

Historia y estado actual del futuro estándar Web Map Tiling Service del OGC

Joan Masó¹, Núria Julià¹ y Xavier Pons².

¹Centre de Recerca Ecològica i Aplicacions Forestals (CREAF)
Universitat Autònoma de Barcelona (UAB)
Facultad de Ciencias, 08193 Bellaterra (Barcelona)
Joan.Maso@uab.cat, n.julia@creaf.uab.cat

²Departament de Geografia
Universitat Autònoma de Barcelona (UAB)
08193 Bellaterra (Barcelona)
Xavier.Pons@uab.cat

Resumen

Web Map Service (WMS) es uno de los servicios de Open Geospatial Consortium (OGC) más ampliamente usado, pero a pesar de su éxito, muchas de las implementaciones existentes tienen graves problemas y dificultades para trabajar con datos de gran resolución o de gran volumen de datos.

En este artículo se expone el proceso de elaboración y el estado actual de un nuevo estándar OGC, el Web Map Tiling Service (el WMTS), del que los autores son los editores. Este nuevo estándar ha de permitir resolver estas limitaciones del WMS, acelerando las transacciones entre servidores y clientes.

El WMTS está basado en un modelo piramidal de teselas (*tiling model*), en el que los mapas son prerenderizados y fragmentados a un conjunto predeterminado de escalas. Incorpora tres peticiones: *GetCapabilities*, *GetTile* y *GetFeatureInfo*. Este servicio acelera la respuesta del servidor, ya que los datos están prerenderizados y mejora el rendimiento de los clientes, ya que estos conocen la estructuración de los datos y todas las peticiones son cacheables. El texto describe el debate existente sobre las codificaciones de las peticiones en RESTful, KVP y SOAP.

Palabras clave: OGC, WMS, WMTS, *Tiling*, Tesela, Fragmento, prerenderizar, piramidal.

1 Introducción

Open Geospatial Consortium, Inc.® (OGC) es un consorcio internacional sin ánimo de lucro, formado por un conjunto de empresas, agencias gubernamentales y universidades, dedicado a desarrollar especificaciones de interfaces para promover y facilitar el uso global de la información espacial. De todos los servicios *web* de OGC, el más conocido y usado es el servicio de mapas o *Web Map Service* (WMS), un estándar internacional (ISO 19128) que define un servicio para producir mapas de forma dinámica a partir de datos georeferenciados. Según este estándar un mapa es una representación pictórica de la información geográfica en forma de archivo de imagen digital apto para ser visualizado en la pantalla de un ordenador. Se definen tres operaciones básicas; una que permite conocer los metadatos del servicio, otra que permite obtener un mapa con un ámbito geográfico y dimensiones conocidas, y una última, opcional, que permite obtener información de los objetos particulares mostrados en el mapa [1].

El WMS es un estándar usado ampliamente en todo el mundo, ya que permite obtener de forma fácil y rápida información geográfica de cualquier tipo sin necesidad de ningún software específico. A pesar de su éxito, existen algunas limitaciones prácticas; muchas de las implementaciones existentes tienen graves problemas para trabajar con datos geográficos voluminosos o de gran resolución; responder de forma dinámica a peticiones de datos complejos puede ser muy costoso en tiempo si no se disponen de una estructura e organización interna de los datos geográficos que facilite su procesamiento y renderización.

En la actualidad existen diferentes aproximaciones que intentan solucionar estos problemas; todas ellas se basan en el uso de estructuras interna optimizadas. Algunos desarrolladores de *software*, como *LizardTech*, han optado por el uso de formatos de datos avanzados tipo *wavelet*, como *MrSID*, etc.; otros, como *OnEarh* del *Jet Propulsion Laboratory* (JPL) de la NASA o *MiraMon* del *Centre de Recerca Ecològica i Aplicacions Forestals* (CREAF) de la UAB, han optado por realizar una implementación óptima de los datos geográficos basada en una prerenderización y fragmentación de los datos, donde cada uno de estos fragmentos o teselas se conoce como *Tile*.

Todas estas aproximaciones, basadas en mantener formatos o estructuras internas más optimizadas y con los datos prerenderizados, aceleran la respuesta del servidor, haciendo más viables los servicios *web* de mapas complejos. Pero aún no son suficientes; el cliente WMS puro ignora y no participa de dichas

optimizaciones, ya que no conoce como se organizan los datos y no los puede cachear. Clientes basados en tiles, como *Google Maps*, han demostrado la utilidad de los servicios de caché de peticiones URL para acelerar aún más el servicio, reduciendo el tráfico de red y eliminando peticiones redundantes. En este sentido, el grupo de trabajo del WMS de OGC, en el que los autores participan, decidió editar un nuevo estándar que ha de permitir aprovechar al máximo todas estas optimizaciones.

Este artículo expone el proceso de elaboración y el estado actual de un nuevo servicio alternativo y complementario al WMS, el *Web Map Tiling Service* (WMTS) de OGC, del que los autores de este artículo son los editores; basado en un modelo interno y externo de teselas (*tiling model*), optimizable tanto para el servidor como para el navegador.

2 Antecedentes, origen e historia

En un principio, se realizó una evaluación de las diferentes implementaciones existentes de este tipo de servidores, compatibles con WMS o no, basados en *tiles*, como *OSGeo*, *OnEarth*, *Google Maps*, *MiraMon*,... Algunas de ellas, como *OSGeo* y *OnEarth* disponen de una sintaxis concreta para extender al cliente las optimizaciones internas de los datos del servidor, otras como *MiraMon*, disponen de optimizaciones en el servidor basadas en tiles, pero estas no son totalmente aprovechables y cacheables para los clientes. A continuación se detallan las principales características de algunas de estas implementaciones.

2.1 *Tile Map Service* de *OSGeo*

El *Tile Map Service* de *OSGeo* es un servicio separado del WMS que permite acceder de forma directa a fragmentos de mapas prerenderizados a un conjunto de escalas fijas [2].

En este servicio, cada capa o *TileMap* es una representación completa de un mapa, formada por un conjunto de imágenes prerenderizadas a escalas fijas (*TileSet*) y cortadas en fragmentos regulares (*tiles*) [2].

El acceso a los recursos se hace vía interfície REST (*Representational State Transfer*). Existe un servicio raíz (*Root resource*) que describe que servicios y versiones se han implementado; a partir de esta información se puede obtener una descripción detallada de los metadatos del servicio (*TileMapService resource*); y a partir de estos solicitar los metadatos de cada capa (*TileMap resource*).

Un *TileMap resource* describe los patrones de corte de un mapa a partir del sistema de coordenadas, ámbito del mapa, origen de coordenadas y tamaño de los fragmentos, y el formato y tamaño de los píxeles para cada conjunto de fragmentos prerenderizado a una escala fija (*TileSet*) [2]. A partir de aquí se puede acceder directamente al fragmento o *Tile* deseado mediante una URL, es decir, se accede directamente a imágenes enlazadas que no requieren ningún procesamiento y que son totalmente cacheables.

2.2 Tiled WMS de OnEarth

Al igual que en el servicio *Tile Map Service* de *OSGeo*, en el *Tiled WMS* de *OnEarth*, cada capa es prerenderizada y cortada a un conjunto de escalas predeterminadas [3]; pero a diferencia del anterior, este servicio es una extensión del WMS, en el se define una nueva petición, *GetTileService*, que permite obtener los patrones de corte de cada capa (*TilePattern*); pero no define ninguna sintaxis ni servicio para solicitar los fragmentos; solamente se informa de cómo se han preparado las capas, los fragmentos serán solicitados de la forma habitual en un servidor WMS, mediante peticiones *GetMap*, no introduce ningún protocolo adicional, de forma que un servidor *Tiled WMS* puede funcionar normalmente como un WMS.

Esta idea es la misma usada en otro servicio de *OSGeo*, el WMS-C [4], que se ha aprovechado para intentar ampliar el estándar WMS en lo que se conoce como *WMS Cached profile*; éste está siendo usado por algunos servidores nacionales como es el caso del servidor de mapas de PNOA. El WMS-C mejora el rendimiento, sobretudo en cuanto al servidor, pero no resuelve todos los problemas. Al no existir una sintaxis explícita para solicitar los fragmentos, se debe usar la petición *GetMap* en codificación KVP (*Key Value Par*); haciendo que las peticiones no sean realmente cacheables por los clientes, ya que solo que el ámbito de la petición varíe en algún decimal o que el orden de los parámetros no sea el mismo la petición es considerada diferente por la mayoría de sistemas de caché.

2.3 Origen e Historia del *Web Map Tiling Service* de OGC

Considerando todas las implantaciones existentes y sus ventajas e inconvenientes, el grupo de trabajo del WMS elaboró un artículo de discusión interno (07-057r1) en el que el modelo de teselas era incorporado al estándar WMS como una extensión a la que se dio el nombre de *Tiled WMS*. En esta extensión se definieron nuevas peticiones que permitían conocer la organización de los fragmentos (*DescribeTiles*) y su posterior solicitud (*GetTile*) [5].

Todo esto fue avanzando y cogiendo forma y, en este momento, lo que empezó como una extensión del WMS, actualmente está siendo reelaborado como un nuevo servicio estándar de OGC, el *Web Map Tiling Service* (07-057r5) [6], que en breve entrará en la fase de RFC (*Request For Comments*).

Este estándar, el *Web Map Tiling Service* del OGC, ha sido elaborado basándose principalmente en el *Tiled Map Service* de *OSGeo* y en el *Tiled WMS* de *OnEarh*, intentando armonizar los diferentes aspectos que describen, dotándolo de las principales ideas y requerimientos de los servicios *web* de OGC y asegurando la compatibilidad e interoperabilidad con *Google Maps*, el KML (*Keyhole Markup Language*) y el JPIP (*JPEG 2000 Interactive Protocol*), protocolos y servicios usados ampliamente.

3 Modelo actual en desarrollo: *Web Map Tiling Service*

Un WMTS es un servicio escalable y cacheable, que usa un modelo de teselas (*tiling model*) parametrizado de tal manera que un cliente puede hacer peticiones de un conjunto discreto de valores y recibir rápidamente del servidor fragmentos de imágenes prerenderizadas (*tiles*) que no requieren de ninguna manipulación posterior.

Cada una de las capas (*layers*) de un servidor WMTS sigue estructuras piramidal de escalas, en la que cada escala o nivel de la pirámide es una rasterización y fragmentación de los datos geográficos a una escala o tamaño de píxel concreto.

Una capa puede estar disponible en varios sistemas de coordenadas, y tener diferente ámbito en función de éstos. Una *TileMatrixSet* es un conjunto de rasterizaciones y fragmentaciones de la capa a diferentes escalas predeterminadas

para un sistema de referencia y ámbito concreto. Una capa tendrá tantos *TileMatrixSet* como sistemas de referencia y ámbitos para los cuales este disponible. Cada una de las rasterizaciones y fragmentaciones de la capa a una escala determinada, sistema de coordenadas y ámbito que conforman la *TileMatrixSet*, se conoce como *TileMatrix* (ver figura 1).

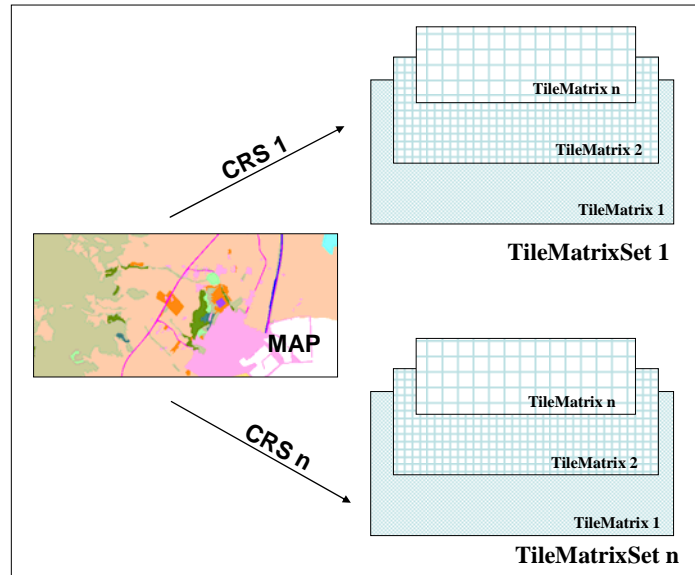


Figura 1: Modelo piramidal de teselas de una capa de un servidor WMTS.

Cada *TileMatrix* define el patrón de corte de la capa en teselas para un sistema de referencia concreto, un ámbito y una escala; indicando el origen de la matriz en el sistema de coordenadas (*TopLeftPoint*), el tamaño de la matriz en número de fragmentos (*MatrixWidth* y *MatrixHeight*) y el tamaño de cada uno de sus fragmentos o *tiles* en número de píxeles (*TileWidth* y *TileHeight*) (ver figura 2).

La interfaz WMTS especifica dos operaciones básicas que pueden ser solicitadas por un cliente: la operación de *GetCapabilities* y la de *GetTile* (ver figura 3).

La operación de *GetCapabilities*, obligatoria en todos los servicios *web* OGC, permite al cliente solicitar y recibir del servidor los metadatos del servicio; que en este caso son una descripción detallada de las capas disponible en el servidor y de los patrones de matrices de fragmentación que siguen.

Una vez conocidas las capacidades de un servidor, la operación de *GetTile* permite al cliente solicitar y recibir una tesela de una de sus capas, a una escala, estilo de visualización y formato determinado.

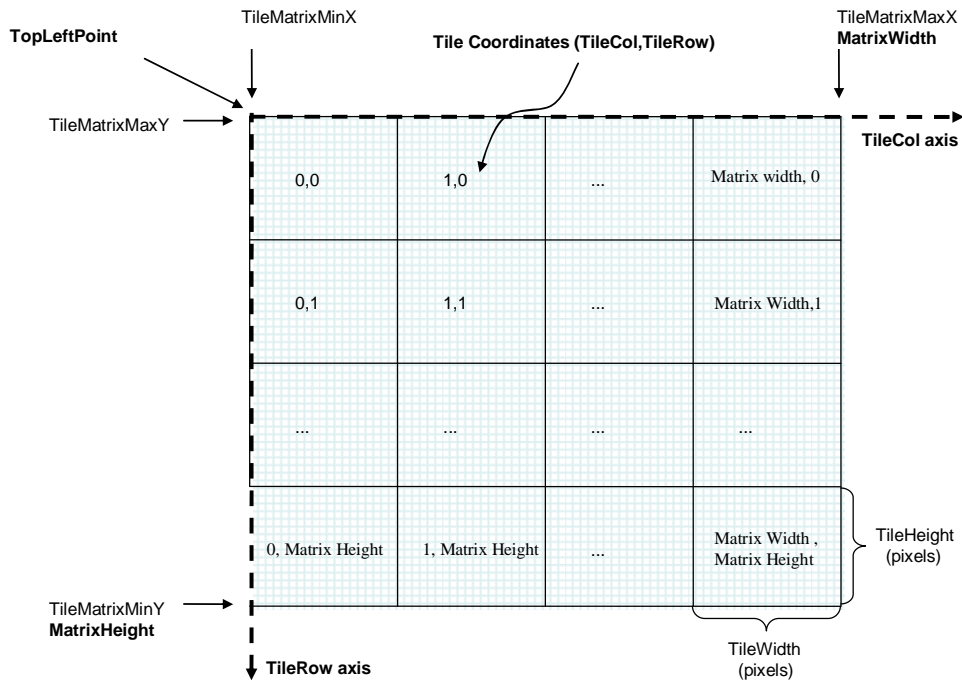


Figura 2: Espacio de una matriz de teselas (TileMatrix)

La especificación del WMTS, también define otra operación, en este caso opcional, el *GetFeatureInfo*, que permite obtener la información sobre el contenido de un píxel particular del mapa.

Las peticiones del servicio WMTS pueden realizarse en codificación GET y POST con KVP (*Key Value Par*) o en codificación RESTful. La codificación KVP fácil de especificar y muy usada, permite realizar cualquiera de las peticiones del servicio de forma eficaz, pero tiene el inconveniente de que si se altera el orden de los parámetros, la petición ya no se cachea. La codificación RESTful, permite cachear e identificar de forma unívoca cualquiera de los recursos del servicio WMTS, el documento de los metadatos del servicio, cada tile y cada píxel de un tile son identificados como recursos, correspondiendo a cada uno una URL.

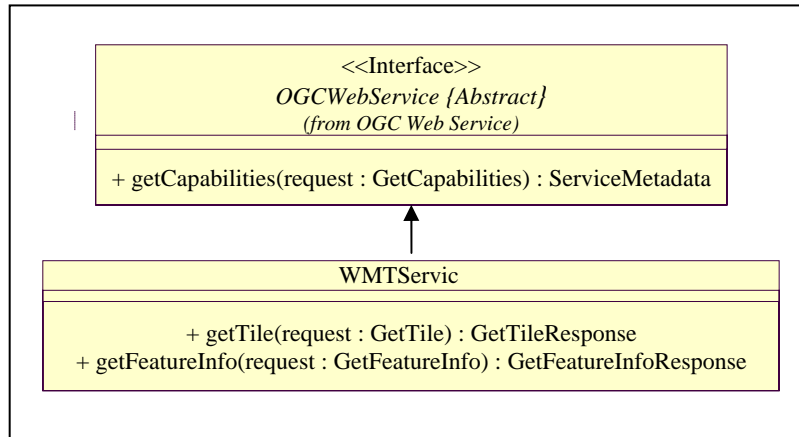


Figura 3: Diagrama UML de la interficie WMTS.

A continuación (figura 4) se muestran los principales parámetros de una petición de capacidades en codificación KVP:

PARÁMETRO (nombre y valor o ejemplo de valor)	USO	DESCRIPCIÓN
SERVICE=WMTS	Obligatorio	Servicio al que se hace referencia, en este caso WMTS
REQUEST=GetCapabilities	Obligatorio	Petición que se solicita, en este caso <i>GetCapabilities</i>
VERSION=1.0.0	Opcional	Versión del servicio a la que se hace referencia

Figura 4: Principales parámetros de una petición de capacidades de un servicio WMTS en KVP.

Un ejemplo de esta petición seria:

`www.maps.org/maps.cgi?REQUEST=GetCapabilities&SERVICE=WMTS`

La misma petición en codificación RESTful seria una dirección URL que se expresaría como:

`http://www.maps.org/1_0_0/WMTSCapabilities.xml`

A continuación (*figura 5*) se muestran los principales parámetros de una petición para solicitar un fragmento de un mapa en codificación KVP:

PARÁMETRO (nombre y valor o ejemplo de valor)	USO	DESCRIPCIÓN
SERVICE=WMTS	Obligatorio	Servicio al que se hace referencia, en este caso WMTS
REQUEST=GetTile	Obligatorio	Petición que se solicita, en este caso <i>GetTile</i>
VERSION=1.0.0	Obligatorio	Versión del servicio a la que se hace referencia
LAYER=mcsc	Obligatorio	Identificador de la capa
STYLE=level1	Opcional	Identificador del estilo de visualización
TILEMATRIXSET=cat	Obligatorio	Identificador del patrón de corte
FORMAT=image/jpeg	Obligatorio	Formato de salida del fragmento solicitado
SCALE=250k	Obligatorio	Identificador de la escala solicitada
TILECOL=1	Obligatorio	Índice de la columna del fragmento en la matriz
TILEROW=4	Obligatorio	Índice de la fila fragmento en la matriz

Figura 5: Principales parámetros de una petición GetTile de un servicio WMTS en KVP.

Un ejemplo de esta petición sería:

```
www.maps.org/maps.cgi?REQUEST=GetTile&SERVICE=WMTS&VERSION=1.0.0&LAYER=mcsc&STYLE=level1&TILEMATRIXSET=cat&FORMAT=image/jpeg&SCALE=250k&TILECOL=1&TILEROW=4
```

La misma petición en codificación RESTful sería una dirección URL que se expresaría como:

```
http://www.maps.org/1_0_0/mcsc/level1/cat/250k/4/1.jpg
```

Un servidor WMTS debe ser escalable y soportar diferentes tamaños de teselas, escalas y alineamientos de los fragmentos, aunque por muchos que soporte siempre será en un número limitado, ya que se trata de un servidor de fragmentos

pre-renderizados. En última instancia, es el cliente quien deberá realizar las transformaciones de coordenadas y el reescalado de los fragmentos. Como las funcionalidades de los clientes pueden ser muy limitadas, y para facilitar la combinación de mapas de diferentes servidores en un mismo cliente y aumentar la interoperabilidad entre servidores, se ha definido unos conjuntos de combinaciones estándares de grupos de escalas y sistema de coordenadas conocidos como *WellKnown ScaleSets*.

Para eliminar redundancias y reducir el tamaño de los documentos de capacidades, y como diferentes capas pueden tener un mismo patrón de corte, se ha diseñado un mecanismo para compartir la definición de los conjuntos de matrices de teselas. Estos pueden definirse independientemente y desde cada capa enlazar a uno de ellos mediante su identificador.

4 Discusión y conclusiones

En las pruebas realizadas hasta el momento, estos servidores se han mostrado muy útiles y eficaces para capas de ámbitos mundiales o muy extensos, para imágenes de satélite o aéreas de gran resolución, mejorando notablemente la velocidad de las comunicaciones entre servidores y clientes; pero aun quedan cuestiones que resolver.

Una de ellas es la necesidad de las peticiones *GetFeatureInfo* o consultas por localización. Algunos argumentan que estas peticiones no son necesarias ya que esta información se puede extraer de otros servicios como el WFS o el WCS; otros, entre los cuales se encuentran los autores, piensan que son imprescindibles para obtener información de lo que se está visualizando y que eliminándolas se está ligando el WMTS con el WFS y el WCS, obligando a que los clientes sean multiservicio, cuando en realidad todos los servicios OGC deberían poder funcionar totalmente independientes. Otro punto a favor de mantener estas peticiones es la disponibilidad de las fuentes, los servicios WFS y WCS, son, en realidad, servicios de descarga de datos geográficos; estos datos pueden estar protegidos y no ser accesibles a todo el público o inexistentes; además pueden provenir de diversas fuentes, o estar en formatos desconocidos o difíciles de interpretar por un cliente simple, lo que dificulta aún más su uso.

En los servicios web existen actualmente dos tendencias: las arquitecturas orientadas a recurso (ROA) y las arquitecturas orientadas a servicio (SOA). La

primera persigue la identificación de recursos y sus relaciones; utiliza una codificación RESTful para acceder a ellos. La segunda, persigue la descripción de servicios y operaciones y sus partidarios prefieren en una codificación denominada SOAP. Ambas aproximaciones tienen partidarios y detractores. Los partidarios del ROA defienden una arquitectura más simple basada en la web tradicional y las URLs mientras que los partidarios del SOA defienden una mayor estandarización y concreción en la descripción de las operaciones y parámetros.

Para evaluar estas cuestiones el experimento de interoperabilidad OWS-6 (OGC Web Services, Phase 6), incorpora un apartado para evaluar el WMTS. En este se elaborarán informes técnicos y se implementarán varios servicios WMTS en codificación REST y SOAP que deberán armonizarse con las actuales codificaciones KVP.

Otro problema intrínseco de este tipo de servicios es debido a las etiquetas o la toponimia donde la posición de algunos elementos del mapa depende de la proximidad de elemento al borde de la representación impidiendo que iconos o textos queden cortados. Por construcción interna del WMTS no es directo aplicar estas optimizaciones u otras que necesiten de algún mecanismo de rasterización dinámico.

Referencias

- [1] de la Beaujardiere, Jeff (ed.): OpenGIS® Web Map Server Implementation Specification 1.3.0, OGC Document #06-042, <http://www.opengeospatial.org/standards/wms>, (2006).
- [2] OSGEO Tile Map Service Specification version 1.0, http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification (2007).
- [3] OnEarth WMS tiled extension , <http://onearth.jpl.nasa.gov/tiled.html> (2007).
- [4] OSGEO WMS Tiling Caching, http://wiki.osgeo.org/wiki/WMS_Tile_Caching (2007).
- [5] Masó, J. y Julià, N.(ed.): OpenGIS® Tiled WMS Discussion Paper 0.3, OGC Document #07-057r1 (2007).
- [6] Masó, J., Julià, N. y Pomakis, K. (ed.): Candidate OpenGIS® Web Map Tiling Server Implementation Standard 0.5.7, OGC Document #07-057r5 (2008).