Creación de directorios de recursos web con LDAP y RDF

Jose A. Senso y Pedro Hípola
Departamento de Biblioteconomía y Documentación. Universidad de Granada. jsenso@ugr.es phipola@ugr.es
mayo 2007

Resumen

Sobre la base de que la recuperación de información en bases de datos con recursos web descritos con metadatos ofrece mejores resultados que la que se realiza únicamente sobre texto completo, el presente trabajo pretende mostrar un modelo de trabajo con procedimientos generales que pueden ser aplicables a cualquier sistema de descripción y recuperación de información con metadatos en entornos distribuidos. Para ello se propone emplear RDF (Resource Description Framework) y LDAP (Lightweight Directory Access Protocol) y se describen los pasos seguidos en la realización de un proyecto real.

Palabras clave

Directorios web, Sistemas de metadatos, RDF, Resource Description Framework, LDAP, Lightweight Directory Access Protocol, Recuperación de información, Dublin Core, Web semántica.

1. Introducción

A pesar de las innumerables ventajas de los metadatos, son todavía pocos los sistemas de recuperación que se valen de ellos para enriquecer las consultas en bases de datos. Las causas por las que esto sucede tienen mucho que ver con dos dificultades evidentes:

- La existencia de múltiples sistemas de metadatos, con diferentes orientaciones y distintas formas de implementarse hace que cada uno de ellos se convierta en una maraña de códigos y sintaxis de difícil comprensión.
- El poco uso de los metadatos ha ayudado a que no existan modelos de trabajo que permitan establecer cómo se tiene que actuar para diseñar, crear, mantener y recuperar información en un sistema de consulta basado en metadatos.

Ambos motivos tienen como principal origen lo relativamente reciente de todas las tecnologías que están involucradas, en mayor o menor medida, en el entorno de trabajo de los metadatos.

El presente trabajo muestra los pasos seguidos en la creación de un sistema de recuperación de recursos web empleando metadatos en un modelo de trabajo con procedimientos generales.

Para llevar a cabo este objetivo lo ideal es diseñar un prototipo, que hace las veces de modelo de trabajo, que debería tener las siguientes características deseables:

- Compatibilidad total. Teniendo en cuenta que se pretende establecer un modelo de trabajo futuro es necesario que el prototipo permita que los recursos descritos se puedan migrar de un sistema a otro, independientemente del conjunto de metadatos utilizado o del mecanismo de recuperación ideado.
- Normalización. Todas las tecnologías implicadas deberían de tener un elevado grado de normalización con la finalidad de facilitar

la integración con otras aplicaciones –presentes o futuras- y los desarrollos posteriores.

A partir de las características, y teniendo en cuenta que lo se quiere lograr en formalizar un sistema de información de recursos web, el siguiente paso debe ser el definir las tecnologías que se utilizarán de base para su desarrollo. En realidad las opciones se deben centrar en localizar un sistema de metadatos que garantice al máximo la compatibilidad de los recursos descritos y, además, contar con un mecanismo de recuperación lo suficientemente flexible y fiable como para trabajar con esta nueva forma de información.

Desde el año 1999 hemos estado trabajando en experimentos de este tipo realizados dentro del proyecto SARAC, puesto en marcha por personal de RedIRIS y de la Universidad de Granada. Para obtener información detallada sobre este proyecto se pueden consultar los siguientes documentos:

http://www.ugr.es/~phipola/Proyecto%20Sarac.ppt
http://www.rediris.es/si/iris-search/coord/jt2000/jt2000-iris-search.ppt
http://sarac.rediris.es

2. RDF

Fue puesto en marcha en agosto de 1997 bajo los auspicios del World Wide Web Corsortium (W3C) con el fin de crear un formato que permitiera alcanzar la compatibilidad entre los diversos sistemas de metadatos, suministrando para ello una arquitectura genérica de metainformación. Con este objetivo se decidió utilizar el lenguaje XML como sistema de comunicación [1].

El primer borrador público data del 2 de octubre de 1997 (en agosto fue cuando se reunió por primera vez el grupo de trabajo que se encargaría del desarrollo del formato) y, tras diferentes esbozos, correcciones y propuestas, el 17 de febrero de 1999 aparece la última versión publicada como Recomendación W3C.

Tal y como afirma Hjelm [2], RDF es un formato que tiene como origen dos ramas recientes de la Documentación. Por un lado se encuentran los metadatos –al ser éste un sistema que, además de servir como modelo de metadatos, es capaz de interconectar sistemas entre sí— y, por otro, de la representación del conocimiento –encarnada ahora en el concepto de "semantic Web" [3]—.

La capacidad que tiene RDF para procesar metadatos facilita la interoperabilidad entre diversas aplicaciones, proporcionando un mecanismo perfecto para el intercambio de información a través del Web. Tal y como se afirma en la Recomendación W3C, RDF tiene distintas áreas de aplicación; como la recuperación de recursos (proporcionando mejores prestaciones a los motores de búsqueda), la catalogación en bibliotecas digitales (especificando también las relaciones de contenido disponibles en un sitio Web determinado), los agentes inteligentes (facilitando el intercambio de conocimiento), en sistemas de gestión de propiedad intelectual (expresando políticas de privacidad de un determinado objeto)...[4].

Todas las opciones de trabajo de este lenguaje le confieren las características de neutral (al no estar ligado explícitamente a ningún otro sistema de metadatos), expresivo (las etiquetas son muy intuitivas y se adivina fácilmente su posible contenido), familiar (su base SGML lo convierte en asequible para las personas relacionadas con HTML) y simple de procesar (al ser texto ASCII). Por último, no se puede obviar que detrás de RDF se encuentra una institución como el Consorcio W3, lo que le confiere cierto rango de estándar de facto.

3. LDAP

En lo que respecta al motor de indexación y recuperación, la elección siempre es algo más compleja, ya que como condición fundamental y como convicción personal, estos autores defienden que debe ser un software libre. La

primera opción en el desarrollo del prototipo fue Harvest [5]. Un motor de búsqueda muy extendido en Internet que funciona sobre máquinas Unix y/o Linux, que tiene una gran potencia y ha ofrecido unos buenos resultados en los últimos años –especialmente en recuperación a texto íntegro-.

El principal problema que planteó su uso, y por el cual fue desechado, se centraba en la escasa interacción con XML. Harvest utiliza un sistema de metadatos, SOIF (Summary Object Interchange Format), para realizar las búsquedas. Este formato no es compatible con RDF, por lo que era necesario generar un programa que convirtiera las descripciones RDF a formato SOIF. Una vez hecho esto, no existía seguridad alguna de que los resultados fuesen coherentes, ya que también existía un problema de compatibilidad con la información de salida.

Al final se optó por utilizar LDAP (Lightweight Directory Access Protocol) [6], ya que es extensible (adaptándose perfectamente a la sintaxis RDF propuesta para el prototipo), escalable (con lo cual se garantiza una recuperación fiable independientemente del número de registros que compongan la base de datos) y distribuido. Además, es capaz de realizar consultas con diferentes grados de complejidad.

Se trata de un estándar abierto que permite gestionar directorios basándose en sistemas X500[1]. De hecho, LDAP está considerado como la evolución lógica del antiguo sistema de directorio de OSI. Tanto es así, que a veces es denominado X500 Lite.

[1] El sistema X500 es excesivamente complejo y requiere contar con grandes recursos (principalmente en lo que se refiere al servidor) así como con una estructura OSI totalmente compatible. Por el contrario, LDAP funciona sobre cualquier ordenador personal y, además, es totalmente compatible con el protocolo TCP/IP, con lo que su integración dentro de Internet es total. En cuanto a la interconexión, ambos sistemas tienen un grado aceptable de compatibilidad. Lo que en la práctica supone que los mismos datos se pueden gestionar perfectamente en los dos sistemas pero, como desventaja, LDAP no es capaz de ejecutar algunas de las funciones propias de X500.

El uso fundamental que se le da a este protocolo es la gestión de información personal. En la práctica, aunque las bases de datos de este tipo suelen contener mucha información (números telefónicos, nombres y apellidos completos, direcciones postales, de correo electrónico...) se utilizan con mucha frecuencia ya que, por un lado, soportan con gran capacidad de respuesta un elevado volumen de tráfico y, por otro, la información que contienen no suele variar con frecuencia.

Desde que se generalizó su uso con X500, los directorios se han convertido en herramientas muy comunes en entornos informáticos, ya que contienen la misma información que se puede obtener en soporte papel de los miembros de una organización pero, gracias a su diseño, son capaces de soportar múltiples consultas en paralelo a través de diferentes servidores. En realidad el sistema funciona de forma parecida a los servidores de dominio DNS (Domain Name Service) [7].

En la mayoría de las ocasiones, los servicios de directorio LDAP sólo son accesibles para los miembros de una Intranet (en parte debido a la necesidad de privacidad que tienen muchos centros con respecto a determinado tipo de información sensible de la institución). No obstante, cada vez son más los servicios de directorio que se están utilizando para gestionar información en texto completo, abriendo, de esta forma, un nuevo campo de actuación hasta ahora reservado a grandes empresas que necesitan de plataformas muy potentes para trabajar.

Al igual que la mayoría de sistemas que han triunfado en Internet, LDAP está desarrollado sobre una arquitectura cliente-servidor, en la que un cliente LDAP se conecta a un servidor capaz de soportar ese mismo protocolo para solicitar o proporcionar información concreta sobre un objeto. Un objeto puede ser una persona que forma parte de una institución, un recurso de Internet o, en su concepto más amplio, un documento a texto completo.

En el caso de que se trate de una solicitud de información (es decir, una consulta), el servidor responde a esa petición enviando la búsqueda a

otro servidor que se encargará de ejecutarla. Si, por el contrario, lo que se pretende es introducir nueva información sobre un objeto, el servidor LDAP analiza la sintaxis de la misma y decide, sobre la base de ese análisis y a las reglas estipuladas por el administrador del sistema, aceptar o no la información suministrada con el fin de incorporarla al directorio.

Las principales ventajas que se obtienen del uso de LDAP en este prototipo son:

- Consolidación de la información. Todos los recursos de internet pueden ser fusionadas en un solo directorio, independientemente del origen de los datos.
- Gran facilidad para implementar y la coherencia de sus APIs, con lo que el número de aplicaciones y gateways que puedan ofrecer servicios LDAP pueden aumentar con el tiempo.
- Se trata de un sistema normalizado.
- Es extensible.
- Escalable, con lo que puede crecer conforme surjan nuevas formas de trabajo sobre la misma base.
- Gestión de usuarios, estableciendo mecanismos que permiten controlar sus acciones mediante la concesión o no de determinados privilegios.
- Es distribuido, lo que puede facilitar por un lado la creación de mirrors que descongestionen el tráfico en el servidor principal y, por otro, que su gestión se puede llevar a cabo a través de diferentes puntos.
- Es capaz de realizar consultas de tipo complejo, utilizando diferentes tipos de operadores booleanos, de proximidad, de adyacencia...

Dentro de cualquier sistema LDAP nos encontramos con una terminología común:

- Por un lado se encuentran las entradas, unidades particulares de un directorio (algo parecido a un registro dentro de cualquier tipo de base de datos).
- Cada entrada se identifica por un Distinguished Name (DN) que está compuesto por el nombre de la entrada en cuestión más la ruta de nombres que permiten rastrear la entrada desde atrás hasta la parte superior de la jerarquía del directorio.
- Los atributos de una entrada son los fragmentos de información asociados con dicha entrada (siguiendo la comparación con las bases de datos, este elementos se correspondería con los campos).
- Y, por último, LDIF (LDAP Data Interchange Format). Se trata de un fichero con formato ASCII que se utiliza para exportar e importar entradas LDAP al servidor.

4. Metodología de trabajo

Una vez decidida la tecnología a aplicar queda la fase de configuración. Por un lado será necesario estructurar las descripciones que se quieren realizar y asignarles una etiqueta meta correspondiente. Por otro habrá que preparar el software de indización y recuperación para que pueda gestionar toda la información almacenada.

Para la primera fase, y dado que se ha optado por trabajar con RDF, es imprescindible decidir qué sistema de metadatos es más conveniente para la descripción así como las etiquetas a emplear. En algunas ocasiones será incluso ineludible el proceso de crear nuevas etiquetas que permitan describir mejor los recursos web. Para el prototipo aquí descrito, por ejemplo, se optó por emplear las etiquetas Dublin Core [8] más alguna que otra implementada adhoc.

Decidirse por un conjunto de metadatos u otro (o incluso decidir si se quiere partir de cero y crear un sistema propio) es, sin duda alguna, el paso más comprometido que tiene la creación de un sistema de información de estas características. Hay que tener en cuenta que esta decisión determinará tanto cómo será la descripción del recurso como su nivel de profundidad. En muchas ocasiones se trata de un problema más político que técnico o informático.

Una vez se tiene la lista de etiquetas a utilizar es necesario contar con una herramienta que permita validar las descripciones RDF finales y que garantice que la información contenida en la

base de datos es uniforme y normalizada. Para ello es más que recomendable utilizar una DTD (Document Type Definition). Se podría definir DTD como una lista de normas que describen, de forma precisa, la composición y estructura de datos de un documento SGML/XML [9].

Un documento XML puede estar, con respecto a su correspondiente DTD, bien formado (es decir, que cumple con las normas gramaticales básicas de la DTD) y validado (cuando, además de bien formado, satisface todas las reglas estructurales de dicha DTD). Una de las ventajas fundamentales que tienen las DTDs dentro del XML es que no son obligatorias. Esta característica convierte al XML sin DTD en un documento libre en cuanto a estructura, facilitando de esa forma la proliferación de información en este formato. Por el contrario, cuando se cuenta con una DTD específica, el documento final debe cumplir con todas las características gramaticales y estructurales especificadas, por lo que se garantiza la plena compatibilidad del documento final.

Existen varias DTDs públicas, creadas por diferentes instituciones para fines propios, pero que se pueden utilizar como la base para otras DTDs más específicas. En algunos casos permiten realizar documentos que pueden ser visualizados através de un navegador web, como es el caso de CDF (Channel Definition Format) [10], MathML (Mathematical Markup Language) [11], XHTML (Extensible Hypertext Markup Language) [12] o del mismo RDF y, en otras ocasiones, es necesaria la utilización de un plug-in específico para poder ver el resultado final, como ocurre con VML (Vector Markup Language) [13] o SVG (Scalable Vector Graphics) [14].

La segunda fase supone la configuración del software para que pueda consultar los recursos descritos, indicarlos y, por fin, permitir su recuperación. Como ya se ha comentado anteriormente, LDAD se puede utilizar como si fuera una base de datos común, ya que es capaz de trabajar con gran cantidad de datos resolviendo, a la vez y de forma rápida, numerosas peticiones simultáneas. Además, la visualización de los resultados se puede realizar directamente sobre el cliente LDAP o bien, utilizando un gateway LDAP, a través de una página web.

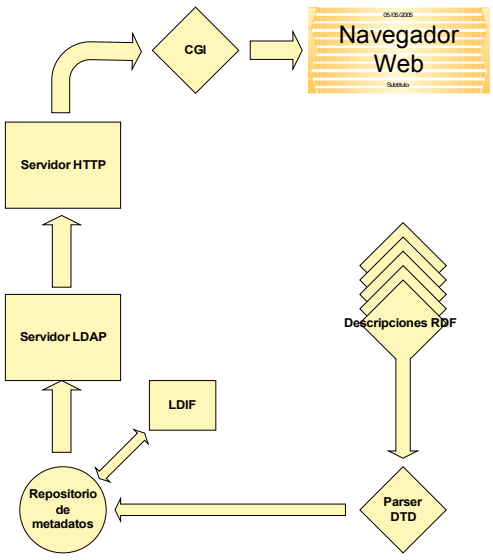
El primer paso dentro de la configuración del programa se centra en definir un objeto que tendrá como propiedades (atributos) cada una de las etiquetas definidas en la fase anterior. Para ello nos valimos de las bases establecidas por el trabajo de Hamilton [15].

Después, y teniendo en cuenta que lo se pretende es que el sistema sea los más automatizado posible, lo ideal es crear una herramienta que transforme los ficheros RDF resultantes de la descripción a ficheros LDIF (LDAP Data Interchange Format), que son con los que trabaja LDAP. Este programa se puede realizar en cualquier lenguaje de programación (PERL, Python...) y alimenta a LDAP de los recursos necesarios para realizar la consulta.

El último paso se centrará en hacer accesible toda la información almacenada de manera amigable para el usuario. Para ello existen diversas librerías gratuitas basadas en DSML (Directory Services Markup Language) [16] que permiten acceder a LDAP y publicar los datos allí contenidos vía web.

En definitiva, el esquema general sería el siguiente:

Figura 1.



5. Conclusiones

Si bien es cierto que el modelo de trabajo propuesto se ha "enmascarado" dentro de la estructura generada para un proyecto concreto, no se puede negar que también se ha plasmado un esqueleto claro de cómo se debe trabajar con metadatos (en concreto RDF) para generar un sistema de información que aglutine recursos web. A modo de recordatorio, mencionar los pasos que han dado pie a dicho sistema de información: a) diseño del sistema de trabajo; b) selección, justificación e implementación del conjunto de metadatos a utilizar; c) especificación del número de etiquetas, sintaxis y contenido que deben albergar; d) elección o no de una DTD, posterior justificación y diseño; e) designación de las herramientas para la edición, trabajo y gestión de ficheros en el formato de metadatos seleccionado y f) elección del motor de búsqueda e indexación y posterior configuración.

Agradecimientos

Los autores desean expresar su agradecimiento al equipo técnico de RedIRIS que se ha encargado de la implementación del modelo de trabajo aquí expuesto: Carlos Fuentes, Diego R. López, José Manuel Macías, Javier Masa y Jesús Sanz de las Heras.

También agradecemos el trabajo de las personas que durante estos años se han encargado de la alimentación y mantenimiento de la base de datos sobre la que han ido operando los sucesivos prototipos del sistema SARAC: Jesús Domínguez, María J. Rodríguez Gálvez y Miguel A. Cabello.

Referencias

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

[2] Hjelm, Johan. Creating the semantic web in RDF. New York: Wiley Computer Publishing; John Wiley & sons, Inc., 2001.

[3] Berners-Lee, T., Hendler, J. and Lassila, O. The semantic web. Scientific American, 284(5):34-43, May 2001.

[4] Brickley, Dan y Guha, R. V. Resource Description Framework (RDF) schema specification. [Página Web] 27 marzo 2000. http://www.w3.org/TR/PR-rdf-schema

[5] http://harvest.sourceforge.net/

[6] http://www.ietf.org/rfc/rfc1777.txt?number=1777

[7] Black, Uyless. Redes de ordenadores: protocolos, normas e interfaces. Madrid: Prentice-Hall, 1993. ISBN: 84-7897-151-3.

[8] http://es.dublincore.org/

[9] Maler, Eve y El Andaloussi, Jeanne. Developing SGML DTDs: from text to model to markup. New Jersey: Prentice Hall, 1996. ISBN: 0-13-309881-8.

[10] http://www.w3.org/2004/CDF/

[11] http://www.w3.org/Math/

[12] http://www.w3.org/MarkUp/

[13] http://www.w3.org/TR/1998/NOTE-VML-19980513

[14] http://www.w3.org/Graphics/SVG/

[15] Hamilton, Martin et al. Representing the Dublin Core within X.500 and LDAP. septiembre 2001. http://runner.ascs.muni.cz/DC_in_LDAP.txt

[16] http://www.dsml.org/