GeoCrawler y gvSIG: un Tándem para la Generación Automática de Metadatos

Arturo Beltran Fonollosa, Joaquín Huerta, Laura Díaz, Carlos Granell

Resumen

Hoy en día la información juega un papel fundamental en la sociedad donde vivimos. Se desea que la información esté disponible a nivel global y llegue fácilmente al mayor número de personas posible en un entorno colaborativo. Para ello resulta esencial organizar, publicitar y facilitar el acceso a dicha información. En ese artículo, se propone una metodología para la generación automática de metadatos mediante la cual podremos describir los recursos, el primer paso hacia nuestro propósito. En la generación automática de metadatos gvSIG puede jugar un papel muy importante al brindarnos la posibilidad de recopilar información durante el proceso de creación de los datos. Finalmente, para poner en práctica la metodología propuesta se presenta una primera versión de GeoCrawler, una aplicación de generación y publicación masiva de metadatos.

Palabras clave: GeoCrawler, gvSIG, metadatos, generación, publicación

Centro de Visualización Interactiva, Departamento de Lenguajes y Sistemas Informáticos. Universitat Jaume I de Castelló, Campus Riu Sec, 12071 Castelló de la Plana, {arturo.beltran, huerta, laura.diaz, carlos.granell}@uji.es

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el proyecto "CENIT España Virtual", subvencionado por el CDTI en el programa "Ingenio 2010" a través del Centro Nacional de Información Geográfica (CNIG), por la Generalitat Valenciana mediante su programa de "Beques per a estades en centres d'investigació fora de la Comunitat Valenciana" y por la Universitat Jaume I y la Fundacion Bancaja mediante el programa de ayudas predoctorales para la formación de personal investigador.

1. Introducción

Hoy en día la información juega un papel fundamental en la sociedad donde vivimos, llegando incluso al punto de la dependencia. Esto ha motivado, y ha sido motivado por, la era digital en la que nos encontramos inmersos. La cantidad de sistemas de información que manejamos actualmente es incontable: Bibliotecas Digitales, SIG/IDEs, directorios y buscadores de internet, etc. Todos ellos motivados por el deseo de que la información esté disponible a nivel global y llegue fácilmente al mayor número de personas posible en un entorno colaborativo. Para ello resulta esencial organizar, publicitar y facilitar el acceso a dicha información.

Es en este contexto en el que los metadatos cobran sentido y resultan ser de gran importancia, pues para que un recurso sea encontrado como resultado de una búsqueda debemos ser capaces de describirlo según sus propiedades. Por lo tanto, los metadatos juegan un papel fundamental en cualquier sistema de información que podamos imaginar, permitiéndonos indexar o catalogar los recursos en base a la descripción de sus características (tipo de dato, contenido, origen, calidad, fecha de creación, etc.) y a su contexto, para posteriormente poder ser encontrados. El problema reside en que la tarea de generación de metadatos resulta tediosa y poco gratificante, siendo necesario dedicar gran cantidad de tiempo y recursos tanto económicos como humanos. Por ello se considera necesario investigar técnicas y metodologías que permitan generar la mayoría de estos metadatos de forma automática. Con el fin de abordar este problema, se propone una metodología para la generación automática de metadatos que nos permitirá describir y documentar los recursos. Uno de nuestros objetivos es desarrollar un motor de generación automática de metadatos basado en dicha metodología y aplicarlo de forma práctica varias aplicaciones.

Por otra parte, analizando el contexto actual nos damos cuenta de que es común encontrar en nuestras máquinas o en servidores multitud de datos que no son accesibles al público. Concretamente y centrándonos en el campo de los Sistemas de Información Geográfica (SIG), existe todavía la necesidad de compartir conjuntos de datos almacenados en máquinas y totalmente ocultos al mundo, anulando así la posibilidad de compartirlos y reutilizarlos. El problema aparece cuando se desea hacer esos datos accesibles y fácilmente descubribles, ya que, debido al arduo proceso de creación de metadatos, normalmente encontraremos los datos sin documentar, lo que en entornos habituales de SIG hace difícil su búsqueda. En este contexto de las Infraestructuras de Datos Espaciales (IDE), los metadatos son el mecanismo necesario para describir y documentar datos y junto con los Servicios de Catálogo con los elementos clave para el descubrimiento de información (Nogueras et al., 2005) (Díaz et al., 2007) (Craglia et al., 2007). De este modo, en los últimos años, directivas como INSPIRE[1] exigen a las organizaciones gubernamentales la documentación, publicación y mantenimiento de metadatos y sus servicios de descubrimiento relacionados (Craglia et al., 2007), por lo que ya no es solo deseable, sino requerido. Así nos encontramos en un escenario de múltiples datos sin documentar ni catalogar y la imposición de publicarlos. Todo un reto que deberemos afrontar con la mayor eficiencia posible.

Basándonos en el contexto actual, y con la idea de aplicar de forma práctica la metodología de generación automática de metadatos que se propone para corroborar su

[1] http://inspire.jrc.ec.europa.eu

validez, planteamos una aplicación que llamamos GeoCrawler. GeoCrawler será una aplicación de generación y publicación masiva de metadatos, es decir, nos permitirá recopilar, describir, catalogar y publicar los recursos disponibles en la máquina en la que se ejecuta. De este modo, de forma integrada, intentará dar solución a la necesidad actual de documentar y publicitar nuestros recursos locales de una forma fácil y transparente al usuario, para permitir el acceso global de forma ordenada y eficiente.

Tal y como se plantea la metodología de generación de metadatos, ésta incorpora diversas técnicas de generación de metadatos, que abarcan desde la extracción de metadatos de los propios datos a la inferencia de nuevos metadatos. En este primer prototipo se han probado las técnicas orientadas a la extracción, en concreto a la extracción de metadatos explícitos en los propios datos y su contenido. Además se está trabajando en la colaboración con gvSIG que nos brindará la posibilidad de recopilar información durante el proceso de creación de los datos. De este modo, podemos obtener información muy valiosa, por ejemplo el proceso de creación para poder replicar los resultados más adelante, costes asociados (computacional, temporal, económico, etc.), un histórico de modificaciones o el autor de los datos con exactitud. Consideramos que esta información es muy importante y raramente tenida en cuenta. Además, debemos destacar que es una fuente de información "volátil", pues solo dispondremos de ella en el momento en que los datos son creados y por ello debemos recolectarla y almacenarla en ese momento.

2. Antecedentes: Metodologías de generación de metadatos

Una vez hemos comprendido la importancia que tienen los metadatos para describir recursos de cualquier naturaleza, vamos a intentar averiguar de dónde y cómo podemos obtenerlos.

Debemos tener en cuenta que nos encontraremos ante recursos de muy variada condición: geodatos en cualquier formato, cualquier tipo de recurso multimedia (archivos de audio, video o imágenes, documentos de texto, etc.). Por otra parte, debemos considerar la posibilidad de que los recursos pueden ser creados en el instante en que estamos recopilando sus metadatos o pueden ser recursos existentes desde hace mucho tiempo. Por todo ello debemos buscar una solución lo más genérica posible, que pueda adaptarse a cualquiera de los escenarios resultante de la combinación de las condiciones que acabamos de comentar.

Ampliando las ideas reflejadas en (Beard, 1996), un taller sobre geodatos donde se repasan los posibles métodos para la compilación de metadatos, podemos destacar nueve metodologías o técnicas para la generación de metadatos:

1. Introducción manual por teclado
2. Extracción de metadatos del propio dato
3. Extracción de metadatos a partir del contenido
4. Recolección en el proceso de creación de los datos
5. Aprovechamiento del contexto
6. Búsqueda (look-up) desde una tabla de referencia
7. Medición del valor

8. Computación del metadato
9. Inferencia del metadato

El primer método lo conocemos bien, ya que es el método por defecto en la mayoría de los casos hoy en día: el usuario edita una ficha (XML) empleando un editor más o menos sofisticado de metadatos del tipo MetaD[2] o CatMDEdit[3] (FGDC, 2008). El problema es la cantidad de tiempo y recursos necesarios que supone este método, por ello resulta un método poco eficiente. Para entenderlo basta con imaginar una situación en la que debemos crear una ficha de cada uno de los libros de una biblioteca. Aparte de lo laborioso de la tarea podemos encontrarnos problemas como la falta de información de alguno de los ejemplares o cometer errores a la hora de copiar los datos a la ficha. Hasta ahora, nos hemos conformado con este primer método y no hemos intentado buscar soluciones más eficientes.

El segundo método resulta bastante obvio, los propios datos nos pueden proporcionar gran cantidad de información sobre ellos mismos si los analizamos bien. Por ejemplo, cualquier archivo de cualquier sistema operativo contendrá información como la fecha de creación y modificación o el tamaño que ocupa en disco.

El tercer método también resulta bastante obvio, el contenido de los datos es una fuente de información muy importante. Analizando los datos podemos encontrar información relevante de forma explícita, por ejemplo, en un correo electrónico podemos encontrar información como el remitente y el destinatario o la fecha de envío.

El cuarto método es la recolección de información durante el proceso de creación de los datos. Debemos destacar que ésta es una fuente de información "volátil", pues solo dispondremos de ella en el momento en que los datos son creados y por ello debemos extraer y almacenar toda la información posible en ese preciso momento o la perderemos para siempre. Consideramos que la información que se puede sacar del proceso de creación de los datos es muy importante y raramente tenida en cuenta. Mediante este método, podemos obtener información muy valiosa como el proceso de creación para poder replicar los resultados más adelante, costes asociados (computacional, temporal, económico, etc.), un histórico de modificaciones o el autor de los datos con exactitud.

El quinto método trata de aprovechar la información del contexto en que los datos son creados o explotados. Del contexto de creación podemos extraer información como la organización o empresa responsable de los datos y la temática de los mismos, ya que los datos que genera una empresa dedicada al análisis del estado de la bolsa probablemente serán de carácter económico. Podemos operar de forma similar con el contexto de explotación obteniendo información como la temática o la calidad que suelen tener los recursos que ofrece cierta empresa. Resumiendo, podemos deducir algunos metadatos que nos faltan sobre un recurso, basándonos en el conjunto de recursos que se encuentran en el mismo contexto.

El sexto método supone que un elemento de metadatos se crea a través de una correspondencia con otro, como en el caso de la derivación de la caja envolvente (4

[2] http://www.geoportal-idec.net/geoportal/cas/metad.jsp
[3] http://catmdedit.sourceforge.net/

coordenadas del BBOX) en un topónimo, a través de un servicio de nomenclátor (*gazetteer*[4]).

El séptimo método supone que, en el proceso de creación de los datos, un sensor u otro mecanismo nos proporcione información relevante. En el mismo momento en el que se están creando los datos se podrían realizar mediciones por ejemplo de elevación o temperatura, y colocar ese valor en la ficha de metadatos de forma automática. De este modo, de forma automática, podemos añadir información adicional y de alto valor a los datos que estamos creando.

El octavo método se centra en el cálculo de un elemento de metadatos empleando los geodatos en sí. En este sentido hay muchas líneas de investigación abiertas que abarcan un amplio abanico de posibilidades. Podemos encontrar desde diferentes técnicas para realizar un análisis/procesado de documentos de texto o páginas web para averiguar su tema principal, a otras técnicas que emplean los propios geodatos para, por ejemplo, determinar la provincia de un pueblo por cálculos topológicos.

El noveno y último método es la inferencia de metadatos a partir de otros metadatos o de los geodatos (Margaritopoulos, 2008). Según Beard supone el mejor método -de hecho en algunas situaciones el único- para la creación de metadatos *post hoc*, es decir, documentando geodatos ya existentes. Un ejemplo sería inferir la época de los geodatos por el metadato temperatura, a lo mejor recogido por el séptimo método que hemos explicado, de manera que una regla establecería que para temperatura inferior a 15 grados en Tenerife supongamos invierno. Beard también señaliza algo que debe ser obvio hoy en día, de hecho está captado en (Goodchild, 2007): que la creación de estos metadatos inferidos solapa ampliamente a los campos de investigación de la minería de datos y de la recuperación de datos.

Como podemos observar, todos estos métodos nos proporcionarán información importante sobre los recursos que hoy en día estamos dejando escapar. Mientras la mayoría de los métodos son aplicables durante todo el ciclo de vida de los datos, otros métodos sólo serán aplicables en el momento en que los datos son creados. Debemos hacer una mención especial de ellos dado que la mayoría de veces la información que no recogemos en ese momento se pierde para siempre, y alguna de esta información puede resultar esencial para conseguir una correcta descripción de los recursos.

3. Metodología propuesta para la generación de metadatos

La metodología que se propone es una combinación de todos los métodos descritos anteriormente orquestados de forma eficiente. Empezaremos por obtener toda la información relevante que podamos obtener del propio dato y su contenido de acuerdo al segundo y tercer método. A esto le sumaremos la información común perteneciente al contexto (método 5), que previamente habrá sido configurada o revisada por el usuario para todos los datos pertenecientes a dicho contexto. En este momento deberemos considerar la posibilidad de recopilar información del proceso de creación de los datos (si existe) y ver si es posible realizar alguna medición que nos proporcione información

[4] http://en.wikipedia.org/wiki/Gazetteer

relevante (métodos 4 y 7). Llegados a este momento, dispondremos ya de una base de información, y es en base a ella que aplicando el resto de métodos deductivos (6, 8 y 9) podremos ampliarla. Finalmente, nunca deberemos olvidar el ofrecer al usuario la posibilidad de introducir o modificar la información, aunque la idea es que éste gane confianza en la metodología en base a la observación de resultados aceptables y acabe por no participar en el proceso de generación de metadatos.

Esta metodología, nos permitirá mejorar progresivamente la generación automática de metadatos y la calidad resultante de éstos tal y como se vayan aplicando y mejorando los diferentes métodos que la componen. Además, la metodología propuesta tiene en cuenta e intenta recopilar información que actualmente pasa desapercibida y no por ello deja de ser importante, como es la que proviene del proceso de creación. En consecuencia, el resultado de aplicar esta metodología será obtener más metadatos, de mejor calidad y corrección, más completos y con reducida participación por parte del usuario. Con lo que estamos atacando directamente el mayor problema que tenemos en la generación de metadatos: lo tediosa y costosa que resulta la tarea actualmente.

4. GeoCrawler

4.1. Descripción

Como hemos comentado anteriormente, en el contexto actual, encontramos en nuestras máquinas o en servidores multitud de datos que no son accesibles al público. El problema aparece cuando se desean hacer esos datos accesibles y fácilmente descubribles, dado que normalmente encontraremos los datos sin documentar. Así nos encontramos en un escenario con millones de datos sin documentar ni catalogar y la imposición (por directivas como INSPIRE) de publicarlos, todo un reto que deberemos afrontar con la mayor eficiencia posible.

La aproximación más obvia al problema es poner cierto número de operarios en cada una de las organizaciones y que estos vayan describiendo y documentando los datos de forma manual. Posteriormente se generará un catalogo en base a las descripciones de los datos y se pondrá a disposición del público. No creemos que ésta sea una solución viable por la cantidad de tiempo y recursos que la tarea exige. Además con gran probabilidad los metadatos generados no serán de gran calidad. Como mínimo serán incompletos por falta de información que se ha perdido desde su creación, y probablemente serán inexactos por información introducida de forma errónea y precipitada.

En este contexto nace la idea de GeoCrawler, una aplicación que intentará dar solución a la necesidad actual de documentar y permitir el acceso global a nuestros recursos locales de una forma fácil, ordenada y eficiente. Tras la investigación que se ha realizado centrada en idear una metodología adecuada para la generación automática de metadatos de recursos de cualquier naturaleza, se pretende aplicar de forma práctica para corroborar su validez. Para ello, y pensando en las necesidades actuales, se ha pensado en crear una aplicación que nos permita recopilar, describir, catalogar y en un futuro publicar los recursos disponibles en la máquina en la que se ejecuta. GeoCrawler

ofrece una solución integrada que nos permitirá documentar y compartir esos recursos de una forma lo más transparente posible para el usuario.

4.2. Arquitectura general

En esta sección presentamos una primera aproximación de la arquitectura de GeoCrawler. Esta arquitectura ha sido diseñada para cubrir los requerimientos funcionales descritos en secciones anteriores y para implementar y cumplir los requisitos de la metodología de generación automática de metadatos propuesta. Por todo ello, hemos diseñado una arquitectura de tres capas (Eckerson & Wayne, 1995). Esta arquitectura es de tipo cliente-servidor en la cual la interfaz de usuario, la lógica del proceso funcional (reglas de negocio), el almacenamiento de los datos y el acceso a los mismos son desarrollados y mantenidos como módulos independientes, y a menudo en plataformas diferentes.

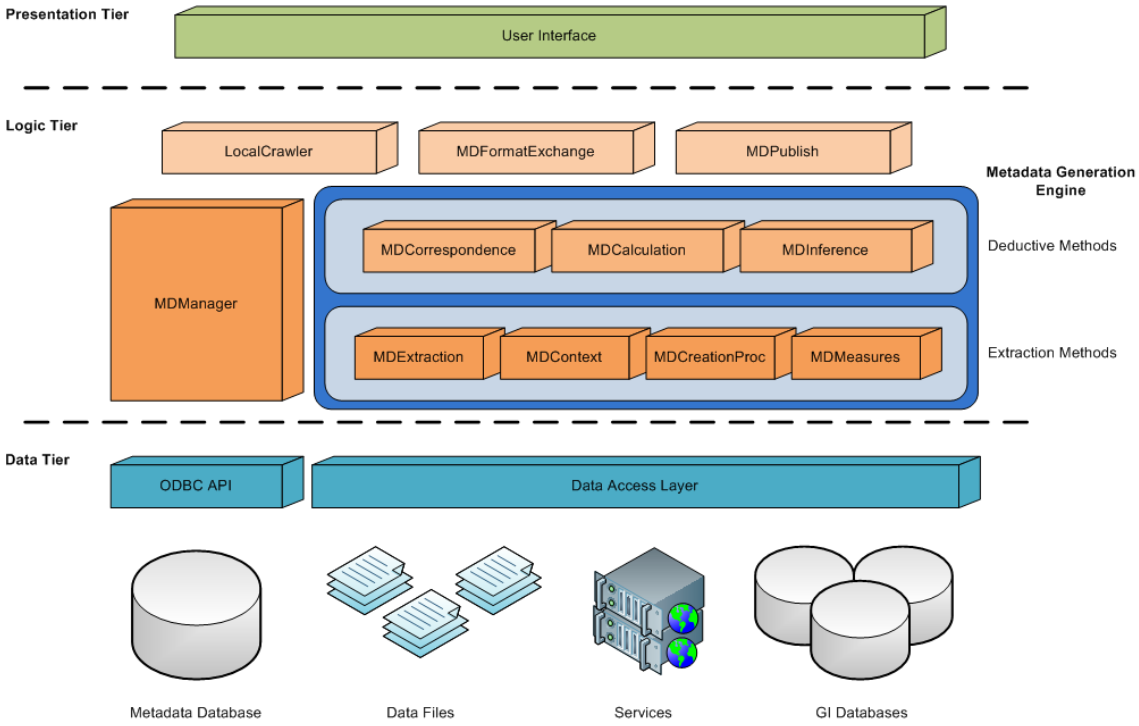

Figura 1: Arquitectura General de GeoCrawler

Como podemos ver en la Figura 1, en la parte inferior de la figura y el nivel más bajo de la aplicación encontramos la capa de datos (*Data Tier*), esta capa incluye una base de datos para almacenar los metadatos generados y los mecanismos de acceso a los propios recursos a tratar, incluyendo ficheros, servicios y bases de datos. Respecto al acceso a los diferentes tipos de recursos, es deseable tener una plataforma que permita acceder a cualquier tipo de recurso de una forma homogénea, esto estará incluido en el componente *Data Access Layer*. La principal función de este componente será ofrecer a la aplicación una interfaz bien definida que nos permita acceder a la mayor cantidad de información posible acerca del contenido de los recursos, sin tener que preocuparnos por su tipo y/o que *driver* de los que incluye debemos usar para obtener dicha información. Por otra parte, esta capa incluye también una base de datos que nos

permitirá gestionar los metadatos generados por la aplicación, en este caso, el acceso lo proporcionará la interfaz estándar ODBC[5]. Este tipo de diseño mantiene los datos neutrales e independientes de la lógica de negocio, mejorando la escalabilidad y el rendimiento.

La siguiente capa, que encontramos justo encima de la capa de datos, es la capa lógica (*Logic Tier*) que contiene las reglas de negocio. Es la encargada de controlar la funcionalidad de la aplicación realizando un procesamiento detallado.

En la parte más baja de esta capa podemos encontrar la pieza clave de la aplicación: el motor de generación de metadatos (*Metadata Genaration Engine*). El MGE incluye las implementaciones de los diferentes métodos de generación de metadatos propuestos, tanto los métodos orientados a la extracción como los métodos deductivos. Debemos destacar que este componente se implementará de una forma lo más desacoplada posible a la aplicación, con la idea de que pueda ser aprovechado en cualquier aplicación que requiera generación automática de metadatos.

Acompañando al MGE que acabamos de describir podemos encontrar el gestor de metadatos (*MDManager*) cuya funcionalidad es orquestar la generación de metadatos de forma eficiente, gestionar los metadatos generados y proporcionarlos al resto de componentes.

En la parte más alta de esta capa encontramos el módulo *LocalCrawler* que implementará la funcionalidad de un *crawler* permitiéndonos generar una lista de los recursos locales disponibles tras explorar los directorios configurados. Al mismo nivel, tenemos el módulo *MDFormatExchange*, responsable de importar y exportar metadatos en formatos estándar y manejar la transformación entre ellos. Finalmente, vemos el módulo *MDPublish* que usando los metadatos generados, normalmente en un formato estándar, implementará la funcionalidad de publicación. Este módulo de publicación incorporará varios medios de publicación, de una forma integrada y automática.

En la parte superior de la figura, representando el nivel más alto de la aplicación, encontramos la capa de presentación (*Presentation Tier*). Esta capa muestra la información proporcionada por las capas inferiores a través de una interfaz de usuario gráfica. Esta interfaz de usuario, además, permitirá a los usuarios interactuar, configurar y manejar la aplicación.

Para terminar, debemos destacar que debido al diseño modular, si se considera interesante incorporar nueva funcionalidad a la aplicación ya sea a nivel general o de generación de metadatos se podrán agregar con facilidad nuevos módulos que la implementen. Por otra parte, este tipo de arquitectura, nos permite actualizar o cambiar cualquier módulo o capa de forma independiente, sin que el resto de componentes de la aplicación se vean afectados. Esto resultará muy útil si queremos reutilizar algunos componentes en otras aplicaciones. Por ejemplo podemos integrar el MGE en cualquier aplicación que desee incorporar la funcionalidad de generación automática de metadatos.

[5] http://en.wikipedia.org/wiki/Open_Database_Connectivity

5. Resultados: Una primera versión de GeoCrawler

Ahora que ya tenemos claro qué queremos hacer y una arquitectura diseñada con ese fin, veamos una primera aproximación a la solución. Esta primera versión de GeoCrawler tiene una funcionalidad muy limitada: por el momento sólo se ha trabajado en tener una aplicación residente básica, en tareas de *crawling* y en la extracción de metadatos de los propios datos y su contenido para ficheros de datos vectoriales. A continuación podemos ver algunas capturas de pantalla de la aplicación:

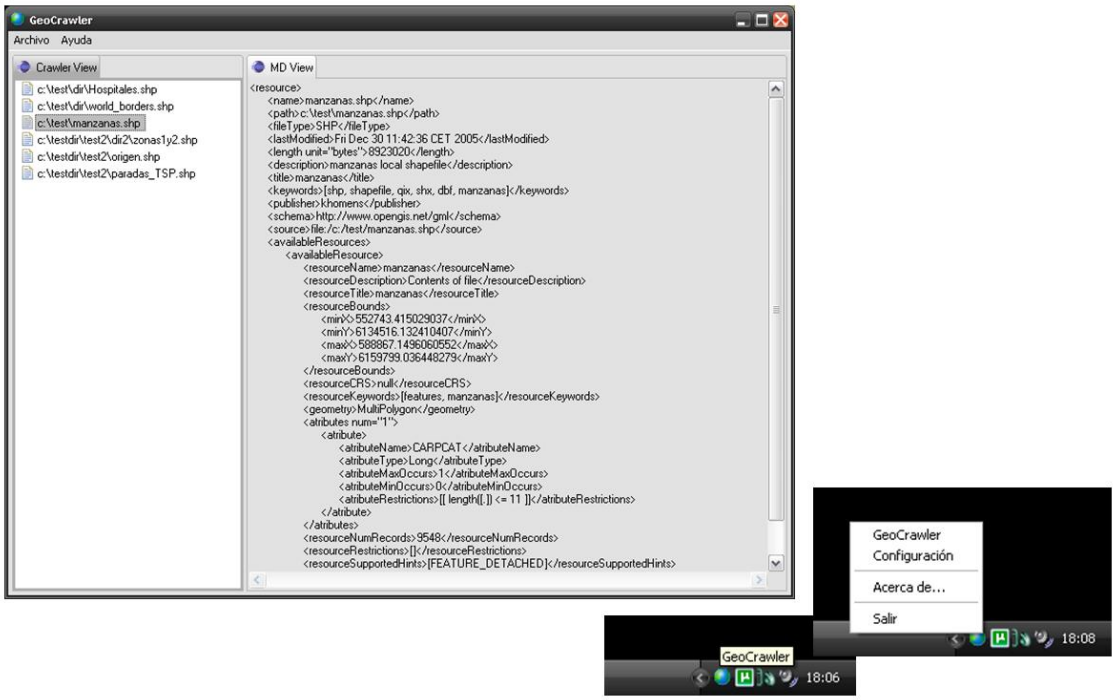

Figura 2: Una primera versión de GeoCrawler

En el ámbito de la máquina local, un *crawler* es un programa que inspecciona el contenido de uno o varios directorios de una forma metódica y automatizada. La idea es explorar el contenido de los directorios, que el usuario ha configurado, de la máquina local en la que se ejecuta la aplicación y crear un listado de los recursos disponibles susceptibles de ser georreferenciados y, en general, de todos los recursos cuyo tipo sea soportado. En base a este listado GeoCrawler, posteriormente, podrá generar de forma automática toda la información (metadatos) de cada uno de los recursos, utilizando para ello la implementación de la metodología de generación automática de metadatos propuesta. Y en base a estas descripciones, finalmente, publicar los recursos.

La parte más interesante de esta primera versión, y que supone el inicio de la puesta en práctica de la metodología de generación automática de metadatos propuesta, es un primer desarrollo de las técnicas orientadas a la extracción, en concreto a la extracción de metadatos explícitos en los propios datos y su contenido, usando para ello las librerías del proyecto *GeoTools*[6].

[6] http://www.geotools.org/

El resultado obtenido por este módulo (*MDExtraction*), ha sido el conjunto de información que hemos conseguido extraer de los recursos mediante *GeoTools* y el sistema operativo. Para comprender cuál es la información que hemos conseguido extraer, podemos ver el significado de cada etiqueta en la Tabla 1. Por cuestiones de claridad y posterior procesado de la información, provisionalmente, el resultado de este módulo se visualiza en formato XML[7]. En la Figura 3 podemos ver un ejemplo del XML generado para un recurso de tipo *shapefile*[8] que contiene todos los metadatos que hemos conseguido extraer tanto del propio fichero como de su contenido.

Etiqueta	Significado
resource	Inicia la descripción de un recurso
name	Nombre del recurso
path	Ruta absoluta al fichero del recurso
fileType	Formato del fichero del recurso
lastModified	Fecha de la última modificación del recurso
length	Tamaño del recurso
description	Descripción del recurso
title	Título del recurso
keywords	Lista de palabras clave relacionadas con el recurso
publisher	Persona responsable del recurso
schema	URI al esquema del tipo del recurso
source	Fuente del recurso
availableResources	Conjunto de recursos disponibles dentro del recurso
availableResource	Inicia la descripción de un recurso interno
resourceName	Nombre del recurso interno
resourceDescription	Descripción del recurso interno
resourceTitle	Título del recurso interno
resourceBounds	Caja envolvente del recurso interno
minX	Mínima longitud de la caja envolvente
minY	Mínima latitud de la caja envolvente
maxX	Máxima longitud de la caja envolvente
maxY	Máxima latitud de la caja envolvente
resourceCRS	Sistema de referencia del recurso interno
resourceKeywords	Lista de palabras clave relacionadas con el recurso interno
geometry	Tipo de geometría del recurso interno
atributes	Conjunto de atributos del recurso interno
atribute	Inicia la descripción del atributo
atributeName	Nombre del atributo
atributeType	Tipo de datos del atributo
atributeMaxOccurs	Número máximo de ocurrencias del atributo
atributeMinOccurs	Número mínimo de ocurrencias del atributo
atributeRestrictions	Lista de restricciones del atributo
resourceNumRecords	Número de registros del recurso interno
resourceRestrictions	Lista de restricciones del recurso interno
resourceSupportedHints	Lista de pistas soportadas por el recurso interno

Tabla 1: Significado de las etiquetas de los metadatos extraídos

[7] http://www.w3.org/XML/
[8] http://es.wikipedia.org/wiki/Shapefile

```
<resource>
     <name>world_borders.shp</name>
     <path>D:/ExampleData/datos_vectorial/world_borders.shp</path>
     <fileType>SHP</fileType>
     <lastModified>Thu Dec 23 07:16:48 CET 2004</lastModified>
     <length unit="bytes">6664344</length>
     <description>world_borders local shapefile</description>
     <title>world_borders</title>
     <keywords>[shp, shapefile, qix, shx, dbf, world_borders]</keywords>
     <publisher>usuario</publisher>
     <schema>http://www.opengis.net/gml</schema>
     <source>file:/D:/ExampleData/datos_vectorial/world_borders.shp</source>
     <availableResources>
          <availableResource>
               <resourceName>world_borders</resourceName>
               <resourceDescription>Contents of file</resourceDescription>
               <resourceTitle>world_borders</resourceTitle>
               <resourceBounds>
                    <minX>-180.0</minX>
                    <minY>-90.0</minY>
                    <maxX>180.0</maxX>
                    <maxY>83.623596</maxY>
               </resourceBounds>
               <resourceCRS>null</resourceCRS>
               <resourceKeywords>[features, world_borders]</resourceKeywords>
               <geometry>MultiPolygon</geometry>
               <atributes num="4">
                    <atribute>
                         <atributeName>CAT</atributeName>
                         <atributeType>Long</atributeType>
                         <atributeMaxOccurs>1</atributeMaxOccurs>
                         <atributeMinOccurs>0</atributeMinOccurs>
                         <atributeRestrictions>[[ length([.]) <= 16 ]]</atributeRestrictions>
                    </atribute>
                    <atribute>
                         <atributeName>CNTRY_NAME</atributeName>
                         <atributeType>String</atributeType>
                         <atributeMaxOccurs>1</atributeMaxOccurs>
                         <atributeMinOccurs>0</atributeMinOccurs>
                         <atributeRestrictions>[[ length([.]) <= 80 ]]</atributeRestrictions>
                    </atribute>
                    <atribute>
                         <atributeName>AREA</atributeName>
                         <atributeType>Double</atributeType>
                         <atributeMaxOccurs>1</atributeMaxOccurs>
                         <atributeMinOccurs>0</atributeMinOccurs>
                         <atributeRestrictions>[[ length([.]) <= 15 ]]</atributeRestrictions>
                    </atribute>
                    <atribute>
                         <atributeName>POP_CNTRY</atributeName>
                         <atributeType>Double</atributeType>
                         <atributeMaxOccurs>1</atributeMaxOccurs>
                         <atributeMinOccurs>0</atributeMinOccurs>
                         <atributeRestrictions>[[ length([.]) <= 15 ]]</atributeRestrictions>
                    </atribute>
               </atributes>
               <resourceNumRecords>3784</resourceNumRecords>
               <resourceRestrictions>[]</resourceRestrictions>
               <resourceSupportedHints>[FEATURE_DETACHED]</resourceSupportedHints>
          </availableResource>
     </availableResources>
</resource>
```

Figura 3: Ejemplo de XML generado por MDExtraction

6. Trabajo Futuro

Como hemos comentado anteriormente, el desarrollo presentado en este artículo sólo es una mínima primera versión de la aplicación que planteamos: GeoCrawler. Sin embargo, sí se han presentado una completa metodología para la generación automática de metadatos y una arquitectura general para la aplicación GeoCrawler que contempla la implementación de esta metodología dentro de un motor de generación de metadatos (MGE). De forma clara, el trabajo futuro debe centrarse en completar la implementación

de la aplicación, prestando especial interés y centrando el esfuerzo en el MGE y en la posterior publicación de los recursos mediante diferentes técnicas.

Como en la mayoría de las cosas debemos empezar trabajando en la base, y la base de esta aplicación es el acceso a los recursos. En la primera versión que hemos presentado en este artículo el acceso a los recursos se realizaba mediante el uso de las librerías del proyecto *GeoTools*. Pero, al intentar ampliar el número de tipos de recursos con lo que trabajar, nos hemos dado cuenta de que *GeoTools* no nos ofrece lo que necesitamos: una plataforma que nos permita obtener información y acceder de forma lo más homogénea posible a recursos heterogéneos. Tras explorar diferentes opciones se decidió optar por la que nos ofrecía el proyecto gvSIG, esta consiste en reutilizar su capa de acceso a datos (DAL) en nuestro proyecto. Sin embargo, para que el DAL de gvSIG sea la plataforma de acceso a datos que todos deseamos habrá que trabajar en ella y habrá que ampliar su funcionalidad, especialmente en la parte relacionada con proporcionar información (metadatos) sobre el recurso. Ya se está trabajando en ello en colaboración con la gente del proyecto gvSIG.

Por otra parte, a corto plazo, se quiere continuar trabajando en el MGE, específicamente en la recolección de información durante el proceso de creación de los datos. Como ya hemos comentado anteriormente, debemos destacar que ésta es una fuente de información "volátil", pues solo dispondremos de ella en el momento en que los datos son creados y por ello es de especial importancia extraer y almacenar toda la información posible en ese preciso momento o la perderemos para siempre. Es en este punto donde gvSIG puede jugar un papel muy importante, pues es durante la creación de los datos cuando podemos obtener toda la información de ese proceso. Por lo tanto, se va a trabajar para conseguir que gvSIG anote información en los procesos de creación de datos para posteriormente poder recopilarla y almacenarla de forma pertinente.

7. Conclusiones

En este artículo se plantea la creación de una aplicación de generación y publicación masiva de metadatos, cuyo nombre es GeoCrawler. Se pretende que esta aplicación, de forma sinérgica, trate distintos tipos de recursos, diferentes métodos de generación de metadatos y diferentes estrategias de publicación. El objetivo de esta aplicación es recopilar, describir, catalogar y publicar todos los recursos disponibles en la máquina en la que se ejecuta. Para ello, en primer lugar se ha propuesto una metodología para la generación automática de metadatos, con la que se pretende poder describir de forma completa y veraz los recursos para posteriormente poder ser publicados. Por otra parte, se ha propuesto el diseño de una arquitectura general para GeoCrawler, permitiendo así empezar a desarrollar la funcionalidad de *crawler* e iniciar la puesta en práctica de la metodología de generación automática de metadatos propuesta.

Además, en este artículo se presenta una primera versión de GeoCrawler, en la que por el momento sólo se ha trabajado en tener una aplicación residente básica, en tareas de *crawling* y en la extracción de metadatos de los propios datos y su contenido para ficheros de datos vectoriales. Esta primera versión pese a tener una funcionalidad muy limitada nos ha permitido extraer algunas conclusiones. En primer lugar podemos concluir que la arquitectura modular diseñada para GeoCrawler cubre todas nuestras

necesidades iniciales y que es altamente adaptable y escalable. En segundo lugar, se ha empezado a desarrollar la funcionalidad de *crawler* obteniendo buenos resultados. Tras explorar de forma metódica y automatizada los directorios configurados por el usuario somos capaces de generar un listado de rutas absolutas correspondientes a los recursos encontrados. En base a este listado GeoCrawler, posteriormente, podrá generar de forma automática toda la información (metadatos) de cada uno de los recursos, utilizando para ello la implementación de la metodología de generación automática de metadatos propuesta (MGE). Finalmente, se ha iniciado la puesta en práctica de la metodología de generación automática de metadatos propuesta desarrollando las técnicas orientadas a la extracción, en concreto a la extracción de metadatos explícitos en los propios datos y su contenido, usando para ello las librerías del proyecto *GeoTools* y la información que nos brinda el sistema operativo. Los resultados obtenidos se consideran bastante positivos, dado que hemos logrado extraer gran cantidad de información de recursos de naturaleza vectorial. Sin embargo, al intentar ampliar el rango de formatos de recursos soportados nos hemos dado cuenta de que necesitamos algo más de lo que *GeoTools* nos ofrece. Con el fin de tener una plataforma que nos permita obtener información y acceder de forma lo más homogénea posible a recursos heterogéneos hemos empezado a trabajar en la ampliación de funcionalidad del DAL de gvSIG.

Uno de los métodos que forman parte de la metodología de generación de metadatos propuesta consiste en recopilar información durante el proceso de creación de los datos. Es en este punto donde gvSIG puede jugar un papel muy importante, proporcionando información muy valiosa, por ejemplo el proceso de creación para poder replicar los resultados más adelante, costes asociados (computacional, temporal, económico, etc.), un histórico de modificaciones o el autor de los datos con exactitud. Consideramos que esta información es muy importante y raramente tenida en cuenta. Además, debemos destacar que es una fuente de información "volátil", pues solo dispondremos de ella en el momento en que los datos son creados y por ello debemos recolectarla y almacenarla en ese momento.

Referencias Bibliográficas

Badawia, M. (2008). *Automatic metadata generation applications: a survey study*. En International Journal of Metadata, Semantics and Ontologies, Vol. 3, No.4, 2008.

Beard, K. (1996). *A Structure for Organizing Metadata Collection*. En Third International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, New Mexico.

Craglia, M., Kanellopoulos, I., Smits, P. (2007). *Metadata: where we are now, and where we should be going*. En 10th AGILE International Conference on Geographic Information Science 2007.

Díaz, L., Martín, C., Gould, M., Granell, C., Manso, M.A. (2007). *Semi-automatic metadata extraction from imagery and cartographic data*. In International Geoscience and Remote Sensing Symposium (IGARSS 2007), pages 3051–3052. IEEE CS Press.

Eckerson, Wayne, W. (1995). *Three tier client/server architecture: Achieving scalability, performance, and efficiency in client server applications*. Open Information Systems 10, 20(2), 1995.

FGDC Metadata Working Group (2008). *ISO Metadata Editor Review*. http://www.fgdc.gov/metadata/iso-metadata-editor-review. Fecha consulta: 30/11/2009.

Goodchild, M. (2007). *Citizens as voluntary sensors: Spatial data infrastructure in the world of Web 2.0*. International Journal of Spatial Data Infrastructures Research, 2:24–32, 2007.

Greenberg, J., Spurgin, K., Crystal, A. (2006). *Functionalities for automatic-metadata generation applications: a survey of metadata experts opinions*. En International Journal of Metadata, Semantics and Ontologies, Vol. 1, No. 1, pp. 3-20.

Liu, J. (2007). *Metadata and its Applications in the Digital Library: Approaches and Practices*. Libraries Unlimited, London, pp.143-149.

Margaritopoulos M., Margaritopoulos, T., Kotini, I., Manitsaris, A. (2008). *Automatic metadata generation by utilising pre-existing metadata of related resources*. En International Journal of Metadata, Semantics and Ontologies, Vol. 3, No.4, 2008.

Nogueras, J., Zarazaga, F. J., Béjar, R., Álvarez, P.J., Muro, P.R. (2005). *OGC catalog services: a key element for the development of spatial data infrastructures*. Computers and Geosciences, 31(2):199–209, 2005.