exacta.

El resultado de una consulta PLQL es un conjunto de URIs apuntando a documentos, posiblemente con metainformación describiéndolos. La recuperación de cada documento es una tarea que debe realizar el cliente PLQL. Si la consulta incluye claúsulas aproximadas, entonces el resultado es un conjunto ordenado de URIs determinadas de acuerdo a las cláusulas aproximadas. La especificación define diferentes niveles de funcionalidad que llevan asociados subconjuntos de funcionalidades y diferentes poder de expresividad, permitiendo así que cada repositorio se adapte a sus necesidades. Así, se han distinguido 3 niveles de consulta, y cuatro niveles de resultados de consulta. El protocolo SQI ofrece un método, setQueryLanguage, que permite a una fuente establecer el lenguaje y el nivle de las consultas, usando para ello la URI que identifica el nivel.

A- Niveles de consulta.

Nivel 0

Consiste en una búsqueda aproximada de conjuntos de términos, de manera que los términos de búsqueda deben encajar con los metadatos o el contenido textual de los objetos de aprendizaje. Todos los terminos deben encajar para que el objeto sea seleccionado. La implementación del ranking es delegada al repositorio sobre el que se realiza la búsqueda, de manera que si ningún ranking es explicitamente especificado por el cliente PLQL, entonces el ranking puede estar basado en la relevancia de las palabras clave. Observar que este nivel es independiente de todo esquema de metadatos, que por una parte supone una limitación del poder de expresividad, pero es ventajoso en el contexto de búsquedas federadas.La síntaxis de este nivel viene definido por la siguiente gramática:

0-1: PLQLQuery ::= approximateClause
0-2: approximateClause ::= operand | '(' approximateClause ')' | approximateClause 'and' approximateClause
0-3: operand ::= term1 | term2 | integer | real
0-4: term1 ::= charString1
0-5: term2 ::= charstring2
0-6: charString1 ::= Cualquier secuencia de caracteres que no incluya ninguno de los siguientes: espacio en blanco, tabulador, '(', ')', '=', '<', '>','"', '/', '\', '.' Si al final de la secuencia está la palabra reservada “OR”, entonces se retorn dicho token.
0-7 charString2 ::= Doble comillas encerrando una secuencia conteniendo cualquier caracter salvo dobles comillas(a menos que esté precedido por un backslash '\'. Todos los backslashes que no preceden a un doble comillas son

parte de la cadena de caracteres.
0-8: integer ::= [0-9]+
0-9: real ::= [0-9]*\.[0-9]+

Este nivel se identifica por la URI: http://www.prolearn-project.org/PLQL/l0.

Nivel 1

Introduce la notición de caminos, separados por puntos, de manera que se pueda navegar por ellos. Asume que los metadatos son LOM, DC y MPEG-7, de manera que las raices de los camino solo pueden ser uno de los siguientes: 'dc', 'lom', and 'mpeg-7'. Así en este nivel se mezclan los criterios de búsqueda exacta y aproximada. Cuando en una misma consulta se mezclan criteriso exactos y aproximados, entonces las claúsualas exactas son ejecutadas en primer llugar, produciendo un conjunto de resultados, los cuales son filtrados ejecutando las cláusulas aproximadas. También estas claúsulas son las responsables de ordenar los resultados. Si las claúsulas exactas no encajan con ningún metadato, un código de error es retornado, para así modificar la consulta. Sin embargo también ante estos casos, es posible especificar un comportamiento alternativo, de manera que la consulta de claúsulas exactas se convierta en una consulta aproximada por palabras claves. Si esto ocurre se retorna un código especial, para indicar esta situación. Este nivel solo soporta conjunción, lo que significa que todas las claúsulas exactas deben ser consideradas, y solo se permite predicados de igualdas en las claúsulas exactas. Es posible usar paréntesis entre claúsulas, pero no cambiarán la semántica de la consulta.

Respecto a la síntaxis, consiste en una ampliación de las producciones del nivel 0, y una sustitución de otras(aquellas que coinciden en número):
1-1: PLQLquery ::= clause
1-10: clause ::= exactclause | approximateclause | '('clause')' | clause 'and' clause
1-11: exactclause ::= standard'.'path operator operand
1-12: path ::= term1 | path '.' path
1-13: operator ::= '='
1-14: standard ::= 'dc' | 'lom' | 'mpeg-7'

Este nivel se identifica por la URI: http://www.prolearn-project.org/PLQL/l1.

Nivel 2

Este nivel tiene en cuenta la naturaleza jerárquica de los metadatos de un objeto de aprendizaje, de manera que se introduce nueva expresividad tanto en las cláusulas exactas como en las aproximadas. Para claúsulas exactas, se hace posible el uso de la disyunción y todos los predicados convencionales de comparación. Además en las expresiones de caminos, se puede usar los paréntesis. Esto permite recorrer la estructura jerárquica y parar en los nodos, en los cuales se expresan diferentes criterios sobre los descendientes. También en este nivel se permite el uso de espacios de nombres sobre 'lom', 'dc', and 'mpeg-7'. Para claúsulas aproximadas, se introduce el uso del símbolo “exact” para indicar que hay que encajar exactamente una palabra.

Respecto a la síntaxis, consiste en una ampliación de las producciones del nivel 1, y una sustitución de otras(aquellas que coinciden en número):

2-1: PLQLquery ::= clause
2-2: approximateClause ::= operand | '(' approximateClause ')' | approximateClause boolean approximateClause
2.10: clause ::= exactclause | approximateClause | clause 'and' clause | '('clause')'
2-11: exactclause ::=pathexpr | '(' exactclause ')' | exactclause boolean exactclause
2-13: operator ::= '=' | '>' | '>=' | '<' | '<=' | 'exact'
2-14: standard ::= 'dc' | 'lom' | 'mpeg-7' | term1
2-15: pathexpr ::= path operator operand | pathExp
2-16: pathExp ::= path '.' pathExp | '(' selector boolean selector ')'
2-17: selector ::= path operator operand | selector boolean selector | '(' selector ')' | '(' selector boolean selector ')'
2-18: boolean ::= 'and' | 'or'

Este nivel se identifica por la URI: http://www.prolearn-project.org/PLQL/l2.

B- Niveles de resultados de consulta.

Además de normalizar el formato de la consulta intercambiada, es necesario normalizar el formato de los resultos devueltos para satisfacer el protocolo SQI. Se definen cuatro niveles que aumentan incrementalmente el detalle:
Nivel 0: Se retorna el número de resultados.
Nivel 1: Los resultados deben contener también las URIs de los objetos de aprendizaje, los cuales pueden ser recuperados por el cliente. Si el sistema de ranking está disponible en el lado del servidor, entonces los resulados son retornados de manera ordenada, de manera que se presentan en primer lugar los mejores resultados.
Nivel 2: Información de los metadatos puede ser añadida al conjunto resultado. Se espera al menos el título, autor y lenguaje del objeto.
Nivel 3: Un valor de ranking numérico es añadido a cada resultado, así como un identificador del método de ranking que haya sido usado.
Es necesario siempre especificar el nivel de resultado deseado, y opcionalmente especificar dos parámetros:

- El método de ranking a ser usado en el servidor para extraer los resultados.
- El nombre de estándar a ser usado(LOM, DC, MPEG-7, ...) cuando se retornan los metadatos con los resultados.

La síntaxis es la siguiente:

1: PLQLRES ::= 'http://www.prolearn-project.org/PLRF/' level ['/' standard '/' method]
['/' standard '/' method] es opcional en el nivel 0, y obligatorio en el resto.
2: level ::= '0'|'1'|'2'|'3'
3: standard ::= 'dc'| 'lom' | 'mpeg-7' | term1
4: method ::= string

2.5.2. Otras especificaciones.

Existe dos alternativas importantes a PLQL:

- CQL (the Contextual Query Language) [CQL]. Este lenguaje fue un punto de partida para definir PLQL, pero difiere en algunos aspectos. CQL se usa a menudo en combinación con el protocolo SRU/W para búsquedas sobre bibliotecas. Su objetivo era ofrecer un lenguaje de consultas muy expresivo que pudiera ser leido y escrito fácilmente. No soporta estructuras de metadatos jerárquicas, pues en el ámbito de las bibliotecas se usan esquemas de metadatos planos como Dublin Core, MARC o Zthes. Otra diferencia se encuentra en la presentación de los resultados, mientas que en CQL se ordenan los resultados sobre los campos de los metadatos, en PLQL ofrece un sistema de ranking más complejo no limitado a campos específicos. Aunque el ranking puede ser implementado en CQL si un campo de los metadatos contiene un valor de ranking.
- QEL (the Query Exchange Language) [Nejdl et al., 2002] es un lenguaje de consultas RDF que está expresado usando sintaxis Datalog. Fue implementado para la red Edutella, una red P2P que proporcionaba búsqueda interoperable entre repositorios de objetos de aprendizaje. Es independiente del esquema de metadatos. Retorna variables acotadas que encajan con objetos, predicados y temas RDF, lo cual difiere de PLQL, el cual está diseñado para recuperar instancias de metadatos en vez de resultados parciales.

En la siguiente tabla se comparan las tres especificaciones.

	QEL	CQL	PLQL
Independiente del esquema	Sí	Sí	Sí
Soporte avanzado para consultar metadatos jerárquicos	Sí	No	Sí
Recuperación jerárquica de metadatos multivalorados	No	Sí	Sí
Aproximación en capas	Sí	Sí	Sí
Soporte para combinar claúsulas con variables	Sí	No	No
Resultados formateados	Variables acotadas	Instancias	Instancias

Tabla 26.Comparativa de PLQL con otras especificaciones

2.6. Modelos globales

La naturaleza distribuida de las instituciones y su tendencia a administrar sus contenidos en repositorios distribuidos, hace necesario disponer de modelos de referencia para crear y gestionar estos repositorios. Así han ido apareciendo diferentes modelos de referencia que tratan de estandarizar las funcionalidades y características de un repositorio digital. En esta sección se describe brevemente el principal modelo de referencia, IMS DRI.

2.6.1. IMS Digital Repositories Interoperability(IMS DRI)

IMS DRI [IMS, 2003] es una especificación de normas y recomendaciones, que facilita un esquema funcional de una arquitectura del sistema y de un modelo de referencia completo para la interoperabilidad de repositorios. En este esquema se definen ocho funciones relevantes, repartidas en dos niveles: a) A nivel del repositorio y b)A nivel de manejo de los recursos, y cuya descripción se muestra en la siguiente tabla:

Nombre de la función	Descripción
Exponer	Se necesita un mensaje asíncrono para su activación. Este servicio será invocado por el proveedor de servicios para proporcionar la respuesta correspondiente a la petición de colectar, buscar o alertar.
Colectar	Los repositorios que quieran proveer este servicio, deben implementar el manejador correspondiente.
Buscar	IMS DRI recomienda el uso de XQuery para soportar las búsquedas. Para permitir la conexión entre repositorios que no implementan totalmente el sistema XQuery se utiliza un conjunto de plantillas. Los repositorios registran sus servicios de búsqueda indicando si soportan totalmente la búsqueda XQuery o sólo el subconjunto de plantillas.
Alertar	IMS DRI recomienda el uso de envío de mensajes de alerta ante la aparición de nuevos metadatos coincidentes con las solicitudes.
Enviar	Función que permite mover un objeto (metadatos y objeto de aprendizaje) a un repositorio.
Almacenar	Se necesita un mensaje asíncrono para su activación. Este servicio será invocado por el proveedor de servicios para proporcionar la respuesta a la petición de envío.
Solicitar	Proporciona una función que permite solicitar los objetos a entregar a un cliente. IMS DRI recomienda que el protocolo de transferencia para la entrega de objetos sea SOAP o FTP.
Entregar	Se necesita un mensaje asíncrono para su activación. Este servicio será invocado por el proveedor de servicios para proporcionar el conjunto de resultados.

Tabla 27.Arquitectura de funciones

En la figura se muestra un esquema del modelo de referencia planteado por IMS DRI, en el que se pueden observar tres entidades:

- Roles (entidades coloreadas en verde).
- Componentes para la administración de los recursos (entidades recuadradas en amarillo).
- Servicios (entidades recuadradas en azul).

Las líneas rojas indican las interacciones entre las principales funciones de la arquitectura planteada.

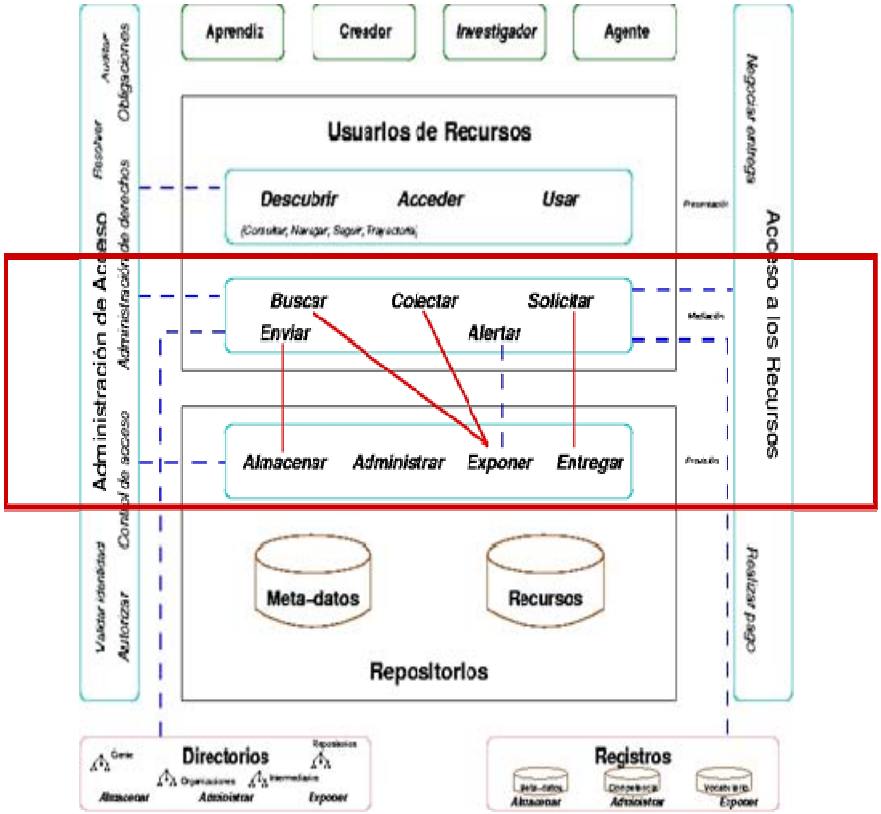

Figura 11.Arquitectura del protocolo IMS DRI

Los roles considerados en esta especificación son los siguientes:

- **Alumno.** Se define cómo una persona que sigue un curso o que está dentro de un proceso de aprendizaje. Generalmente un alumno usará los diferentes servicios Web que un sistema de enseñanza pueda ofrecer y necesitará los diferentes recursos educativos para poder llevar a cabo su formación.

- **Creador.** Es una persona encargada de la creación de los objetos educativos, de los cursos, del secuenciamiento de aprendizaje, o de los programas de las asignaturas.

- **Buscador de información.** Este rol lo componen aquellas personas que buscan información en los repositorios a través de los diferentes servicios de búsqueda. Tanto un alumno cómo un creador pasan a ser buscadores de información en el momento que buscan dentro de los diferentes recursos de aprendizaje que hay

dentro de un almacén. Es importante tener en cuenta que aquellos que se engloban dentro de este rol no tienen porqué estar dentro del proceso de aprendizaje.

- **Agente.** Es una aplicación inteligente capaz de emular el comportamiento de cualquiera de los otros roles. Una vez realizada su tarea, un agente procesará los resultados obtenidos bien de manera automática, bien mediante la intervención de alumnos, creadores o investigadores.

Mediante esta especificación, se consigue acceder a los contenidos almacenados en repositorios desde cualquier aplicación software que sea compatible con la especificación IMS DRI. Sin embargo el problema comienza cuando se tienen que realizar búsquedas sobre los contenidos almacenados. Aunque la utilización de metadatos trata de mitigarlo, la especificación IMS DRI propone introducir un componente intermedio que podría implementar las siguientes funciones para facilitar las búsquedas:

- **Un traductor.** Debe ser capaz de traducir una petición de búsqueda de un formato a otro para hacerla así comprensible a todos los repositorios.
- **Un convertidor.** Debe encargarse de hacer que todos los metadatos incluidos en cada objeto de aprendizaje sean útiles para facilitar las búsquedas.
- **Un encargado.** Debe pasar una petición de búsqueda a los diferentes almacenes y de gestionar las respuestas.

En la siguiente tabla se observan de forma resumida los pares de funciones descritos por la especificación IMS DRI y las recomendaciones tecnológicas que proporciona para la implementación de las mismas.

Pares de funciones	**Recomendación tecnológica**
Search / Expose (Buscar / Exponer).	Dos posibles lenguajes de consultas: XQuery y Protocolo Z39.50. Aunque depende de la naturaleza física de los repositorios.
Gather / Expose (Colectar / Exponer).	No realiza recomendaciones. Sugiere utilizar el modelo OAI para la función “Gather”.
Alert / Expose (Alertar / Exponer).	No realiza recomendaciones. El equipo de desarrollo debe seleccionar la solución.
Submit / Store (Enviar / Almacenar).	La utilización de IMS CP como paquetes adjuntos en mensajes SOAP
Request / Deliver (Solicitar / Entregar).	No realiza recomendaciones. El equipo de desarrollo debe seleccionar la solución.

Tabla 28.Funcionalidades de pares.

A continuación, se detalla la funcionalidad de los pares de funciones propuestos para la gestión completa de un repositorio:

Search / Expose (Buscar / Exponer). El usuario especifica los atributos que servirán para discriminar sobre los metadatos de los objetos de aprendizaje. La respuesta es un conjunto de metadatos.

Se dispone de un dominio muy extenso y heterogéneo en cuanto a las diferentes tecnologías empleadas en el almacenamiento. Para solventar esto, se propone una capa intermedia encargada de mediar entre las peticiones y las diferentes formas en que se encuentran almacenadas las respuestas.

Las recomendaciones tecnológicas que realiza IMS DRI para el lenguaje de consultas empleado es:

- XQuery para la búsqueda realizada sobre los metadatos en formato XML.
- Protocolo Z39.50 para la búsqueda en los contenedores de información.

La especificación IMS DRI propone como metodología de búsqueda óptima para localizar recursos en los distintos repositorios distribuidos, el modelo de búsqueda federada. Con anterioridad describe la especificación, la figura de un intermediario para facilitar las búsquedas y la gestión de los resultados.

En la siguiente figura se puede observar el esquema de funcionamiento propuesto por IMS DRI para las búsquedas. Un usuario solicita una búsqueda para la que proporciona los criterios que considere oportuno. El intermediario recibe la solicitud y lanza las búsquedas sobre los intermediarios de cada repositorio. Estos, se encargan de "exponer" los resultados para la búsqueda solicitada y el intermediario se encarga de proporcionárselos al usuario que realizó la búsqueda. Posteriormente, también se encarga de solicitar el objeto al repositorio que lo contiene y facilitárselo al usuario.

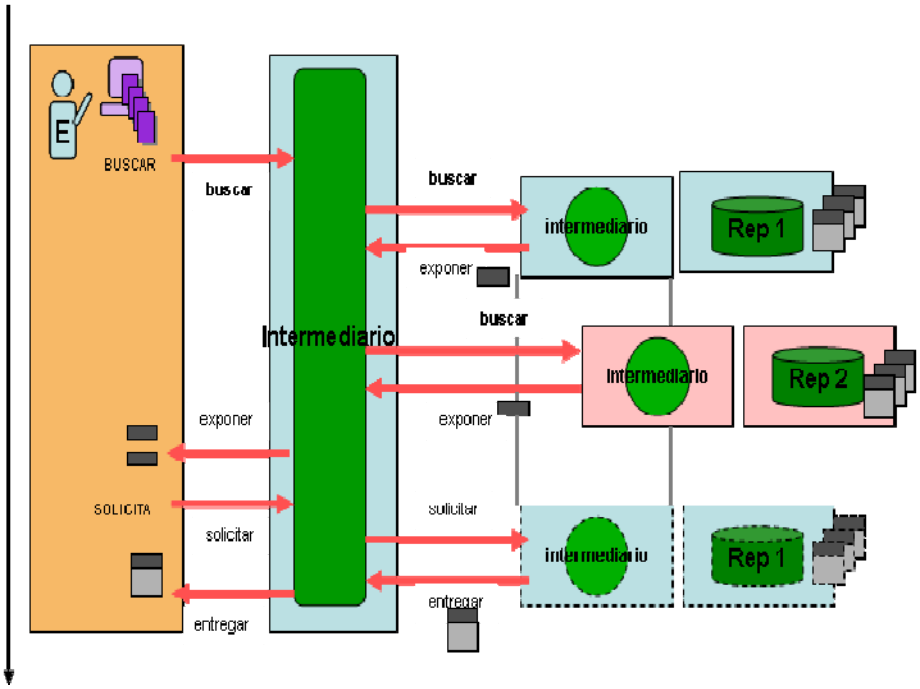

Figura 12.Esquema de Buscar/Exponer

Gather / Expose (Colectar / Exponer). Corresponde a la activación periódica del mecanismo de búsqueda. Esta funcionalidad proporciona la forma de escribir los meta datos que van a servir para las búsquedas, la forma de agruparlos para facilitar los sondeos futuros y la manera en que se tienen que agregar para formar nuevos repositorios (estos almacenes estarán disponibles para las funciones de

búsqueda y alerta). Esta funcionalidad interactúa con el repositorio de dos maneras diferentes. La primera consiste en solicitar metadatos del repositorio (pull), mientras que en la segunda ofrece al almacén meta datos para que sean almacenados (push). La siguiente figura muestra el modelo de referencia de la funcionalidad Gather (Colectar) para una solicitud de meta datos.

En este caso, IMS DRI no realiza ninguna recomendación tecnológica concreta para desarrollar las funcionalidades. Sugiere para la implementación de la función colectar "pull", que el modelo de OAI (Open Archive Initiative) ofrece la funcionalidad suficiente. Para la función colectar "push" considera que es un caso básico de mensajes de alerta y por tanto no se especifica en profundidad.

Alert / Expose (Alertar / Exponer). La especificación contempla esta funcionalidad cómo un posible componente de un repositorio digital o un servicio intermedio encargado de mandar correos electrónicos. IMS DRI no realiza ninguna recomendación al respecto de cómo implementarlo.

Submit / Store (Enviar / Almacenar). Esta funcionalidad hace referencia a la forma de almacenar un objeto en un almacén y la forma que tomará una vez almacenado para hacer posible su recuperación. El lugar desde el cual se coge el objeto para su almacenamiento puede ser otro repositorio, un sistema enseñanza, el disco duro del desarrollador, o cualquier punto de la red. Generalmente, la recuperación y almacenamiento de objetos serán llevados a cabo mediante el protocolo FTP, por lo que esta especificación tiene las siguientes propuestas para evitar las debilidades de esta forma de comunicación:

- Añadir al protocolo mecanismos de encriptación para asegurar la integridad de los contenidos.
- No ofrecer acceso directo al servidor mediante FTP por los problemas de seguridad que esto conlleva.
- Asumir que el protocolo FTP no ofrece ningún mecanismo que asegure que un objeto ha sido copiado en su totalidad de un punto de la red al almacén.

Dentro de esta funcionalidad hay que tener en cuenta las recomendaciones propuestas en IMS CP (Content Packaging). Esta especificación dice: "La interoperabilidad de los sistemas implica que puedan importar, exportar, agregar y desagregar paquetes de contenidos". IMS CP propone utilizar un paquete en formato comprimido que contiene el objeto educativo, su registro de metadatos, y un manifiesto que describe los contenidos del paquete. Por tanto, si se quiere mantener la coherencia en las diferentes especificaciones, se tiene que utilizar este formato para intercambiar contenidos entre los sistemas educativos y los almacenes.

La especificación IMS DRI recomienda que la función de recuperación de objetos educativos debe cumplir que los paquetes se envíen utilizando mensajes SOAP con ficheros adjuntos que tendrán el formato que IMS Content Packaging propone. La función de almacenamiento de objetos educativos permitirá que dichos objetos se guarden según el formato propuesto por IMS CP.

A continuación se muestra una figura que muestra la diferencia en la comunicación entre almacenes que cumplen esta especificación y aquellos que no la cumplen.

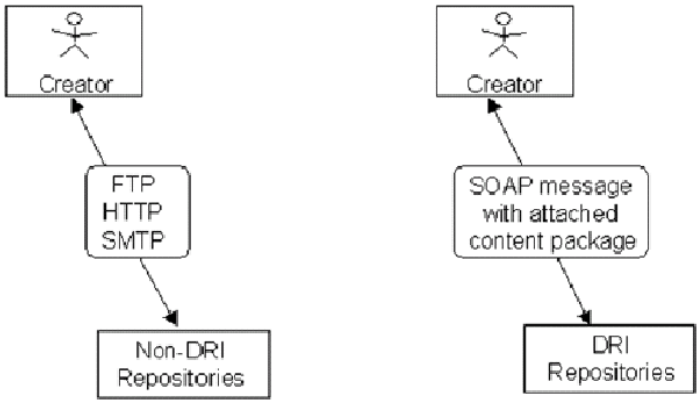

Figura 13.Diferencias en la comunicación entre almacenes.

Request / Deliver (Solicitar / Entregar). La función Request es la petición de acceso a un recurso que realiza un usuario del sistema una vez lo ha localizado gracias a los meta datos que lleva asociados. Deliver se refiere a la respuesta que le da el repositorio, que le otorga o le niega el acceso al recurso.
En este caso, IMS DRI no ofrece ninguna recomendación tecnológica respecto a la funcionalidad referida, dejando este aspecto sin especificar y abierto a la solución tecnológica que cada desarrollador considere oportuno utilizando protocolos genéricos HTTP y FTP.

2.6.2. Otras especificaciones.

La principal alternative a IMS DRI es CORDRA (Content Object Repository Discovery and Registration/Resolution Architecture) [CORDRA], el cual proporciona un modelo de referencia para repositorios de objetos de aprendizaje. Define entre otras cosas cómo una federación de repositorios de objetos de aprendizaje debería ser diseñado y deplegado en la red. En el nivel más bajo, los repositorios de contenido, publican sus metadatos en el registro de metadatos de CORDRA. CORDRA dicta el uso de un registro intermedio de registro en el cual metadatos de diferentes comunidades son agrupados, aunque no está claro qué metadatos deberían encontrarse en este registro. Lo mismo ocurre con respecto al registro principal de registros, en el nivel más alto de los registros. El registro podría contener referencias a registros intermedios, permitiendo una búsqueda federada en el nivel más alto de CORDRA o por el contrario permitir una recolección de los metadatos de todos los contenidos como si fuera un sistema centralizado. En la figura siguiente se muestra un esquema de los registros de CORDRA.

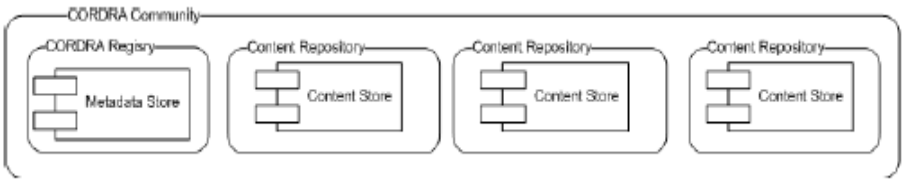

Figura 14.Esquema de registros CORDRA

3. Proyectos de implementación de repositorios digitales interoperables.

En este capitulo se describen brevemente 3 casos de estudio sobre proyectos que han intentado crear repositorios digitales interoperables.

3.1. Proyecto Elena

ELENA[ELENA] es un proyecto de la Comunidad Europea, cuyo objetivo principal es permitir la creación de un "Espacio Inteligente para el Aprendizaje" (Smart Learning Space). Este espacio inteligente es un sistema distribuido que permite dar soporte a la gestión de entrega y consumo de recursos educativos heterogéneos. El término "Espacio" es usado como un sinónimo para red y el término "inteligente" es usado para hacer referencia a la mediación inteligente de recursos educativos (cursos, contenido educativo, etc.) basada en el perfil del usuario y las técnicas de inteligencia artificial.
Los componentes principales de esta infraestructura son:

- API para establecimiento de sesión y realización de consultas sincronías y asíncronas basada en SQI (Simple Query Interface) que define los servicios que un repositorio puede tener disponibles para recibir y responder consultas de otros repositorios.
- Un modelo semántico común para la descripción de recursos educativos, con el fin de facilitar las búsquedas.
- Uso de un lenguaje común para el formato de las consultas que se enviarán a cada uno de los nodos educativos.
- Nodos que contienen recursos educativos. Algunos de los nodos conectados en la red de ELENA son: Amazon, Clix, EducaNext, Lason, Seminarshop.com, Iteach you, ULI.

En la siguiente imagen se muestra un diagrama de los nodos que conforman la red de ELENA, en el que se puede observar que todos ellos utilizan el mismo interfaz (API propuesta por SQI) para acceder al espacio de aprendizaje.

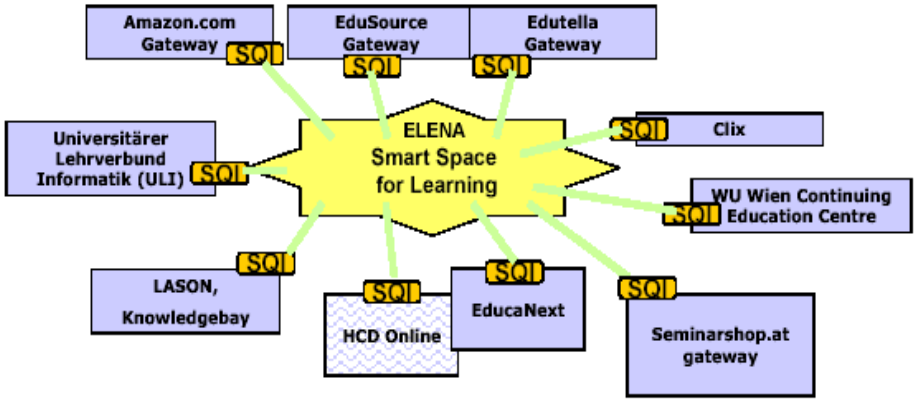

Figura 15.Red Elena

Esta red de nodos está soportada por Edutella. Se trata de una infraestructura de red P2P para repositorios educativos que permite conectar sistemas heterogéneos. Edutella está basado en la tecnología JXTA que permite conectar servidores heterogéneos que describen sus recursos como metadatos.

El proyecto ELENA debe concretar los aspectos que deja abiertos a la interpretación la especificación SQI. Por un lado la concreción del esquema común para la descripción de recursos educativos y por otro lado, el lenguaje común de consultas entre repositorios digitales distribuidos.
El foco central de investigación y desarrollo del proyecto ELENA se encuentra en la semántica que se va a emplear para describir los recursos educativos con el fin de proveer un marco flexible para la integración de los recursos. Actualmente se emplea un esquema en RDF basado en Dublín Core, LOM, Vcard y OpenQ.
Este esquema contiene los atributos necesarios para describir un recurso educativo. Contiene atributos comunes tanto para los materiales como los servicios educativos.
Por otro lado, para la integración de los datos, ELENA sigue una arquitectura basada en mediadores, ya descrita en el apartado correspondiente al modelo SQI de interoperabilidad.
En esta arquitectura varias fuentes de datos están "envueltas" por una capa de software, denominada adaptador o “wrapper”, el cual traduce entre el lenguaje, modelos y conceptos de la fuente de datos y el lenguaje, modelo y conceptos del mediador.
Los mediadores obtienen la información a partir de una o más fuentes de datos o de otros mediadores, es decir, de los componentes que están por debajo de él, y proporcionan información a las componentes que están por encima.
En el proyecto ELENA, los recursos educativos permanecen en las fuentes y el mediador es el responsable de proporcionar a los usuarios la apariencia de estar realizando consultas sobre un único esquema global. Cuando un mediador recibe una consulta, éste la descompone en subconsultas sobre los distintos repositorios.
El lenguaje de consultas común empleado es el QEL, un lenguaje de consultas basado en Datalog o RDF. El formato de entrega de resultados es RDF. El mediador debe ser capaz de propagar las consultas a través de los repositorios de recursos educativos.
El proyecto ELENA ilustra a través de la documentación pertinente, las distintas maneras en las que los nodos que componen la red de repositorios educativos, se han integrado implementando los “wrapper” apropiados. En la siguiente figura, se representa como se produce la integración de un nodo en la red de ELENA.
La arquitectura correspondiente está basada en la creación de un “wrapper” que permite la comunicación entre el SQI y la base de datos del nodo. A través de la interfaz SQI, el “wrapper” recibe las consultas en formato QEL enviadas a través de la red de ELENA, las transforma a lenguaje SQL y reenvía las consultas hacia la base de datos relacional. Los resultados son retornados en formato RDF. Gráficamente se observa en la siguiente figura.

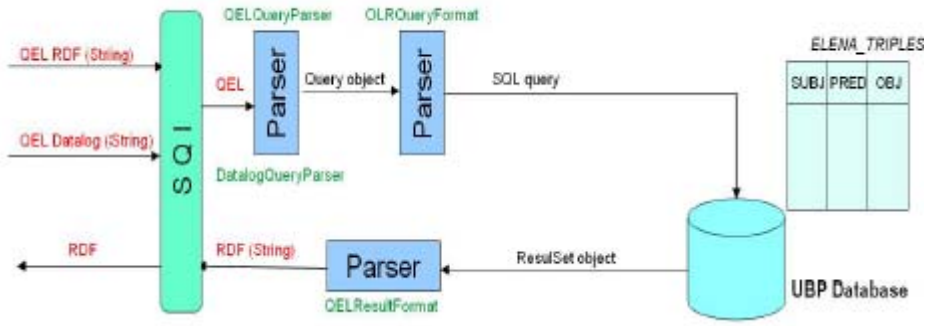

Figura 16.Comunicación SQI-Base de Datos

Por otro lado, el proceso de interoperabilidad semántica, se consigue en la integración almacenando la descripción del recurso educativo en forma de metadatos almacenados en una base de datos relacional.

Primero se construye un archivo RDF que contiene la descripción del recurso educativo en forma de metadatos, basándose en un esquema o plantilla en XML/RDF que define los atributos obligatorios y opcionales necesarios para describir los recursos. El esquema está basado en Dublin Core, LOM, OpenQ, and VCard. Por último, el archivo RDF es parseado para generar las tripletas correspondientes, almacenadas en la base de datos relacional.
La implementación del interfaz SQI de la figura anterior depende de cada uno de los nodos que desee conectarse a la red heterogénea. La red Edutella ha creado una librería (edutella-sqi.jar) en Java que implementa todos los métodos definidos por SQI y que permite el envío de consultas a repositorios.
En la siguiente imagen, se puede observar la particularización del esquema anterior.

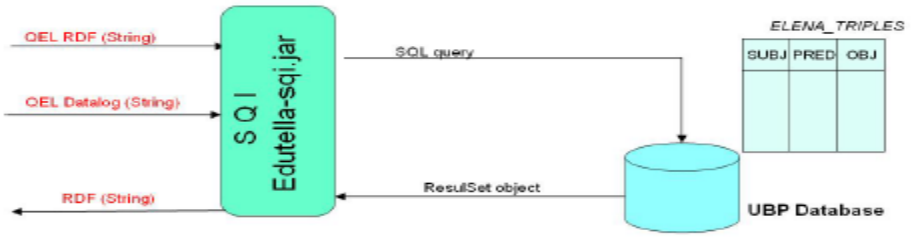

Figura 17.Interoperabilidad semántica

3.2. Proyecto Edusource

El proyecto EduSource [ECL] pretende unir los repositorios canadienses de objetos de aprendizaje heterogéneos a través de la infraestructura necesaria para permitir la interoperabilidad entre todos ellos. Uno de los grandes logros alcanzados por el proyecto, ha sido crear una red abierta orientada a usuarios, organizaciones y servicios. Para ello, EduSource se fundamenta en tres pilares básicos:

- Un protocolo claramente definido
- Perfectamente capaz de trabajar con herramientas, repositorios y servicios
- Creación de una capa que permita la interconexión con sistemas existentes

Para poder comprender el gran esfuerzo realizado por el proyecto EduSource en la interoperabilidad de repositorios digitales, es importante comprender la variedad de comunidades a las que pretende dar soporte. En la siguiente figura se puede observar la infraestructura de la que debe disponer para dar soporte a las distintas comunidades.

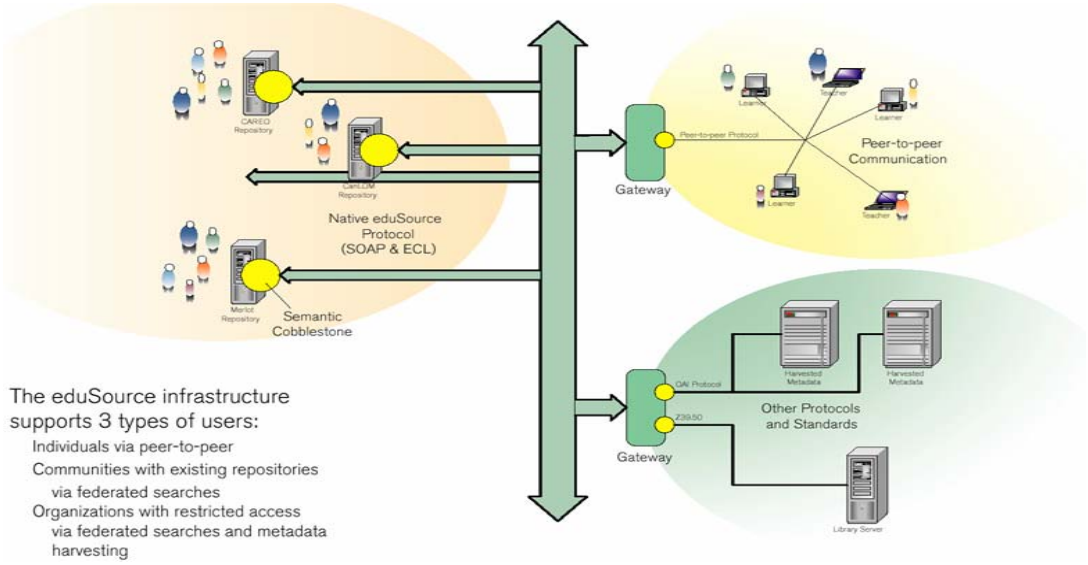

Figura 18.Infraestructura Edusource

Las distintas comunidades reflejadas en la imagen anterior son:

- Repositorios existentes con búsquedas federadas. Este tipo de repositorios suele pertenecer a organismos oficiales y centros educacionales o académicos. Normalmente estos repositorios dan servicio a toda su comunidad de usuarios. Por ejemplo, repositorios albergados en universidades sirven a toda la comunidad de alumnos y profesores del entorno. Normalmente estos repositorios ofrecen sus servicios a través de un portal web. Dentro de este portal se dispone de funcionalidades de búsqueda y creación de contenidos. Es habitual que cada repositorio tiene su esquema particular de metadatos asociados a los objetos de aprendizaje.

- Repositorios individuales. Corresponden a la red formada por los repositorios individuales que se comunican a través de protocolos peer-to-peer (P2P). Los repositorios poseen la funcionalidad necesaria para almacenar y manipular los distintos objetos de aprendizaje confeccionados por cada usuario particular.

- Repositorios de datos recolectados. (“Harvested metadata”). Los servidores de tipo metadatos recolectados son una alternativa a las búsquedas federadas. En lugar de estar constantemente enviando peticiones de búsqueda a todos los repositorios de contenidos como hacen las búsquedas federadas, el modelo recolector, recupera la información de metadatos de los distintos repositorios en un “nodo” central y realiza sobre ese “nodo” las búsquedas.

- Otro tipo de repositorios. EduSource hace hincapié en la posibilidad de poder conectar cualquier otra iniciativa en repositorios distribuidos. Estas conexiones deben ser bidireccionales, de forma que se permita a usuarios de eduSource realizar búsquedas en repositorios externos y viceversa.

La definición del protocolo de comunicaciones apropiado juega un papel determinante en la construcción de un sistema de estas características. El protocolo permite la comunicación entre los miembros, las aplicaciones y los servicios. Es importante notar que para que eduSource se convirtiese en un sistema abierto era necesario que cimentase sus protocolos en los distintos estándares y especificaciones existentes. De esta forma, el proyecto eduSource, crea EduSource Communication Layer (ECL) siguiendo las especificaciones y recomendaciones proporcionadas por IMS DRI. Recordando las directrices que marcaba IMS DRI en el desarrollo de su especificación, es necesario matizar que ECL ha tenido que tomar decisiones de implementación ante la falta de especificación al respecto.

En la siguiente figura, se muestra el entorno de trabajo que plantea ECL.

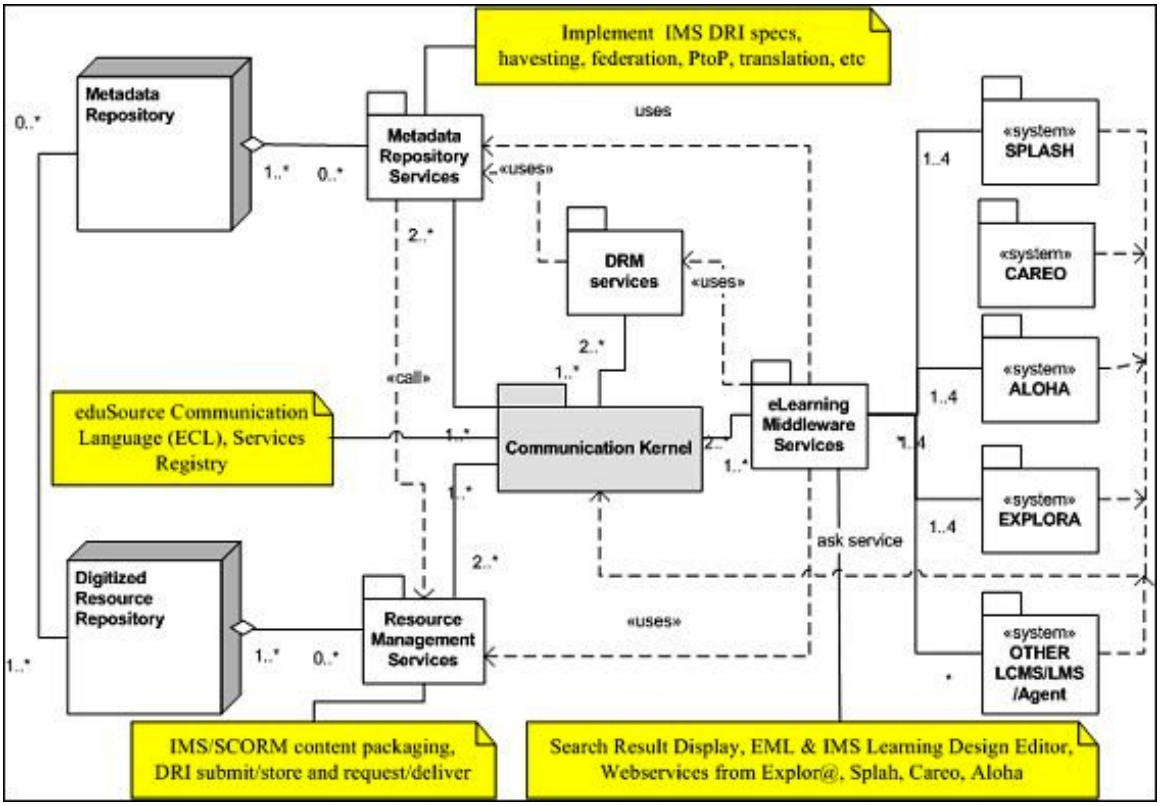

Figura 19.Entorno de trabajo ECL

ECL tuvo en cuenta los siguientes criterios para la implementación apropiada del protocolo:

- EduSource es una red heterogénea formada por repositorios existentes y aún no concebidos, compuesta por repositorios individuales y por interfaces de aplicaciones.

- ECL debe ser capaz de evolucionar durante toda la vida del proyecto teniendo en cuenta todas las iniciativas nuevas posibles que interfieran con la definición del protocolo.

- ECL debe soportar cualquier nuevo servicio no existente actualmente. Algunos de estos servicios podrían requerir un tratamiento asíncrono, tales como búsquedas en el protocolo peer-to-peer.

- ECL es un protocolo completo. Para poder incorporar nuevas definiciones, debe ser rápido y fácil de usar y contar con una capa preconfigurada de funcionamiento.

- Las soluciones para permitir la conexión entre eduSource y otras iniciativas debe ser fácil de mantener y de actualizar si se produce algún cambio en el protocolo definido por esa otra iniciativa.

ECL se encuentra muy cerca en su definición de las pautas marcadas por la recomendación IMS DRI y utiliza SOAP como capa de comunicación. Las funciones básicas definidas en la especificación IMS DRI son definidas como servicios. Repositorios digitales o herramientas que se conectan a la red ECL pueden implementar algunos de estos servicios y registrarlos en el registro de eduSource (por ejemplo UDDI).

Dentro de la definición de ECL se tienen dos grandes ámbitos. Por un lado, proporcionar las herramientas necesarias para crear un nuevo repositorio y permitir que se una a la red existente y por otro lado, proporcionar los componentes necesarios para que un repositorio ya existente pueda formar parte de la red que conforma ECL.

Creación de un repositorio nuevo

Siguiendo esta línea, ECL proporciona la herramienta RIB Repository in the Box. Se trata de un conjunto de objetos, componentes y servicios disponibles para ser instalados en un servidor y trabajar como un repositorio de la red ECL.

- Proporciona una interfaz web para que todos los actores puedan iniciar cualquier acción y ser informados sobre los resultados.

- Ofrece una forma estándar de conectarse con otros repositorios para poder buscar metadatos, obtener metadatos como resultado a las búsquedas y obtener los recursos (objetos de aprendizaje) asociados a esos metadatos.

- Proporciona un conjunto por defecto de herramientas que pueden ser utilizadas localmente para etiquetar contenidos, empaquetar contenidos, etc.

El caso de uso que representa las distintas funcionalidades que tiene un usuario que puede crear un repositorio, se representan en la siguiente figura.

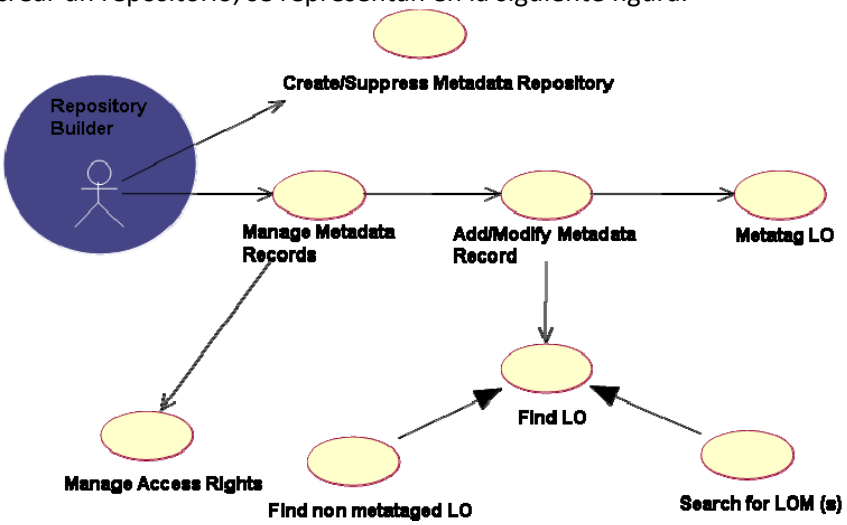

Figura 20.Casos de uso de creación de un repositorio

Incorporación de un repositorio existente a la red eduSource

Para evitar que la complejidad del protocolo definido, pueda impedir su implantación en los repositorios ya existentes, eduSource proporciona un conector que posee implementada la definición del protocolo ECL (ECL connector). Este conector posee un API estándar para permitir la conexión de un repositorio existente a la red eduSource. El conector ECL sólo necesita que los repositorios existentes que quieran incorporarse a la red eduSource le proporcionen los manejadores para soportar los servicios que quieran publicar al resto de repositorios, esto puede ser tan simple como desarrollar e implementar los servicios en cada repositorio.

En la siguiente figura se puede observar un esquema del conector ECL.

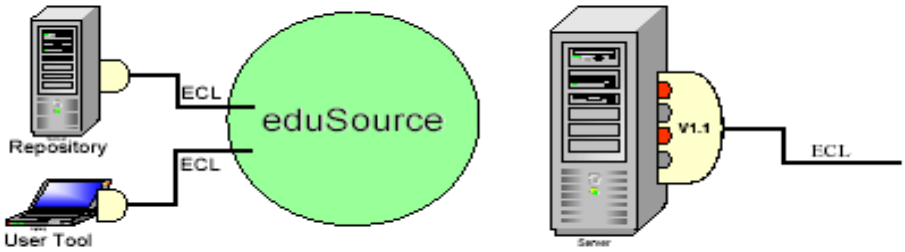

Figura 21.Esquema del conector ECL

El conector facilita la sincronización de las modificaciones introducidas en su implementación durante la evolución de la definición del protocolo. La mayoría de las veces los repositorios no se deben preocupar en absoluto de los cambios producidos en el protocolo sino que simplemente deben actualizar la versión software del conector. Cambios en el protocolo ECL pueden ser detectados y actualizados por las nuevas versiones del conector lo que hace muy interesante la implementación del protocolo al permitir cambios dinámicamente en el desarrollo del conector. Es importante destacar que un API utiliza un lenguaje de programación específico mientras que un protocolo se define a través de un lenguaje genérico. Esto implica que a pesar de que existe una versión del conector desarrollada en Java, se puede desarrollar el conector, siguiendo las especificaciones del protocolo en cualquier otro lenguaje de programación.

En líneas generales, el proceso de funcionamiento que sigue un conector ECL es el siguiente:

- El conector ECL recoge la petición que se recibe, normalmente a través del protocolo SOAP.
- El manejador de servicios crea una petición Java.
- Los manejadores que reciben las solicitudes, envían una petición al gestor de servicios.
- Se crean los objetos de búsqueda Java y se ejecutan dentro del gestor de servicios.
- En los repositorios “propietarios” de la red eduSource se disparan los métodos de búsqueda.
- Los resultados (registros en formato LOM) se adjuntan a mensajes SOAP.
- Por último, se envían los metadatos LOM a través de la tecnología SOAP al ECL cliente.

A pesar de que el protocolo definido por eduSoruce proporciona una solución cómoda y flexible para interconectar repositorios, es poco probable que repositorios estables e iniciativas para el desarrollo de los mismos, conviertan sus protocolos de comunicación a la definición proporcionada por ECL. Por tanto, la capacidad del proyecto eduSource de poder conectarse con otros protocolos ya existentes y establecidos es otro de los puntos importantes a tener en cuenta.

EduSource encamina el problema de la interoperabilidad con otros protocolos, proporcionando un segundo tipo de conector, llamado ECL Gateway. La función principal del ECL Gateway es actuar de mediador entre el protocolo ECL y los protocolos de comunicación utilizados por los repositorios existentes.

En la siguiente figura se puede observar un diagrama del esquema.

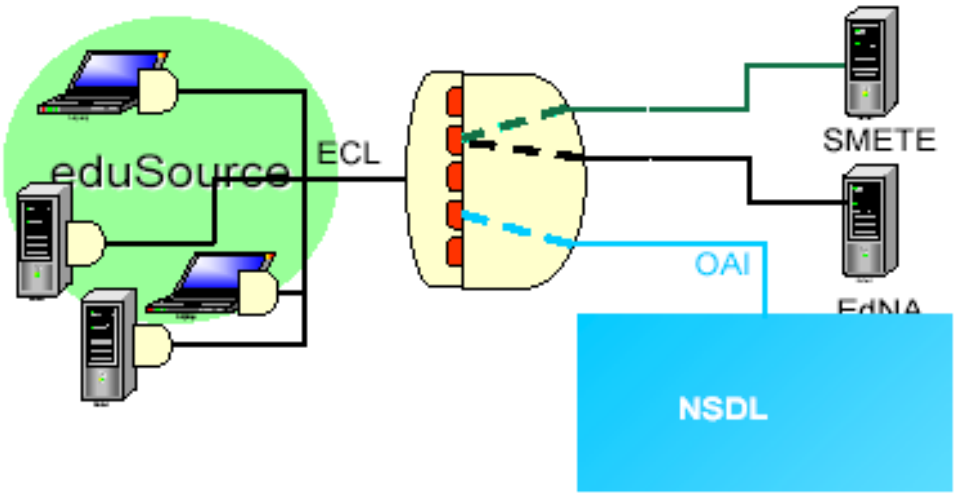

Figura 22. Esquema de funcionamiento.

A uno de los lados se tiene el conector ECL, en el otro lado del ECL Gateway se tiene un entorno de trabajo compuesto por una serie de manejadores en cadena que consiguen la conversión entre el protocolo ECL y el protocolo existen de la red externa.

El entorno de trabajo definido por ECL Gateway permite realizar el mapeo entre protocolos a cuatro niveles distintos:
Nivel 1. Protocolos de comunicación tales como http, SOPA, XML-RPC, Peer-to-Peer.
Nivel 2. Lenguajes de comunicación. Por ejemplo, ECL, OAI, POOL.
Nivel 3. Metadatos. Especificaciones y recomendaciones proporcionadas por IMS, CanCore, Dublín core.
Nivel 4. Ontologías, es decir, vocabulario de metadatos.

Gráficamente se puede observar en la siguiente figura.

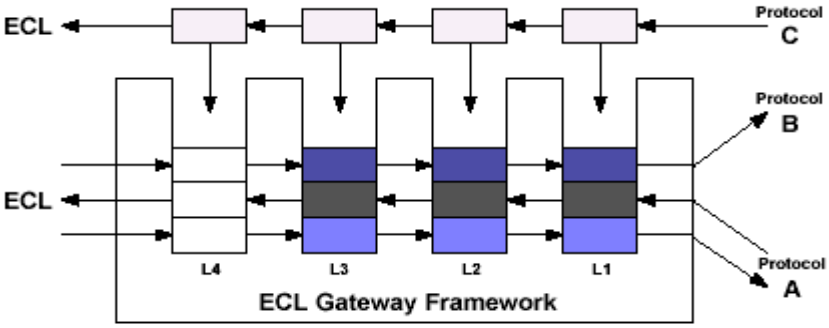

Figura 23. ECL Gateway Framework

Normalmente, ECL Gateway se ejecuta en una máquina dedicada exclusivamente y proporciona servicio a todos los miembros de la red de eduSource. La gran ventaja que presenta colocar la funcionalidad del ECL Gateway en una máquina únicamente, en lugar de replicarlo en cada uno de los participantes externos, radica en la facilidad para actualizar el software en caso de cambios en el protocolo. En este caso, se actualiza el software en un único punto, permitiendo a los sistemas externos seguir comunicándose a través del protocolo ECL sin necesidad de modificar nada en sus sistemas. En la actualidad, la red de eduSource permite la interconexión de distintos modelos de

repositorios, a través directa del conector ECL o a través del mediador ECL Gateway. Por ejemplo, el repositorio Explora de la TéléUniversité en Montreal, el repositorio CAREO en Calgary o el repositorio LO en la universidad de Athabasca. Por otro lado, los repositorios EdNA, SMETE y repositorios acordes a OAI se conectan a la red eduSource a través de ECL Gateway.En la siguiente figura se muestra un esquema de cómo se disponen los ECL Gateway con los conectores ECL.

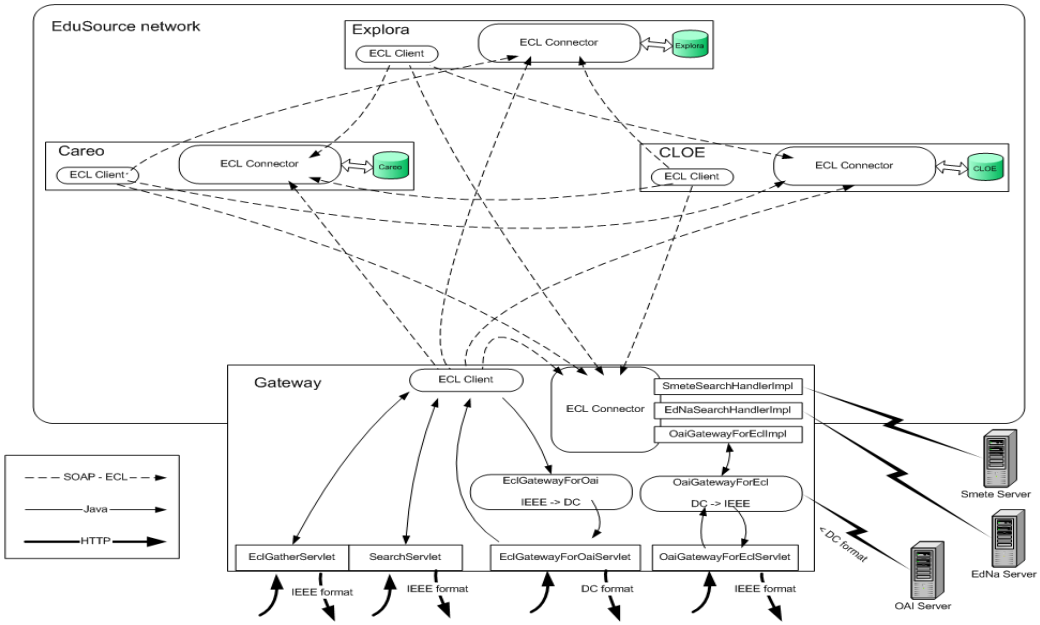

Figura 24. ECL Gateway y conectores ECL

Por último, el siguiente esquema muestra la arquitectura propuesta por el proyecto eduSource.

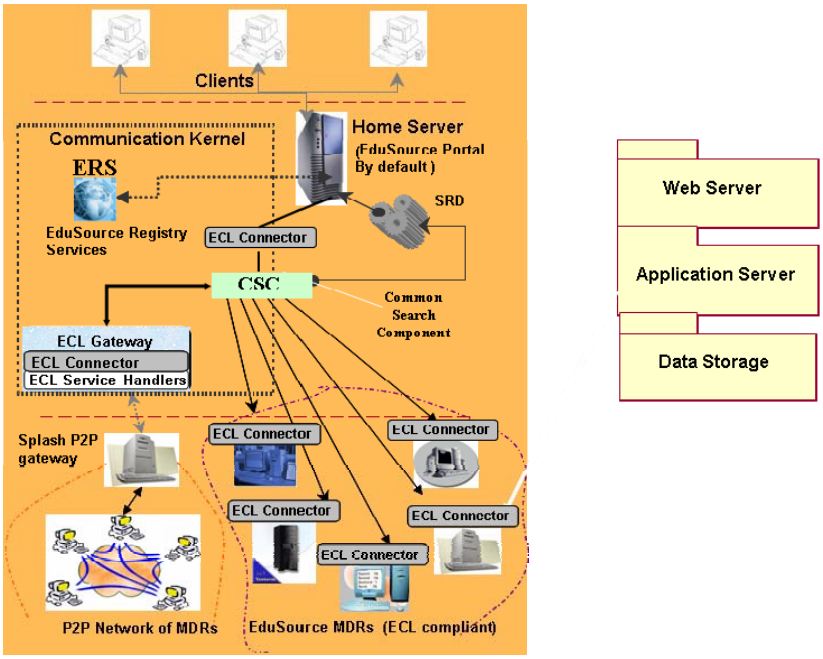

Figura 25. Arquitectura proyecto Edusource

3.3. Proyecto Agrega

El proyecto Agrega[AGREGA] tenía como objetivo crear una federación de repositorios digitales con nodos instalados en cada una de las comunidades autónomas de España. Cada nodo permite almacenar objetos digitales SCORM 2004[SCORM] etiquetados con metadatos LOM-ES[Sarasa-2008]. Estos nodos permiten realizar diferentes operaciones a los usuarios en función de perfil que tengan definido tales como búsqueda, visualización o creación de nuevo material.

El diseño de Agrega se realizo pensando en la necesidad de poder incorporar en la federación de repositorios a otras federaciones de repositorios que existen en el mundo. Por otro lado también se tuvo en cuenta que un repositorio digital es una componente de construcción que debe ser fácil de integrar con otras herramientas u entornos, con la finalidad de crear entornos de educación online más complejos. Es por ello que Agrega trató de cubrir los siguientes objetivos:

- Posibilidad de que otras entidades se puedan federar con Agrega, de forma que se comparta el material digital de ambas entidades.
- Posibilidad de que otras entidades puedan incorporar en su funcionalidad algunos de los servicios implementados en los nodos, como si de servicios propios se tratara, y viceversa, usando el concepto de mashup.
- Posibilidad de incorporar a los repositorios, material educativo digital recuperado de la red, de una forma sencilla.
- Posibilidad de integración de Agrega con otras herramientas o sistemas de orden superior tales como LMS (Learning Management Systems) o bien portales educativos o de otra temática.
- Posibilidad de extender Agrega para que gestione material otros formatos distintos a SCORM 2004.

Cada nodo de la federación se ha implementado usando Java (J2EE) y siguiendo un modelo basado en una Arquitectura Orientada a Servicios (SOA) donde el papel del proveedor de servicios lo interpreta el Nodo de Objetos Educativos Digitales y el papel consumidor lo interpretaran las Aplicaciones Clientes, según el esquema siguiente:

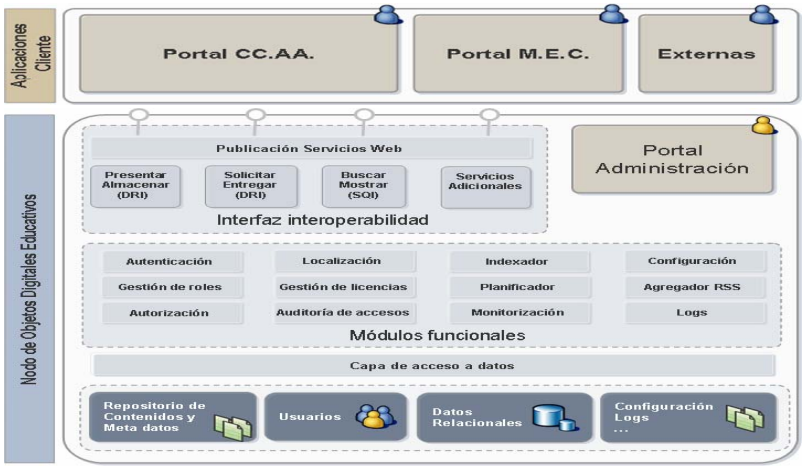

Figura 26. Estructura de un nodo Agrega.

Cada nodo alberga un repositorio de contenidos que almacena contenidos empaquetados de acuerdo a SCORM 2004, y catalogados usando el perfil de aplicación LOM-ES. Además cada nodo dispone de un conjunto de componentes funcionales básicos tales como:

autenticación, autorización, etc, y publica un conjunto de servicios Web, en lo que se denomina Interfaz de Interoperabilidad, para facilitar la integración con otros sistemas. También incorpora un portal de administración que integra todas las herramientas de gestión. La arquitectura funcional del nodo está inspirada en el estándar IMS DRI y las búsquedas de contenidos se realizan usando la especificación SQI. El modelo de referencia de la especificación IMS DRI se ha implementado a través de un servicio denominado DRI, cuya definición arquitectónica se puede ver en la figura 25.

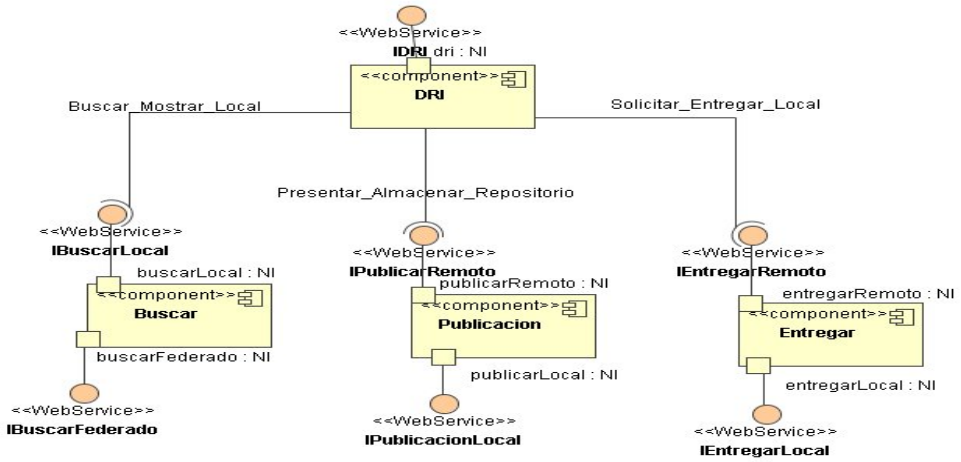

Figura 25. Arquitectura de la implementación de IMS DRI

En esta figura se pueden distinguir los servicios internos utilizados por el subsistema DRI, así como los tres servicios de la especificación, que pone a disposición del exterior. En este sentido para que un nodo sea interoperable y cumpla el protocolo DRI debe al menos de permitir almacenar objetos en él, buscarlos y obtener un objeto. Para almacenar objetos se ha definido el método presentar_almacenar y para recuperarlos el método solicitar_entregar, cuya descripción funcional se puede ver en las tablas:

Nombre Método	presentarAlmacenar		solicitarentregar	
Tipo retornado	Void		DataHandler	
Parámetros	Name	Type	Name	Type
	sessionID	String	sessionID	String
	pif	DataHandler	mec	String
Errores	NO_SUCH_SESSION METHOD_FAILURE		NO_SUCH_SESSION METHOD_FAILURE	

Tabla 29.Metodos IMS DRI

Con el fin de facilitar la interoperabilidad y que otros repositorios o sistemas puedan integrarse con Agrega, todos los servicios del subsistema DRI, se han publicado al exterior en forma de servicios web.

La búsqueda y obtención de objetos se realiza a través de SQI. El servicio SQI expone todos los métodos necesarios para realizar desde el exterior cualquier tipo de consulta sobre la plataforma, y para interoperar con otros repositorios digitales. Para ello dispone de una serie de métodos definidos, y mostrados en la tabla de más abajo. Cuando se realiza una query desde un origen ya sea de forma síncrona o asíncrona el nodo destino

debe devolver un conjunto de registros. La Query debe realizarse en un lenguaje previamente especificado mediante el método setQueryLanguage . Los resultados deben devolverse en un formato que responda a un esquema previamente conocido establecido mediante el método setReultsFormat. En caso de errores se devolverá una excepción. En Agrega los lenguajes permitidos son: VSQI y LQS, y el formato de vuelta debe responder al esquema que da forma al LOM-ES.

Método	Significado
Query Parameter Configuration	
setQueryLanguage	Establece el formato en el que realizar la query al nodo.
setResultsFormat	Este método permite definir el formato de resultados que se desea
setMaxQueryResults	Define el número máximo de resultados que una query puede devolver.
setMaxDuration	Este método establece un tiempo de time-out, sólo en caso que se desee operar de forma asíncrona.
Synchronous Query Interface	
setResultsSetSize	Define el máximo número de resultados que pueden devolverse en un conjunto simple de resultados.
synchronousQuery	Emplaza una consulta en el target.
getTotalResultsCount	Número total de resultados existentes de la última query realizada
Asynchronous Query Interface	
asynchronousQuery	Este método permite a la fuente enviar una query al target, mientras los resultados son retornados en un método asíncrono.
setSourceLocation	Este método debe ser llamado antes de enviar una query en modo asíncrono.
queryResultsListener	Este método inicializa el target, enviando los resultados a la fuente

Tabla 30.Métodos SQI

Por último mencionar que Agrega implementa el protocolo OAI-PMH (Open Archives Iniciative Protocol for Metadata Harvesting) para facilita la publicidad de los recursos digitales etiquetados que se encuentran disponibles en el repositorio. En Agrega se cubre el framework de exposición de metadatos implementando los métodos: GetRecord, Identify ,ListIdentifiers, ListMetadataFormats, ListRecords y ListSets. En este sentido el sistema devuelve una lista de metadatos en formato Dublin Core, tras haber realizado la transformación desde LOM-ES (sólo devuelve los contenidos digitales públicos y que hayan cambiado desde la última vez que se solicitaron)

3.4. Otras fuentes de información.

El portal EDRENE[EDRENE] ha realizado recientemente un estado del arte sobre los repositorios educativos digitales interoperables que existen a nivel europeo, el cual está accesible de manera gratuita desde su sitio web:

http://edrene.org/results/currentState/index.html

En dicho informe se hace un respaso país por país, de las iniciativas que existen en este momento.

4. La especificación "IMS Learning Object Discovery and Exchange"(IMS LODE)

IMS Learning Object Discovery and Exchange (LODE)[LODE 2010] es una especificación que facilita de manera global el descubrimiento, intercambio y recuperación de objetos de aprendizaje reutilizables. Para ello no define nuevos protocolos, sino que toma como base las especificaciones ya existentes que facilitaban parcialmente el intercambio y reutilización de objetos de aprendizaje tales como IMS Question and Test Interoperability, IMS Common Cartridge, IMS Content Packaging, IMS Simple Sequencing o IMS Learning Design, o aquellas que facilitaban el descubrimiento tales como IMS Learning Resource Metadata o IMS Vocabulary Definition Exchange. Así pues la especificación parte de los siguientes supuestos:

- Los objetos de aprendizaje se describen mediante metadatos del tipo IEEE LOM o Dublin Core. Para describir adecuadamente todos los aspectos de un objeto de aprendizaje puede que sean necesarias múltiples instancias de metadatos.
- Los metadatos pueden unirse para crear catálogos de objetos de aprendizaje sobre los que realizar búsquedas, de manera que la consulta de los catálogos se convierte en el camino principal para obtener la información necesaria para buscar objetos de aprendizaje, evaluar su utilidad y recuperarlos.
- Los catálogos de metadatos se almacenan en repositorios.
- Se pueden construir búsquedas sobre los catálogos de metadatos, usando APIs estándar tales como el SQI (Simple Query Interface) o SRU (Search/Retrieve with URL).
- Una forma de crear catálogos de metadatos es mediante el harvesting o recolección de los metadatos almacenados en los repositorios, usando protocolos como OAI-PMH (Open Archives Initiative – Protocol for Metadata Harvesting)

4.1. Modelos de datos en IMS LODE

La especificación IMS LODE consta de 3 modelos de datos:

- IMS LODE Information for Learning Object eXchange (LODE ILOX), es un modelo para organizar los metadatos de los objetos de aprendizaje para ser usados en el intercambio de datos. Permite estructurar los metadatos recolectados o recuperados en una búsqueda de manera que puedan exponerse y combinarse metadatos de diferentes orígenes(por ejemplo metadatos LOM, anotaciones Web 2.0 y metadatos de accesibilidad). Además permite hacer explicito diferentes usos del contenido.
- IMS LODE Learning Object Repository Registry data model (LODE Registry), es un modelo para colecciones de objetos de aprendizaje, usado para descubrir y configurar el acceso a dichas colecciones. Permite descubrir repositorios de objetos de aprendizaje en base a los propiedades de las coleccciones de objetos de aprendizaje y metadatos gestionadas, y sobre los protocolos soportados para exponer los metadatos. En este sentido proporciona la información necesaria para permitir una configuración automatizada de las aplicaciones de búsqueda y recolección que quieran acceder al repositorio.
- IMS LODE Context Set for the Contextual Query Language (LODE CQL), es un modelo para los atributos de los objetos de aprendizaje, usado para buscar mediante consultas significativas desde el punto de vista educativo.

La Figura 26 esquematiza el uso de la especificación IMS LODE. En la figura se puede observar un cliente LODE que descubre y recupera objetos de aprendizaje, los cuales se encuentran descritos mediante metadatos almacenados en repositorios. Los metadatos son expuestos para que sean usados por los protocolos de búsqueda o de recolección de metadatos (SQI, SPI, OAI-PMH). Con el fin de obtener acceso a los metadatos, el primer paso es descubrir los repositorios en los cuales están almacenados. El modelo LODE Registry permite describir repositorios, colecciones de objetos de aprendizaje, metadatos y protocolos soportados por un repositorio. De esta forma, los repositorios se registran en un registro central que puede ser accedido por los clientes LODE. Cuando se consulta el registro, se obtienen descripciones de los repositorios que contienen toda la información necesaria para conectar automáticamente con los repositorios y conseguir acceder a las colecciones de metadatos. Además, para aquellos repositorios que soportan protocolos de búsqueda tales como SQI o SRU, el modelo LODE CQL permite expresar consultas en términos de atributos de objetos de aprendizaje. Con independencia del protocolo usado para obtener los metadatos, ya sea por búsqueda o por recolección, usando LODE ILOX se puede organizar las diferentes instancias de metadatos para asegurar que toda la información necesaria para acceder a los objetos de aprendizaje se encuentra y está bien organizada.

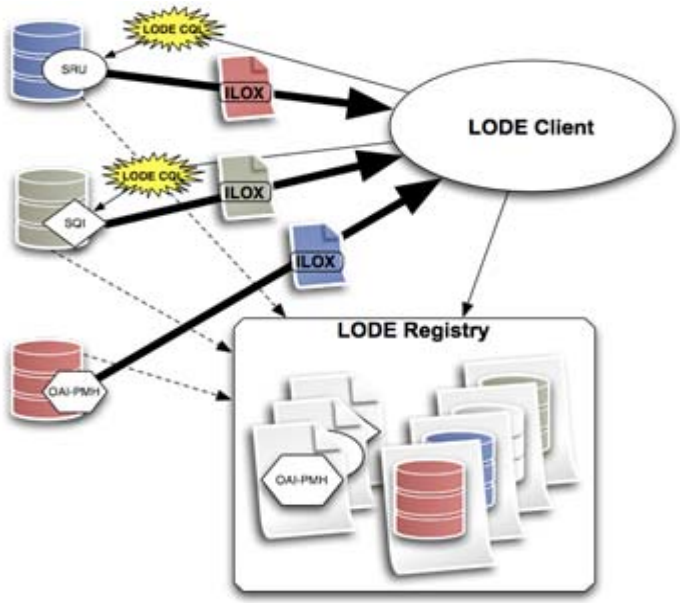

Figura 26. Esquema del uso de IMS LODE

La especificación define un conjunto de casos de uso cubiertos mediante los modelos de datos definidos. Cada caso de uso se identifica numéricamente, y pueden consultarse en el apéndice A. La tabla 31 muestra a modo de resumen los casos de uso cubiertos, los modelos de datos usados y las especializaciones de los casos de uso.

Caso de uso			Modelo de datos
Navegar			ILOX
Buscar	Busqueda federada		ILOX+Registry
	Búsqueda centralizada		
Consultar	Consulta básica		CQL
	Consulta avanzada		
	Por taxonomía	Criterio competencia	
		Criterio curriculum	
	Por tipos de metadatos específicos	Criterio accesibilidad	
		QTI	
		LOM	
Modo de consulta	Asincróna		Registry
	Síncrona		
Resultados búsqueda	Formato del resultado	LOM	ILOX
		Accesible	
		QTI	
Exposición metadatos	Registro en catálogo		ILOX
	Actualización en catálogo		
Descubrimiento repositorios	Describir en registro		Registry
	Actualizar en registro		
Anotación/ Recomendación/ Puntuación	Etiquetar		ILOX
Usos del contenido	Previsualizar		ILOX
	Ensamblar		
	Descargar		
	Utilización remota		
	Incluir en curso		
	Visualizar registro de metadatos completo		
	Seleccionar versión		
	Importar a entorno de aprendizaje		

Tabla 31. Resumen de casos de uso contemplados

4.2. IMS LODE LEARNING OBJECT REPOSITORY REGISTRY(LODE REGISTRY)

El objetivo de este modelo de datos es facilitar el acceso a las colecciones de objetos de aprendizaje almacenadas en los repositorio, y el intercambio de contenidos.Para ello es necesario disponer de metainformación mínima:

- Mecanismos de acceso a las colecciones.
- Evaluación de las colecciones.
- Descripción de las colecciones.
- Condiciones de acceso a las colecciones.
- Configuración automática para acceder a las colecciones.

El modelo ofrece una framework independiente de la configuración local de un registro , que cubre la metainformación mínima antes comentada. La framework definida se basa por una parte en el modelo ISO 2146[ISO, 09] que ofrece un modelo genérico para registros de contenidos independiente de cualquier protocolo o plataforma particular, y por otra parte en el perfil del proyecto ASPECT [ASP, 09] del modelo ISO 2146 que define una forma de describir registros de objetos de aprendizaje.

4.2.1. Elementos básicos de IMS LODE Registry.

Los elementos contemplados en el modelo son:

- **Objetos de aprendizaje**. Es cualquier entidad que pueda ser usada para enseñar o aprender. Usualmente se trata de objetos digitales.
- **Metadatos.** Es información que describe a un objeto de aprendizaje en un formato entendible por la maquina. En si mismo son objetos de aprendizaje.
- **Colecciones.** Es una agrupación de objetos de aprendizaje, gestionada conforme a un conjunto de normas como políticas de acceso o directivas acerca de que puede incluirse en una colección. Las colecciones pueden ser colecciones de contenido (objetos de aprendizaje y sus metadatos asociados) o colecciones de metadatos(solo los metadatos de los objetos de aprendizaje). Además una colección puede incluir otras colecciones o ser derivadas de otras colecciones. Sin embargo las relaciones de inclusión o derivación están restringidas, dado que colecciones de metadatos no pueden incluir o ser incluidas, o derivar o ser derivadas de colecciones de contenidos. Por otro lado observar que:
  - Un objeto de aprendizaje puede estar disponibles en diferentes colecciones.
  - Una colección de metadatos puede estar restringida a objetos de aprendizaje de determinadas colecciones de contenido.
  - Los metadatos de una colección de metadatos podrían ser distintos de los metadatos que acompañan a un elemento de una colección de contenido.
- **Repositorios.** Es el software especializado para gestionar colecciones. Observar que un repositorio no tiene porqué contener completamente una colección, dado que podría extenderse sobre varios repositorios, y de la misma forma un repositorio podría contener múltiples colecciones.
- **Partes.** Entidades que interactuan con las colecciones con el objetivo de llevar a cabo determinadas actividades. En las partes se incluye tanto a los usuarios finales (consumidores) como a los responsables de proporcionar las

colecciones(proveedores). En los proveedores se incluye a las partes que generan la colección, las que mantiene el repositorio y aquellas que gestionan el repositorio

- **Servicios.** Los repositorios soportan servicios que permiten a las partes acceder a los ítems de las colecciones que albergan. Los usuarios interactúan con los servicios a través de puntos de acceso a lo servicios. Los servicios siguen uno o más protocolos, y están sujetos a determinadas políticas de acceso que describen quién puede acceder y cuándo. Estos servicios incluyen: el Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH) [OAI, 04], el Simple Query Interface (SQI) [SQI, 05], el Search/Retrieval via URL(SRU) [SRU, 08], y el Simple Publishing Interface (SPI) [SPI, 08].

4.2.2. Modelo conceptual de IMS LODE Registry.

El modelo conceptual de IMS LODE Registry considera un registro como una agregación de colecciones, partes, servicios y actividades. Por tanto contempla entidades para las colecciones (**collection**) que pueden ser de metadatos(**Metadata collection)** o de contenidos(**Content collection)**, para las partes (**parties**) responsables de una colección y para los servicios (**services**) que interactúan con la colección. Los servicios se modelan mediante puntos de acceso al servicio (**targets**) , de manera que dos puntos de acceso al mismo servicio se modelará como dos entidades distintas. La descripción de los servicios incluye tanto información del protocolo(**protocol)** como información sobre las políticas de acceso(**Access policy**). Por otro lado, los detalles de configuración para los puntos de acceso al servicio individuales son modelados mediante descripciones de implementación del protocolo (**protocol implementation description**), y son particulares del protocolo usado .Observar que dos puntos de acceso pueden usar el mismo protocolo, pero pueden tener diferentes configuraciones para ese protocolo. En la figura 27 se representan las relaciones entre las entidades consideras en el modelo de datos.

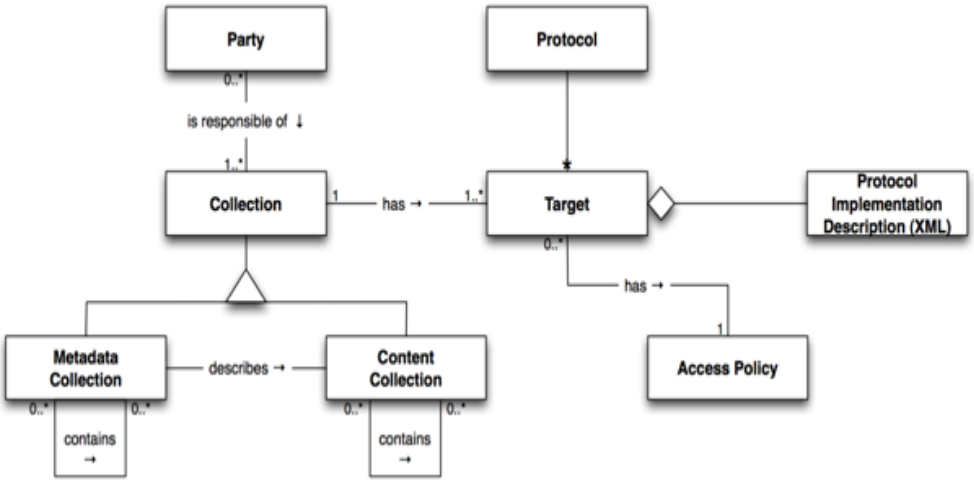

Figura 27. Relaciones entre las entidades consideradas en IMS Registry

Observar que el modelo no considera la entidad actividad, aunque si presupone un conjunto de actividades nucleares: creación de elementos de una colección, gestión de una colección y de los servicios del repositorio para la colección, descubrimiento y acceso a los elementos de una colección y a las propias colecciones. El objetivo de todas las actividades es interactuar con los elementos de una colección, sin embargo el modelo tampoco considera ese tipo de entidades pues ya existen especificaciones para tal fin.

4.2.3. Federaciones y búsqueda de colecciones.

Una federación es un agrupamiento de repositorios que permite a los usuarios acceder al contenido de cualquiera de los repositorios participantes. Para ello es necesario la existencia de un registro central donde se encuentren registrados todos los repositorios participantes. Existen dos modelos de federación, dependiendo de cómo el contenido de los repositorios se puede localizar:

- **Búsqueda federada**. Las peticiones para descubrir elementos de las colecciones, son llevadas de manera separada sobre cada repositorio participante y los resultados son agregados. Dado que las peticiones son separadas para cada repositorio, y el rol del registro es solo para agregar los resultados de la búsqueda, los repositorios pueden permanecer autónomos en sus políticas y en los perfiles de metadatos usados.
- **Búsqueda centralizada**. Las descripciones de los elementos buscables son copiadas desde los repositorios participantes en el repositorio central. Esto es realizado a través de un proceso de harvesting o recolección de metadatos, y puede ser implementado si los índices de los repositorios participantes son importados periódicamente al repositorio central, o si los repositorios participantes usan para iniciar la copia un protodolo de tipo push tal como SQI [SQI, 05] o SWORD [SWO, 08]. Las peticiones para el descubrimiento de elementos de las colecciones son realizadas sobre el repositorio central, el cual actúa de mediador del acceso del usuario al contenido de la federación.

El modelo LODE Registry soporta ambos tipo de federación, bajo el modelo de búsqueda federada se asume que la búsqueda será el principal servicio para interactuar con los repositorios, y bajo la búsqueda centralizada se asume que el principal servicio será el harvesteado o push. Ambos tipos de interacción pueden ser descritos a través de los metadatos de los servicios que se incluyen dentro de los metadatos de las colecciones, de manera que los metadatos de los servicios pueden ser usados para configurar las federaciones automáticamente. Observar que los metadatos de las colecciones facilitan la creación de las federaciones, dado que éstos se encuentran incluidos en los metadatos de los repositorios participantes. Así estos metadatos están escritos en términos de las colecciones que los repositorios publican, y cuyos metadatos pueden ser almacenados y añadidos al registro central.

Otro aspecto en el que intervienen los metados de las colecciones es en el propio **descubrimiento de las colecciones**. Por una parte a los usuarios les permite identificar las colecciones relevantes y por otra parte a las partes responsables gestionar el acceso a dichas colecciones. Los metadatos también permite a los usuarios navegar por las referencias de las colecciones de otras colecciones, por descripción (colecciones de metadatos describiendo colecciones de contenido) o por inclusión (coleccion conteniendo otras colecciones). Si las colecciones descritas son de acceso abierto, entonces las federaciones pueden unirse , permitiendo descubrir contenido a través de un rango de repositorios sin negociación explícita. Sin embargo, un elemento clave para que sea posible el descubrimiento a través de estos repositories es que se usen los mismos metadatos para describir sus ítems. Por último, las descripciones de las colecciones también permiten que las colecciones sean descubiertas automáticamente por un registro, siempre y cuando las colecciones estén disponibles en una localización que el registro conozca. Mientras esta situación presupone alguna negociación, el descubrimiento dinámico de colecciones permite a los registros ser más flexibles en el rango de federaciones soportadas. Observar que sin una estructura de federación, el acceso real necesitaría ser acordado en base a un acuerdo uno a uno

4.2.4. *Modelo de metainformación de IMS LODE Registry.*

El modelo de metainformación de IMS LODE Registry se describe en la figura 28(todos los metadatos pueden consultarse en el apéndice B).

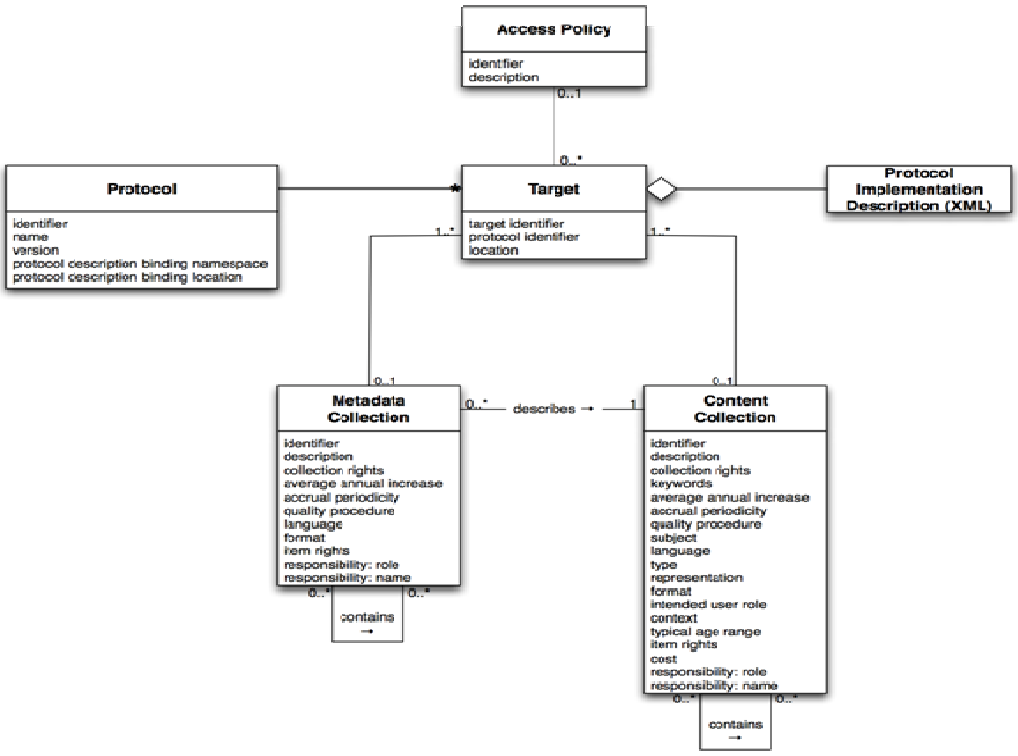

Figura 28. Modelo de metainformación de IMS LODE Registry.

En el modelo se establecen las siguientes relaciones entre sus elementos:

- Las colecciones de contenido y metadatos están asociadas con uno o más puntos de acceso a servicio.
- Una colección podría contener o estar contenida en cero o más colecciones del mismo tipo(metadatos en metadatos, contenidos en contenidos).
- Una colección de contenido podrías ser descrita por cero o más colecciones de metadatos, y una colección de metadatos podría describir cero o más colecciones de contenido. Observar que existe una diferencia entre metadatos describiendo colecciones de ítems, y metadatos describiendo los ítems que contienen. Aunque los metadatos se solapen, no son idénticos.
- Un punto de acceso a servicio está asociado con una única política de acceso. Sin embargo una política de acceso podría aplicarse a varios puntos de acceso a servicios.
- Un punto de acceso a servicio contiene una única descripción de la implementación del protocolo, puesto que un punto de acceso a servicio está definido para ser accedido a través de un solo protocolo. Un protocolo distinto significa un punto de acceso a servicio distinto.
- Para todas las relaciones, el objeto que hace referencia pueden referirse a los objetos relacionados por valor (incorporando la descripción del objeto relacionado en su propia descripción), o bien por referencia(añadiendo el identificador del objeto relacionado). Por ejemplo un punto de acceso a servicio, podría identificar su política de acceso por valor, lo que significa que incluye la descripción de la política de acceso en la descripción del punto de acceso a servicio, o bien por referencia, añadiendo el identificador para la política..

En la descripción del modelo se usan un conjunto de tipos de datos inspirados en IEEE LOM [LOM, 02]:

- **CharacterString:** texto , podría tener un número máximo de caracteres definidos.
- **LangString**: uno o más CharacterString con indicación del lenguaje que tiene cada CharacterString.
- **Identifier**: un tipo generic en IEEE LOM. En este modelo, toma la forma de un par formado por un **Entry** (CharacterString), identificador en si mismo, y un **Catalog** (CharacterString), indicando el espacio de nombres del identificador.
- **VocabularyTerm**: un par formado por un **Value** (CharacterString), dando un término tomado de un vocabulario fijado, y un **Source** (CharacterString), identificando el vocabulario fijado.
- **DateTime**: fecha especificada acorde con la ISO8601 [ISO8601, 04]

Observar que:

a) Muchas propiedades de las colecciones, son derivadas de las propiedades de sus ítems. A causa de que las colecciones no son homogéneas,las propiedades de una colección a menudo se aplican solo a un subconjunto de todos los ítems. Para reflejar esto, el modelo usa el tipo de datos **Property**, el cual establece la fortaleza de las propiedades en una colección. Un tipo Property tiene un **Source** y un **Value**, con el mismo significado que un **VocabularyTerm**, y un cuantificador **Strength**, el cual toma uno de los valores siguientes "**some**", "**most**", and "**all**". Esto significa que la propiedad

se aplica a algunos, muchos, o todos los items de una colección. El cuantificador es opcional, y el valor asumido por defecto es "some".

b) El modelo también usa los tipos de datos **Integer** y **anyURI**, ambos con el significado aceptado en XMLSchema. [XML, 04].

A continuación se describe el modelo:

Atributos para las colecciones

Las colecciones son modeladas usando Dublin Core Collections Application Profile (DCCAP) [DCCAP, 07] con contribuciones tomadas de la ISO 2146 y de IEEE LOM. Observar que aunque las colecciones de contenido y metadatos son ambas colecciones, sus atributos son modelados de forma diferente, para establecer diferencias contextuales entre los dos.

a) Colecciones de contenido.

Los siguientes atributos describen una colección de contenido como una entidad.

Identifier	Identificador de la colección de contenido	Identifier	Obligatorio
Description	Una descripción de la colección	LangString	Obligatorio
Collection Rights	Descripción de los derechos que se aplican a la colección como una entidad	LangString	Opcional
Keywords	Lista de palabras clave que describen a la colección	LangString	Opcional
Average Annual Increase	Incremento medio anual del número de objetos de aprendizaje en la colección. Puede ser un número negative.	Integer	Opcional
Accrual Periodicity	Atributo que usa un vocabulario controlado de DCCAP [DCCAP, 07] para describir con que frecuencia nuevos objetos de aprendizaje son añadidos o eliminados de la colección.	VocabularyTerm	Opcional

Tabla 32. Metadatos para colecciones de contenido

También se han definido atributos que describen propiedades a nivel de colección de los ítems de una colección de contenido.Es por ello que todos tienen el tipo de datos Property. Para poder describir correctamente las propiedades, todas las propiedades pueden ocurrir cero o más veces.

Quality Procedure	Procedimiento de calidad aplicado a los ítems de la colección.	Property	Opcional
Subject	Temas cubiertos por los objetos de aprendizaje de la colección.	Property	Opcional
Language	Lenguaje de los objetos de aprendizaje de la colección.	Property	Opcional
Type	Tipo de recurso educativo de los objetos de aprendizaje de la colección.	Property	Opcional
Representation	Representación en la que están disponibles los objetos de aprendizaje.	Property	Opcional
Format	Formato técnico de los objetos de aprendizaje de la colección.	Property	Opcional
Intended User Role	Rol previsto para los usuarios de los objetos de aprendizaje de la colección.	Property	Opcional
Context	El contexto de uso de los objetos de aprendizaje a la colección.	Property	Opcional
Typical Age Range	La edad media de los usuarios de los objet os de aprendizaje de la colección	Property	Opcional
Item Rights	Los derechos que se aplican a los objetos de aprendizaje de la colección. Es independiente de los derechos a nivel de colección, los cuales gobiernan el acceso a la colección de metadatos.	Property	Opcional
Cost	Coste de los objetos de aprendizaje de la colección.	Property	Opcional

Tabla 33. Metadatos para atributos para ítems.

b) Colecciones de metadatos.

Los siguientes atributos describen una colección de metadatos como una entidad.

Identifier	Identificador de la colección de metadatos..	Identifier	Obligatorio
Description	Una descripción de la colección de metadatos.	LangString	Obligatorio
Collection Rights	Descripción de los derechos que se aplican a la colección de metadatos como una entidad	LangString	Opcional
Keywords	Lista de palabras clave que describen a la colección	LangString	Opcional
Average Annual Increase	Incremento medio anual del número de instancias de metadatos en la colección. Puede ser un número negative.	Integer	Opcional
Accrual Periodicity	Atributo que usa un vocabulario controlado de DCCAP [DCCAP, 07] para describir con que frecuencia las instancias de metadatos son añadidas o eliminadas de la colección.	VocabularyTerm	Opcional

Tabla 34. Metadatos para colecciones de metadatos.

También se han definido atributos que describen propiedades de las instancias de metadatos de la colección. Todos tienen el tipo de datos Property, y pueden ocurrir cero o más veces.

Quality Procedure	Procedimiento de calidad applicado a los metadatos de la colección.	Property	Opcional
Language	Lenguaje de los metadatos de la colección.	Property	Opcional
Format	Formato de los metadatos de la colección.	Property	Opcional
Item Rights	Derechos que se aplican a los metadatos de la colección.	Property	Opcional

Tabla 35. Metadatos para atributos de instancias de metadatos.

Atributos para las Partes.

Describen las partes mediante la combinación de dos atributos: la responsabilidad (Responsibility) y los detalles de contacto(Contact Details).

Responsibility	Responsabilidad de la parte con respecto a la colección	VocabularyTerm	Obligatorio
Contact Details	VCARD [VCARD, 98] describiendo a la parte	CharacterString	Obligatorio

Tabla 36. Metadatos para atributos para las partes.

Atributos para los servicios.

a) Puntos de acceso a los servicios.

Describen los puntos de acceso a los servicios que ofrecen los repositorios para acceder a las colecciones.

Target Identifier	Identificador único del punto de acceso.	Identifier	Obligatorio
Protocol Identifier	Identificador del protocolo soportado por el punto de acceso.	Identifier	Obligatorio
Location	Localización del punto de acceso.	URI	Opcional
<extension>	Descripción XML de los parámetros del protocol soportados por el punto de acceso. Esta descripción es una instancia válida del XSD referenciado por el atributo Protocol Description Binding Location del protocolo identificado por el <Protocol Identifier>.	XML	Obligatorio

Tabla 37. Metadatos para puntos de acceso a los servicios.

b) Descripción de implementación del protocolo.

Describen las propiedades de una implementación específica del protocolo soportado por el punto de acceso, tales como si se usa SQI asíncrono o síncrono, o la elección del lenguaje de consultas. Esta descripción debería proporcionar toda la información necesaria para configurar un cliente para acceder al punto de acceso dado. Las descripción es específica para el protocolo, y debería ser una instancia válida del XSD referido en el atributo "Protocol Description Binding Location" del correspondiente protocolo.

c) Protocolo.

Describe la interface del servicio para interactuar con el punto de acceso. Debe incluir un puntero a la descripción de la implementación del protocolo.Observar que uno o más puntos de acceso pueden tener el mismo protocolo.

Identifier	Un identificador único de protocolo.	Identifier	Obligatorio
Name	Nombre del protocolo.	CharacterString	Obligatorio
Version	Versión del protocolo.	CharacterString	Opcional
Protocol Description Binding Name Space	Espacio de nombres usado para describir una implementación particular del protocolo.	URI	Obligatorio
Protocol Description Binding Location	Locación del XSD	URI	Obligatorio

Tabla 38. Metadatos para el protocolo.

d) Políticas de acceso.

Se describen con dos atributos:

Identifier	Identificador único de política	Identifier	Obligatorio
Description	Descripción de la política de acceso	LangString	Obligatorio

Tabla 39. Metadatos para políticas de acceso.

Observar que este modelo es genérico, y requiere que algunos aspectos sean desarrollados antes de poder ser usado. Concretamente es necesario:

- Especificar un tipo de Identificador.
- Seleccionar que atributos serán obligatorios, pues por defecto todos los atributos son opcionales.
- Seleccionar los vocabularios controlados para los diferentes atributos de tipo propiedad.
- Añadir los bindings XSD para las diferentes descripciones de protocolo.

4.3. IMS LODE LEARNING OBJECTO EXCHANGE(ILOX)

El objetivo final de la especificación LODE es especificar que se desea hacer con un objeto de aprendizaje que ha sido descubierto en una búsqueda (tal como se describe en los casos de uso). Para ello es esencial poder describir algunos aspectos de los objetos de aprendizaje (tales como las versiones, los formatos,etc) mediante diferentes instancias de metadatos. Con los estándares de metadatos usados (Dublin Core o LOM) no es posible describir los aspectos de un objeto en una única instancia de metadatos, y si se utiliza más de una instancia para describirlos no hay forma de relacionarlas. El modelo ILOX permite que cada aspecto de un objeto de aprendizaje sea descrito con distintas instancias de

metadatos, proporcionando un mecanismo para identificar qué aspecto se describe y que relaciones existen entre las descripciones.

4.3.1. Fundamentos.

Una materialización es un mecanismo de abstracción usado cuando se modela información para representar las relaciones entre una clase de categorías y una clase de ítems más concretos. Los atributos de las clases abstractas son heredados por las clases que los materializan.

El FRBR (Functional Requirements for Bibliographic Records) [FRBR, 98] es un modelo de representación de información que usa una materialización jerárquica basada en los siguientes conceptos:

- FRBR work es una creación distinguible de tipo intelectual o artística.
- FRBR expression es cada una de las versiones de un FRBR work.
- FRBR manifestation es cada una de las implementaciones de un FRBR expression de un FRBR work.
- FRBR ítem es cada una de las copias de un FRBR ítem.

El modelo FRBR puede ser útil para describir los aspectos de un objeto de aprendizaje relevantes para diferentes contextos. En la figura 29, se muestra un ejemplo de aplicación del FRBR a un objeto de aprendizaje sobre nutrición donde:

1. El FRBR work es el objeto de aprendizaje sobre nutrición.
2. Las versiones del objeto de aprendizaje en inglés y francés son FRBR expressions del FRBR work
3. Los paquetes en IMS Content Package y en IMS Common Cartridge de cada FRBR expression anterior considerada son FRBR manifestations.
4. Cada copia del paquete IMS Common Cartridge de la versión inglesa del objeto de aprendizaje es un FRBR ítem de dicho FRBR manifestation.

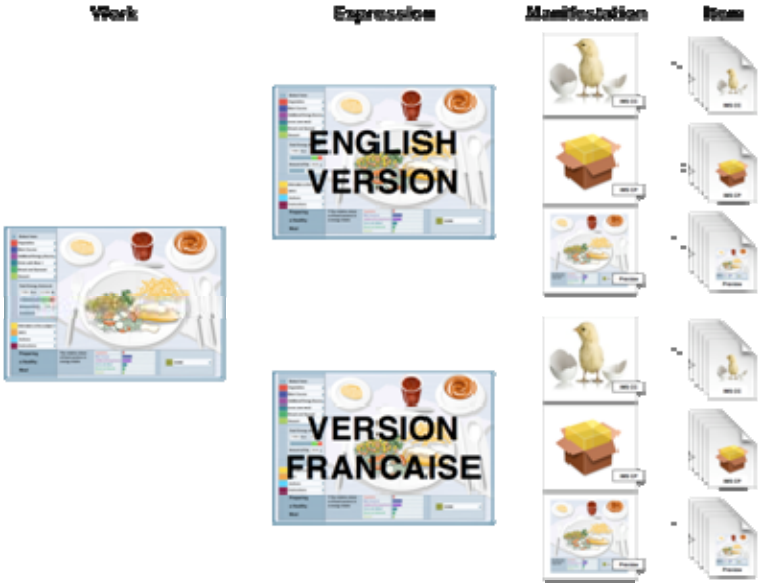

Figura 29. Ejemplo de aplicación del FRBR a un objeto de aprendizaje

4.3.2. Modelo conceptual

El modelo de datos ILOX estructura la descripción de un objeto de aprendizaje como una materialización jerárquica de los aspectos FRBR de un objeto de aprendizaje. En la figura 30 se muestra un diagrama de clases del modelo

- La clase LO Copy es una materialización de la clase LO Package, y modela el aspecto FRBR ítem de un objeto de aprendizaje
- La clase LO Package es una materialización de la clase LO Version, y modela el aspecto FRBR manifestation de un objeto de aprendizaje
- La clase LO Version es una materialización de la clase Learning Object, y modela el aspecto FRBR expression de un objeto de aprendizaje
- La clase Learning Object modela el aspecto FRBR work de un objeto de aprendizaje.

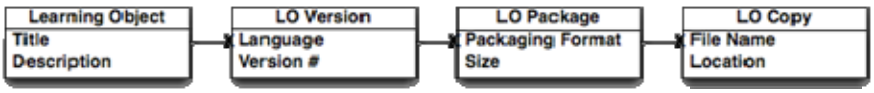

Figura 30. Diagrama de clases modelando las materializaciones.

En el modelo, los aspectos FRBR son usados de la siguiente forma.

- Work. Es una visión abstacta de un objeto de aprendizaje que captura los puntos comunes entre todas las posibles variaciones de este objeto.
- Expressions. Son usadas para capturar información específica de las diferentes versiones, drafts, traducciones y localizaciones de los objetos de aprendizaje.
- Manifestations. Son usadas para capturar información específica de la forma que una expression de un objeto de aprendizaje es codificada y presentada.
- Items. Son usados para capturar información específica de las copias concretas de un objeto de aprendizaje

Una instancia ILOX puede finalizar en cualquier nivel de la jerarquía dependiendo del nivel de concreción o abstracción que sea necesario. Así mantener descripciones de objetos de aprendizaje a:

- Nivel Work, permite una entrada por cada objeto de aprendizaje sin distinción entre versiones del mismo objeto de aprendizaje.
- Nivel Expression, permite una entrada por cada versión de un objeto de aprendizaje sin distinción entre diferentes formatos de una versión de un objeto de aprendizaje dado, y sin poder decidir a que work las expressions pertenecen.
- Nivel Manifestation, permite una entrada por cada formato de un objeto de aprendizaje sin distinción entre diferentes copias de un objeto de aprendizaje, y sin poder decidir a que work o expression, los manifestation pertenecen.
- Nivel ítem, permite una entrada por cada copia de un objeto de aprendizaje, sin poder decidir a que work, expression o manifestation, los ítems pertenecen.

En cada nivel de la jerarquía, existe un patrón común usado para modelizar el correspondiente aspecto FRBR del objeto de aprendizaje. En la figura ,se muestra el diagrama de clases del patrón para un nivel FRBR dado(clase "LO at FRBR level #n"), y en la que aparecen:

- Identificadores opcionales(modelizado por el atributo Identifier)
- Descripciones consistentes de metadatos del nivel específico(modelizado por la clase Description)
- Información adicional del nivel específico(modelizado por el atributo "Level specific features").
- Información acerca del nivel FRBR inmediatamente más bajo (modelizado por la clase "LO at FRBR level #n-1")

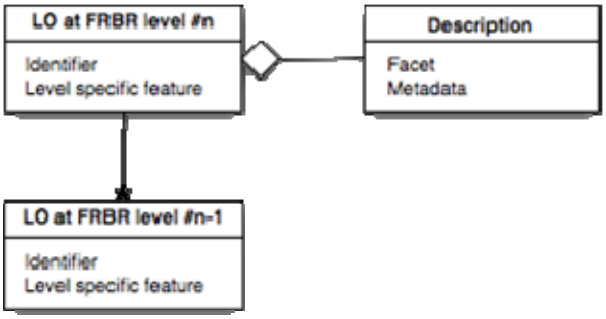

Figura 31. Diagrama de clases del patrón de un aspecto FRBR

Observar:

- Algunos tipos de información son típicos de un nivel de materialización FRBR. Por ejemplo el lenguaje de un objeto es una característica de una expression(un objeto puede ser traducido en diferentes lenguajes sin convertirse en un diferente work), aunque diferentes manifestations de una misma expression se esperan que estén en el mismo lenguaje.
- Hay información que puede aparecer en múltiples niveles de materialización. Por ejemplo, los derechos de acceso pueden aplicarse a todas las copias de un work, o podrían ser especificados a un manifestation particular.
- La misma información de un nivel de materialización más bajo sobreescribe a la información que aparece en un nivel más alto.Por ejemplo los derechos de acceso pueden ser configurados para un work como entidad, pero derechos de acceso más específicos pueden ser especificados como una excepción a nivel de manifestation.

4.3.3. Tipos de datos

El modelo se basa en dos tipos XML primitivos: normalizedString y anyURI , con los que se construye los tres tipos básicos del modelo:

- El tipo ***Identifier*** es el tipo de los identificadores de elementos, y consta de dos subelementos:
  - **catalog** que indica el espacio de nombres del identificador(normalizedString).
  - **entry** que es la cadena del identificador en si misma(normalizedString)

- El tipo **ValueFromVocab** es usado por los elementos que contienen vocabulario controlado, y consta de dos subelementos:
  - **vocabularyID** que identifica a un vocabulario controlado (normalizedString).
  - **value** que contiene un término tomado de un vocabulario controlado(normalizedString)
- El tipo Metadata proporciona puntos de extensión para los registros de metadatos XML embebebidos dentro de la estructura ILOX. Consta de dos subelementos:
  - **schema** que identifica el esquema de metadatos para el registro XML embebido. El esquema es identificado por su espacio de nombre(anyURI))
  - **metadata** que son los metadatos propiamente dichos.

Asi mismo existen dos tipos de datos derivados:

- Eltipo **Description,** usados para describir cada nivel FRBR de un objeto de aprendizaje con metadatos específicos del nivel. Constan de dos subelementos:
  - **Facet** que indica qué faceta de un aspecto FRBR dado de un objeto de aprendizaje es descrito(ValueFromVocab).
  - **Metadata** que contiene una descripción de metadatos de un aspecto de un FRBR dado del objeto de aprendizaje(metadata).

  Observar:
  - Cada nivel puede tener multiples descripciones de metadatos, y cada una de ellas con su propia faceta para diferenciar entre ellas. En ILOX no se define ningún vocabulario controlado para las facetas, y se espera que los perfiles de aplicación de LODE proporcionen los vocabularios controlados a usar. Estos vocabularios probablemente diferirán en cada perfil de aplicación y en cada nivel FRBR.
  - Un elemento faceta se puede usar para distinguir entre los proveedores de metadatos indicando por ejemplo si es información generada por el usuario o son metadatos del proveedor, o para ndicar el propósito del registro de metadatos embebido, indicando por ejemplo “descripción general”, “descripción de derechos”, “perfil de accesibilidad”.
  - La cantidad de información que se adjunto a un elemento description varía según el escenario que se considere. Por ejemplo cuando se recolecciona información, puede que interese intercambiar descripciones del objeto de aprendizaje tan completas como sea posible, pero sin embargo cuando se está buscando, podría ser interesante limitar la cantidad de información transportada que va a ser mostrada al usuario.
  - A causa de la importancia de los derechos de los metadatos, las licencias disponen de su propio faceta denominada licenseType, que incluye un identificador de licencia y la descripción de los metadatos.La faceta de la licencia puede aplicarse a cualquier nivel, y aplicarse a todas las materializaciones en los niveles más bajos, a menos que sea sobreescrita por una faceta de licencia distinta. Además más de una licencia podría ser aplicada al mismo objeto.

4.3.4. El aspecto Work.

Es uno de los posibles elementos raíz de una instancia ILOX, y se trata de una vista abstracta de un objeto de aprendizaje que trata de capturar aquellos aspectos comunes a todas las versiones de un objeto de aprendizaje. Consta de:

- Uno o más identificadores opcionales(Identifier Type).
- Uno o más descripciones opcionales(Description Type).
- Al menos una **expression** ILOX.

Observar:

- El identificador únicamente identifica a un ***work*** propiamente dicho, y no a una versión, formato o copia particular del objeto de aprendizaje.
- La descripción debería ser usada para describir las diferentes facetas de un ***work***, de manera que información específica de una versión particular, formato o copia del objeto no debería aparecer a este nivel.

4.3.5. El aspecto Expression

Una ***expression*** ILOX puede ser un elemento raíz de una instancia ILOX o bien ser un subelemento de un ***work***. Se usan para capturar información específica de las diferentes versiones, drafts, traducciones y localizaciones de los objetos de aprendizaje. Consta de:

- Uno o más identificadores opcionales(Identifier Type).
- Uno o más descripciones opcionales(Description Type).
- Al menos un manifestation ILOX.
- Uno o mas dimensión opcionales(Dimension Type) que explicitan los critrios por los que las ***expressions*** de un mismo ***work*** difieren. Este elemento es opcional si la ***expression*** es un elemento raíz, y es obligatorio si la expression es un subelemento de un ***work***.

Observar:

- El identificar únicamente identifica a una ***expresssion*** propiamente dicho(una versión dada de un objeto de aprendizaje). Cada identificador diferirá de los identificadores de otras ***expression*** del mismo objeto de aprendizaje y del identificador del ***work***.
- La descripción debería ser usada para describir las diferentes facetas de una ***expression***. Además si la ***expression*** es un elemento raíz, entonces la información del aspecto ***work*** debería ser expresada a nivel de la ***expression***.
- El elemento **Dimension** es del tipo **Dimension**, el cual consta de dos subelementos:
  - **name**, que indica el nombre de la dimensión de la expression(ValueFromVocab). Por ejemplo podría ser “lenguaje”, indicando que las diferentes expressions del objeto de aprendizaje están expresadas en diferentes lenguajes.
  - **parameter**, define la posición actual de una expression en la dimensión que está siendo considerada(ValueFromVocab). Por ejemplo el valor actual del lenguaje.

4.3.6. El aspecto Manifestation

Un ***manifestation*** ILOX puede ser la raiz de una instancia ILOX o un subelemento de una ***expression***. Son usados para capturar información específica de la forma en que una expression de un objeto de aprendizaje es codificado y presentado. Consta de:

- Uno o más identificadores opcionales(Identifier Type).
- Uno o más descripciones opcionales (Description Type).
- Al menos un ***ítem*** ILOX.
- Un ***nombre*** opcional para el manifestation (ValueFromVocab). Es opcional si el manifestation es la raiz de la instancia ILOX, y obligatorio si un subelemento de una expression. Algunos posibles nombrs son:
  - preview: Una previsualización de una expression.
  - thumbail: Un thumbail de una expression.
  - metadata in: Una descripción de metadatos completa del objeto de aprendizaje.
  - experience: Es un manifestation de un objeto de aprendizaje que está disponibel online y puede ser visualizado desde un navegador web.
  - package in: Es un manifestation de un objeto de aprendizaje que está empaquetado de acuerdo un formato de empaquetamiento de contenido dado.
- Un parámetro opcional que proporciona información adicional del tipo de manifestation (ValueFromVocab). Solo puede aparecer un parámetro, y sus posibles valores dependen del nombre dado:
  - preview.
    No está definido.
  - thumbail.
    Todos los tipos MIME para imágenes, es decir del estilo "image/"
  - metadata in.
    Todos los espacios de nombre válidos de esquemas de metadatos usados en un registro completo de metadatos.
  - Experience.
    Con este nombre es opcional el uso de un parámetro. Pero si se usa, debe contener el tipo MIME de los ficheros referidos en el elemento "item.location.uri". También son aceptables los tipos MIME basados en IANA registration[IANA, 07]
  - package in.
    Es el vocabulario controlado usado para describir los diferente formatos de paquete. Se muestra en la tabla 40.

Termino	Descripción
imscp_v1p0	IMS Content Packaging Version 1.0
imscp_v1p1	IMS Content Packaging Version 1.1
imscp_v1p1p1	IMS Content Packaging Version 1.1.1
imscp_v1p1p2	IMS Content Packaging Version 1.1.2
imscp_v1p1p3	IMS Content Packaging Version 1.1.3
imscp_v1p1p4	IMS Content Packaging Version 1.1.4
imscc_v1p0	IMS Common Cartridge Version 1.0
imscc_v1p1	IMS Common Cartridge Version 1.1
scorm_v1p0	SCORM Version 1.0
scorm_v1p1	SCORM Version 1.1
scorm_v1p2	SCORM Version 1.2
scorm_v2004	SCORM 2004
imsqti_v1p2lite	IMS Question & Test Interoperability Version 1.2 Lite
imsqti_v1p2	IMS Question & Test Interoperability Version 1.2
imsqti_v1p2p1	IMS Question & Test Interoperability Version 1.2.1
imsqti_v2p0	IMS Question & Test Interoperability Version 2.0
imsqti_v2p1	IMS Question & Test Interoperability Version 2.1

Tabla 40. Vocabulario para formatos de paquete

Observar:

- El identificar únicamente identifica a un ***manifestation*** propiamente dicho(un formato dado de una versión dada de un objeto de aprendizaje). Cada identificador diferirá de los identificadores de otros ***manifestation*** del mismo objeto de aprendizaje y del resto de identificadores del ***work*** y de las ***expressions***.
- La descripción debería ser usada para describir las diferentes facetas de un ***manifestation***. Además si el ***manifestation*** es un elemento raíz, entonces la información del aspecto ***work*** y de las ***expressions*** debería ser expresada a nivel del ***manifestation***.
- Cada combinación nombre+parámetro es única. Es decir, que sólo es posible tener diferentes ***manifestations*** de una misma ***expression*** con el mismo nombre si tienen diferentes valore de parámetro.

4.3.7. El aspecto Item.

Un ***item*** ILOX puede ser la raíz de una instancia ILOX o bien ser un subelemento de un ***manifestation***. Se usan para capturar información específica de las copias concretas de los objetos de aprendizaje. Constan de:

- Uno o más identificadores opcionales(Identifier Type).
- Una o más despcripciones opcionales(Description Type).
- Una característica específica del ***ítem*** denominado ***location***(Location Type) que proporciona información acerca de la localización actual del ***ítem***.

Observar:

- El ***ítem*** es nivel más bajo de la jerarquía ILOX, y no necesita de ningún elemento de modelización de un nivel menor.
- El identificador identifica únicamente a un ***ítem***(una copia de un objeto de aprendizaje). Cada identificador será diferente de los identificadores del resto de copias, y del resto de identificadores de ***manifestations***, ***expressions*** y ***works***.
- La descripción debería ser usada para describir las diferentes facetas del ***ítem***. Además si el ***ítem*** es un elemento raíz,la información de los aspectos del ***work***,

expression y ***manifestation*** del objeto de aprendizaje son expresadas a nivel de ***ítem***.

- El elemento ***location*** es del tipo ***Location***,el cual consta de:
  - ***URI*** de tipo anyURL. Es un elemento obligatorio que debe contener la información de cómo alcanzar el recurso tal como la página de autenticación, o información sobre el modelo de distribución. Debe tratarse de una localización alcanzable del tipo URL, URL persistente, o cualquier otra entidad que permita alcanzar dicha localización.
  - ***Metadata*** de tipo Metadata Type.

Observar que si la localización representada en el elemento URI es la localización del ítem propiamente dicho, el elemento Metadata no es usado. Por otro parte, si la URI no contiene la localización del objeto de aprendizaje(por ejemplo en objetos de aprendizaje comerciales con acceso controlado), entonces el elemento Metadata es usado para describir el modelo de distribución ofrecido por el proveedor del objeto.

4.3.8. Perfilado del modelo de datos

Este modelo es genérico, y requiere que algunos aspectos sean desarrollados antes de poder ser usado. Concretamente es necesario:

- Seleccionar el elemento raíz ILOX(Work, Expression, Manifestation, or Item).
- Hace obligatorios uno o más de los elementos opcionales ILOX.
- Seleccionar los vocabularios controlados para los elementos facet y description.
- Seleccionar el esquema de metadatos permitido para los diferentes puntos de extensión de metainformación que permite el esquema de ILOX.

4.4. IMS LODE CONTEXT SET FOR CQL

Las búsquedas de objetos de aprendizaje siguen el modelo definido en CQL[CQL, 08],de manera que se usan los atributos de los objetos de aprendizaje como términos de las búsquedas. Así el modelo se caracteriza por:

- Los atributos de un objeto de aprendizaje están disponibles para buscar a través de los denominados **puntos de acceso (access point)**, existiendo uno por cada atributo diferente considerado(por ejemplo "competencia" es un punto de acceso para descubrir contenido basado sobre las competencias soportadas por el contenido). Las consultas de una búsqueda pueden ser construidas combinando uno o más puntos de acceso. En la tabla 41 aparecen los metadatos referidos a los puntos de acceso.

Punto Acceso	Definición	Fuente Definición
Registro completo	Índice del texto complete de los metadatos para un recurso.	—
Título	Nombre dado al recurso	DC
Autor	Entidad principal de hacer el recurso.	DC
Publicador	Cualquier entidad responsablle de hacer disponible el recurso.	DC
Contribuidor	Cualquier entidad responsible de hacer contribuciones al recurso.	DC
Palabras clave	Tópicos del objeto de aprendizaje	DC
Disciplina	Clasificación de un objeto de aprendizaje de acuerdo a su uso por los departamentos, facultades o unidades en una organización	LOM (CanCore): 9. Classification donde proposito = disciplina
Competencia	Clasificación de un objeto de aprendizaje de acuerdo a su uso para desarrollar habilidades, conocimientos, tareas y resultados del aprendizaje	LOM (IMS RDCEO): 9. Classification donde propósito = competencia
Lenguaje	Lenguaje del recurso	DC
Coste	Si el recurso requiere pagar algo.	LOM: 6.1 Cost
Licencia	Identificador de una licencia bajo la cual el recurso se hace disponible.	—
Tipo de recurso de aprendizaje	Tipo específico del objeto de aprendizaje.	LOM: 5.2 Learning Resource Type
Papel predefinido para el usuario final	Usuarios principales para los cuales está diseñad el objeto de aprendizaje.	LOM: 5.5 Intended End User Role
Contexto	Entorno principal dentro del cual se espera que el aprendizaje y uso del objeto de aprendizaje tenga lugar.	LOM: 5.6 Context
Edad títpica	Edad del típico usuario final	LOM: 5.7 Typical Age Range
Fecha	Fecha de publicación	DC
Versión	Versión del objeto de aprendizaje.	LODE: ILOX Expression dimension parameter donde nombre de dimensión= versión
Nombre del manifestation	Nombre del manifestation.	LODE: ILOX Manifestation name
Parámetro del manifestation	Parámetro del manifestation.	LODE: ILOX Manifestation parameter

Tabla 41. Metadatos referidos a los puntos de acceso.

- Se definen los **modificadores (modifiers)** para ciertos puntos de acceso con el objetivo de hacer más específicas las consultas en caso de que los puntos de acceso puedan contener múltiples valores (por ejemplo el modificador language puede ser usado en conjunción con el punto de acceso title para construir una búsqueda que solo considere títulos en francés).En la tabla 42 aparecen los metadatos referidos a los modificadores.

Modificador	Definición	Fuente Definición
Lenguaje	Restringe la búsqueda a valores en el lenguaje especificado.	LOM
Fuente	Restringe la búsqueda a valores del vocabulario específicado.	bib.classAuthority [CQLBIB, 09]. restriction condiciona los vocabularios controlados a las autoridades de clasificación bibliográfica como las listadas en [MARC, 03].

Tabla 42. Metadatos referidos a los modificadores.

- No es obligatorio que los puntos de acceso tengan un único valor por objeto, de manera que si no se especifica ningún modificador, todos los posibles valores del punto de acceso son considerados cuando se lleva a cabo la búsqueda.
- Se define un **criterio de ordenación(sort criteria)**, que se utiliza para indicar como ordenar el resultado generado por la búsqueda. En la tabla 43 aparecen los metadatos referidos al criterio de ordenación.

Criterio Ordenación	Definición	Fuente Definición
Fecha Modificación	Ordenado en orden inverso de la última modificación de fecha/hora.	—
Calificación	Ordenado en orden inverso de la calificación media de los usuarios.	—
Relevancia	Ordenado en orden de relevancia del criterio de búsqueda especificado.	—

Tabla 43. Metadatos para indicar el criterio de ordenación

Observar que:

- La implementación de los puntos de acceso a través de índices de búsqueda enlaza los puntos de acceso a un protocolo de búsqueda existente, de manera que el modelo de datos puede ser enlazado a otros protocolos de búsqueda. Sin embargo el modelo de datos está diseñado para ser usado a través de CQL [CQL, 08], y sus puntos de acceso y modificadores corresponden a los índices CQL y modificadores de la relación, a través de un binding con CQL. Este binding enlaza los puntos de acceso a índices desde cuatro conjuntos contextuales separados.No se espera que todos procedan de un simple conjunto contextual CQL.
- La semantica de los puntos de acceso y los modificadores se basa en el modelo de LOM [LOM, 02], pero no existe ninguna obligatoriedad acerca de que los metadatos deban estar codificados en LOM. De hecho cualquier esquema de metadatos que pueda ser adaptado para coincidir con la semántica de los puntos de acceso, puede ser utilizado.
- Las fuentes de las definiciones siguen el orden jerárquico: Dublin Core [DCME, 08] donde sea aplicable, semántica de LOM donde Dublin Core no es lo suficiente específico, CanCore [CANCORE, 04] o IMS RDCEO [RDCEO, 02]donde LOM no es definitivo(por ejemplo en algunos de sus vocabularios).
- Algunos de los puntos de acceso corresponden directamente a campos de LOM que están condicionadas a un vocabulario. Aunque estos vocabularios no están incluidos en el modelo, se espera que cualquier perfil de la especificación los use por defecto.
- Los metadatos subyacentes a los puntos de acceso puede ser refinados con el fin de limitar los valores de los vocabularios. En particular el identificador utilizado para el punto de acceso de la Licencia debe ser adaptado a un esquema particular

.

5. Desarrollo de un caso de uso de IMS LODE para el contexto de un repositorio digital de material de lógica matemática.

En este capítulo se presenta un caso de uso de la especificación en el contexto de un repositorio digital de material de lógica matemáticas. Para ello se va introduciendo el ejemplo, y cómo se aplica la especificación a cada uno de los aspectos.

a) IMS LODE Registry

Caso de uso:

Medatato	**Valor**
identifier	Nombre Catálogo:Uoc-Catalog Valor:Meta-02
description	"Colección de problemas de lógica de enunciados"
collectionRights	Creative Commons Reconocimiento-Compartir Igual
keywords	Logica de enunciados, Deducción natural
averageAnnualIncrease	100
accrualPeriodicity	freq:annual [DCCAP]
qualityProcedure	No se aplica procedimiento de calidad, all [UOC-Vocabulario]
language	Castellano,most[UOC-Vocabulario]
format	text/plain,all[Mime Types]
itemRights	No hay derechos definidos, all[UOC-Vocabulario]
responsible	Contacto Autor[UOC Vocabulario], Antonio Sarasa
target	Nombre Catálogo:URI Valor:http://www.uoc.es/lógica-enun-oaiphm
toContentCollection	Nombre Catálogo:Uoc-Catalog Valor:Content-02
contains	Nombre Catálogo:Uoc-Catalog Valor:Meta-01
isContained	Nombre Catálogo:Uoc-Catalog Valor:Meta-03

<metadataCollection>
<identifier>
<catalog> Uoc-Catalog </catalog>
<entry> Meta-02</entry>
</identifier>
<description>
<language> es </language>
<string> Colección de problemas de lógica de enunciados </string>
</description>
< collectionRights >
<language> es </language>
<string> Creative Commons Reconocimiento-Compartir Igual </string>
</ collectionRights >
< keywords >
<language>es </language>
<string> Logica de enunciados </string>
</ keywords >
< keywords >
<language>es </language>
<string> Deducción natural </string>
</ keywords >
< averageAnnualIncrease > 100 </ averageAnnualIncrease >
< accrualPeriodicity >
<vocabularyID> DCCAP </vocabularyID>
<value> freq:annual </value>
</ accrualPeriodicity >
< qualityProcedure >
<vocabularyID>UOC-Vocabulario </vocabularyID>
<value> No se aplica procedimiento de calidad </value>
< strength > all </ strength >
</ qualityProcedure >
< language >
<vocabularyID> UOC-Vocabulario </vocabularyID>
<value> castellano </value>
< strength > most </ strength >
</ language >
< format >
<vocabularyID> Mime Types </vocabularyID>
<value> text/plain</value>
< strength > all </ strength >
</ format >
< itemRights >
<vocabularyID>UOC-Vocabulario </vocabularyID>
<value> No hay derechos definidos</value>
< strength >all </ strength >

```
</ itemRights >
< responsible >
    <responsibility>
        <vocabularyID> UOC-Vocabulario</vocabularyID>
        <value> Contacto autor</value>
    </responsibility >
    < vcard > BEGIN:VCARD VERSION:2.1 N:Antonio Sarasa FN:4758
              ORG:Universitat Oberta de Catalunya TITLE:Consultor
              END:VCARD
    </ vcard >
</ responsible >
< target >
  < targetIdentifier >
        <catalog>URI </catalog>
        <entry> http://www.uoc.es/lógica-enun-oaiphm </entry>
  </ targetIdentifier >
</ target >
< toContentCollection >
    <contentCollectionIdentifier>
            <catalog> Uoc-Catalog </catalog>
            <entry> Content-02</entry>
    </contentCollectionIdentifier>
</ toContentCollection >
< contains >
    <metadataCollectionIdentifier>
            <catalog> Uoc-Catalog </catalog>
            <entry> Meta-01 </entry>
    </metadataCollectionIdentifier>
</ contains >
< isContained >
    <metadataCollectionIdentifier>
            <catalog> Uoc-Catalog </catalog>
            <entry> Meta-03</entry>
    </metadataCollectionIdentifier>
</ isContained >
</metadataCollection>
```

Caso de uso:

Medatato	**Valor**
identifier	Nombre Catálogo:Uoc-Catalog Valor:Content-02
description	“Colección de problemas de lógica de enunciados”
collectionRights	Creative Commons Reconocimiento-Compartir Igual
keywords	Logica de enunciados, Deducción natural
averageAnnualIncrease	100
accrualPeriodicity	freq:annual [DCCAP]
qualityProcedure	No se aplica procedimiento de calidad, all [UOC-Vocabulario]
language	Castellano ,most[UOC-Vocabulario]
format	text/plain,all[Mime Types]
itemRights	No hay derechos definidos, all[UOC-Vocabulario]
responsible	Contacto Autor[UOC Vocabulario], Antonio Sarasa
target	Nombre Catálogo:URI Valor:http://www.uoc.es/lógica-enun-search
subject	Deducción natural, all[UOC-Vocabulario]
type	Objeto de aprendizaje, all[UOC-Vocabulario]
representation	imscp_v1p0, all[UOC-Vocabulario]
intendedUserRole	Estudiante, all[UOC-Vocabulario]
context	Grado informatica, all[UOC-Vocabulario]
typicalAgeRange	20-50, all[UOC-Vocabulario]
cost	No tiene coste, all[UOC-Vocabulario]
toContentCollection	Nombre Catálogo:Uoc-Catalog Valor:Meta-02
contains	Nombre Catálogo:Uoc-Catalog Valor:Content-01
isContained	Nombre Catálogo:Uoc-Catalog Valor:Content-03

```
<contentCollection>
  <identifier>
    <catalog> Uoc-Catalog </catalog>
    <entry> Content-02 </entry>
  </identifier>
  <descrition>
    <language> es </language>
    <string> Colección de problemas de lógica de enunciados </string>
```

```
</description>
 < collectionRights >
    <language> es </language>
    <string> Creative Commons Reconocimiento-Compartir Igual </string>
</ collectionRights >
 < keywords >
    <language> es </language>
    <string> Logica de enunciados </string>
</ keywords >
 < keywords >
    <language> es </language>
    <string> Deducción natural </string>
</ keywords >
 < averageAnnualIncrease > 100 </ averageAnnualIncrease >
 < accrualPeriodicity >
    <vocabularyID> DCCAP  </vocabularyID>
    <value> freq:annual </value>
</ accrualPeriodicity >
 < qualityProcedure >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> No se aplica procedimiento de calidad </value>
    < strength >all </ strength >
</ qualityProcedure >
 < language >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value>Castellano</value>
    < strength > most</ strength >
</ language >
 < format >
    <vocabularyID> Mime Types </vocabularyID>
    <value> text/plain </value>
    < strength > all </ strength >
</ format >
 < itemRights >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> No hay derechos definidos </value>
    < strength > all </ strength >
</ itemRights >
 < responsible >
     <responsibility>
         <vocabularyID> UOC Vocabulario </vocabularyID>
         <value> Contacto Autor </value>
     </responsibility >
     < vcard > BEGIN:VCARD VERSION:2.1 N:Antonio Sarasa FN:4758
                 ORG:Universitat Oberta de Catalunya TITLE:Consultor
```

```
                END:VCARD
    </ vcard >
</ responsible >
 < target >
  < targetIdentifier >
        <catalog> URI </catalog>
        <entry> http://www.uoc.es/lógica-enun-search </entry>
  </ targetIdentifier >
</ target >
 < subject >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> Deducción  natural </value>
    < strength > all </ strength >
</ subject >
 < type >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> Objeto de aprendizaje </value>
    < strength > all </ strength >
</ type >
 < representation >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> imscp_v1p0</value>
    < strength > all </ strength >
</ representation >
 < intendedUserRole >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> Estudiante </value>
    < strength > all </ strength >
</ intendedUserRole >
 < context >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> Grado informatica </value>
    < strength > all </ strength >
</ context >
 < typicalAgeRange >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> 20-50</value>
    < strength > all </ strength >
</ typicalAgeRange>
 < cost >
    <vocabularyID> UOC-Vocabulario </vocabularyID>
    <value> No tiene coste </value>
    < strength > all </ strength >
</cost>
< toMetadataCollection>
```

```
    <metadataCollectionIdentifier>
        <catalog> Uoc-Catalog </catalog>
        <entry> Meta-02 </entry>
    </metadataCollectionIdentifier>
  </ toMetadataCollection >
  < contains >
    <contentCollectionIdentifier>
        <catalog> Uoc-Catalog </catalog>
        <entry> Content-01</entry>
    </contentCollectionIdentifier>
  </ contains >
  < isContained >
    <contentCollectionIdentifier>
        <catalog> Uoc-Catalog </catalog>
        <entry> Content-03 </entry>
    </contentCollectionIdentifier>
  </ isContained >
</contentCollection>
```

Caso de uso:

Medatato	**Valor**
identifier	Nombre Catálogo:Uoc-Catalog Valor:Protocol-01
name	“Simple Query Interface”
version	1.0
protocolDescriptionBindingNamespace	http://www.imsglobal.org/services/lode/imslosqi-1p0_v1p0
protocolDescriptionBindingLocation	http://www.imsglobal.org/services/lode/imslosqi-1p0_v1p0.xsd

```
<protocol>
  <identifier>
    <catalog>  Uoc-Catalog </catalog>
    <entry> Protocol-01  </entry>
  </identifier>
  <name>  Simple Query Interface </name>
  <version> 1.0  </version>
   < protocolDescriptionBindingNamespace>
    http://www.imsglobal.org/services/lode/imslosqi-1p0_v1p0
  </protocolDescriptionBindingNamespace>
  < protocolDescriptionBindingLocation>
  http://www.imsglobal.org/services/lode/imslosqi-1p0_v1p0.xsd
  </protocolDescriptionBindingLocation>
</protocol>
```

Caso de uso:

Medatato	**Valor**
identifier	Nombre Catálogo:Uoc-Catalog Valor:Target-01
protocolIdentifier	Nombre Catálogo:Uoc-Catalog Valor:OAIPMH
location	http://www.uoc.es/lógica-enun-target-oaiphm
protocolImplementationDescription	Instancia de la implementación del servicio de OAI-PMH
responsible	Contacto Técnico [UOC Vocabulario], Antonio Sarasa
accessPolicy	Nombre Catálogo:Uoc-Catalog Valor:Access-01
collection	Nombre Catálogo:Uoc-Catalog Valor:Meta-02

< target >

```
<identifier>
  <catalog> Uoc-Catalog </catalog>
  <entry> Target-01</entry>
<identifier>
<protocolIdentifier>
  <catalog> Uoc-Catalog </catalog>
  <entry> OAIPMH </entry>
<protocolIdentifier>
<location> http://www.uoc.es/lógica-enun-target-oaiphm </location>
< protocolImplementationDescription>
   Instancia de la implementación del servicio de OAI-PMH
< /protocolImplementationDescription>
< responsible >
  <responsibility>
    <vocabularyID> UOC Vocabulario </vocabularyID>
    <value> Contacto Técnico </value>
  </responsibility >
< vcard > BEGIN:VCARD VERSION:2.1 N:Antonio Sarasa FN:4758
        ORG:Universitat Oberta de Catalunya TITLE:Consultor
        END:VCARD
</ vcard >
</ responsible >
<accessPolicy>
  <accessPolicyIdentifier>
    <catalog> Uoc-Catalog </catalog>
    <entry> Access-01</entry>
  <accessPolicyIdentifier>
</ accessPolicy >
<collection>
```

```
<metadataCollection>
  < metadataCollectionIdentifier >
    <catalog> Uoc-Catalog </catalog>
    <entry> Meta-02</entry>
  </ metadataCollectionIdentifier >
</metadataCollection>
</ collection >
```

</ target >

Caso de uso:

Medatato	**Valor**
identifier	Nombre Catálogo:Uoc-Catalog Valor:Access-01
description	Acceso libre

<accessPolicy>

```
<identifier>
  <catalog> Uoc-Catalog </catalog>
  <entry> Access-01 </entry>
<identifier>
< description >
  <language> es </language>
  <string> Acceso libre </string>
</ description >
```

</ accessPolicy >

b) IMS LODE ILOX

Caso de uso:

Medatato			**Valor**
identifier			Nombre Catálogo:Uoc-Catalog Valor:Work-01
description			Rights[Uoc-Vocabulario] Esquema:http://ltsc.ieee.org/xsd/LOM
expression	identifier		Nombre Catálogo:Uoc-Catalog Valor:Expression-01
	dimension		Rights 1.0
	Description		Rights[Uoc-Vocabulario] Esquema:http://ltsc.ieee.org/xsd/LOM
	manifestation	identifier	Nombre Catálogo:Uoc-Catalog Valor:Manifestation-01
		name	Rights
		parameter	1.0
		description	Rights[Uoc-Vocabulario] Esquema:http://ltsc.ieee.org/xsd/LOM
		Ítem	Nombre Catálogo:Uoc-Catalog Valor:Item-01 Uri: http://www.uoc.es/rights Esquema:http://ltsc.ieee.org/xsd/LOM Rights[Uoc-Vocabulario] Esquema:http://ltsc.ieee.org/xsd/LOM

```
<work>
  <identifier>
    <catalog> Uoc-Catalog </catalog>
    <entry> Work-01 </entry>
  <identifier>
  <description>
      <facet>
        <vocabularyID> Uoc-Vocabulario </vocabularyID>
        <value> Rights </value>
      </facet>
      <metadata>
        <schema> http://ltsc.ieee.org/xsd/LOM </schema>
        /*Instancia de metadatos LOM para Rights*/
      </metadata>
  <description>
 <expression>
  <identifier>
    <catalog> Uoc-Catalog </catalog>
    <entry> Expression-01</entry>
  <identifier>
  <dimension>
   <name> Rights </name>
   <paremeter> 1.0</parameter>
  </dimension>
  <description>
      <facet>
        <vocabularyID> Uoc-Vocabulario </vocabularyID>
        <value> Rights </value>
      </facet>
      <metadata>
        <schema> http://ltsc.ieee.org/xsd/LOM </schema>
        /*Instancia de metadatos LOM para Rights*/
      </metadata>
  <description>
  <manifestation>
     <identifier>
       <catalog> Uoc-Catalog </catalog>
       <entry> Manifestation-01 </entry>
     <identifier>
     <name> Rights </name>
    <paremeter>1.0</parameter>
    <description>
      <facet>
        <vocabularyID> Uoc-Vocabulario </vocabularyID>
        <value> Rights </value>
```

```
        </facet>
        <metadata>
          <schema> http://ltsc.ieee.org/xsd/LOM </schema>
          /*Instancia de metadatos LOM para Rights*/
        </metadata>
      <description>
      <item>
        <identifier>
          <catalog> Uoc-Catalog </catalog>
          <entry> Item-01 </entry>
        <identifier>
        <location>
          <uri> http://www.uoc.es/rights</uri>
         <metadata>
          <schema>http://ltsc.ieee.org/xsd/LOM </schema>
          /*Instancia de metadatos LOM para Rights*/
         </metadata>
        </location>
        <description>
         <facet>
          <vocabularyID> Uoc-Vocabulario </vocabularyID>
          <value> Rights </value>
         </facet>
         <metadata>
          <schema>http://ltsc.ieee.org/xsd/LOM </schema>
           /*Instancia de metadatos LOM para Rights*/
         </metadata>
        <description>
      </item>
      </manifestation>
  <expression>
</work>
```

6. Conclusiones

El trabajo ha abordado un estudio del arte sobre las especificaciones que cubren aquellos aspectos que son necesarias cuando se implementa un repositorio digital, profundizando especialmente en la especificación IMS LODE. Respecto a ese estudio se puede concluir:

a) El estudio del arte realizado pone de manifiesto que se han desarrollado un número importante de especificaciones que estandarizan diferentes aspectos de un objeto de aprendizaje, tales como su empaquetamiento, sus metadatos, protocolos para poder consultar y recuperar objetos de un repositorio,...Sin embargo estas especificaciones que resultan ser muy útiles de manera individual en el aspecto que cubren, no resultan útiles cuando se tratan de utilizar para implementar un repositorio digital. La razón de ello es que no están pensadas para ser usadas de manera coordinada.

b) IMS LODE es una especificación que viene cubrir la necesidad de disponer de una única especificación que coordine de una manera coherente las especificaciones y recomendaciones que son necesarias tener en cuenta cuando se quiere implementar de manera estándar un repositorio. En este sentido:
   - Define una forma de estructura y de organizar los repositorios en forma de colecciones de metadatos y contenidos, y de cómo estas colecciones se relacionan.
   - Define una forma de especificar servicios sobre los metadatos y colecciones definidas.
   - Define una forma un mecanismo de versionado sobre los objetos de aprendizaje.
   - Establece pautas para poder recuperar objetos de aprendizaje de un repositorio digital.

Respecto a los trabajos futuros que podrían derivarse de éste, se refieren al aspecto práctico del mismo:

- Implementación de una herramienta de edición que permitiera editar los metadatos que se especifican en IMS LODE.
- Implementación de un repositorio digital usando los principios definidos en la especificación.
- Realizar una evaluación de las bondades que presenta la forma de organizar el acceso y recuperación definido en la especificación.

Bibliografía

[AGREGA] Proyecto Agrega: http://www.proyectoagrega.es

[Atom] http://tools.ietf.org/html/rfc5023

[ASP,09]Design of Data Model and Architecture for a Registry of Learning Object Repositories and Application Profiles, D. Massart, N. Smith & R. Tice, ASPECT Project, March 2009. http://aspect.eun.org/sites/default/files/docs/ASPECT_D2p2.pdf

[Blinco et al., 2005] Blinco, K., Dovey, M., Hatala, M., Mason, J., Massart, D., Robson, R., Sanderson, R., Simon, B., Ternier, S., Thorne, S., Tice, R., van Assche, F., and Wilson, S. (2005). IMS query services white paper. URL:http://www.imsglobal.org/query/imsQueryServices.html.

[Broisin-2005] J. Broisin, P. Vidal, M. Meire, and E. Duval. Bridging the Gap between Learning Management Systems and Learning Object Repositories:Exploiting Learning Context Information. In ELETE 2005, IEEE Computer Society, p. 478483, Lisbon, July 2005.

[CANCORE, 04] CanCore Guidelines for the Implementation of Learning Object Metadata (IEEE 1484.12.1-2002), N. Friesen, S. Fisher, & A. Roberts, April 2004. http://www.cancore.ca/en/guidelines.html

[CORDRA] CORDRA (Content Object Repository Discovery and Registration/Resolution Architecture): Technical Introduction and Overview: http://www.lsal.cmu.edu/lsal/expertise/projects/cordra/intro/intro-v1p00.html, 2004.

[CQL]CQL: Contextual Query Language (SRU Version 1.2 Specifications). Last accessed 12 November 2007 at <http://www.loc.gov/standards/sru/specs/cql.html>.

[CQL, 08] CQL: Contextual Query Language (SRU Version 1.2 Specifications), Library of Congress. http://www.loc.gov/standards/sru/specs/cql.html

[CQLBIB, 09] CQL Profile for Bibliographic Searching, Library of Congress. Version 1.0. July 2009. http://www.loc.gov/standards/sru/resources/cql-bibliographic-profile.html

[DCCAP, 07] Dublin core collections application profile, Dublin Core Collection Description Task Group, March 2007. http://dublincore.org/groups/collections/collection-application-profile/

[DCME, 08] Dublin Core Metadata Element Set, Dublin Core Metadata Initiative, Version 1.1, January 2008. http://dublincore.org/documents/dces/

[Dripps] Dripps D., Casey J., Proven J. “The Technical Landscape of Digital Repositories”. http://trustdr.ulster.ac.uk/work_in_progress/workpackages/WP2-1/The_Tech_Landscape_WP2-1_30.doc

[Dublin Core]The Dublin Core Metadata Initiativa: <http://dublincore.org/>.

[Duval-2002] Duval, E., Hodgins, W., Sutton, S. & Weibel, S. L. , “Metadata Principles and Practicalities”, D-LibMagazine, April 2002.

[Eap, 2003] Eap, T. (2003). The communication middleware for collaborative learning object repository network. PhD thesis, Simon Fraser University, Irvine.

[Eap et al., 2004] Eap, T., Hatala, M., and Richards, G. (2004). Digital repository interoperability: design, implementation and deployment of the ECL protocol and connecting middleware. In WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters , pages 376_377, New York, NY, USA. ACM Press.

[ECL] ECL: eduSource Communication Layer. Interoperability for digital repositories. http://ecl.iat.sfu.ca/

[ELENA] ELENA Project http://www.elena-project.org/

[EDRENE] http://edrene.org/

[Hatala et al., 2004a] Hatala, M., Richards, G., Eap, T., and Willms, J. (2004a). The edusource communication language: implementing open network for learning repositories and services. In SAC '04: Proceedings of the 2004 ACM symposium on Applied computing, pages 957_962, New York, NY, USA. ACM Press.

[Hatala et al., 2004b] Hatala, M., Richards, G., Thorne, S., and Merriman, J. (2004b). Closing the Interoperability Gap: Connecting Open Service Interfaces with Digital Repository Interoperability. In Lorenzo Cantoni and Catherine McLoughlin, editors, Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2004, pages 78_83, Lugano, Switzerland. AACE.

[IANA, 07] MIME Media Types, IANA, March 2007. http://www.iana.org/assignments/media-types/

[IMS, 2003] IMS (2003). IMS Digital Repositories Interoperability - Core Functions Information Model.

[ISO,09] Information and documentation—Registry Services for Libraries and Related Organisations. ISO TC 46/SC 4. http://www.iso.org/iso/catalogue_detail.htm?csnumber=44936

[ISO8601, 04] Data elements and interchange formats—Information interchange—Representation of dates and times. ISO 8601:2004. International Organization for Standardization. December 2004.
http://isotc.iso.org/livelink/livelink/4021199/ISO_8601_2004_E.zip?func=doc.Fetch&nodeid=4021199

[Lagoze] C. Lagoze, H. Van de Sompel, M. Nelson, S. Warner. The Open Archives Initiative Protocol for Metadata Harvesting – Version 2.0 <http://www.openarchives.org/OAI/openarchivesprotocol.html>.

[Lagoze-2007] Lagoze, Carl; De Sompel, Herbert van. Compound information objects: the OAIORE Perpective, May 28, 2007. Consultado en 14072008. http://www.openarchives.org/ore/documents/CompoundObjects200705.html

[Littlejohn-2003] Littlejohn A., “Reusing Online Resources: A Sustainable Approach to eLearning”, (Ed.). Kogan Page, London,2003

[LODE 2010] http://www.imsglobal.org/LODE/spec/imsLODEv1p0bd.html

[LOM]IEEE Standard for Learning Object Metadata: <http://ltsc.ieee.org/wg12/>.

[LOM, 02] Standard for Information Technology—Education and Training Systems—Learning Objects and Metadata, W. Hodgins et al., IEEE Learning technology Standards Committee, December 2002. http://ltsc.ieee.org/wg12/materials.html

[MARC, 03] Source Codes for Classification, Library of Congress, Network Development and MARC Standards Office, August 2003. http://www.loc.gov/marc/sourcecode/classification/classificationsource.html

[Manepalli et al., 2006] Manepalli, G., Jerez, H., and Nelson, M. (2006). FeDCOR: An institutional CORDRA registry. D-Lib Magazine, 12(2).

[McCallum, 2006] McCallum, S. H. (2006). A look at new information retrieval protocols: SRU, OpenSearch/a9, CQL, and XQuery. In World Library and Information Congress: 72nd IFLA General Conference and Council. IFLA, IFLA.

[Nejdl et al., 2002] Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmér, M., and Risch, T. (2002). EDUTELLA: a P2P networking infrastructure based on RDF. In WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 604_615, New York, NY, USA. ACM Press.

[OAI, 04] The open archives initiative protocol for metadata harvesting version 2.0, Document Version 2004/10/12T15:31:00Z, C. Lagoze, H. Van de Sompel, M. Nelson, & S. Warner. Also available as http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm, 2002.

[OAI-PMH] OAI-PMH http://www.openarchives.org/OAI/openarchivesprotocol.html

[Payette and Lagoze, 1998] Payette, S. and Lagoze, C. (1998). Flexible and Extensible Digital Object and Repository Architecture (FEDORA). In Research and Advanced Technology for Digital Libraries: Second European Conference, ECDL'98, pages 41_59, Heraklion, Crete, Greece. Springer Berlin / Heidelberg.

[PENS, 2005] PENS (2005). Guidelines for Package Exchange Noti_cation Services.

[Polsani-2003] Polsani, P. R. “Use and Abuse of Reusable Learning Objects”. Journal of Digital information,3(4). 2003.

[RDCEO, 02] IMS Reusable Definition of Competency or Educational Objective—Information Model, IMS Global Learning Consortium, October 2002. http://www.imsglobal.org/competencies/rdceov1p0/imsrdceo_infov1p0.html

[RDF] http://www.w3.org/RDF/

[RDFa] http://www.w3.org/TR/rdfa-syntax/

[Sanderson et al., 2005] Sanderson, R., Young, J., and LeVan, R. (2005). SRW/U with OAI: Expected and Unexpected Synergies. D-Lib Magazine, 11(2).

[Sarasa-2008] Sarasa,A.,Canabal, J.M, Sacristán, J.C. “LOM-ES: Un perfil de aplicación de LOM”. Spedece 2008. Salamanca.

[Simon et al., 2005] Simon, B., Massart, D., van Assche, F., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., and Miklos, Z. (2005). A Simple Query Interface for Interoperable Learning Repositories. In Proceedings of the 1st Workshop on Interoperability of Web-based Educational Systems, pages 11_18.

[SCORM] SCORM. Sharable Content Object Reference Model: http://www.adlnet.gov/scorm/

[SWO, 08] SWORD: Simple Web-service Offering Repository Deposit,J. Allinson, S. François & S. Lewis. Ariadne 54, January 2008. http://www.ariadne.ac.uk/issue54/allinson-et-al/

[SPI, 08] A simple publishing interface for learning object repositories, S. Ternier, D. Massart, F. Van Assche, N. Smith, B. Simon, & E. Duval. Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008 (Vienna, Austria), AACE, June 2008, pp. 1840–1845. http://www.editlib.org/p/28625

[SQI,05] A simple query interface specification for learning repositories, CEN Workshop Agreement (CWA 15454). B. Simon, D. Massart, F. Van Assche, S. Ternier, & E. Duval. Also available from ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/WS-LT/CWA15454-00-2005-Nov.pdf, November 2005.

[SRU, 08]SRU Version 1.2 Specifications, Library of Congress. http://www.loc.gov/standards/sru

[Staples et al., 2003] Staples, T., Wayland, R., and Payette, S. (2003). The FedoraProject. An Open-source Digital Object Repository Management System. D-Lib Magazine, 9(4).

[Ternier and Duval, 2003] Ternier, S. and Duval, E. (2003). Web services for the ARIADNE knowledge pool system. In Proceedings of the 3rd Annual ARIADNE Conference, pages 1_9. ARIADNE Foundation.

[Ternier et al., 2008] Ternier, S., Massart, D., Campi, A., Guinea, S., Ceri, S., and Duval, E. (2008). Interoperability for Searching Learning Object Repositories: The ProLearn Query Language. D-Lib Magazine, 14(1/2).

[Van de Sompel-2004] H. Van de Sompel, M. L. Nelson, C. Lagoze, S. Warner. Resource Harvesting within the OAIPMH Framework, DLibMagazine, Volume 12, Number 12, December 2004, <doi:10.1045/december2004vandesompel>.

[XML, 04] XML Schema Part 2: Datatypes Second Edition,World Wide Web Consortium, October 2004. http://www.w3.org/TR/xmlschema-2/

Apendices

Apendice A: Casos de uso

Use Case Title:	**Professor searches for colleagues' resources within Lionshare**
Use Case Local ID:	LODE001
Brief Description:	Professors in the astronomy department often do searches on the Internet for such items as images of galaxies, papers on astronomy that they can refer to in class, and charts of the solar system. The professors mentioned at a meeting how frustrating it was to try to find astronomy-related items online, as often the search results are cluttered with things such as astrology sites, company names that are astronomy-related, etc. Sorting through such results to find useful items can be a problem for professors with limited time. Along with this, several professors mentioned that they often spend a lot of time searching for something, and then find out after the fact that a colleague has what they are looking for on their computer. It would be great, the professors conclude, if they had something like a Web site where they could search for each other's resources. However, they have limited computer expertise and little time for such an involved project.
Level:	Summary
Actors:	Professors

Use Case Title:	**Register repository metadata with the FRED registry ()Federated Repositories for Education)**
Use Case Local ID:	LODE002
Brief Description:	New repository is registered with registry, and thereby joins the federation. Accountability and access metadata is registered for repository.
Level:	Summary
Actors:	Primary: Local Repository Manager, Registry Manager

Use Case Title:	**Pull metadata of item into the FRED registry**
Use Case Local ID:	LODE003
Brief Description:	Metadata for an item is pulled from repository into registry, and exposed by registry to end users for discovery.
Level:	Summary
Actors:	Primary: Repository Manager, Registry Manager, End User

Use Case Title:	**Update metadata of item in the FRED registry**
Use Case Local ID:	LODE004
Brief Description:	The metadata record for an item is updated; it sits alongside the old metadata record as a new version.
Level:	Summary
Actors:	Primary: Registry, Metadata Manager (~ Repository Manager)

Use Case Title:	Discover item through the FRED registry (Search)
Use Case Local ID:	LODE005
Brief Description:	Item registered in registry is discovered by end user through full-text search of meta- data or specific metadata fields
Level:	Summary
Actors:	Primary: End User, Registry

Use Case Title:	Add repository to the FRED directory roster
Use Case Local ID:	LODE006
Brief Description:	The directory manager adds a local repository to their roster of repositories
Level:	Summary
Actors:	Primary: Directory Manager

Use Case Title:	Discover item through the FRED roster (Full text Federated Search)
Use Case Local ID:	LODE007
Brief Description:	Item in an (ad hoc) federated repository is discovered by end user through full-text search of local repository content
Level:	Summary
Actors:	Primary: End User

Use Case Title:	Searching remote LODE repositories through a European Digital Library web service
Use Case Local ID:	LODE008
Brief Description:	LODE repositories are described within a European Digital Library web service as collections. These descriptions allow end-users to select them for searching. Search results are retrieved from LODE repositories and displayed in the web service interface. The descriptions should contain information about how repositories are accessed (protocols like SRU, Z39.50, etc.) and which formats the received metadata is in.
Level:	Summary
Actors:	Primary: European Digital Library web service Secondary: LODE repositories, end users
Stakeholders & Interest:	Stakeholder: LODE Interest: Repositories exposed through different platforms reaching a broader audience
Stakeholders & Interest:	Stakeholder: European Digital Library Interest: Provide European Digital Library users with searchable content

Use Case Title:	**Harvesting and indexing LODE repositories with a European Digital Library harvesting and indexing service**
Use Case Local ID:	LODE009
Brief Description:	A European Digital Library harvesting service retrieves LODE repositories metadata and index them into a central index. Metadata are harvested in a format compliant for search within a European Digital Library web service.
Level:	Summary
Actors:	Primary: European Digital Library harvesting and indexing service Secondary: LODE repositories
Stakeholders & Interest:	Stakeholder: LODE Interest: >From within a central index the LODE repositories are part of a fast search and retrieve system giving convenient access to end users.
Stakeholders & Interest:	Stakeholder: European Digital Library Interest: Provide European Digital Library users fast and convenient access to new content

Use Case Title:	**Federated search within Lornet**
Use Case Local ID:	LODE010
Brief 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 LOM records. The goal is to create a service for an Infoseeker (teacher, student, course developer or software agent) to search for a learning object/resource across multiple repositories, providing an aggregated result list of metadata records in return. The Infoseeker must have access to the Internet. The Federator possesses a (possibly empty) result list of learning objects from a variety of repositories, which meet the search criterion provided by the Infoseeker.
Level:	Summary
Actors:	Primary: **Federator**: a software agent responsible for providing the interaction between the Infoseeker and the Federated Search System (FSS). Secondary: **Infoseeker**: a person (or a software application) wanting to search a network of metadata repositories. Most often the Infoseekers 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 return a set of Metadata Records.

Use Case Title:	Harvested search within Lornet
Use Case Local ID:	LODE011
Brief 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. To provide Infoseekers with a way to use a Harvester as a search service and enable resource Providers (commercial and non-commercial) with a simple way to make their resources available for use through the eduSource network. The Infoseeker 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. The Infoseeker has located a resource through a Harvester from at least one learning resource Provider.
Level:	Summary
Actors:	Primary: **Harvester:** a client application that issues OAI-PMH compliant requests and makes metadata available to the Infoseeker. Secondary: **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.

Use Case Title:	Instructor conducts search by metadata value
Use Case Local ID:	LODE013
Brief Description:	Instructor searches a digital repository for content by a specific value of a metadata field. A successful result returns a list of objects that met search criteria
Level:	Primary task
Actors:	Primary: Instructor Secondary: Digital repository, metadata indexer, SME (subject matter expert), instructional designer, web content developer
Stakeholders & Interest:	Stakeholder: Institution the provides content Interest: Discoverability of and reuse of their content
	Stakeholder: Instructor Interest: Finding existing content to use as part of their offering
	Stakeholder: SME and/or metadata indexer Interest: Providing metadata information that supports discovery by instructors

Use Case Title:	**Alocom powerpoint plugin searches a learning object repository**
Use Case Local ID:	LODE014
Brief Description:	A plugin for MS powerpoint (and similar applications) enables authors to reuse fragments of presentations available in a repository. 1. Alocom sends a query to a repository. This query contains keywords and the type of learning object the author is interested in (examples, images, definitions, ...) 2. The repository returns a list of learning objects that match the query. 3. The author selects an object and adds it to her/his presentation
Level:	Summary
Actors:	Primary: The author is in the process of creating a powerpoint presentation, reusing existing learning objects. Secondary: The learning object repository that hosts reusable assets. These can be images, single slides, definitions, examples, animations, etc.
Stakeholders & Interest:	Stakeholder: ARIADNE Interest: making reuse of small LO possible.

Use Case Title:	**ARIADNE federated search engine**
Use Case Local ID:	LODE015
Brief Description:	The ARIADNE federated search engine federates search requests through standards such as SQI and LOM to learning object repositories such as the repositories available in ProLearn and GLOBE. 1. A client tool sends a query (synchronously or asynchronously) to the federated search engine. This tool can additionally limit the search to a subset of the repositories available. 2. The federated search engine forwards the query (preferably asynchronously) to the repositories that support the query language. 3. The repository return results to the federated search engine. 4. The federated search engine ranks the results according to a ranking algorithm. 5. The federated search responds to the client tool.
Level:	Summary
Actors:	Primary: A client tool issuing a search. Secondary: The federated search engine, responsible for federating a query Secondary: A repository that is part of federation. Such a repository can implement a native interface to a repository or host metadata, harvested from several repositories. Secondary: A registry documenting repositories (Search interface, harvest interface, publish interface, etc.)
Stakeholders & Interest:	Stakeholder: ARIADNE/GLOBE Interest: Enabling search engines to reach a broader mass of educational content, unlocking the deep web.

Use Case Title:	VLE (Virtual Learning Environment) searches a learning object repository
Use Case Local ID:	LODE016
Brief Description:	Loosely couple search and obtain LO's from repositories. 1. A client tool sends a query (synchronously or asynchronously) to the federated search engine. This tool can additionally limit the search to a subset of the repositories available. 2. The federated search engine forwards the query (preferably asynchronously) to the repositories that support the query language. 3. The repository return results to the federated search engine. 4. The federated search engine ranks the results according to a ranking algorithm. 5. The federated search responds to the client tool.
Level:	Summary
Actors:	Primary: A VLE issuing a search or trying to obtain content Secondary: A repository or federated search engine responding to the search.
Stakeholders & Interest:	Stakeholder: ARIADNE Interest: Enable content to be open, reachable beyond the borders of a repository. Avoid content to be locked up inside a single VLE. Making coupling with other VLEs easy.

Use Case Title:	Disclosure and social contribution: Organization level (NIME)
Use Case Local ID:	LODE017
Brief Description:	Many Japanese universities and colleges have some traditions to share their knowledge with their regional community as a "social contribution". They provide their face-to- face lectures and/or on-line course materials for the community. For example, the member universities of Japanese Opencourseware Consortium also understand their significance of their movement in the similar contexts. Japanese government promotes the new "e-Japan Strategic Plan II", and one of the objectives is to contribute to the global knowledge-based society by providing Japanese content (open or profit) to overseas. Nowadays, the Japanese universities and colleges should promote their social contributions in more international and global fashions. Even if they provide their content in English or in other foreign languages, it is very difficult for overseas potential users to find their homepages. So, they register their metadata at NIME-glad gateway, and give more search opportunities from overseas using GLOBE federated search functions. At the moment, the five organizations, ARIADNE in EU, education.au limited-EdNA in Australia, LORNET in Canada, MERLOT in the United States and NIME in Japan, join the GLOBE. The users in each organization can search the other overseas databases cross-organizationally using the federated search services.
Level:	Summary

Use Case Title:	Resource detail information browsing (Institute for Information Industry)
Use Case Local ID:	LODE020
Brief Description:	Users get full descriptions of the resource, which was chosen in search result pages, in learning object metadata (LOM) format.
Level:	Primary Task
Actors:	Primary: public user Secondary: teacher, student, learning object author or provider
Stakeholders & Interest:	Stakeholder: National Science & Technology Program Office for e-Learning Government departments, ex. National Palace Museum, Overseas Compatriot Affairs Commission, R.O.C (Taiwan), Ministry of Education ... Interest: National Science & Technology Program Office for e-Learning

Use Case Title:	Basic search (Institute for Information Industry)
Use Case Local ID:	LODE021
Brief Description:	The basic function of the search.
Level:	Primary Task
Actors:	Primary: public user Secondary: teacher, student, learning object author or provider
Stakeholders & Interest:	Stakeholder: 1. National Science & Technology Program Office for e-Learning 2. Government departments, ex. National Palace Museum, Overseas Compatriot Affairs Commission, R.O.C (Taiwan), Ministry of Education ... Interest: National Science & Technology Program Office for e-Learning

Use Case Title:	Advanced search (Institute for Information Industry)
Use Case Local ID:	LODE022
Brief Description:	Users can specify more restrictive conditions to make a more accurate query.
Level:	Sub-function

Use Case Title:	Advanced search (Institute for Information Industry)
Actors:	Primary: public user Secondary: teacher, student, learning object author or provider
Stakeholders & Interest:	Stakeholder: 1. National Science & Technology Program Office for e-Learning 2. Government departments, ex. National Palace Museum, Overseas Compatriot Affairs Commission, R.O.C (Taiwan), Ministry of Education ... Interest: National Science & Technology Program Office for e-Learning

Use Case Title:	**Federated search within EdNA**
Use Case Local ID:	LODE023
Brief Description:	This use case describes at a high level, the tasks used to perform federated search across a number of repositories and present the results of that search to an end user. The use case is based on an existing federated search. There is a distinction drawn between distributed search and federated search. In this case, distributed search refers to a search across multiple repositories where there may be no agreement on how that search is performed. In this 'federated search', there is a collection of repository owners that have agreed to provide a search across their collection of repositories and have agreed to a common set of behaviors/guidelines. The use case describes a federated search request as distinct from harvesting metadata for searching. However, any or all of the repositories in this federation may contain harvested metadata from other repositories.
Level:	Primary task
Actors:	**Primary:** *Searcher*. The primary actor is typically a person wishing to query multiple repositories within a single search request. The searcher usually generates the search request from a web-form. It is possible that the search request can be generated from another program. **Secondary:** *Repository.* Repositories will have an API that can be queried as part of the search request. *Federated Search Provider (FSP):* The federated search provider is typically a software program that is called to process search requests. The federated search provider also performs administrative and operational functions related to federated search. *Access Provider (AP)*: An access provider provides access to federated search functions. There may be multiple access providers for a federation. Each repository owner may provide access to the federated search function. Other website/portal owners that do not have a repository may also act as access providers. Each provider may configure their access point differently (e.g., limiting the number of repositories, using different performance policies, presenting results differently etc.).
Stakeholders & Interest:	Stakeholder: *Federation:* A collection of repository owners that have agreed to allow their repositories to be searched collectively according to an agreed set of rules/ guidelines. *Repository Provider*: Owners of a repository. In addition to agreeing to a common set of guidelines for the federation, an individual repository may have additional constraints that they need to abide by.

Use Case Title:	**Discovery within Digital Marketplace**
Use Case Local ID:	LODE024
Brief Description:	Search/Browse for Finding – What materials are available that could satisfy my needs in teaching the course and my students' needs in learning? On demand search and browse services will provide users hitlists of materials for them to evaluate if the material(s) satisfy their "finding" requirements. >From an initial request, the search/browse functionality must allow the user to easily navigate to the item level and the item-level metadata should allow the user to have a 70% confidence level that the materials will satisfy their requirement that the material can satisfy their teaching needs and their students' learning needs. Professor Plum logs into his RLS during the summer to begin to build the collection of resources he will want his students to use in the Biology 101 course he's teaching in the fall (1). It's been 3 years since he taught the introductory level course so he's interested in reviewing what's available in the field. Within the LMS website, he goes to the page for building his resource list and clicks on "Search for Resources". He types in a key concept he'll be covering in the course and a hit list of materials from 6 different publishers is generated along with free materials from MERLOT. The descriptions of the materials includes title, author, abstract, publisher collateral, type of resource (book, article, multimedia, etc.), indication of its ability to be rendered in an accessible (section 508 compliant) format, and the different delivery formats and prices (hard copy text book, custom book, eBook to own, eBook to rent).
Level:	Summary
Actors:	Primary: Faculty

Use Case Title:	**Preview within Digital Marketplace**
Use Case Local ID:	LODE025
Brief Description:	Previewing for Use – Of the materials I've discovered, which ones do I want to evaluate more deeply for selection in my course? Once users have created a limited collection of prospective course content, they will evaluate the resources in more detail. Previewing the content in preparation for the final selection is an important service for faculty. At the completion of previewing the materials, faculty will have information about the resource and about their own needs to support a confident decision to select materials and label them as "my materials for my course" – personalizing the materials in the process of reusing materials. Professor Plum selects 10 different resources to review in more detail. He clicks on the PREVIEW button and a window pops ups indicating that since he is a faculty in good standing at CSULB, he will have full electronic access to the eBook for a 72-hour period, starting whenever he wishes. After previewing 10 materials, he selects 5 for his course, a textbook, and a chapter from another book, a tutorial on using EXCEL, and 2 multimedia simulations. He also gets to preview the net-generation handbook as well.
Level:	Summary
Actors:	Primary: Faculty

Use Case Title:	Selection, assembly, and annotation within Digital Marketplace
Use Case Local ID:	LODE026
Brief Description:	Professor Plum saves his selections of materials for his students and writes notes (annotations) about the resources he’s selected to use. With the textbook, Professor Plum decides that only 8 of the 14 chapters will be used in the course so he chooses the custom publishing option to create a book that’s more relevant to the course learning. He also adds the chapter from another book to create the custom course “textbook”. As Professor Plum views the resource list he remembers that he created some content when last taught the course that students found especially useful. By using the Digital Marketplace Authoring / Assembly service, he adds this content to the CSU repository and then adds it to his resource list. He also allows other CSU faculty to view and use this content. He notices that the custom textbook, and tutorial can be rendered in an accessible format but the 2 multimedia simulations are only 80% accessible. Professor Plum contacts the campus Center for Students with Disabilities to learn what he needs to do to provide alternative curriculum to the visually impaired student he’ll have in his class. Finally Professor Plum examines the “student view” of the resource list and sees that the textbook is offered in an eBook-to-own version for 50% of the hard copy text and the eBook-to-rent is only $15.99 for the semester. With all these options for access to the materials, he’s hoping all his students will use the materials.
Level:	Summary
Actors:	Primary: Faculty

Use Case Title:	Choosing, rendering, and buying content within Digital Marketplace
Use Case Local ID:	LODE027
Brief Description:	When Jane Student gets access to the RLS for her Biology 101 course, she navigates to the Resource List to check out what she’ll need to buy. As a student with a vision disability, she has had a challenge of getting the materials in a format she can use in a timely manner. She reviews the resource list and sees that the textbook and tutorial are in an accessible format and is pleased. She then reviews the different types of style sheets CSULB has certified has rendering the content in an accessible manner. She likes the choices and decides on the size, contrast, colors, and layout that suits her needs. Jane is considering becoming a biology major so she decides to put the eBook-to-buy in her shopping cart and the tutorial in her shopping cart. She buys the resources online with her credit card and stores the resources in her campus ePortfolio. For the two multimedia resources, there’s a note for her stating that the CSULB Center for Students with Disabilities will provide an aid to work with Jane on the portions of these resource that are not accessible to her. In the 4th week of the semester, Jane realizes she’s having trouble with one of the key concepts in biology. She goes to the Digital Marketplace in her RLS and searches for additional materials that might do a better job in helping her learn the concept. She finds a student workbook that has the background information she needs and it can be rendered in the accessible format she prefers. Jane buys it online.
Level:	Summary
Actors:	Primary: Student

Use Case Title:	**Federated search within the Learning Resource Exchange**
Use Case Local ID:	LODE028
Brief Description:	A user sends a query to the Learning Resource Exchange and gets results 1. The user enters search criteria in its system 2. User's system transforms user input into a valid query 3. User's system sends the query to the BS 4. The BS propagates the query to all the repositories participating in the LRE 5. Each repository that supports the query language used, processes the query and sends the results back to the BS asynchronously 6. The BS routes the results to User's system 7. The user gets the results
Level:	Summary
Actors:	A system connected to the Learning Resource Exchange A user of the system The Learning Resource Exchange (LRE) The Brokerage System (BS). All the repositories participating in the LRE

Use Case Title:	**Federated discovery of repositories within the Learning Resource Exchange**
Use Case Local ID:	LODE029
Brief Description:	A harvester sends an Identify query to the Learning Resource Exchange and gets the description of the participating repositories: 1. The harvester sends the Identify query to the BS 2. The BS propagates the query to the repositories participating in the LRE 3. Each repository receives the query and sends its description to the BS 4. The BS routes the repository descriptions to the harvester 5. The harvester receives the descriptions of the participating repositories
Level:	Summary
Actors:	The Learning Resource Exchange (LRE) The Brokerage System (BS) All the repositories participating in the LRE A harvester connected to the Learning Resource Exchange

Use Case Title:	Federated harvesting within the Learning Resource Exchange
Use Case Local ID:	LODE030
Brief Description:	The harvester sends a ListRecords query to the Learning Resource Exchange and gets metadata updates: 1. The harvester sends the ListRecords, which contains a timestamp query to the BS 2. The BS propagates the query to the repositories participating in the LRE 3. Each repository receives the query and returns metadata updates 4. Metadata that have been updated after the timestamp 5. Ids of Metadata that have been removed after the timestamp 6. The BS routes the metadata to the harvester 7. The harvester receives metadata updates
Level:	Summary
Actors:	The Learning Resource Exchange (LRE) The Brokerage System (BS) All the repositories participating in the LRE A harvester connected to the Learning Resource Exchange

Use Case Title:	Search LORs to get results sorted by relevance
Use Case Local ID:	LODE031
Brief Description:	Tutor searches one or more LORs, and gets back results sorted in order of relevance to their search criteria
Level:	Summary
Actors:	Primary: Tutor Secondary: search client (e.g. part of LMS system), repository system(s), search gateway
Stakeholders & Interest:	Stakeholder: Tutor Interest: looking for packages of learning content suitable for a lesson for their students

Use Case Title:	Tutor refines their search criteria using properties exposed by the target repositories
Use Case Local ID:	LODE033
Brief Description:	The target repositories support a common set of metadata fields and vocabularies, such as educational level, which are made available to the tutor as possible search constraints within the search interface of the LMS. The tutor restricts the search to a specific educational level, and sees a more relevant list of results.
Level:	Summary
Actors:	Primary: Course Tutor
Stakeholders & Interest:	Stakeholder: Course Tutor

Use Case Title:	Tutor browses repositories by subject classification
Use Case Local ID:	LODE034
Brief Description:	Repository system(s) expose the classification system(s) used to categorize the collections they hold. A searching system reads these classifications systems, and offers up to a course tutor to explore the content available.
Level:	Summary
Actors:	Primary: Course Tutor

Use Case Title:	Tutor selects content objects for inclusion in course
Use Case Local ID:	LODE035
Brief Description:	From a list of search results the tutor selects one or more content objects for inclusion in their course.
Level:	Sub-function
Actors:	Primary: Tutor Secondary: search client (e.g. part of LMS system), repository system(s), search gateway
Stakeholders & Interest:	Stakeholder: Tutor Interest: looking for packages of learning content suitable for a lesson for their students

Use Case Title:	Students is shown latest version of a content object
Use Case Local ID:	LODE036
Brief Description:	Students accessing the lesson in the LMS see the latest versions of the selected content objects, requested from the host repository and parsed/played by the LMS
Level:	Summary
Actors:	Primary: Course Tutor

Use Case Title:	LMS downloads copy of a content package from LOR
Use Case Local ID:	LODE037
Brief Description:	From inside their LMS, a tutor selects one or more content objects for inclusion in a course. These are downloaded from the relevant repository by the LMS, and imported into the course.
Level:	Summary
Actors:	Primary: Course Tutor

Use Case Title:	Tutor previews content stored in an LOR
Use Case Local ID:	LODE039
Brief Description:	From a list of search results, a tutor chooses to preview each content object they are interested in.
Level:	Summary
Actors:	Primary: Course Tutor

Use Case Title:	Tutor reads full catalogue record
Use Case Local ID:	LODE040
Brief Description:	From a list of search results, a tutor chooses to view more information (metadata) about the content object.
Level:	Summary
Actors:	Primary: Course Tutor

Use Case Title:	Resource access
Use Case Local ID:	LODE046
Brief Description:	Users can access the resource they chose in three kinds of way: 1. link to the original website of the resource 2. access the resource immediately 3. download the resource to users' local hard drive space.
Level:	Sub-function
Actors:	Primary: public user Secondary: teacher, student, learning object author or provider
Stakeholders & Interest:	Stakeholder: 1. National Science & Technology Program Office for e-Learning 2. Government departments, ex. National Palace Museum, Overseas Compatriot Affairs Commission, R.O.C (Taiwan), Ministry of Education .. Interest: National Science & Technology Program Office for e-Learning

Use Case Title:	Display Objects
Use Case Local ID:	LODE052
Brief 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. The goal is to create a service for an Infoseeker (teacher, student, course developer or software agent) to display the objects/resources referenced by a metadata record. The Infoseeker must have access to some LORs, both physically and logically through attaining appropriate rights according to the DRM infrastructure. The Launcher has displayed items in a content package, or a single object, corresponding to the Infoseeker's request.
Level:	Summary
Actors:	Primary: 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: Infoseeker: a person (or a software application) wanting to search a network of metadata repositories.

Use Case Title:	**Tutor chooses specific version of a content object to use in a course**
Use Case Local ID:	LODE055
Brief Description:	Content objects in a repository are subject to regular updates. When they select one of these content objects for use in the course, a tutor is given the choice to always use the latest version of the content object, or to apply a specific version of the content object.
Level:	Summary
Actors:	Primary: Course Tutor

Use Case Title:	**Querying learning resources based on a rating or other annotation**
Use Case Local ID:	LODE056
Brief Description:	A user is interested in all the learning objects that are rated at 4 and higher in all the repositories of the federation. This could be one of the criteria in the advanced search. Ratings information could additionally be used to rank results.
Level:	Summary
Actors:	Primary: End-users Secondary: Repository owners who are interested in filtering resources that are of a certain rating.
Stakeholders & Interest:	Stakeholder: Repositories who have implemented ratings or review criteria for learning resources. Interest: Ratings can be binary votes, on a scale or multi-attribute. Moreover, in some repositories they are done by expert and in some by end-users.

Apendice B: IMS LODE Registry.

Metadatos de protocolo.

Nr	Name	Description	Multiplicity	Order	Value space	Data type	Note	Example
P.1	Identifier	A unique identifier of the protocol.	1..1	-	-	-	Mandatory data element.	-
P.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
P.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this protocol. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
P.2	Name	The name of the protocol.	1..*	Unordered	A valid protocol name.	CharacterString	Mandatory data element.	“Simple Query Interface”
P.3	Version	The version of the protocol.	0..1	-	A valid protocol version.	CharacterString	Optional data element.	“1.0”
P.4	Protocol Description Binding Namespace	The namespace of an XSD used to describe a particular implementation of the protocol.	1..1	-	A valid XML namespace.	URI	Mandatory data element.	“http://explain.z3950.org/dtd/”, “http://www.imsglobal.org/services/lode/imslosqi-1p0_v1p0”
P.5	Protocol Description Binding Location	The location of an XSD used to describe a particular implementation of the protocol.	1..1	-	A valid URL.	URL	Mandatory data element.	“http://fire.eun.org/xsd/registry/imslosqi-1p0_v1p0.xsd”

Metadatos puntos de acceso a servicio.

Nr	Name	Description	Multiplicity	Order	Value space	Data type	Note	Example
T.1	Identifier	A globally unique label that identifies the target of a collection	1..1	-	-	-	Mandatory data element. (Is referred to by C.12.1 Target Identifier)	-
T.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this target. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "URI"
T.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this target. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“101.1021323”, “http://foo.org/1234”
T.2	Protocol Identifier	The identifier of the protocol supported by the target.	1..1	-	-	-	Mandatory data element. Refers to protocol identified by P.1 Protocol.Identifier	-
T.2.1	Catalog	The name or designator of the identification or cataloguing scheme for this protocol. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "URI"
T.2.2	Entry	The value of the identifier within the identificati	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“101.1021323”, “http://foo.org/1234”

		on or cataloguing scheme that designates or identifies this protocol. A namespace specific string.						
T.3	Location	The location of the target	0..*	Unordered	A valid URL.	URL	Optional data element	“http://foo.org/1234”
T.4	Protocol Implementation Description	An XML description of the protocol parameters supported by the target end-point.	1..1	-	A valid metadata instance	A valid instance of the XSD referenced at the Protocol Description Binding Location P.5 of the protocol identified by the Protocol Identifier T.2.	Mandatory data element	A ZeeRex instance, an OAI-PMH description instance
T.5	Responsible	Parties responsible for the service target	0..*	Unordered	-	-	Optional data element	-
T.5.1	Responsibility	The responsibility the party has with regard to the target	1..1	-	No controlled vocabulary provided	VocabularyTerm	Mandatory child data element	“administrative contact”, “technical contact”
T.5.2	Contact Details	A description of the party	1..1	-	A VCARD [VCARD, 98] description.	CharacterString	Mandatory child data element	BEGIN:VCARD VERSION:2.1 N:kapukod;Hétszin FN:4758 ORG:Bubba Gump Shrimp Co. TITLE:Shrimp Man TEL;CELL;VOICE: 2536 END:VCARD
T.6	Access Policy	Access policy applicable to the collection	0..1	-	-	-	Optional data element	-
T.6.1	Access Policy Identifier (option 1)	A unique identifier of the policy	1..1	-	-	-	Mandatory child data element (Mutually exclusive with T.6.2 Access Policy Description)	-
T.6.1.1	Catalog	The name	1..1	-	Repertoir	normalizedStr	Mandatory	“hdl”, "URI"

		or designator of the identification or cataloguing scheme for this access policy. A namespace scheme.			e of ISO/IEC 10646-1:2000	ing	data element.	
T.6.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this access policy. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“101.1021323”, “http://foo.org/1234”
T.6.2	Access Policy (option 2)	A description of the policy	1..1	-	-	-	Mandatory child data element (Mutually exclusive with T.6.1 Access Policy Identifier)	-
T.6.2.1	Access Policy Identifier	A unique identifier of the policy	1..1	-	-	-	Mandatory child data element. (Is referred to by T.6.1 Access Policy Identifier)	-
T.6.2.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this access policy. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "URI"
T.6.2.1.2	Entry	The value of the identifier within the identification or cataloguing scheme	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“101.1021323”, “http://foo.org/1234”

		that designates or identifies this access policy. A namespace specific string.						
T.6.2.2	Description	A description of the access policy	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	(“en”, “This target is behind a firewall. Access can be obtained on request by sending an email to our technical contact. ”)
T.7	Collection	Collection exposed by the target	0..1	-	-	-	Optional data element	
T.7.1	Content Collection (option 1)	Content collection exposed by the target	1..1	-			Mandatory child data element (Mutually exclusive with T.7.2 Metadata Collection)	
T.7.1.1	Content Collection Identifier	Identifier of Content collection exposed by the target	1..1	-			Mandatory child data element. Refers to identifier of C.1	
T.7.1.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this content collection. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
T.7.1.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this content collection. A namespace	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”

		specific string.						
T.7.2	Metadata Collection (option 2)	Metadata collection exposed by the target	1..1	-			Mandatory child data element (Mutually exclusive with T.7.1 Content Collection)	
T.7.2.1	Metadata Collection Identifier	Identifier of Metadata collection exposed by the target	1..1	-			Mandatory child data element. Refers to identifier of M.1	
T.7.2.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this metadata collection. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
T.7.2.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this metadata collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”

Metadatos colección de contenido.

Nr	Name	Description	Multiplicity	Order	Value space	Data type	Note	Example
C.1	Identifier	A globally unique label that identifies this collection of learning objects.	1..1	-	-	-	Mandatory data element.	-
C.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
C.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this learning object collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
C.2	Description	A description of the collection.	1..1	-	-	LangString	Mandatory data element.	(“en”, “Collection of Algebra Assessments for 3rd grade”)
C.3	Collection Rights	A description of the rights that apply to the collectio	0..1	-	-	LangString	Optional data element.	(“en”, ”Creative Commons Attribution Share-Alike 3.0”)

		n as a whole.						
C.4	Keywords	A list of keywords for describing the collection as a whole.	0..*	Unordered	-	LangString	Optional data element.	(“en”, “assessment”), (“en”, “mathematics”)
C.5	Average Annual Increase	The average annual increase in the number of learning objects in the collection. This can be a negative number.	0..1	-	-	Integer	Optional data element.	“100”, “-50”
C.6	Accrual Periodicity	How often new learning objects are added to (or removed from) the collection.	0..1	-	No controlled vocabulary provided	VocabularyTerm	Optional data element. Possible controlled vocabulary is DCCAP [DCCAP, 07]: http://dublincore.org/groups/collections/frequency/2007-03-09/	“freq:annual”, “freq:threeTimesAMonth”, “freq:semiweekly”
C.7	Quality Procedure	The quality procedure(s) applied to the items in the collection.	0..*	Unordered.	No controlled vocabulary provided.	Property	Optional data element.	(“no quality procedure”, “all”) (“peer-reviewed”, “some”)
C.8	Language	The language(s) of the learning objects in the collection.	0..*	Unordered	ISO-639.	Property	Optional data element.	(“en”, “all”), (“fr”, “some”)
C.9	Format	The technical format(s) of the learning objects in the	0..*	Unordered	A valid MIME type.	Property	Optional data element.	“application/swf”.

		collection.						
C.10	Item Rights	The rights that apply to the learning objects in the collection.	0..*	Unordered	No vocabulary provided.	Property	Optional data element. This is independent of Collection-level Rights, which govern access to collection.	(“All rights reserved, copyright big money limited 2010”, “some”)
C.11	Responsible	Parties responsible for the collection	0..*	Unordered	-	-	Optional data element.	-
C.11.1	Responsibility	The responsibility the party has with regard to the collection.	1..1	-	No vocabulary provided.	VocabularyTerm	Mandatory child data element.	“administrative contact”, “technical contact”
C.11.2	Contact Details	A VCARD description of the party.	1..1	-	A valid VCARD [VCARD, 98]	CharacterString	Mandatory child data element.	BEGIN:VCARD VERSION:2.1 N:kapukod;Hétszin FN:4758 ORG:Bubba Gump Shrimp Co. TITLE:Shrimp Man TEL;CELL;VOICE:2536 END:VCARD
C.12	Target	Service end-point for the collection	0..*	Unordered	-	-	Optional data element	-
C.12.1 (option 1)	Target Identifier	A globally unique label that identifies a target of the collection (end-point for services offered)	1..1	-	-	-	Mandatory data element. (Mutually exclusive with C12.2 Target Description)	-
C.12.1.1	Catalog	The name or designator of the identific	1..1	-	Repertoire of ISO/IEC 10646	normalizedString	Mandatory data element.	“hdl”, "URI"

		ation or catalogu ing scheme for this target. A namespa ce scheme.			- 1:2000			
C.12. 1.2	Entry	The value of the identifie r within the identific ation or catalogu ing scheme that designat es or identifie s this target. A namespa ce specific string.	1..1	-	Repert oire of ISO/I EC 10646 - 1:2000	normalize dString	Mandatory data element.	“101.1021323”, “http://foo.org/1 234”
C12. 2 (opti on 2)	Target Descripti on	A descripti on of a target of the collectio n (end-point for services offered)	1..1	-	-	-	Mandatory child data element. (Mutually exclusive with C12.1 Target Identifier) (For further expansion, see T)	
C.13	Subject	The subject(s) covered by the learning objects in the collectio n	0..*	Unord ered	No contro lled vocab ulary provid ed.	Property	Optional data element.	(“Algebra”, “most”)
C.14	Type	The learning resource type(s) (cf. LOM 5.2) of the learning objects in the collectio n	0..*	Unord ered	No contro lled vocab ulary provid ed.	Property	Optional data element.	(“Assessment”, “most”)
C.15	Represen tation	The represen	0..*	Unord ered	No contro	Property	Optional data element.	(“imscp_v1p0”, “some”)

		tation of the learning objects of the collection (cf. ILOX Manifestation Names and Parameters).			lled vocabulary provided.			
C.16	Intended User Role	The intended end-user role for the learning objects in the collection.	0..*	Unordered	No controlled vocabulary provided.	Property	Optional data element.	(“Learner”, “most”)
C.17	Context	The context of use of the learning objects in the collection.	0..*	Unordered	No controlled vocabulary provided.	Property	Optional data element.	(“Compulsory education”, “all”)
C.18	Typical Age Range	The typical age range of the users of the objects in the collection.	0..*	Unordered	No controlled vocabulary provided.	Property	Optional data element.	(“10-14”, “some”)
C.19	Cost		0..*	Unordered	No controlled vocabulary provided.	Property	Optional data element.	(“no”, “most”)
C.20	To Metadata Collection	Related metadata collections	0..*	Unordered				
C.20.1	Metadata Collection Identifier (option 1)	Identifier of related metadata collection	1..1	-			Mandatory child data element (Mutually exclusive with C.20.2 Metadata Collection) (Refers to identifier of M.1)	
C.20.1.1	Catalog	The name or designat	1..1	-	Repertoire of ISO/I	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”

		or of the identification or cataloguing scheme for this metadata collection. A namespace scheme.			EC 10646-1:2000			
C.20.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this metadata collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
C.20.2	Metadata Collection (option 2)	Description of related metadata collection	1..1	-			Mandatory child data element. For further expansion, see M	
C.21	Contains	Content collections that this content collection contains	0..*	Unordered				
C.21.1	Content Collection Identifier (option 1)	Identifier of contained content collection	1..1	-			Mandatory child data element (Mutually exclusive with C.21.2 Content Collection) (Refers to identifier of C.1)	
C.21.1.1	Catalog	The name or designator of the identific	1..1	-	Repertoire of ISO/IEC 10646	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”

		ation or cataloguing scheme for this content collection. A namespace scheme.			- 1:2000			
C.21.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this content collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646 - 1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
C.21.2	Content Collection (option 2)	Description of contained content collection	1..1	-	-	-	Mandatory child data element. For further expansion, see C	-
C.22	Is Contained	Content collections that this content collection is contained by	0..*	Unordered				
C.22.1	Content Collection Identifier (option 1)	Identifier of containing content collection	1..1	-			Mandatory child data element (Mutually exclusive with C.22.2 Content Collection) (Refers to identifier of C.1)	
C.22.1.1	Catalog	The name or designator of the identification or cataloguing	1..1	-	Repertoire of ISO/IEC 10646 - 1:2000	normalizedString	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”

		scheme for this content collection. A namespace scheme.						
C.22.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this content collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
C.22.2	Content Collection (option 2)	Description of containing content collection	1..1	-	-	-	Mandatory child data element. For further expansion, see C	-

Metadatos colección de metadatos.

Nr	Name	Description	Multiplicity	Order	Value space	Data type	Note	Example
M.1	Identifier	A globally unique label that identifies this collection of metadata.	1..1	-	-	-	Mandatory data element.	-
M.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
M.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this metadata collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
M.2	Description	A description of the collection.	1..1	-	-	LangString	Mandatory data element.	(“en”, “Collection of Algebra Assessments for 3rd grade”)
M.3	Collection Rights	A description of the rights that apply to	0..1	-	-	LangString	Optional data element.	(“en”, ”Creative Commons Attribution Share-Alike 3.0”)

		the collection as a whole.						
M.4	Keywords	A list of keywords for describing the collection as a whole.	0..*	Unordered	-	LangString	Optional data element.	(“en”, “assessment”), (“en”, “mathematics”)
M.5	Average Annual Increase	The average annual increase in the number of metadata objects in the collection. This can be a negative number.	0..1	-	-	Integer	Optional data element.	“100”, “-50”
M.6	Accrual Periodicity	How often new metadata objects are added to (or removed from) the collection.	0..1	-	No controlled vocabulary provided.	VocabularyTerm	Optional data element. A possible vocabulary for this element is DCCAP [DCCAP, 07] http://dublincore.org/groups/collections/frequency/2007-03-09/	“freq:annual”, “freq:threeTimesAMonth”, “freq:semiweekly”
M.7	Quality Procedure	The quality procedure(s) applied to the items in the collection.	0..*	Unordered.	No controlled vocabulary provided.	Property	Optional data element.	(“no quality procedure”, “all”) (“peer-reviewed”, “some”)
M.8	Language	The language(s) of the learning objects in the collection.	0..*	Unordered	ISO 639	Property	Optional data element.	(“en”, “all”), (“fr”, “some”)
M.9	Format	The technical format(s	0..*	Unordered	A valid MIME type.	Property	Optional data element.	“application/swf”.

		) of the metadata objects in the collection.						
M.10	Item Rights	The rights that apply to the metadata objects in the collection.	0..*	Unordered	No vocabulary provided.	Property	Optional data element. This is independent of Collection-level Rights, which govern access to collection.	(“All rights reserved, copyright big money limited 2010”, “some”)
M.11	Responsible	Parties responsible for the collection	0..*	Unordered	-	-	Optional data element.	-
M.11.1	Responsibility	The responsibility the party has with regard to the collection.	1..1	-	No vocabulary provided.	VocabularyTerm	Mandatory child data element.	“administrative contact”, “technical contact”
M.11.2	Contact Details	A VCARD description of the party.	1..1	-	A valid VCARD [VCARD, 98]	CharacterString	Mandatory child data element.	BEGIN:VCARD VERSION:2.1 N:kapukod;Hétszin FN:4758 ORG:Bubba Gump Shrimp Co. TITLE:Shrimp Man TEL;CELL;VOICE:2536 END:VCARD
M.12	Target	Service end-point for the collection	0..*	Unordered	-	-	Optional data element	-
M.12.1 (option 1)	Target Identifier	A globally unique label that identifies a target of the collection (end-point	1..1	-	-	-	Mandatory data element . (Mutually exclusive with M.12.2 Target Description)	-

		for services offered)						
M.12.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this target. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“hdl”, "URI"
M.12.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this target. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“101.1021323”, “http://foo.org/1234”
M.12.2 (option 2)	Target Description	A description of a target of the collection (end-point for services offered)	1..1	-	-	-	Mandatory child data element. (Mutually exclusive with M.12.1 Target Identifier) (For further expansion, see T)	-
M.13	To Content Collection	Related content collections	0..*	Unordered	-	-	Optional element.	-
M.13.1	Content Collection Identifier (option 1)	Identifier of related content collection	1..1	-	-	-	Mandatory child data element (Mutually exclusive with M.20.2 Content Collection) (Refers to identifier of	-

							C.1)	
M.13.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this content collection. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
M.13.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this content collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
M.13.2	Content Collection (option 2)	Description of related content collection	1..1	-	-	-	Mandatory child data element. For further expansion, see C	-
M.14	Contains	Metadata collections that this metadata collection contains	0..*	Unordered	-	-	Optional data element.	-
M.14.1	Metadata Collection Identifier (option 1)	Identifier of contained metadata collection	1..1	-	-	-	Mandatory child data element (Mutually exclusive with M.21.2 Metadata Collection) (Refers to	-

							identifier of M.1)	
M.14.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this metadata collection. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
M.14.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this content collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
M.14.2	Metadata Collection (option 2)	Description of contained metadata collection	1..1	-	-	-	Mandatory child data element. For further expansion, see M	-
M.15	Is Contained	Metadata collections that this content collection is contained by	0..*	Unordered	-	-	-	-
M.15.1	Metadata Collection Identifier (option 1)	Identifier of containing metadat	1..1	-	-	-	Mandatory child data element (Mutually exclusive	-

		a collection					with M.22.2 Metadata Collection) (Refers to identifier of M.1)	
M.15.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this metadata collection. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“hdl”, "ISBN", "lre", "URI",”doi”
M.15.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this metadata collection. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalized String	Mandatory data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
M.15.2	Metadata Collection (option 2)	Description of containing metadata collection	1..1	-	-	-	Mandatory child data element. For further expansion, see M	-

Apendice C: IMS LODE CQL

Nr	Name	Description	Multiplicity	Order	Value space	Data type	Note	Example
Root	**Work**	**An abstract view of a learning object that captures the commonalities between all the possible variations of this object.**	-	-	-	-	**With expression, manifestation, and item, work is one of the four possible root elements for an ILOX instance.**	-
1	Identifier	A globally unique label that identifies the learning object work.	0..*	Unordered	-	-	Optional data element.	-
1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“hdl”, "ISBN", "lre", "URI",”doi”
1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this learning object work. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
2	Description	A multi-faceted descriptio	0..*	Unordered	-	-	Optional data element.	-

		n of the LO work under considera tion.						
2.1	Facet	The name of the facet of the LO work that is described .	0..1	-	No vocabulary defined.	ValueFromVocab	Optional data element. If absent, it is recommend ed to provide a default value.	“default”, “accessibility”, “rights”
2.2	Metada ta	A metadata instance describin g the facet defined in element 2.1 Descripti on.Facet.	1..1	-	-	-	Mandatory child data element.	-
2.2.1	Schem a	The namespac e of the schema used for the descriptio n.	0..1	-	A valid URI.	URI.	Optional data element. When used, this URI is the namespace of the schema used to define the metadata at the extension point 2.2.2.	“http://ltsc.ieee.org/xsd/LOM”
2.2.2	<extension>	The metadata instance containin g the descriptio n.	1..1	-	A valid metadata instance.	A metadata instance used to describe the LO work facet defined in element 2.1 Description.Facet. This should be a valid instance of the schema identified by the namespace provided in element 2.2.1 Description.Metadata.Schema.	Mandatory child data element.	A LOM instance, a Dublin Core instance,
Root / 3	Expres sion	The expressio ns (versions) of the LO work.	1..*	Unor dered	-	-	Can either be a mandatory data element of Work or the ILOX root element.	-
3.1	Identifi er	A globally	0..*	Unor dered	-	-	Optional data	-

		unique label that identifies this learning object expression.					element.	
3.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“hdl”, "ISBN", "lre", "URI",”doi”
3.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this LO expression. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
3.2	Dimension	The criteria (e.g., language, version, accessibility) used to distinguish between the different expressions of a learning object work.	0..*	unordered	-	-	Dimension is optional when Expression is the ILOX root element and mandatory otherwise.	-
3.2.1	Name	The name of the dimension (or criteria) according to which	1..1	-	No vocabulary defined.	ValueFromVocab	Mandatory child data element.	“language”, “version”

		the different expressions of the work differ.						
3.2.2	Parameter	The actual value of the dimension criteria for this expression.	0..1	-	No vocabulary defined.	ValueFromVocab	Mandatory child data element.	“en”, “fr”, “1.0”, “beta 6”
3.3	Description	A multi-faceted description of the LO expression under consideration.	0..*	Unordered	-	-	Optional data element.	-
3.3.1	Facet	The name of the facet of the LO expression that is described.	0..1	-	No vocabulary defined.	ValueFromVocab	Optional data element. If absent, it is recommended to provide a default value.	“default”, “rights”
3.3.2	Metadata	A metadata instance describing the facet defined in element 3.3.1 Description.Facet.	1..1	-	-	-	Mandatory child data element.	-
3.3.2.1	Schema	The namespace of the schema used for the LO expression description.	0..1	-	A valid URI.	URI.	Optional data element. When used, this URI is the namespace of the schema used to define the metadata at the extension point 3.3.2.2.	“http://ltsc.ieee.org/xsd/LOM”
3.3.2.2	<extension>	The metadata	1..1	-	-	A metadata instance used to describe the LO	Mandatory child data	A valid LOM instance

		instance containing the description.				manifestation facet defined in element 3.3.1 Expression.Description. Facet. This should be a valid instance of the schema identified by the namespace provided in element 3.3.2.1 Expression.Description. Metadata.Schema.	element.	
Root / 3.4	Manifestation	The manifestations (embodiments) of the LO expression.	1..*	Unordered	-	-	Can either be a mandatory data element of Expression or the ILOX root element.	-
3.4.1	Identifier	A globally unique label that identifies this learning object manifestation.	0..*	Unordered	-	-	Optional data element.	-
3.4.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“hdl”, "ISBN", "lre", "URI",”doi”
3.4.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this LO manifestation. A namespace specific string.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”
3.4.2	Name	The name of the LO	1..1	-	A controlled	ValueFromVocab	Dimension is optional	“package in”, “experience”

		manifestation.			vocabulary including at least the values provided in Section 4.6.1.		when Manifestation is the ILOX root element and mandatory otherwise.	
3.4.3	Parameter	A parameter further defining the manifestation.	0..1	-	The value space of this element depends on the value of element 3.4.2 Manifestation.Name. See section 5 for details.	ValueFromVocab	Usage depends on the value of element 3.4.2 (Manifestation.Name)	Value depends on the value of element 3.4.2 (see Section 4.6.2).
3.4.4	Description	A multi-faceted description of the LO manifestation under consideration.	0..*	Unordered	-	-	Optional data element.	-
3.4.4.1	Facet	The name of the facet of the LO manifestation that is described.	0..1	-	No vocabulary defined.	ValueFromVocab	Optional data element.	“default”, “rights”
3.4.4.2	Metadata	A metadata instance describing the facet defined in element 3.4.1 Description.Facet.	1..1	-	-	-	Mandatory child data element.	-
3.4.4.2.1	Schema	The namespace of the schema used for the LO manifestation description.	0..1	-	A valid URI.	URI.	Optional data element. When used, this URI is the namespace of the schema used to define the metadata at	“http://ltsc.ieee.org/xsd/LOM”

							the extension point 3.4.2.2.	
3.4.4.2.2	<extension>	The metadata instance containing the description.	1..1	-	-	A metadata instance used to describe the LO manifestation facet defined in element 3.4.1 Manifestation.Description.Facet. This should be a valid instance of the schema identified by the namespace provided in element 3.4.2.1 Manifestation.Description.Metadata.Schema.	Mandatory child data element.	A valid LOM element.
Root/ 3.4.5	Item	The items (copies) of an LO manifestation.	1..*	Unordered	-	-	Can either be a mandatory data element of Manifestation or the ILOX root element.	-
3.4.5.1	Identifier	A globally unique label that identifies this learning object item.	0..*	Unordered	-	-	Optional data element.	-
3.4.5.1.1	Catalog	The name or designator of the identification or cataloguing scheme for this entry. A namespace scheme.	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“hdl”, "ISBN", "lre", "URI",”doi”
3.4.5.1.2	Entry	The value of the identifier within the identification or cataloguing scheme that designates or identifies this learning object	1..1	-	Repertoire of ISO/IEC 10646-1:2000	normalizedString	Mandatory child data element.	“DB123456”, "2-7342-0318", "LEAO875", “http://foo.org/1234”

		item. A namespace specific string.						
3.4.5.2	Location	The location of the LO item.	1..1	-	-	-	Mandatory data element.	-
3.4.5.2.1	URI	The URL of the LO item or of a resolution service that can provide it.	1..1	-	A valid URL	URI	Mandatory data element.	“http://fire.eun.org/LREMAPv4p0.pdf”
3.4.5.2.2	Metadata	A piece of metadata further describing the LO item location.	0..1	-	-	-	Optional data element.	-
3.4.5.2.2.1	Schema	The namespace of the schema used for the description of the LO item location.	0..1	-	A valid URI.	URI.	Optional data element. When used, this URI is the namespace of the schema used to define the metadata at the extension point 3.4.5.2.2.2.	“http://ltsc.ieee.org/xsd/LOM”
3.4.5.2.2.2	<extension>	The metadata instance that actually contains the description.	1..1	-	-	This element can only be used when A metadata instance used to describe the location. This should be a valid instance of the schema identified by the namespace provided in element 3.4.5.2.2.1 Item.Location.Description.Schema.	Mandatory child data element.	A DREL expression instance.
3.4.5.3	Description	A multi-faceted description of the copy (item) of the LO item	0..*	Unordered	-	-	Optional data element.	-

		under consideration.						
3.4.5.3.1	Facet	The name of the facet of the LO item that is described.	0..1	-	No vocabulary defined.	ValueFromVocab	Optional child data element.	
3.4.5.3.2	Metadata	A metadata instance describing the facet defined in element 3.4.5.3.1 Description.Facet.	1..1	-	-	-	Mandatory child data element.	-
3.4.5.3.2.1	Schema	The namespace of the schema used for the description.	0..1	-	A valid URI.	URI.	Recommended data element. When used, this URI is the namespace of the schema used to define the metadata at the extension point 3.4.5.3.2.2.	“http://ltsc.ieee.org/xsd/LOM”
3.4.5.3.2.2	<extension>	The metadata instance containing the description.	1..1	-			Mandatory data element.	A LOM instance.