OAC: recolección y agregación de metadatos heterogéneos para un proveedor de servicios OAI

JosuKa Díaz, Inés Jacob, Joseba Abaitua, Fernando Quintana, Jon Fernández, Txus Sánchez, Garikoitz Echebarría

Grupo DELi (http://www.deli.deusto.es/)
[1]Dpto. de Ingeniería del Software, [2]Dpto. de Filología Inglesa, [3]Tecnológico Deusto
Universidad de Deusto
Apartado 1 - 48080 Bilbao
josuka@eside.deusto.es[1], ines@eside.deusto.es[1], abaitua@fil.deusto.es[2], fquintan@eside.deusto.es[3], jonferna@tecnologico.deusto.es[3], jesanche@tecnologico.deusto.es[3], gecheba@eside.deusto.es[3]

Josu Azpillaga

CodeSyntax
(http://www.codesyntax.com/)
Azitaingo Industrialdea 3K
20600 Eibar (Gipuzkoa)
jazpillaga@codesyntax.com

Resumen

Se presenta una implementación del protocolo OAI-PMH para el servidor web Zope, que incluye el repertorio completo de funciones: proveedor de datos, recolector de metadatos, pasarela estática y proveedor de servicios. Con el objetivo inicial de mejorar la gestión de metadatos documentales para grandes corpora de textos multilingües del grupo DELi, la implementación en la actualidad sirve de referencia para la normalización de criterios de catalogación del patrimonio cultural en el País Vasco. Desde el punto de vista técnico, el uso de Zope como soporte web para OAI se ha revelado muy apropiado, dada la arquitectura de base de datos para documentos que aplica este servidor web, lo que permite usos novedosos del protocolo OAI-PMH en cuestiones como la diseminación selectiva y la consulta de metadatos.

Palabras clave: OAI; interoperabilidad y formatos de metadatos; diseminación de metadatos; servicios sobre OAI; Zope.

1. Introducción

El proyecto OAC (*Open Archives Cataloger*), desarrollado por el grupo de investigación DELi de la Universidad de Deusto y la empresa CodeSyntax desde otoño de 2003, tiene por objeto difundir las bondades de OAI (*Open Archives Initiative* [16]) y facilitar la adopción del protocolo por parte de cualquier institución, o grupo editor que desee compartir sus metadatos. El proyecto surge para dar respuesta a necesidades propias de DELi y de CodeSyntax, pero se está utilizando como ejemplo de normalización de criterios para la catalogación en el marco del *Plan de digitalización, preservación y difusión del patrimonio cultural vasco*, promovido por el Departamento de Cultura del Gobierno Vasco, actualmente en fase de estudio. En opinión de los autores, el estándar OAI ofrece un modelo muy apropiado para que grupos heterogéneos de creadores, preservadores o archiveros de contenidos digitales compartan los metadatos de sus colecciones en la forma de un catálogo unificado, actualizado y de fácil mantenimiento.

En diciembre de 2004, durante la *39. Euskal Liburu eta Disko Azoka* (39ª Feria del Libro y Disco Vasco de Durango), DELi y CodeSyntax presentaron un prototipo [3] del sistema con metadatos aportados por varios organismos del País Vasco. Un sencillo motor de búsqueda mostraba de manera clara e intuitiva la funcionalidad final del sistema.

En su parte técnica, el proyecto incluye la programación del repertorio completo de componentes del protocolo OAI-PMH, así como el desarrollo de los módulos particulares de diseminación de metadatos en los agentes de contenidos. Tanto DELi como CodeSyntax usan la tecnología del servidor web Zope [22]. Por ello, los componentes básicos del protocolo OAI-PMH se están desarrollando para este sistema, basándose en algunos desarrollos parciales ya existentes, y bajo postulados de software libre, es decir, el producto final será de libre uso, distribución y mejora.

Las características de Zope resultan especialmente apropiadas en el ámbito de la gestión documental web y se revelan muy eficaces

Actas de las V Jornadas de Bibliotecas Digitales, JBiDi'2005, pp.11-18
ISBN: 84-9732-453-6

para la implantación de la metodología OAI. Por esa razón, se han encontrado posibles aplicaciones novedosas del protocolo, que se salen del objetivo propio de OAI, y que no resultarían tan fáciles de implantar en otros sistemas documentales web.

El trabajo se organiza de la siguiente forma. Primero, se presentan los elementos de OAI necesarios para comprender la aportación del proyecto. A continuación, se describen de forma detallada las características técnicas de la implementación en Zope. En la siguiente sección se muestran las virtudes de Zope como gestor de contenidos, que lo hacen particularmente apropiado para OAI. En la sección quinta se describen dos opciones para diseminar metadatos que la implementación en Zope permite: diseminación selectiva y diseminación por consulta, que extiende la anterior. La comunicación se completa con la mención a un experimento que sondea el aprovechamiento de ontologías bibliográficas como medio de potenciar el descubrimiento y consulta de los registros.

2. Open Archives Initiative

La experiencia acumulada en los repositorios de documentos electrónicos (EPrints, arXiv o NCSTRL) llevó a finales de la pasada década a la creación de un estándar de interoperabilidad, conocido como *Open Archives Initiative* (OAI, [16, 15]), dirigido a facilitar la diseminación eficiente de metadatos.

La infraestructura técnica actual de OAI está recogida en el *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH, [10]), que define un mecanismo para que proveedores de contenidos puedan mostrar sus metadatos y estos puedan ser aprovechados por otras aplicaciones. El protocolo se encuentra actualmente en su versión 2.0.

Una prueba reciente del éxito de OAI es que en marzo de 2005 tanto *CiteSeer* como la *National Library of Australia* (a través de *Google*) lo han adoptado como protocolo de intercambio de información [2, 13].

2.1. Arquitectura

El protocolo OAI-PMH define dos clases de participantes: proveedores de datos (*data providers*, que exponen los metadatos de sus contenidos), y proveedores de servicios (*service providers*, que almacenan conjuntos de metadatos y proveen aplicaciones sobre los mismos).

Los programas necesarios para implementar el protocolo son, a su vez, dos: el recolector de metadatos (*harvester*, usado por un proveedor de servicios para realizar peticiones de metadatos a los proveedores de datos) y el repositorio (*repository*, programa que responde, en el proveedor de datos, a las peticiones de un recolector de metadatos).

El protocolo distingue tres entidades en la constitución de un repositorio: *recurso* (*resource*, entendido como el objeto en sí mismo), *ítem* (o "imagen" del recurso en el repositorio, sobre el que hablan los metadatos, y en particular, con un *identificador único*) y *registro* (*record*, colección de metadatos en un formato específico que describen un ítem). De esta forma, la respuesta de un proveedor de datos a una petición OAI-PMH realizada por un proveedor de servicios se define como un *conjunto de registros* expresados en XML, en un formato (o esquema XSD) definido explícitamente por el protocolo.

Existe la posibilidad de colocar *agregadores* (*aggregators*) recolectando metadatos (registros) de varios proveedores de datos. También está contemplado que un proveedor de datos con un número inferior a 5.000 registros pueda exponer sus metadatos de una forma más simple, mediante una *pasarela estática* (*static repository gateway*): se trata simplemente de un archivo XML con el esquema de registros definido en el protocolo.

En resumen, aspectos cruciales del éxito de OAI-PMH son la sencillez de su arquitectura, el asentamiento en tecnologías de Internet básicas como HTTP o XML, y el enfoque de recolección centralizada de metadatos (aunque no de recursos), opuesta a la más tradicional búsqueda distribuida (*cross search*).

2.2. Formatos de metadatos

Otro aspecto relevante en OAI-PMH es la independencia respecto de los tipos de recursos, lo que se manifiesta en los formatos de expresión de metadatos admitidos. El objeto de recolección en OAI se ha ampliado desde el "documento" (en un sentido clásico) al "recurso" (en el sentido más laxo posible): literalmente "cualquier cosa con identidad" [15].

Este enfoque abre un enorme abanico de posibilidades para la aplicación del protocolo OAI-PMH, pero plantea el problema de elegir un formato adecuado de expresión de metadatos que pueda ser capaz de reflejar las características de *cualquier* recurso. Por supuesto, tal formato no existe, por lo que se eligió un repertorio mínimo, que es *Dublin Core* (DC, [5]). El esquema `oai_dc` del protocolo está basado en los 15 elementos de DC sin calificar y asegura la interoperabilidad básica: un proveedor de datos *puede* diseminar sus metadatos en varios formatos, pero al menos *debe* hacerlo en `oai_dc`.

Ello no es óbice para que se puedan utilizar otros formatos (BibTeX, EAD, MARC, RFC1807, UDDI/WSDL, etc.), bajo la condición de que exista el esquema XSD correspondiente.

2.3. Recolección selectiva

La *recolección selectiva de metadatos* permite a un recolector solicitar un conjunto limitado de registros. Desde un primer momento, se rechazó un enfoque de consulta (*query*), por contradecir el objetivo de simplicidad requerido para la máxima interoperabilidad, y se definieron únicamente dos mecanismos, por fechas y por conjuntos, de los cuales incidiremos en el segundo.

Un repositorio puede organizar sus ítems en conjuntos (*sets*), de forma jerárquica, y un recolector puede solicitar en la petición OAI-PMH únicamente los registros de un determinado conjunto. Para aplicar este mecanismo, cada registro en el repositorio lleva en su cabecera marcas de pertenencia a conjuntos.

El protocolo define una petición especial para que un proveedor de datos muestre su estructura jerárquica de conjuntos: la respuesta consiste en una lista con todos los nodos de la jerarquía a cualquier profundidad.

Es importante recalcar que no hay prácticamente ningún a priori en cuanto a la organización en conjuntos que puede adoptar un repositorio. Específicamente, un ítem puede estar presente en un conjunto, en varios (incluso hermanos, por ejemplo) o en ninguno (dicho exactamente, un registro puede tener cero o más marcas de pertenencia a conjuntos). Por otro lado, la jerarquía no ha de responder necesariamente a un criterio coherente de distribución: por ejemplo, puede haber una partición por tema (“física”, “química”), otra por lengua (“inglés”, “español”), y una tercera aplicarse solo a algunos ítems (“artículos en revista”).

3. Implementación de OAI-PMH en Zope

El proyecto OAC (*Open Archives Cataloger*) incluye en su parte técnica la implementación de la infraestructura OAI completa en el servidor web Zope. La elección de Zope está motivada, entre otras cosas, por su adecuación para el desarrollo de sistemas documentales accesibles en la web, aspecto que se desarrolla con más profundidad en la siguiente sección.

3.1. Implementación de Pentila

La solución propuesta en el proyecto OAC está basada en el *producto* (módulo que extiende el sistema Zope básico) llamado *ZOpenArchives*, creado por la compañía francesa Pentila [17]. Este producto Zope implementa la parte básica de las funcionalidades definidas por OAI-PMH.

La característica más importante de *ZOpenArchives* es su diseño modular y la variedad de entidades OAI-PMH que ofrece. El núcleo del producto, *pyOAIPMH*, está escrito en el lenguaje orientado a objetos Python, es independiente de la plataforma Zope y es el responsable de la lógica de intercambio de metadatos definida por el protocolo OAI-PMH. Por otro lado, existe un nutrido grupo de clases (en el sentido de la programación orientada a objetos) que proporcionan la interfaz adecuada para añadir las funcionalidades de OAI a objetos Zope:

- El proveedor de datos, *zOAIRepository*, tiene como misión responder a las peticiones OAI-PMH, generando la respuesta XML.
- El proveedor de datos Zope, *ZopeOAIServer*, permite que una base datos Zope se convierta en un proveedor de datos (*zOAIRepository*).
- El recolector, *zOAIHarvester*, se encarga de realizar peticiones OAI-PMH a un repositorio de datos; es también el responsable de analizar la respuesta XML obtenida y generar e indexar un objeto de registro por cada recurso recolectado.
- El agregador, *zOAIAggregator*, incluye tantos recolectores (*zOAIHarvester*) como

proveedores de datos se quieran consultar; como se sabe, un agregador es, a su vez, proveedor de datos.

- Los registros *zOAIRecord* almacenan la información relativa a cada recurso recolectado; cada registro cuenta con varias *propiedades* (contenedores de información equivalentes en Zope a los *campos* de una base de datos) que conservan las secciones `<header>`, `<metadata>` y `<about>` de cada registro OAI.

Una de las novedades del enfoque de Pentila es que permite utilizar el formato de almacenamiento de metadatos del agregador o del proveedor de servicios (registros *zOAIRecord*) en la implementación de un proveedor de datos en Zope (*ZopeOAIServer*), con lo cual un proveedor de datos individual podría disponer de funcionalidad de proveedor de servicios, si así se desea (para mantenimiento y evaluación, por ejemplo).

El proyecto OAC añade varios componentes y funciones a *ZOpenArchives* para resolver o mejorar ciertos aspectos que el producto de Pentila deja sin abordar.

3.2. OAC: formatos diversos de metadatos

El producto *ZOpenArchives* admite solo el lenguaje de metadatos exigido por el estándar OAI, es decir, *Dublin Core* sin calificar (formato `oai_dc`). Sin embargo, el propósito original de OAC es la recolección y agregación de metadatos provenientes de fuentes muy diversas y, por lo tanto, heterogéneos en cuanto a su expresión.

Esto ha planteado dos problemas. En primer lugar, ha sido necesario modificar el recolector (*zOAIHarvester*) para que admita metadatos codificados con cualquier esquema XML. En segundo lugar, se ha tenido que habilitar la capacidad para almacenar dichos metadatos (*zOAIRecord*) tanto en el agregador como en el proveedor de servicios, lo cual es un problema conceptualmente más complejo.

La solución OAC, en consonancia con lo expuesto en el apartado 2.2, consiste en definir un superconjunto de metadatos que sea la unión semántica (es decir, sin repetición de elementos que representen la misma información) de los formatos de metadatos seleccionados. En la actualidad, además de DC sin calificar, están incluidos DC calificado, BibTeX [19] y MARCXML [11]. Se ha denominado *lenguaje neutro de metadatos* a este superconjunto (similar al concepto de *lingua franca* utilizado por Chan y Zeng [1, 21]) y es el que usa internamente OAC (con la nueva versión de *zOAIRecord*) para el almacenamiento de los registros OAI.

Existen funciones de conversión entre el lenguaje neutro y cada uno de los demás (DC, BibTeX, MARCXML), en ambos sentidos. Un recolector *zOAIHarvester* usa la función de conversión en *sentido directo* para transformar los metadatos obtenidos (en MARCXML, por ejemplo) al lenguaje neutro. Un proveedor de servicios puede usar la conversión en *sentido inverso* (lenguaje neutro a MARCXML) para las aplicaciones, de manera que la interfaz de estas puedan adaptarse a un formato de metadatos concreto, si así se desea, para un usuario con experiencia en el mismo.

3.3. OAC: nuevos componentes OAI

El proyecto OAC añade nuevos componentes de la arquitectura OAI a la solución *ZOpenArchives*. En primer lugar, se ha incorporado el soporte de pasarelas estáticas (tal como se definieron en el apartado 2.1), necesarias para el proyecto en su primera fase, dado que los proveedores de datos no disponen aún de la infraestructura OAI.

En segundo lugar, se han añadido operaciones básicas de los proveedores de servicios, que incluyen la indización automática de la base de registros (*zOAIRecord*) en Zope y funciones primitivas de consulta, junto con la infraestructura que soportará el desarrollo de la interfaz de usuario de la aplicación.

4. Zope como proveedor de datos OAI

La elección de Zope, a priori un servidor más de aplicaciones web, como agente técnico para OAI parece anecdótica. Sin embargo, la cercanía de Zope al concepto de *gestor de contenidos web* hace que se revele como una plataforma muy adecuada para la implantación en particular de un proveedor de datos, y podría permitir fácilmente otros usos de OAI que se están desarrollando en estos momentos [14, 18] en otros contextos.

Zope organiza un sitio web como una *base de datos transaccional orientada a objetos*, cuyos componentes pueden ser objetos de varias clases (con diferentes funcionalidades, entre otras la de contener otros objetos), y organizados de forma jerárquica (en árbol). Algunos de estos objetos pueden corresponder de forma directa a páginas concretas del sitio web, pero otros se pueden manejar mediante complejas técnicas de gestión dinámica. Por ello, se puede considerar a Zope como un gestor de contenidos, o también en cierto modo, gestor documental.

Es importante reseñar que a cada objeto de la base de datos se le puede asociar una serie de *propiedades*, en la terminología de Zope, pero que no son otra cosa que *metadatos* acerca del objeto. Es posible implementar cualquier esquema de definición de metadatos conocido, o variaciones o combinaciones de los mismos.

Por otro lado, Zope permite incluir, en cualquier punto de su estructura de objetos, un indizador llamado *ZCatalog*, que puede actuar bien sobre el contenido de los objetos, bien sobre los propios metadatos. Ello permite disponer de manera rápida de funciones de búsqueda y filtrado potentes y eficientes.

Una prueba de ello es el sistema SARE-Bi [4] (http://www.deli.deusto.es/SareBi): un gestor documental de textos multilingües (español, euskera, inglés, y otras lenguas) segmentados y alineados, que pueden utilizarse como memorias de traducción (TMX). Cada documento es un objeto de la clase `DeliTei`, anotado con una serie de metadatos (derivados del estándar TEI) y compuesto de varios subdocumentos, cada uno de la clase `DeliLang`, según la lengua. A su vez, cada uno de estos subdocumentos está formado por segmentos textuales, objetos de la clase `DeliSeg`, con el contenido textual completo y alineados en las distintas lenguas mediante una marca específica.

SARE-Bi presenta al usuario funciones de filtrado por los metadatos o de búsqueda según el contenido, mediante la indización del texto completo realizada por *ZCatalog*, que permiten visualizar los textos segmentados y alineados.

La extensión de un sistema documental como SARE-Bi a repositorio o proveedor de datos OAI es técnicamente simple con ayuda del producto *ZOpenArchives* de Pentila y las extensiones OAC recogidas en la sección precedente. En particular, el módulo *ZopeOAIServer* aporta la funcionalidad básica, indizando los metadatos de los documentos en un catálogo orientado a OAI llamado *OAICatalog*. Este catálogo es el que permite construir de manera rápida y eficaz el conjunto de registros XML que conforman la respuesta a un recolector de metadatos.

5. Usos novedosos de OAI en Zope

La arquitectura de Zope permite otros usos novedosos de OAI, y lo que es más importante, por el mismo precio que implementar la funcionalidad OAI estándar.

Imaginemos que el grupo de investigación FOO ha construido en su web en Zope un gestor de sus publicaciones (`http://foo.es/pub/`). Al añadir al gestor la cualidad de repositorio (proveedor de datos) OAI, la dirección anterior ha de responder a las peticiones de los recolectores de metadatos, mediante la URI:

[a] `http://foo.es/pub/oai?...`

Hasta aquí se trata del enfoque OAI tradicional.

5.1. Diseminación selectiva de metadatos

Ahora supongamos que los objetos documentales se han organizado en Zope jerárquicamente, mediante "carpetas": `en` para las publicaciones en inglés, `es` para español y `eu` para euskera. Entonces, la dirección:

[b] `http://foo.es/pub/eu/oai?...`

proporcionaría solamente los metadatos de las publicaciones en euskera.

Es importante entender que el recurso `oai` que se aplica en las dos direcciones [a] y [b] es exactamente el mismo. Esto es posible gracias al mecanismo de Zope denominado *adquisición* [12], que extiende el concepto de *herencia* en los lenguajes orientados a objetos: un objeto Zope adquiere propiedades y comportamientos de su *contexto*. El contexto de un objeto depende tanto de su disposición en el árbol de objetos (herencia), como de la forma en que es invocado en la URI (adquisición). De esta forma, puede decirse que un

objeto puede utilizar los propios recursos que contiene y los que contienen aquellos objetos que aparecen en la URI utilizada para invocarlo. En el caso de [b], por tanto, el programa `oai` es un recurso del objeto `pub`, pero como el objeto `eu` está contenido en `pub`, se le pueden aplicar los recursos de este último.

Esta posibilidad se muestra similar a la recolección selectiva de metadatos que permite el mecanismo de los conjuntos en OAI, explicado en el apartado 2.3: si se dispone de tres conjuntos (`en`, `es`, `eu`), la petición [b] sería equivalente a solicitar los registros del conjunto `eu`.

Este comportamiento puede ser interesante si el sitio `foo` quiere servir a distintos recolectores de metadatos, cada uno con propósitos distintos. Por ejemplo, puede servir con la dirección [a] a un proveedor de servicios especializado en publicaciones científicas de todo tipo, pero también puede servir con la dirección [b] a un servicio de publicaciones en euskera (que no está interesado en las demás).

Este comportamiento lo denominamos *diseminación selectiva de metadatos*, por contraposición a la *recolección selectiva* basada en conjuntos explicada en el apartado 2.3. Sin embargo, una profundización en la comparación de ambas capacidades muestra aspectos positivos y negativos. En cuanto a los primeros, siguiendo con el ejemplo, hay que decir que al proveedor de servicios de publicaciones en euskera le resultaría algo más sencilla esta solución, porque solo ha de anotar la dirección [b] como proveedor de datos. Con el mecanismo de los conjuntos contemplado en OAI, tendría que anotar la dirección genérica [a] del proveedor, y el conjunto `eu` que le interesa solicitar.

En la parte negativa, hay que decir que no siempre es posible reflejar una estructura de conjuntos (en el sentido definido por OAI) de forma jerárquica. La razón es que los conjuntos no tienen que responder al mismo criterio taxonómico. Para seguir con el ejemplo, las publicaciones del grupo FOO podrían dividirse por lenguas, pero también por tipo (libros, artículos) o por año, y es evidente que esta estructura múltiple (los mismos objetos perteneciendo a diferentes carpetas) no puede representarse jerárquicamente de forma directa. Podría hacerse, eso sí, de forma indirecta, almacenando todos los objetos documentales al mismo nivel, y creando carpetas para los diferentes criterios que almacenen únicamente referencias a los objetos reales.

5.2. Diseminación de metadatos por consulta

Lo precedente nos lleva a pensar en el posible significado de la siguiente petición:

```
[c] http://foo.es/pub/
          gomez/paper/eu/oai?...
```

donde `gomez` es uno de los investigadores del grupo FOO. Sin otro referente, podríamos interpretar la dirección [c] como una petición para obtener los documentos del autor `gomez`, que sean artículos de revista (`paper`) y estén escritos en euskera (`eu`), es decir, recuerda a la *consulta* a una base de datos.

La diferencia cualitativa con el caso anterior es que los constituyentes `gomez`, `paper` y `eu` de [c] no han de entenderse como nodos de la jerarquía de la base de datos, sino como calificadores de la petición `oai` al repositorio `pub`, en el cual los objetos documentales pueden encontrarse tanto inestructurados (sin jerarquía) como estructurados según alguna taxonomía, que no tiene que ser exactamente la que reflejan los elementos (autor, tipo, lengua) de la consulta.

Los mecanismos de herencia y adquisición de Zope permiten, por lo tanto, configurar la diseminación de metadatos mediante un conjunto de objetos "filtro" que, si bien no llega a aportar la funcionalidad completa de un lenguaje de consultas, tiene indudables aplicaciones.

Es importante matizar una vez más que el proveedor de datos, el recurso `oai`, no precisa modificación alguna, es decir, sigue incorporando estrictamente la funcionalidad contemplada en el estándar OAI-PMH. No se pretende extender esta capacidad a otros componentes de la arquitectura OAI (recolector, proveedor de servicios), sino habilitar mecanismos simples que aprovechan el propio estándar para manejar de forma flexible un conjunto de recursos y sus metadatos.

6. Perspectivas futuras

Como medio de potenciar la utilización de los catálogos y el descubrimiento de la información

recolectada, se está analizando la utilización de *ontologías* de metadatos bibliográficos, aprovechando que los registros pueden representarse tanto en BibTeX como en DC y que existen ontologías disponibles para ambas versiones de metadatos. Estas ontologías permiten recurrir a la aplicación de reglas de inferencia y descubrimiento basadas en las relaciones conceptuales que se establecen entre los metadatos. Sin embargo, el aprovechamiento es sólo parcial, ya que carecemos de una ontología temática que haga posible las labores de descubrimiento basadas en esquemas clasificatorios de los contenidos, como sugiere Welty [20]. No tenemos noticia de la existencia de una ontología que recoja alguno de los sistemas de clasificación temática de registros bibliográficos (LCC/LCSH, DDC, UDC, IFLA). Estos esquemas facilitarían la realización de búsquedas temáticas, el acceso multilingüe y la interoperabilidad con otros servicios [8].

Descartada momentáneamente la utilización de esquemas de clasificación temática, la cuestión metodológica que nos planteamos es ¿en qué medida una ontología para metadatos estructurales (como la de BibTeX o DC en OWL) puede favorecer el descubrimiento de información bibliográfica? El reto es importante, habida cuenta de la existencia de poderosos mecanismos de consulta únicamente basados en el aprovechamiento de los metadatos y la búsqueda por caracteres (DBLP, *CiteSeer*, *Citebase* y *Scholar* de *Google*) [6].

7. Conclusión

Se ha mostrado que es posible considerar la arquitectura OAI como un criterio adicional en el propio diseño de un sistema de gestión de documentos accesible por web, y que ello aporta la ventaja adicional de facilitar la incorporación de dichos recursos a los sistemas proveedores de servicios previstos por dicha arquitectura. La utilización en particular del servidor Zope como soporte del gestor documental se ha revelado muy adecuada gracias a su arquitectura de base de datos de objetos, y permite la incorporación de sencillos mecanismos como la diseminación selectiva o la diseminación por consulta, que extienden la funcionalidad de un proveedor de datos, sin comprometer en ningún caso el cumplimiento del estándar.

La implementación OAC de la arquitectura OAI se está aplicando en diversos contextos, algunos internos del grupo DELi, como la gestión de grandes corpora de textos multilingües (sistema SARE-Bi) o el conjunto de recursos bibliográficos propios, y otros con proyección externa, como la catalogación de recursos del patrimonio cultural vasco. En todos los casos, están surgiendo cuestiones relacionadas con la heterogeneidad de fuentes y de recursos, lo que ha llevado a la búsqueda de diversas soluciones, como son la definición de una superestructura o lingua franca para la expresión de metadatos, o el uso de ontologías para mejorar el aprovechamiento de los recursos y facilitar su descubrimiento.

8. Agradecimientos

Este trabajo se ha realizado con ayudas del Dpto. de Industria, Comercio y Turismo del Gobierno Vasco y de la empresa CodeSyntax, en el marco de los proyectos "OAC" (*Open Archives Cataloger*, S-OD03UD09) y "OAC-onto" (*Open Archives Cataloger: ontologías y metadatos*, S-OD04UD04), ambos en el programa SAIOTEK. Agradecemos especialmente a Gari Araolaza, Eneko Astigarraga, Josu Azpillaga y Luistxo Fernández (responsables de CodeSyntax) su apoyo constante al proyecto. También deseamos agradecer a Badihardugu, Gaztelupeko Hotsak, Gerediaga Elkartea, Ibinagabeitia Proiektua, Inguma, Lanbide Ekimena y Megadenda la colaboración prestada como agentes de contenidos en euskera, y en particular, la aportación de partes significativas de sus bases de datos documentales para las pruebas del proyecto.

Referencias

[1] Lois Mai Chan. "Metadata Interoperability. A Study of Methodology", *The 3rd China-US Library Conference*, Shanghai (China), 22-25 marzo, 2005, <http://www.nlc.gov.cn/culc/paper/Lois%20Mai%20Chan...Metadata%20Interoperability-A%20Study%20of%20Methodology.pdf>.

[2] CiteSeer.IST. *OAI Compliance*, <http://citeseer.ist.psu.edu/oai.html>.

[3] CodeSyntax. *Demo para un modelo de biblioteca distribuida y abierta*, <http://www.codesyntax.com/oai>.

[4] JosuKa Díaz Labrador, Joseba Abaitua Odriozola, Inés Jacob Taquet, Fernando Quintana Hernández y Garikoitz Araolaza. "Metadata for multilingual content management", *Translating and the Computer 25. Conference Proceedings*, Londres (Reino Unido), 20-21 noviembre, 2003, pp. 151-170, <http://www.deli.deusto.es/AboutUs/Publications#TaTC2003>.

[5] Dublin Core Metadata Initiative (DCMI). *Dublin Core Metadata Initiative*, <http://dublincore.org/>.

[6] Steve Hitchcock, Arouna Woukeu, Tim Brody, Les Carr, Wendy Hall and Stevan Harnad. *Evaluating Citebase, an open access Web-based citation-ranked search and impact discovery service*, Technical Report ECSTR-IAM03-005, School of Electronics and Computer Science, University of Southampton, 2003, <http://opcit.eprints.org/evaluation/Citebase-evaluation/evaluation-report-tr.html>.

[7] Henry N. Jerez, Xiaoming Liu, Patrick Hochstenbach, Herbert Van de Sompel. "The Multi-faceted Use of the OAI-PMH in the LANL Repository", *4th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'04)*, Tuscon, AZ (EE.UU.), 7-11 junio, 2004, pp. 11-20, <http://doi.acm.org/10.1145/996350.996355>.

[8] Traugott Koch, Anna Brümmer, Debra Hiom, Marianne Peereboom, Alan Poulter, Emma Worsfold. "The role of classification schemes in Internet resource description and discovery. Project deliverable D3.2", *DESIRE - Development of a European Service for Information on Research and Education*, EU project, 1997, <http://www.lub.lu.se/desire/radar/reports/D3.2.3/>.

[9] Carl Lagoze, Herbert Van de Sompel. "The Open Archives Initiative: Building a low-barrier interoperability framework", *1st ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL'01)*, Roanoke, VA (EE.UU.), 17-23 junio, 2001, pp. 54-62, <http://doi.acm.org/10.1145/379437.379449>.

[10] Carl Lagoze, Herbert Van de Sompel, Michael Nelson, Simeon Warner. *The Open Archives Initiative Protocol for Metadata Harvesting. Protocol Version 2.0 of 2002-06-14*, 2002, <http://www.openarchives.org/OAI/openarchivesprotocol.htm>.

[11] The Library of Congress. *MARCXML: MARC 21 XML Schema*, <http://www.loc.gov/standards/marcxml/>.

[12] Chris McDonough, Michel Pelletier, Shane Hathaway. *The Zope Developer's Guide (2.4 Edition)*, <http://www.zope.org/Documentation/Books/ZDG/>.

[13] National Library of Australia. *Digital Object Repository*, <http://www.nla.gov.au/digicoll/oai/>.

[14] Michael L. Nelson, Herbert Van de Sompel, Xiaoming Liu, Terry L. Harrison, Nathan McFarland. "mod_oai: An Apache Module for Metadata Harvesting", borrador, 2005, <http://arxiv.org/abs/cs.DL/0503069>.

[15] Open Archives Forum. *OAI for Beginners - the Open Archives Forum online tutorial*, 2003, <http://www.oaforum.org/tutorial/>.

[16] Open Archives Initiative, *Open Archives Initiative*, <http://www.openarchives.org/>.

[17] Pentila. *ZOpenArchives Product*, <http://www.pentila.com/produits/ZOpenArchives/>.

[18] Herbert Van de Sompel, Jeff Young, Thom Hickey. "Using the OAI-PMH ... Differently". *D-Lib Magazine*, volumen 9, números 7/8, 2003, <http://www.dlib.org/dlib/july03/young/07young.html>.

[19] Norman Walsh. *Help on BibTeX. Version 1.0, 12 Apr 94*, <http://www.nwalsh.com/tex/texhelp/BibTeX.html>.

[20] Christopher A. Welty. "The Ontological Nature of Subject Taxonomies", en N. Guarino (ed.) *Formal Ontology in Information Systems*, IOS Press, pp. 317-327, 1998, <http://www.cs.vassar.edu/faculty/welty/papers/fois-98/fois-98-1.html>.

[21] Marcia Lei Zeng, Lois Mai Chan. "Trends and issues in establishing interoperability among knowledge organization systems", *Journal of the American Society for Information Science and Technology*, vol. 55, nº 5, pp. 377-395, 2004. <http://dx.doi.org/10.1002/asi.10387>.

[22] Zope Community, *Welcome to Zope.org*, <http://www.zope.org/>.

Todas las referencias de documentos web han sido visitadas en abril de 2005.