

CatalogConnector. Un cliente CSW para conectar catálogos de metadatos

Victor Pascual¹, Jordi Guimet¹, Wladimir Szczerban¹, Silvia Corcoll¹

⁽¹⁾ Servicio de Sistemas de Información Geográfica y Teledetección (SIGTE), Facultat de Lletres, Universitat de Girona, Plaça Ferrater i Mora, 1, 17071 Girona, correo@correo.es.

⁽²⁾ Departamento de Geografía, Historia e Historia del Arte, Facultat de Lletres, Universitat de Girona, Plaça Ferrater i Mora, 1, 17071 Girona, correo@correo.es.

RESUMEN

Este artículo refleja los problemas de interoperabilidad que existen entre las diferentes implementaciones de catálogos de metadatos que contemplan la especificación CSW [1] (Catalog Service for the Web) de OGC (Open GeoSpatial Consortium) . Esta situación ha llevado al desarrollo de una aplicación cliente para poder lanzar peticiones simultáneas a diferentes catálogos de metadatos, con la intención de poder visualizar los resultados de forma unitaria.

En el artículo se detalla tanto la arquitectura como todo proceso de desarrollo de la aplicación.

Palabras clave: Jornadas, SIG, software libre, Girona.

INTRODUCCIÓN

En los últimos años han proliferado los clientes interoperables, básicamente visores, web o de escritorio que gestionaban múltiples conexiones a diferentes servicios OGC, tales como Web Map Service (WMS) y en menor medida Web Feature Service (WFS).

Sin embargo, este fenómeno, no se ha producido en el campo de los catálogos de metadatos. Pieza fundamental dentro de las Infraestructuras de Datos Espaciales (IDEs).

Entre las diferentes causas que han llevado a la falta de clientes dedicados la conexión de múltiples catálogos OGC, destacaríamos:

- La falta de entendimiento dentro del propio OGC para mantener y unificar las normas de implementación y de perfiles de la especificación OGC CSW.

Esto parece haberse solucionado con la versión 2.0.2, pero pasará algún tiempo hasta que los principales desarrolladores de catálogos de metadatos adopten esta versión.

- La ardua y poco gratificante tarea de crear metadatos
- La complejidad intrínseca que lleva el mantenimiento de un catálogo de metadatos
- Hasta hace muy poco, la poca disponibilidad de programas de código abierto.

Afortunadamente, el escenario futuro tiende a cambiar, gracias en parte a la nueva versión del OGC CSW 2.0.2 y a software libre como GeoNetwork o Deegree.

Se prevé, a corto plazo, la aparición de nuevos nodos integrantes de las IDE que ya empezarán a ofrecer, a demás de servicios WMS y WFS, también servicios CSW.

Todo esto nos ha llevado a desarrollar un cliente que permite conectar catálogos OGC CSW (CatalogConnector), con la misión de lanzar peticiones simultáneas a múltiples catálogos con diferentes implementaciones y perfiles y mostrar todos los resultados en una única ventana.

En esta ponencia se presentará el programa de código abierto “CatalogConnector” desarrollado por la Infraestructura de Datos Espaciales de Cataluña (IDEC) [2] que actualmente estará en funcionamiento dentro del proyecto IdeUnivers [3].

PROYECTO IDEUNIVERS

IdeUnivers es un proyecto europeo dentro del Programa *Interreg IIIB MEDOCC*, el objetivo del cual es la realización de una Infraestructura de Datos Espaciales (IDE) sectorial en el ámbito universitario..

Liderado por la *Secretaria de Telecomunicacions i la Societat de la Informació* (STSI) en colaboración con el *Institut Cartogràfic de Catalunya* (ICC), participan también en el proyecto: la Comunidad Autónoma de Andalucía a través del *Instituto Cartográfico de Andalucía* (ICA), el Centro de Investigación italiano IREA de Milán, el servicio cartográfico de la región italiana Emilia Romagna, y la Universidad Aegean de Grecia.

Cada socio del proyecto tenía que:

- Coordinar los departamentos universitarios
- Crear un su propio geoportal “local”.
- Implementar un visualizador WMS y un catalogo de metadatos CSW

Sólo en Cataluña y Andalucía, actualmente ya hay más de 30 departamentos universitarios y /o centros de investigación, colaborando en el Proyecto IdeUnivers.

La participación de estas entidades tiene dos claros objetivos:

- Elaborar metadatos de la información geográfica disponible y publicarla en el catálogo de metadatos de su nodo regional.
- Publicar las capas cartográficas georeferenciadas en servidores WMS para su integración en la Infraestructura.

La fase final del proyecto, significaba la creación de un geoportal “global” que enlazara con los geoportales “locales”, la conexión entre los catálogos de metadatos y finalmente la creación de un catálogo “global”.

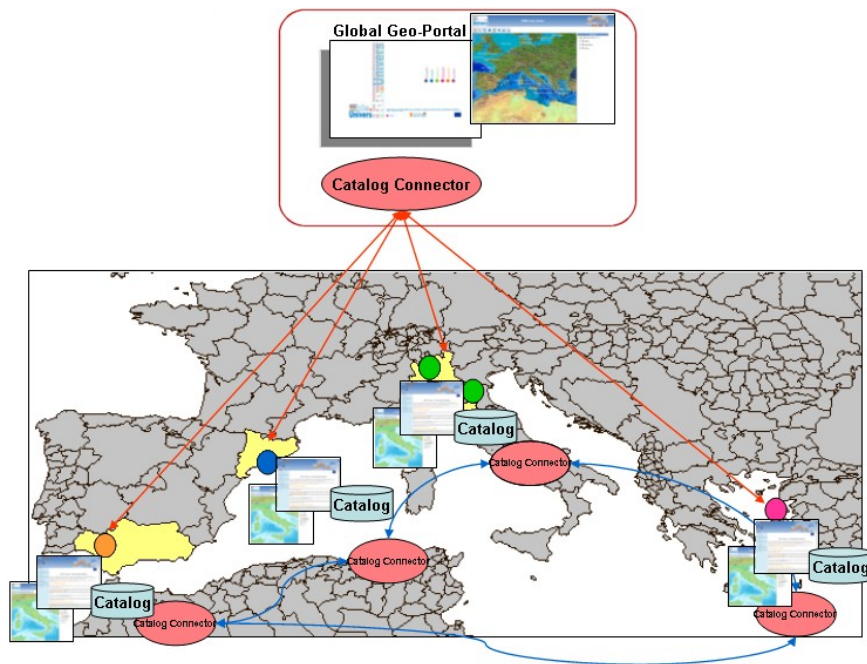


Figura 1: Nodos y geoportales de IdeUnivers

Para el desarrollo de este catálogo “global” se plantearon dos posibilidades:

- La implementación de un catálogo que mediante procesos de “harvesting” replicara los metadatos de los diferentes catálogos “locales”
- La creación de un cliente CSW que consultase de forma simultánea a los diferentes catálogos “locales”.

Finalmente y considerando los fundamentos de la interoperabilidad, como son la no replicación de datos y la conexión entre servicios, se decidió implementar la segunda opción.

CATALOGCONNECTOR, UN CLIENTE CSW

Seguramente una de las especificaciones OGC más extendidas e implementadas es WMS.

Peticiones fáciles de montar tipo KVP [4] vía HTTP-GET, respuestas de tipo raster que no necesitan ser “parseadas” y son directamente interpretadas por el navegador y lo que es más importante, la mayoría de las aplicaciones servidor que utilizan WMS lo han implementado de forma satisfactoria, es decir, que cuando se lanza un petición WMS ya no hace falta conocer a que desarrollador pertenece ese servidor WMS.

En el mundo de los catálogos de metadatos y sus implementaciones CSW, esas condiciones, no se dan: las peticiones son de tipo XML enviadas vía HTTP-POST, las respuestas, de tipo xml también, tienen que ser interpretadas y lo que es más importante, coexisten diferentes perfiles de aplicaciones (AP) dentro de la especificación CSW, lo que hace que cuando hay que interrogar a un catálogo, es recomendable saber de quién es el producto.

Por todo ello, se definieron unas premisas básicas para desarrollar el cliente CSW:

- Tenia que ser interrogado con la misma facilidad que un servicio WMS.
- Tenia que ser fácilmente escalable, sin necesidad de modificar código
- Tenia que interpretar y ofrecer las respuesta en formatos fácilmente manejables

Arquitectura

CatalogConnector es servlet de Java. Su funcionamiento se basa en recibir parámetros, enviar peticiones a servidores de catálogo e interpretar y devolver su respuesta.

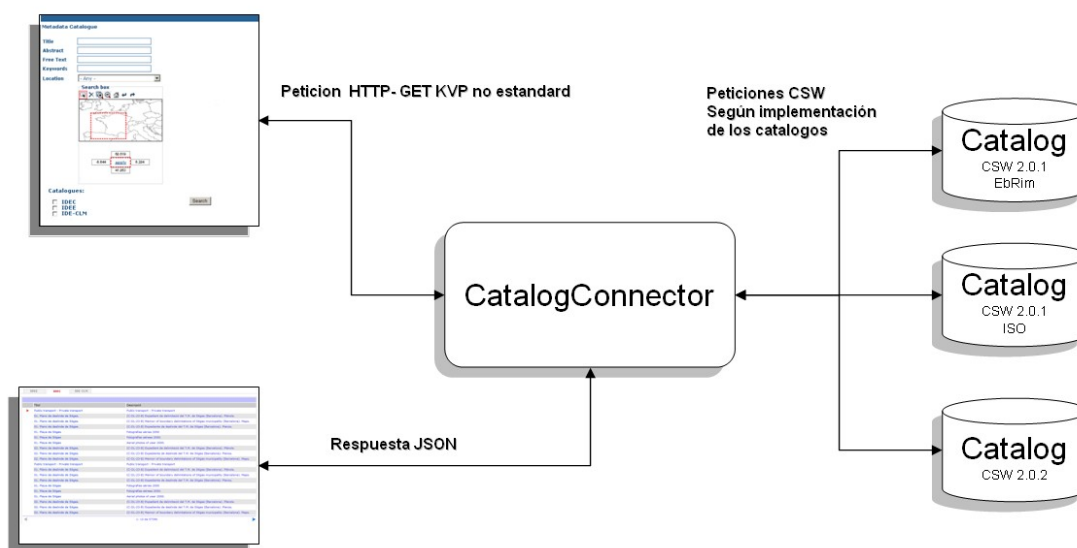


Figura 2. Funcionamiento de CatalogConnector

Se han implementado dos interfaces para las peticiones: *GetCapabilities* y *GetRecords*

GetCapabilities

Devuelve un archivo XML de capacidades dónde se describe el servicio y los catálogos que han sido configurados y que pueden ser interrogados. Diríamos que la concepción es la misma que WMS sustituyendo capas por catálogos.

Dentro de la estructura de directorios de CatalogConnector (dentro de WEB-INF) existe un archivo de configuración dónde se describen las conexiones a catálogos. Para cada catálogo se especifica:

Name: nombre de catálogo

Title: un título

Abstract: breve descripción

Urlcatalog: URL dónde lanzar las peticiones CSW

Product: nombre del productor (geonetwork, indicio ,etc..)

Csw-version: versión CSW soportada

```

<catalogues>
<catalog>
<name>idec</name>
<title>Infraestructura de Dades Espacials de Catalunya</title>
<abstract>IDEC metadata catalog</abstract>
<urlcatalog>http://delta.icc.es/indicio/csw</urlcatalog>
<product>indicio</product>
<csw-version>2.0.1</csw-version>
</catalog>
<catalog>
<name>idee</name>
<title>Infraestructura de Datos Espaciales de España</title>
<abstract>IDEE metadata catalog</abstract>
<urlcatalog>http://idee.unizar.es/csw/servlet/cswservlet</urlcatalog>
<product>catogc_server</product>
<csw-version>2.0.2</csw-version>
</catalog>
</catalogues>

```

Figura 3. Ejemplo archivo de configuración

GetRecords

Permite lanzar peticiones sobre los catálogos y está formada por los siguientes parámetros.

Parámetro	Valor	Descripción
Catalogs	Nombres de los catálogos, según valor "name" de capabilities. Separados por comas	Permite especificar a que catálogos se enviará la petición
Title	texto	Buscará en por título
Subject	texto	Buscará en por tema
Description	texto	Buscará en por descripción
Organization	Nombre organización	Buscará en por organización
AnyText	texto	Buscará en por cualquier texto
Bbox	XMIN,YMIN,XMAX,YMAX	Buscará en por caja de coordenadas
startPosition	número	Posición de búsqueda
maxRecords	número	Valor candidatos devueltos
outputFormat	JSON plain text XML	Formato de la respuesta

Figura 4. Tabla de parámetros

Así por ejemplo, para buscar metadatos por descripción dentro de los catálogos de IDEC e IDEE, lanzaríamos esta petición:

[http://dominio/CatalogConnector/Connector?
Request=GetRecords&Catalogs=idec,idee&description=Barcelona&outputFormat=JSON](http://dominio/CatalogConnector/Connector?Request=GetRecords&Catalogs=idec,idee&description=Barcelona&outputFormat=JSON)

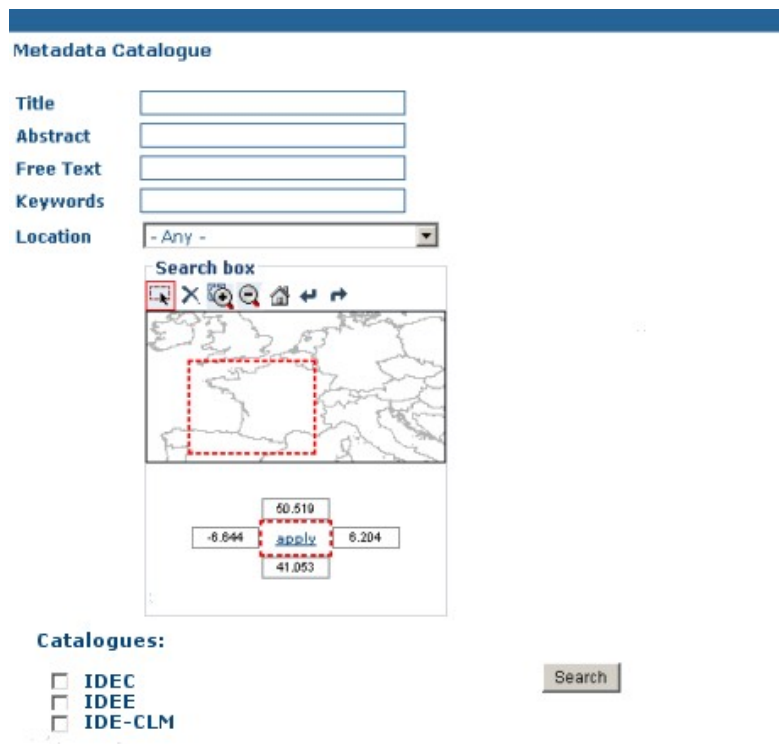


Figura 5. Ejemplo de aplicación web

El constructor de peticiones

El módulo de configuración para poder construir las peticiones hacia los catálogos, está orientado a producto.

Dentro del directorio “WEB-INF/catalogues” se crean subdirectorios con los nombres de diferentes productos de catálogos, como por ejemplo “geonetwork”. Dentro del directorio “geonetwork” se crea otro subdirectorio con la versión CSW soportada (ver figura 6).

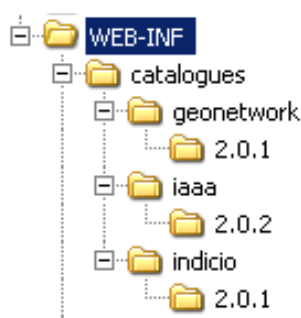


Figura 6. Estructura de directorios

Los nombres de estos directorios tienen que coincidir con los valores “product” y “csw-version” del archivo de configuración.

Finalmente en el interior del directorio de la versión se encuentran los diferentes archivos XML, utilizados para crear las peticiones.

Archivo	Descripción
Body.xml	Cabecera de la petición
Subject.xml	Cláusula “filter” por tema
Title.xml	Cláusula “filter” por título
Abstract.xml	Cláusula “filter” por descripción
Bbox.xml	Cláusula “filter” por caja de coordenadas
Organization	Cláusula “filter” por organización
Any.xml	Cláusula “filter” por cualquier
Response.xsl	Plantilla para leer la respuesta

Figura 7. Tabla archivos para crear la petición

Por defecto se han incluido los productos de *GeoNetwork*, *Catogc_Server* y *Indicio*, pero es de esperar que el cliente vaya creciendo y se añadan otros productos.

Para añadir un nuevo producto sólo es necesario crear un nuevo directorio y replicar esta estructura adaptando el contenido de las peticiones y la respuesta.

El constructor de respuestas

Este punto presentaba varios interrogantes:

- ¿Cómo devolver las respuestas de los servidores? a medida que iban respondiendo o todas de golpe.

Se decidió devolver las peticiones de los servidores a medida que iban respondiendo, de esta forma un posible servidor lento no penalizaría el resto de respuestas

- ¿Qué formato de respuesta utilizar para que fuera fácilmente integrable en una aplicación web?

Se integraron varios formatos de respuesta, pero se utilizaría por defecto JSON, por ser muy ligero y dar total libertad a los desarrolladores web para representar los resultados.

- ¿Cómo gestionar la paginación de los candidatos?

Se optó por gestionar de forma individual para cada catálogo.

Así pues, para optimizar los resultados de CatalogConnector es recomendable utilizar un aplicación web con librerías JSON y que utilice técnicas Ajax para comunicarse con el servidor

IDEE		IDEC		IDE-CLM	
Títol		Descripció			
▶	Public transport - Private transport	Public transport - Private transport			
	01. Plano de deslinde de Sitges.	(C-DL-23-B) Expedient de delimitació del T.M. de Sitges (Barcelona). Plànols.			
	01. Plano de deslinde de Sitges.	(C-DL-23-B) Memoir of boundary delimitations of Sitges municipality (Barcelona). Maps.			
	01. Plano de deslinde de Sitges.	(C-DL-23-B) Expediente de deslinde del T.M. de Sitges (Barcelona). Planos.			
	01. Playa de Sitges	Fotografies aèries 2000			
	01. Playa de Sitges	Fotografias aéreas 2000.			
	01. Playa de Sitges	Aerial photos of year 2000.			
	02. Plano de deslinde de Sitges.	(C-DL-23-B) Expedient de delimitació del T.M. de Sitges (Barcelona). Plànols.			
	02. Plano de deslinde de Sitges.	(C-DL-23-B) Expediente de deslinde del T.M. de Sitges (Barcelona). Planos.			
	02. Plano de deslinde de Sitges.	(C-DL-23-B) Memoir of boundary delimitations of Sitges municipality (Barcelona). Maps.			
	Public transport - Private transport	Public transport - Private transport			
	01. Plano de deslinde de Sitges.	(C-DL-23-B) Expedient de delimitació del T.M. de Sitges (Barcelona). Plànols.			
	01. Plano de deslinde de Sitges.	(C-DL-23-B) Memoir of boundary delimitations of Sitges municipality (Barcelona). Maps.			
	01. Plano de deslinde de Sitges.	(C-DL-23-B) Expediente de deslinde del T.M. de Sitges (Barcelona). Planos.			
	01. Playa de Sitges	Fotografies aèries 2000			
	01. Playa de Sitges	Fotografias aéreas 2000.			
	01. Playa de Sitges	Aerial photos of year 2000.			
	02. Plano de deslinde de Sitges.	(C-DL-23-B) Expedient de delimitació del T.M. de Sitges (Barcelona). Plànols.			
	02. Plano de deslinde de Sitges.	(C-DL-23-B) Expediente de deslinde del T.M. de Sitges (Barcelona). Planos.			
	02. Plano de deslinde de Sitges.	(C-DL-23-B) Memoir of boundary delimitations of Sitges municipality (Barcelona). Maps.			

1 - 10 de 57396

Figura 8. Ejemplo de respuesta por pestañas

REFERENCIAS

- ◆ Catalog Service for the Web: <http://www.opengeospatial.org/standards/cat>
- ◆ Infraestructura de Dades Espacials de Catalunya: <http://www.geoportal-idec.net>
- ◆ IdeUnivers: <http://www.ideunivers.eu>
- ◆ Key value Pair: http://en.wikipedia.org/wiki/Key_Value_Pair