

Learning Objects universal publishing and location Architecture using Web Services

Antonio Ortiz¹, Salvador Otón², Roberto Barchino³

¹ Universidad de Alcalá, Departamento de Ciencias de la Computación,
28871 Alcalá de Henares, Spain
ortiz_baillo@yahoo.es

² Universidad de Alcalá, Departamento de Ciencias de la Computación,
28871 Alcalá de Henares, Spain
salvador.oton@uah.es

³ Universidad de Alcalá, Departamento de Ciencias de la Computación,
28871 Alcalá de Henares, Spain
roberto.barchino@uah.es

Abstract. This paper presents a solution to the learning objects (LO) location and search, that improve their reusability so that, the educational possibilities related to the Information and Communication Technologies (TIC) will be higher. To reach a complete reusability of LOs, it is necessary to reference some standards, that allow us to describe them using the metadata. The problem we have described is not only to be able to reuse LOs over different existing learning platforms, but it is to give a universal solution to its search and reusing. To solve this problem, we need an architecture that allows the possibility for intercommunication between different architectural components and that it allows that developed systems over different platforms and different programming languages can interoperate and the possibility for resources in the Internet to be discovered. This is the Web Services architecture case.

Arquitectura para publicación y localización universal de Objetos de Aprendizaje mediante Servicios Web

Antonio Ortiz Baíllo, Salvador Otón Tortosa, Roberto Barchino Plata

Departamento de Ciencias de la Computación
Escuela Técnica Superior de Ingeniería Informática
Universidad de Alcalá
28871 Alcalá de Henares
ortiz_baillo@yahoo.es, salvador.oton@uah.es, roberto.barchino@uah.es

Resumen. El trabajo presenta una solución a la localización y búsqueda de objetos de aprendizaje (LO: *Learning Object*) que favorezca su reutilización, y que de esta manera, las posibilidades educativas relacionadas con las Tecnologías de la Información y la Comunicación (TIC), sean mayores. Para poder conseguir una completa reutilización de los LO, es necesario recurrir a diferentes estándares, que nos permitan su descripción, utilizando para ello metainformación. El problema planteado, no sólo reside en habilitar una reutilización de LO sobre las diferentes plataformas de aprendizaje existentes, sino proporcionar una solución universal a su búsqueda y reutilización. Para solucionar este problema, necesitamos una arquitectura que posibilite la intercomunicación entre diferentes arquitecturas de componentes, y que sistemas desarrollados en diferentes plataformas y diferentes lenguajes de programación, puedan interactuar, así como que permita el descubrimiento de recursos en Internet, como es el caso de la arquitectura de los servicios Web.

1 Introducción a los sistemas e-learning.

Ya en 1986, colaborando con *Standard Eléctrica*, un equipo de personas encargadas de contribuir a que el progreso técnico se pusiera al servicio del aprendizaje continuo en las empresas, empezaron a diseñar para diferentes clientes, módulos que anteriormente se conocían como “Enseñanza Asistida por Ordenador (EAO)”. La inquietud por las nuevas tecnologías y metodologías educativas, ya había surgido en los años 70, pero podemos decir que el aprendizaje por ordenador comenzó en España entrados los años 80.

Fue en esos primeros años de la década de los 80, cuando comenzó a popularizarse el PC, y el consiguiente concepto de interactividad o comunicación hombre-máquina. La aparición del PC abrió la puerta al aprendizaje individualizado por ordenador. Al principio se pensó, que la difusión del aprendizaje por ordenador sería muy rápida, pero no fue así, aunque el empeño de las empresas productoras de cursos no cejó. Algunas oportunas ayudas oficiales permitieron seguir todavía de cerca el avance tecnológico. Ante las crecientes posibilidades del software y el hardware, se abrió el reto de poner la tecnología al mejor servicio del aprendizaje.

En los 90, se empezó a hacer productos en *CD-I*: por ejemplo, el proyecto *500 años después*, para la *Sociedad Estatal del Quinto Centenario*. También se hicieron proyectos interesantes en *CD-ROM*: por ejemplo, el proyecto *LETRA*, del *Programa ATENEA* del Ministerio de Educación. Ya en la primera mitad de los años 90, numerosas empresas e instituciones realizaron proyectos importantes: Anaya Interactiva, Telefónica I+D, Chadwyck-Healey, BSI Multimedia, Espasa Calpe, Zeta Multimedia, Creatividad y Tecnología, Edicinco, FYCSA... En la segunda mitad de esta década, empezó a extenderse el uso de Internet en las empresas de sectores vinculados a la tecnología, y pronto llegaría el aprendizaje *on line*.

Al comenzar el siglo, las grandes empresas (de TMT, Banca, Servicios...) parecían haber apostado decididamente por el aprendizaje *on line*, incorporando los cursos a sus intranets o redes corporativas, como complemento o alternativa a la tradicional formación continua presencial. Miles de directivos y trabajadores, de diferentes grados de cualificación profesional, se habían sumado en España a este método de aprendizaje autoconducido, ya denominado *e-learning*. Este tipo de formación se caracteriza por dos factores:

- El aprendizaje se podía hacer en cualquier lugar, cualquier hora, con tal de disponer de una conexión a Internet.
- La gestión y control de la formación se centralizaba en un servidor accesible a través de la web, denominado LMS (*Learning Management System*).

En el año 2003, un estudio de *Santillana Formación*, reveló la insatisfacción de los usuarios ante este tipo de metodologías de aprendizaje. La única dificultad que tenían, era la de pasar páginas Web sin ningún tipo de fundamento, y por lo tanto, los alumnos solían abandonar antes de terminar la formación. El cambio se produjo de manera extraordinaria hacia Internet, se crearon multitud de empresas dedicadas exclusivamente a la creación de contenidos formativos, mucho más dinámicos y atractivos y, como no, de LMS's o Plataformas o Herramientas de Gestión del Aprendizaje. En la actualidad nos encontramos en esta situación, donde existen multitud de herramientas, pero que sin duda alguna el mercado aglutinará en unas pocas, tanto de creación de contenidos docentes, como de los propios portales de e-learning [1].

El gran potencial de esta tecnología, reside por lo tanto en la distribución y reutilización de objetos de aprendizaje, entendiéndose como tales, las unidades mínimas en las que se puede organizar el material de formación. Para que la búsqueda, localización y referenciación de LO sea efectiva, es necesario que dicho objeto se construya usando diferentes estándares, como por ejemplo SCORM, cuya principal característica reside en la descripción del LO utilizando para ello metainformación. Solucionado el problema de la reutilización de LO sobre los diferentes LMS existentes, surge ahora otro problema, ¿cómo puedo acceder a otros objetos de aprendizaje fuera de mi plataforma? O justo el ejemplo contrario, ¿cómo puedo yo publicar y hacer accesibles desde el exterior mis objetos docentes?

Para resolver estas dos cuestiones, es donde entran en juego los servicios Web, como motor que posibilita realizar estas acciones de una manera efectiva, a través de estándares abiertos como WSDL, UDDI, SOAP y XML.

2 Los servicios Web son la solución.

Si intentamos centrar el estado actual del desarrollo de aplicaciones basadas en la Web, podemos encontrar una gran cantidad de tecnologías, muchas de ellas incompatibles entre sí, y lo que es peor, inaccesibles a través de Internet. Pero si una conclusión podemos obtener de esto, es que las arquitecturas basadas en tecnología de componentes, están tomando un papel principal dentro del desarrollo Web.

Los servicios Web, se proponen como una alternativa para facilitar la intercomunicación entre diferentes arquitecturas de componentes, ofreciendo una visión de dichas arquitecturas, basada en servicios, totalmente compatible con Internet.

La aparición de los servicios Web, y de las Arquitecturas Orientadas a Servicios (SOA), supone el establecimiento de nuevos mecanismos de comunicación B2B, B2C, B2E. El organismo encargado de definir estos estándares, y asegurar este comportamiento, es el WS-I (*Web Services Interoperability Organization*, [2]). Gracias a este organismo, va a ser posible que sistemas desarrollados en diferentes plataformas y diferentes lenguajes de programación, puedan interactuar.

La Arquitectura Orientada a Servicios, pretende extender la idea de servicio Web, de forma que una invocación, de manera totalmente transparente al usuario, implique la ejecución de más de un servicio Web.

Un servicio Web, va a poner a nuestra disposición una serie de operaciones, invocables mediante HTTP/XML. Esta arquitectura, se basa en una serie de perfiles que realizan las tareas de:

- *Proveedor de servicios*: Componente que se encarga de poner un servicio a disposición de los clientes, y de publicar el *interface* del mismo.
- *Agente de servicio*: Encargado de registrar servicios publicados por un proveedor, así como de informar a posibles consumidores, de qué forma pueden acceder a un servicio Web.
- *Consumidor de servicios*: Elemento que se encarga de invocar a un servicio, y que, también realizará consultas al agente, para obtener información relativa a servicios.

La existencia de estos tres perfiles, implica la aparición de tres características funcionales básicas en el sistema:

- *Transporte*: Componente encargado de definir el formato y protocolos empleados a la hora de intercambiar la información, entre un cliente y un servicio Web. SOAP.
- *Descripción*: Componente encargado de describir el *interface* de un servicio Web, de forma que un cliente, pueda consultar dicha información. WSDL
- *Descubrimiento*: Encargado de realizar la publicación de un nuevo servicio Web, y de responder a búsquedas de servicios publicados. UDDI

La exposición del mecanismo de operación de los servicios Web, ha sido recuperada del "*Java Web Services Tutorial*", Sun Microsystems [3], y de la organización Web Services [4].

Según lo expuesto, podemos concluir que los servicios Web, son un conjunto de estándares, que permiten a los desarrolladores implementar aplicaciones distribuidas, construidas a partir de combinaciones de módulos software, que son llamados desde distintos sistemas distribuidos en regiones geográficas distintas. La arquitectura de los

servicios Web, permite que ciertos servicios sean dinámicamente descritos, publicados, descubiertos e invocados en un ambiente de cómputo distribuido, cualidad necesaria para llevar a cabo nuestro cometido, la búsqueda universal de objetos de aprendizaje.

3 Estándares que conforman los servicios Web.

Los estándares que conforman los servicios Web, son los siguientes:

- *XML: eXtensible Markup Language*, o Lenguaje Extensible de Marcas en Español, fue creado en Febrero de 1998. Revolucionó la forma en la que estructuramos, describimos e intercambiamos información. Hoy en día, todas las tecnologías de servicios Web se basan en XML. Deriva principalmente de SGML y de HTML. Será utilizado por tanto, como elemento para representar toda la información que se trasmite utilizando esta arquitectura, ya que permite la descripción y el formateo de datos, separando claramente el contenido de su presentación, y la extensibilidad del lenguaje, mediante la cual, se conseguirá crear lenguajes específicos al área de negocios. [3, 5]
- *UDDI: Universal Description, Discovery and Integration*. Es un protocolo que permite describir los componentes disponibles de servicios Web. Este estándar permite a las empresas registrarse, y les ayuda a anunciar sus servicios. Es un Framework abierto, independiente de la plataforma. Fue una iniciativa de Microsoft e IBM. UDDI es el protocolo que permitirá buscar y registrar servicios Web. [3, 6]
- *SOAP: Simple Object Access Protocol*. Es un protocolo que permite iniciar las conversaciones con un servicio UDDI. El protocolo SOAP simplifica el acceso a los objetos, permitiendo a las aplicaciones invocar métodos que residen en sistemas remotos. Es un estándar de comunicación basado en esquemas XML, mediante el cual, podemos intercambiar información estructurada y fuertemente tipada, en entornos descentralizados y distribuidos. Es sencillo, extensible y fácil de desplegar en Internet, ya que está basado en estándares (codificación XML, tipado XSD y transporte HTTP), facilitando la interoperabilidad entre múltiples entornos. [3, 7]
- *WSDL: Web Service Description Language*. Es el estándar propuesto para la descripción de los servicios Web, el cual consiste en un IDL de servicio, basado en XML, que defina la interfaz del servicio y sus características de implementación. Este estándar va a permitir por tanto, describir los servicios Web, indicándole a los usuarios la forma de acceder a ellos, y de realizar sus llamadas remotas. [3, 8].

4 La pareja perfecta: servicios Web y Objetos de Aprendizaje.

Se han comentado las principales características de la arquitectura de los servicios Web, así como los protocolos que la conforman. A continuación se explicará cómo vamos a utilizar esta tecnología en beneficio de los objetos de aprendizaje.

En primer lugar, se expondrá cuál es la metodología de funcionamiento de esta arquitectura, basándose en los protocolos que la sostienen.

Después de desarrollar un servicio Web, el primer paso a realizar será el de su descripción. Para ello, se utilizará el estándar WSDL. Gracias a este paso, se podrá desarrollar una descripción que posibilite su almacenamiento y su posterior búsqueda, que es el principal cometido de esta tecnología.

Una vez desarrollado el documento WSDL de descripción del servicio Web, el siguiente paso será almacenarlo en un lugar de acceso común, como es un registro UDDI. Gracias al estándar UDDI, se podrá almacenar la descripción de dicho servicio Web.

Cuando un cliente requiera cierta funcionalidad, podrá recurrir a un registro UDDI para consultar las descripciones de servicios que realicen la funcionalidad deseada. De tener éxito la búsqueda solicitada, gracias al documento WSDL, el cliente sabrá la manera de acceder a ese servicio.

Una vez obtenida toda la información sobre cómo es el acceso a ese servicio Web remoto, sólo queda realizarlo, para ello, contamos con un protocolo como SOAP, que permite realizar esta acción de una manera sencilla.

Esta es la forma, mediante la cual, se permite la localización e invocación de servicios distribuidos, pero no se ha citado la utilidad de XML en este campo. XML es el lenguaje utilizado para el envío y recepción de toda la información del sistema. WSDL es el lenguaje para la descripción del servicio, dicho documento WSDL está escrito en XML, al igual que las peticiones de registro y búsqueda en UDDI. También el acceso remoto al servicio, que se realiza gracias al protocolo SOAP, utiliza XML (a bajo nivel) como lenguaje para la comunicación. Por lo tanto, estamos ante el estándar más importante y fundamental.

De esta forma, se facilita un acceso remoto completo a determinados servicios Web, que proporcionan la funcionalidad deseada, sin que sea necesario el desarrollo de un aplicativo ya existente, ahorrando con ello el consiguiente gasto. Esto es posible, gracias a que los servicios Web permiten la intercomunicación entre diferentes arquitecturas de componentes, y que sistemas desarrollados en diferentes plataformas y diferentes lenguajes de programación, puedan interactuar.

A continuación se muestra un gráfico explicativo de la metodología de funcionamiento que se acaba de comentar:

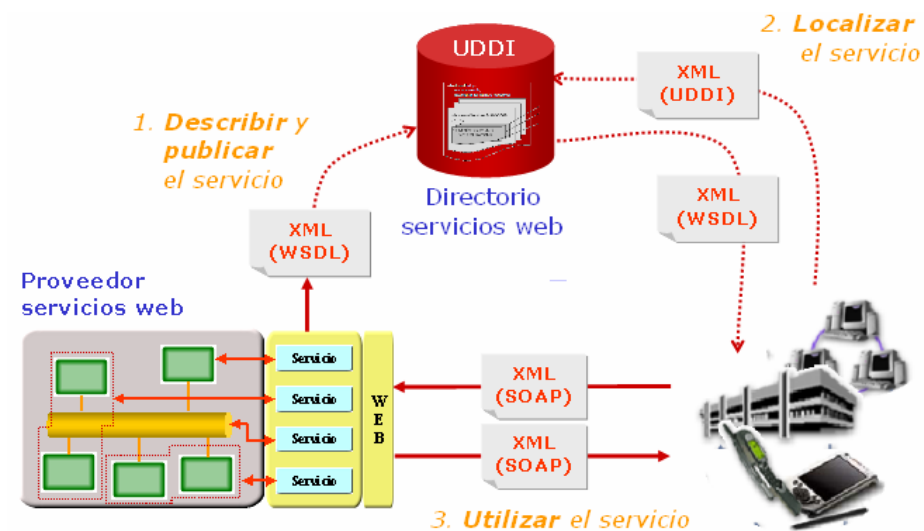


Figura 1. El modelo integral de servicios Web.

Para la generación de los objetos de aprendizaje, se tienen estándares que permiten su reutilización en diversas plataformas de teleformación. Entre estos destacan: LOM (*Learning Object Metadata*) [9] de IEEE, que sirve para caracterizar un objeto de aprendizaje mediante su metainformación, IMS (*Instructional Management System*) [10] creado por Global Learning Consortium, que define como diseñar y empaquetar contenidos docentes, SCORM (*Shareable Content Object Reference Model*) [11] desarrollado por ADL que adopta las especificaciones de IMS y permite la aplicación de esos estándares, y DCMI (*Dublin Core Metadata Initiative*) [12] estándar de catalogación o de metadatos más comúnmente utilizado en las bibliotecas digitales.

Cuando una plataforma de formación quiera dar acceso a los objetos de aprendizaje que almacena en su repositorio, debería crear uno o varios servicios Web que involucran los métodos de búsqueda interna que posea, de forma que se pueden realizar búsquedas tan precisas como sea posible, y que luego se traducen en llamadas a métodos de búsqueda internos del repositorio, cuyo resultado serán los objetos de aprendizaje que coincidan con los patrones de búsqueda especificados.

Se ha comentado, que la información que se registra en UDDI, es una descripción en la que se detalla la manera de contactar con el servicio buscado, y la funcionalidad ofrecida por éste. Por lo tanto, si lo que se pretende es posibilitar una distribución de LO, equiparable a la realizada con servicios Web, se deberá incluir la metainformación que describe al LO en el documento WSDL. De esta manera, se podrá utilizar el motor de búsqueda de servicios Web, UDDI, para buscar objetos de aprendizaje y SOAP para acceder a ellos de una manera remota.

4.1 Dublin Core como estándar para la elaboración de metainformación.

Dublin Core es un conjunto de elementos de metadatos creados con la intención de facilitar la descripción de recursos electrónicos. Las principales características de Dublin Core son: simplicidad, interoperabilidad semántica consensuada internacionalmente, extensibilidad y modularidad de los metadatos en la Web.

Desde el punto de vista práctico de este trabajo, se ha decidido utilizar Dublin Core como estándar para la elaboración de la metainformación, ya que la aplicación de DCMI es de carácter muy general, lo cual favorecerá la descripción de cualquier tipo de elemento. Además DCMI, posee una propuesta sencilla de esta Iniciativa, que se conoce como DCMI Simple, conformada por los siguientes quince elementos, agrupados en tres grandes grupos [13]:

- Elementos relacionados principalmente con el contenido del recurso.
- Elementos relacionados principalmente con el recurso cuando es visto como una propiedad intelectual.
- Elementos relacionados principalmente con la instanciación del recurso.

| <i>Contenido</i> | <i>Propiedad Intelectual</i> | <i>Instanciación</i> |
|------------------|------------------------------|---------------------------|
| Título | Autor | Fecha |
| Claves | Editor | Tipo de Recurso |
| Descripción | Otros Colaboradores | Formato |
| Fuente | Derechos | Identificador del Recurso |
| Lengua | | |
| Relación | | |
| Cobertura | | |

Tabla 1. Lista de Elementos de DCMI Simple

Por lo tanto, gracias a esta pequeña disposición de campos, sería posible una descripción sencilla y eficiente de un objeto de aprendizaje, sin tener que utilizar una gran cantidad de elementos descriptores, sin que esto fuera totalmente necesario. La mayor parte de estos elementos, deberán ser formados con la utilización de vocabularios controlados y sistemas de clasificación formales, para que de esta manera, se favorezcan las búsquedas de objetos docentes, algunos ejemplos de estas recomendaciones se pueden ver en [14, 15, 16].

Para mejorar la búsqueda y accesibilidad de objetos docentes, en el caso que sea necesario, podemos extender algunos de éstos elementos, a través de los calificadores del estándar de metadatos Dublin Core (Dublin Core Qualifiers [17]). El DCMI reconoce dos grandes clases de calificadores:

- Refinación de elementos: Estos calificadores hacen que el significado de un elemento sea más ceñido o más específico. Un elemento refinado comparte el significado del elemento no calificado, pero con un alcance más restrictivo. Si un agente no entiende un término de refinamiento específico para un elemento, debe ser capaz de ignorarlo y tratar el valor del metadato como si estuviese sin calificar.

- Esquema de codificación: 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 con formato, de acuerdo con una notación formal (por ejemplo, “2002-01-01” como la expresión estándar de una fecha). Si un esquema de codificación no es comprensible por un agente, el valor será de todas maneras, útil para un lector humano.

4.2 Afinando más la arquitectura propuesta.

Basándonos en el sistema que se está detallando, el repositorio necesario, debería contener únicamente los metadatos de objetos de aprendizaje. Con esta arquitectura, un cliente podrá buscar objetos de aprendizaje accediendo al registro UDDI, y consultar las respectivas descripciones WSDL, hasta localizar el objeto buscado, a través de la búsqueda en los respectivos campos Dublin Core que se describieron anteriormente. Una vez que se haya localizado, será necesario acceder a las etiquetas *<binding>* y *<service>* también del documento WSDL, para saber cómo y dónde acceder al objeto docente. A través del protocolo SOAP, podremos acceder remotamente a este servicio Web, utilizando mensajería SOAP con archivos adjuntos (SAAJ), y de esta manera transmitir el objeto docente, facilitando así la reutilización, que es la funcionalidad que se está buscando con esta arquitectura, la reutilización de objetos de aprendizaje.

En este punto del trabajo, surge otra cuestión, y es la inexistencia de un estándar común para elaborar la descripción de un objeto de aprendizaje. Entonces, ¿qué resultado daría una búsqueda a través de etiquetas o elementos bajo el estándar LOM en un repositorio con LO descritos utilizando DCMI?

En primer lugar, indicar que sería necesario llegar a un consenso mundial en la determinación de cuáles son los elementos mínimos imprescindibles, para que las búsquedas sean totalmente efectivas.

En segundo lugar, el problema de la transformación entre diferentes sistemas de descripción, tiene una solución sencilla, ya que los metadatos utilizados por el sistema DCMI es un subconjunto de otros metadatos definidos por el estándar LOM. Como muestra de la sencillez que supondría esta transformación, se ha realizado una tabla explicativa con los elementos de ambos estándares y su relación en el Anexo 1. Es importante destacar que tanto IMS como SCORM se basan en LOM para describir su metainformación, por lo tanto, esta relación también es aplicable a estos estándares.

Con la arquitectura que se está presentado, y gracias a la equivalencia que se acaba de comentar, también se puede proporcionar el añadido de transformar el sistema utilizado para describir el objeto de aprendizaje buscado por el cliente, adaptándolo a la plataforma de gestión del aprendizaje, garantizando aún más si cabe, la compatibilidad y reutilización ofrecida por esta arquitectura. De tal forma que el cliente no sólo buscaría un objeto de aprendizaje en particular, sino que también seleccionaría el estándar con el que desearía que fuera descrito.

5 Conclusiones

Para que las posibilidades educativas relacionadas con las Tecnologías de la Información y la Comunicación sean grandes, es preciso que los elementos constituyentes de este modelo docente, los objetos de aprendizaje, sean totalmente reutilizables. Una solución a este problema, es utilizar para su localización y distribución, la arquitectura de los servicios Web, como medio que permite el descubrimiento de recursos en Internet, a través de estándares abiertos.

Uno de los factores que determinarán el auge de los sistemas de formación *on line*, es una estandarización del modelo de identificación de objetos docentes. De lo contrario, esto imposibilitará que las búsquedas de objetos educativos, tengan todo el éxito que se espera de esta tecnología. Para solventar este problema, se presenta una arquitectura abierta que posibilita realizar esta tarea de una manera eficiente, permitiendo una búsqueda universal de objetos docentes, incluso adaptándolos al sistema de descripción solicitado por el cliente, para garantizar así una compatibilidad total con la herramienta de gestión del aprendizaje utilizada.

6 Bibliografía

1. eLearning Workshops. Enebral, J. Dieciocho años del e-learning en España (2004): <http://www.elearningworkshops.com>
2. Web Services Interoperability Organization, <http://www.ws-i.org>
3. Sun Microsystems. (2004), "The Java Web Services Tutorial" <http://java.sun.com/webservices>
4. Web Services ORG. (2002), <http://www.webservices.org>
5. World Wide Web Consortium (W3C), "eXtensible Markup Language (XML)": <http://www.w3.org/XML>.
6. Universal Description, Discovery and Integration (UDDI): <http://www.uddi.org> <http://uddi.microsoft.com/> y <http://www-3.ibm.com/services/uddi/>
7. World Wide Web Consortium (W3C), "Simple Object Access Protocol (SOAP)": <http://www.w3.org/TR/SOAP/> y <http://www.develop.com/soap>
8. World Wide Web Consortium (W3C), "Web Services Description Language (WSDL)": <http://www.w3.org/TR/wsdl>.
9. IEEE Learning Object Metadata (LOM). Institute of Electrical and Electronics Engineers, (2002). <http://ltsc.ieee.org>.
10. IMS Group Management Services WSDL Binding. (2004), <http://www.imsglobal.org>
11. ADL Advanced Distributed Learning Initiative (2004): <http://www.adlnet.org>
12. Dublin Core Metadata Initiative: DCMI Metadata Terms (2003): <http://www.dublincore.org>
13. Hillman, D. Using Dublin Core, Dublin Core Metadata Initiative (2003), <http://dublincore.org/documents/2003/08/26/usageguide/>
14. DCMIType vocabulary, <http://dublincore.org/documents/dcmi-type-vocabulary/>
15. Date and Time Formats, <http://www.w3c.org/TR/Note-datetime>
16. Lengua: Etiqueta DC:Language, <http://www.faqs.org/rfcs/rfc1766.html>
17. Hillman, D. Using Dublin Core, Dublin Core Qualifiers (2003), <http://dublincore.org/documents/2003/08/26/usageguide/qualifiers.shtml>

Anexo 1. Tabla de conversión de elementos DCMI a IEEE LOM

| <i>Campo</i> | <i>Metadato DCMI</i> | <i>Descripción del Campo</i> | <i>Equivalente IEEE LOM</i> |
|---------------------------|----------------------|--|--|
| Título | DC.Title | El nombre dado a un recurso, normalmente proporcionado por el autor. | <i>general.title</i> |
| Autor | DC.Creator | La persona u organización responsable de la creación del contenido intelectual del recurso. | <i>lifecycle.contribute</i> |
| Claves | DC.Subject | Expresa las claves o frases que describen el título o el contenido del recurso. | <i>general.keywords</i> |
| Descripción | DC.Description | Una descripción textual del recurso, como puede ser un resumen o una descripción de su contenido. | <i>general.description</i> |
| Editor | DC.Publisher | La entidad responsable de hacer que el recurso se encuentre disponible en la red en su formato actual. | <i>lifecycle.contribute.role</i> |
| Otros Colaboradores | DC.Contributor | La persona u organización que haya tenido una contribución intelectual significativa en la creación del recurso, pero cuyas contribuciones son secundarias en comparación a las de las personas u organizaciones especificadas en el elemento <i>Creator</i> | <i>lifecycle.contribute</i> con el tipo de contribución especificado en <i>lifecycle.contribute.role</i> |
| Fecha | DC.Date | Una fecha en la que el recurso se puso a disposición del usuario en su forma actual. | <i>lifecycle.contribute.date</i> |
| Tipo de Recurso | DC.Type | La categoría del recurso. | <i>Sin equivalencia</i> |
| Formato | DC.Format | El formato de datos y/o las dimensiones (tamaño, duración...) de un recurso usado para identificar el software y posiblemente, el hardware que se necesitaría para mostrar el recurso. | <i>technical.format</i> |
| Identificador del Recurso | DC.Identifier | Secuencia de caracteres usados para identificar unívocamente un recurso. Por ejemplo la URL del recurso. | <i>general.catalogentry</i> |
| Fuente | DC.Source | Secuencia de caracteres utilizado para identificar unívocamente un trabajo a partir del cual proviene el recurso actual. | <i>relation.source</i> |
| Lengua | DC.Language | Lengua/s del contenido intelectual del recurso | <i>general.language</i> |
| Relación | DC.Relation | 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. | <i>Sin equivalencia</i> |
| Cobertura | DC.Coverage | La característica de cobertura espacial y/o temporal del contenido intelectual del recurso. | <i>general.coverage</i> |
| Derechos | DC.Rights | Una referencia (URL, por ejemplo) para una nota sobre derechos de autor, para un servicio de gestión de derechos o para un servicio que dará información sobre términos y condiciones de acceso a un recurso. | <i>rights.description</i> |