(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

Servidor de mapas interoperable para Internet, una aproximación Java basada en la reutilización de componentes SIG

P. Fernández, R. Béjar, R. López, J. Zarazaga, P.R. Muro-Medrano[1]

Computer Science and Systems Engineering Department
University of Zaragoza
María de Luna 3
50015 Zaragoza, SPAIN

pedrofb@ebro.cps.unizar.es
rbejar@ebro.cps.unizar.es
lrafa@ebro.cps.unizar.es
javy@posta.unizar.es
prmuro@posta.unizar.es
http://iaaa.cps.unizar.es

Resumen: El "Web Map Server Interface Specification" desarrollado por el consorcio OpenGIS en la última parte de 1999, propone un marco estándar para la publicación e intercambio de información geográfica, incrementando la interoperabilidad entre aplicaciones GIS sobre Internet. Por otra parte, la implementación de software para Internet está altamente condicionada por el crecimiento y estandarización de Java como lenguaje de implementación preferido. Los autores ilustran una implementación Java del servidor de mapas propuesto por OpenGIS, tanto en el servidor para [] los servicios de mapas de la especificación como en el lado del cliente, usando Java y el más extendido HTML. El artículo muestra la experiencia...

Palabras clave: *Java, GIS, applet, servidor de mapas en web, orientación a objeto, OpenGIS*

Introducción

El consorcio OpenGIS (en adelante OGC)[2][3] es una organización sin ánimo de lucro dedicada a la promoción de nuevas técnicas para el geoprocesamiento distribuido e interoperable, fundada por las más importantes entidades industriales, gubernamentales y académicas. Recientemente, empujado por el impacto de Internet, el consorcio ha puesto un gran interés en aprovechar las posibilidades abiertas por el web. El siguiente párrafo muestra su idiosincrasia: "Gran cantidad de información geoespacial está disponible en el Web en archivos estáticos, pero es compleja, heterogénea e incompatible.[...] Los interfaces comunes son la única forma de habilitar la superposición y combinación automática de complejas y esencialmente diferentes fuentes de información geográfica sobre Internet, debido a las diferencias de base entre los sistemas GIS [...]".

La necesidad de un tecnología de publicación de mapas en web adaptada a las actuales necesidades en el mundo en el mercado GIS sobre Internet ha condicionado la , y capaz de incorporar los avances tecnológicos de este sector de tan rápida evolución. [9][10][11][15][16]. La coincidencia de nuestros intereses con el OGC en este aspecto, la especificación del interfaz de un servidor de mapas propuesta por OpenGIS ha sido la guía (OpenGIS Web Map Server Interface Specification [1]).

Trabajos anteriores en proyectos de sistemas de información geográfica nos llevaron a tomar la decisión de desarrollar nuestra propia tecnología GIS basada en la plataforma del lenguaje de programación Java, aprovechando la experiencia y el software desarrollado en proyectos anteriores. [5][6][7][19] El servidor de mapas desarrollado utiliza como motor de visualización el componente de visualización GIS desarrollado, aportando además capacidades que han sido desarrolladas pensando en la funcionalidad que el servidor debe ofrecer y que por su generalicidad van a poder ser incorporadas en la siguiente versión del componente. Es tarea de este artículo exponer como se ha construido el servidor de mapas a partir de un software existente, indicando los puntos clave del diseño y señalando los puntos de choque encontrados, así como describir la arquitectura de dos sistemas combinados, uno de propósito general y otro especializado a partir del anterior. En los siguientes puntos del artículo se muestra esta visión.

[1] Autor a quien se debe dirigir la correspondencia.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

Contexto y arquitectura del servidor de mapas

La especificación del servidor de mapas de OpenGIS define una serie de descripciones de servicios relacionados con la producción de mapas que un servidor web conforme con la especificación debe ser capaz de responder. El interfaz común es la manera de conseguir que una aplicación pueda interactuar con los servicios de distintos servidores de mapas que cumplan la especificación de OpenGIS. Cada implementación específica debe proveer la funcionalidad requerida por el interfaz respetando los servicios, métodos, convenciones y nombrado propuestos en la especificación En la figura 1 puede verse como diversos clientes pueden acceder a los servicios de producción de mapas del servidor JWMS, o a los de cualquier otro conforme con la especificación.

La especificación del servidor de mapas tiene tres tipos principales de servicios: producción de mapas, información sobre elementos y capacidades del servidor. Los distintos tipos de clientes que pueden acceder componen estas peticiones a partir de la interacción con el usuario y se encargan de enviarlas y procesar las respuestas del servidor para mostrar los resultados. Las peticiones de servicio de mapas son enviadas al servidor con el objetivo de obtener un mapa sobre alguna zona de interés. La petición estará compuesta por los parámetros que el servidor requiere para generar el mapa, como el sistema de referencia, el tipo de información a incluir o el formato de respuesta. Las peticiones de información de elementos son una extensión de las anteriores. Especificando un pixel sobre el mapa generado el servidor responde con un listado, en texto plano o XML, de todos los elementos geográficos que contienen dicho pixel. Las peticiones de capacidades informan sobre el propio servidor, especificando los servicios, datos y formatos que oferta el servidor de mapas. Herramientas adicionales permiten componer los mapas que serán ofrecidos por el servidor y configurar adecuadamente las capacidades. Una descripción más detallada de la arquitectura desarrollada puede encontrarse en [4].

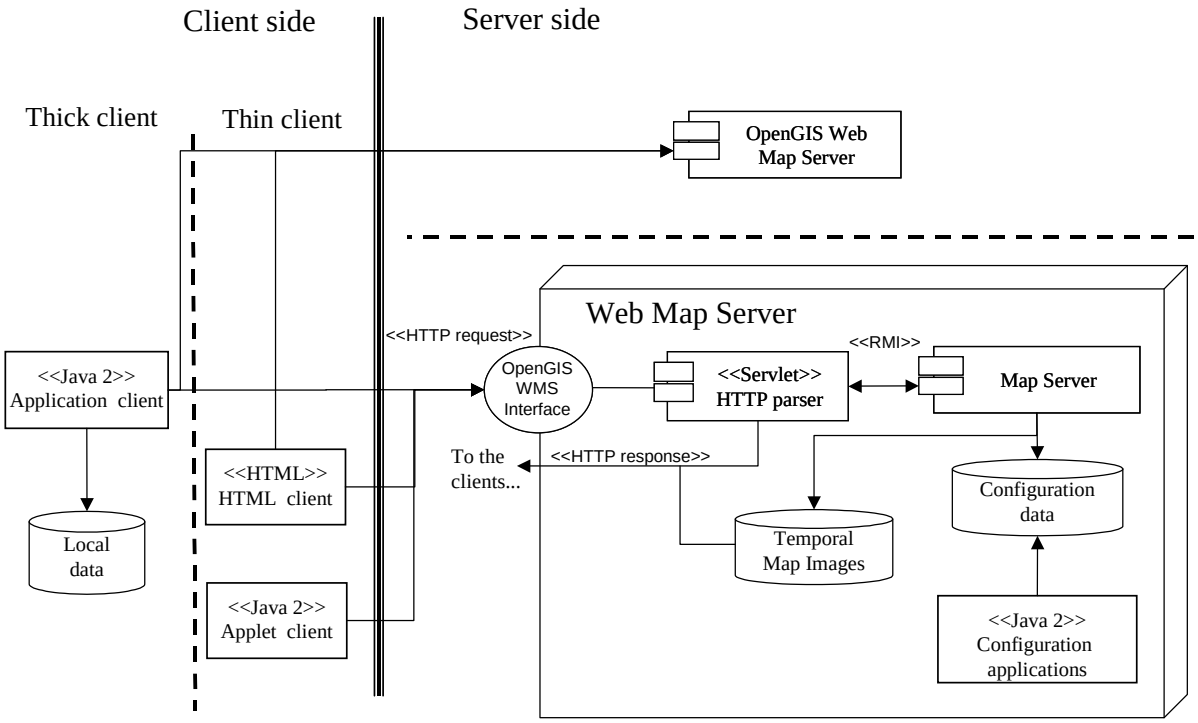


Figura 1: Arquitectura de JWMS

El principal componente del JWMS es el servidor de mapas. Este componente, implementado en Java, ofrece una interfaz similar a la especificada por el OGC para servir mapas en Internet, y es accesible a componentes externos por RMI, el protocolo Java para la gestión de objetos distribuidos.. Un servlet de Java codifica el acceso al servidor de mapas a través del interfaz definido por OpenGIS. El servlet se integra dentro de un servidor web y se encarga de traducir las peticiones HTTP al equivalente RMI del servidor de mapas. Una petición de servicio produce que se carguen y visualicen determinadas capas de información sobre el mapa activo. Una vez generado el mapa, se salva la información contenida en el formato especificado en la petición y el fichero generado es devuelto al cliente como una referencia URL.

El servidor de mapas (MapServer en la figura 1) está compuesto por tres módulos principales, el componente genérico de visualización GIS, el productor de mapas y el gestor de capacidades. El componente de visualización GIS es una librería de clases Java con capacidades de gestión de mapas e información geoespacial, que sigue una

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

arquitectura tipo JavaBean [14][22]. El productor de mapas es el encargado de hacer la traducción de las peticiones que recibe el servidor a visualización de mapas concretos sobre el módulo de visualización GIS. El recibe las peticiones de mapas conformes con la especificación, genera y visualiza el mapa de interés usando el módulo de visualización y salva el contenido del mapa devolviendo al peticionario una referencia o URL al fichero generado. El último módulo, el gestor de capacidades, va a estar a cargo de configurar los servicios y datos disponibles del servidor de mapas y de responder a todas las peticiones de capacidades sobre el servidor de mapas. La información manejada por el gestor se encuentra disponible en un fichero XML.

Componente genérico de visualización GIS

El componente de visualización GIS tiene su antecedente en el incremento de las necesidades del mercado del sistemas de información geográfica. Durante los últimos años, y debido principalmente al crecimiento de la potencia de los ordenadores y del nacimiento de redes de información que permiten compartir información, los usuarios exigen la inclusión en sus sistemas de información tradicionales la capacidad de gestionar gráficamente información geográfica. El componente desarrollado ofrece herramientas para gestionar información geográfica, como visualización de información georeferenciada tipo raster o vectorial, utilidades de zooming o paning sobre los mapas, o selección y visualización de información de los elementos. El componente también dispone de una librería básica de GUI (Graphical User Interface), que permite a otras aplicaciones integrar el módulo sin tener que implementar las utilidades más comunes, como ventanas de visualización de mapas, barras de herramientas, o visores de registros en selección.

El módulo de visualización GIS esta íntegramente desarrollado en Java. Sigue un diseño orientado a objetos fácilmente reusable e incrementable, que ha sido diseñado con el objetivo de ser integrado en múltiples sistemas de información geográfica que tengan unas necesidades de visualización comunes. El módulo aporta capacidades de visualización y gestión de información geoespacial, recuperadas de anteriores proyectos que necesitaron la inclusión de información geográfica, como aplicaciones de mineria, sistemas de seguimiento de vehículos en tiempo real, o gestión de recursos humanos en base a información geográfica (proyectos Leader) [5][6][7][19]. La funcionalidad del módulo ha sufrido un proceso evolutivo, resultado de uso en múltiples proyectos, incrementando sus capacidades y corrigiendo errores conforme a las necesidades específicas de cada proyecto. Sin embargo no pretende ser una librería de propósito general como podría pensarse, sino que sólo cubre las necesidades más comunes que necesitan los sistemas SIG desarrollados por nuestro equipo, reservando el uso de plataformas comerciales para los casos en que se necesiten capacidades mucho más elaboradas, como podría ser por ejemplo el análisis espacial, o la gestión de depósitos de datos espaciales con acceso concurrente.

Las ventajas que ofrece el componente son varias, modularidad, reusabilidad, fácil incremento de sus prestaciones, que permite implementar nuevas funcionalidades adaptadas al problema en el momento en que se necesitan, y por supuesto la libertad en el pago de licencias. Sin embargo no todo es tan bonito como lo pintan. El problema del desarrollo de un componente genérico que pueda ser utilizado en varios proyectos exige un gran esfuerzo de diseño preliminar, cotejando las diferentes aproximaciones comerciales, teniendo en cuenta las limitaciones en cuanto a dinero y personal, y delimitando la funcionalidad que debe ser implementada, para no caer en el error de construir un componente con demasiadas funciones, no lo suficientemente genéricas como para poder ser utilizadas por el abanico de proyectos de interés. A su vez el desarrollo de un proyecto que requiere funciones de visualización GIS no implementadas necesita un esfuerzo similar para discernir cuáles de ellas son lo suficientemente genéricas como para formar parte del componente, de aquellas que sólo van a ser útiles dentro del contexto del proyecto en cuestión, y por tanto realizar un mayor esfuerzo de diseño e implementación sobre las primeras de manera que se puede desarrollar un módulo más potente.

En la figura X se exponen los diagramas de objetos principales del componente de visualización. El *JMapControl* es el objeto principal del diseño. Su misión es representar gráficamente un mapa. Se puede ver como un lienzo donde se dibujan en orden ascendente cada una de las capas de información que contiene. Una capa (*Layer*) es el objeto que contiene la información geográfica y sabe representarla. El objeto capa contiene las características que son comunes a todas las fuentes de información, como zona geográfica representada, escalas máxima y mínima de visualización, o estado de visualización. Distinguimos entre tres fuentes distintas fuentes de información que heredan la funcionalidad del objeto *Layer*, la *ImageLayer* que representa imágenes georeferenciadas, la *MapLayer* que es la encargada de dibujar elementos vectoriales utilizando patrones de dibujo y la *OpenGISLayer*, que es una especialización de la *ImageLayer* y su función es encapsular los métodos de acceso a servidores de mapas conformes con la especificación de OpenGIS. La implementación de los métodos específicos para el pintado de los datos de cada una de los tipos de capas se realiza a través del método abstracto *draw()* que posee pero no implementa la clase *Layer*.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

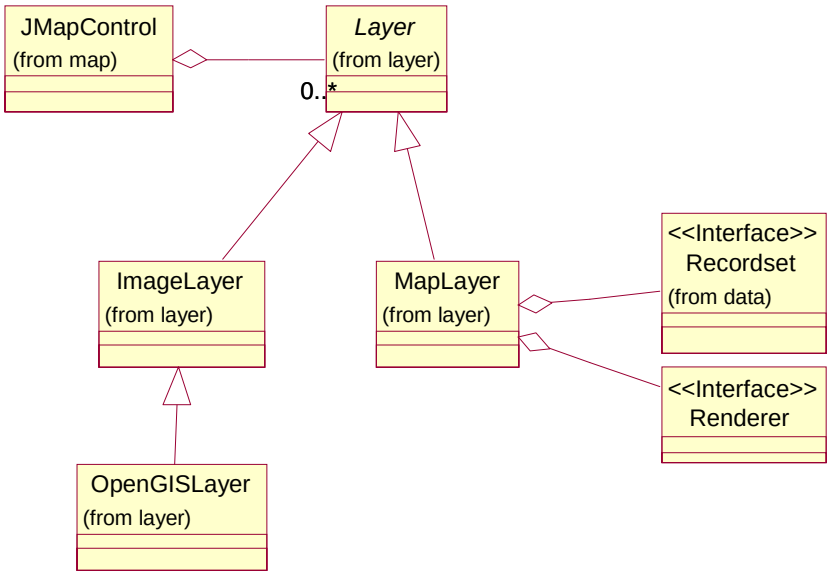


Fig X: Diagrama principal

El JMapControl es el componente principal de la librería, encapsula los aspectos fundamentales de la gestión de mapas, como...La MapLayer esta formada por...

Todos los objetos descritos han sido construidos siguiendo la directrices arquitecturales de los JavaBeans, la arquitectura de componentes Java definida por Sun[13]. El objetivo de un JavaBean reside en ofrecer al mismo tiempo una implementación y una especificación de las propiedades y servicios que dispone un determinado componente gráfico Java. La especificación de propiedades y servicios permite que aplicaciones externas puedan interrogar al JavaBean y obtener la funcionalidad que el componente ofrece. De esta manera un JavaBean, en nuestro caso un *JMapControl* o alguno de los tipos de *Layer*, pueden ser insertados en la barra de herramientas de cualquier editor Java, de manera que el usuario, pinchando y arrastrando con el ratón los componentes, puede configurar la funcionalidad y el aspecto de su aplicación.

Los *JavaBeans* creados para la gestión de mapas utilizan el mecanismo estándar de Java basado en eventos para comunicar algún cambio en su estado (explicar patron sujeto-observador[12]) . Cada bean creado tiene asociados determinados tipos de eventos que puede emitir, que se suman a los ya ofrecidos por ser componentes Java. Por ejemplo el *JMapControl*, como sujeto en el patrón, tendrá observadores que les interese ser notificados en el momento en el que se produzca algún cambio en su estado como en la extensión del mapa visualizado, o al añadir una nueva capa de información. Cada uno de los observadores de los eventos de mapas implementan un interfaz (denominado *Listener* en la terminología Java[14]), de esta manera pueden ser suscritos en la lista de observadores del objeto sujeto y ser notificados cuando se produzca un cambio. En la figura X, los objetos graficos *Scale*, *ZoomButton*, o *MapLegend*, se suscriben a la lista de eventos del *JMapControl* para ser informados de los cambios de la extensión del mapa, coordenadas del ratón, y capas visualizadas respectivamente.

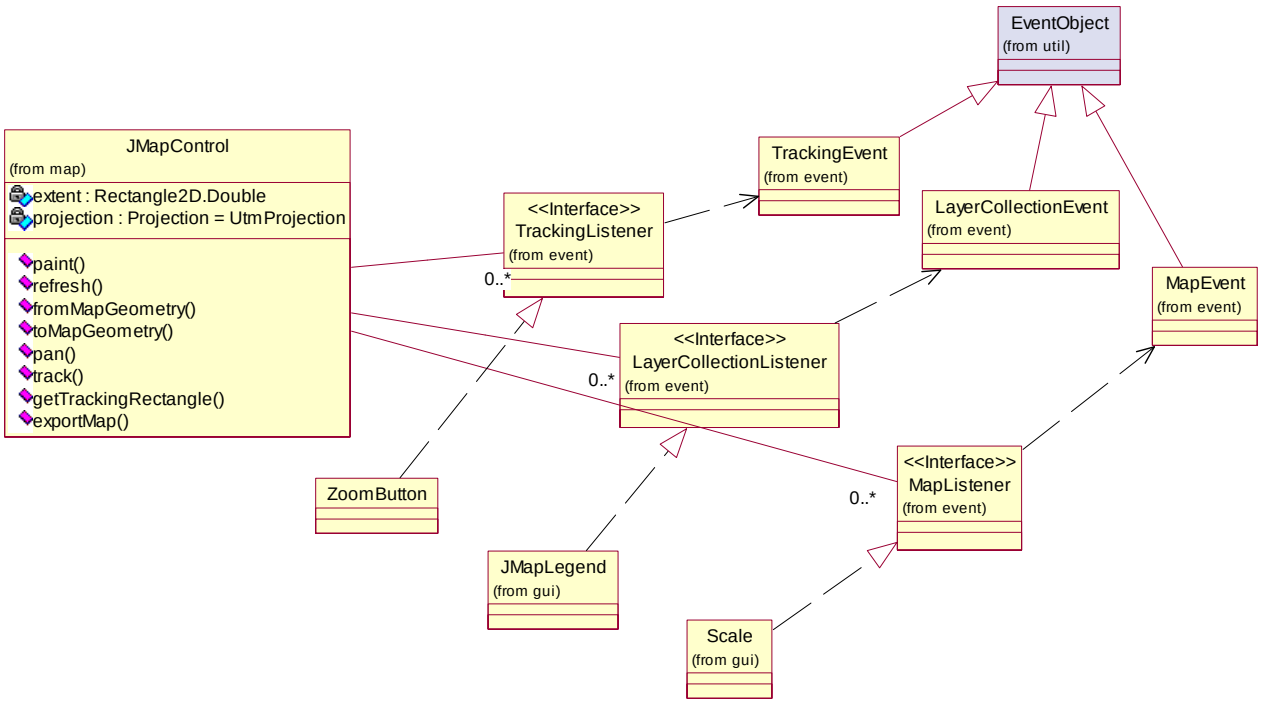


Fig X: JMapControl, Observadores y eventos. (Quiza sobra)
Una explicación más detallada del componente de visualización puede encontrarse en [7]

Componente de servicio de mapas

Jerarquía del servidor

Las capacidades de visualización de información geográfica quedan cubiertas por el componente descrito en el apartado anterior. El trabajo realizado en este proyecto consiste en ampliar dichas capacidades para soportar la producción de mapas acorde con las especificaciones de servicio que exige el interfaz OpenGIS.

La estructura de clases que sigue el servidor se guía por la especificación de OpenGIS de los servicios que debe ofrecer un servidor de mapas, relativos a la producción de mapas, la información de un elemento y la devolución de las capacidades de un servidor. En el diagrama de la Figura X el interfaz Java *MapServer* define los servicios de la especificación de OpenGIS. Tiene tres servicios, *getMap*(), *getFeatureInfo()* y *getCapabilities()* que cumplen respectivamente con las tres funciones anteriores. La implementación del interfaz del servidor de mapas se encuentra codificada en la clase *JMapServer*. Esta clase será la encargada de procesar todas las peticiones recibidas, generar una respuesta y devolver una referencia a un fichero que contendrá el mapa generado, la descripción de los elementos de un cierto pixel, o el fichero XML de descripción de capacidades.

En figura X se detalla la arquitectura y funcionamiento de la clase JMapServer, así como su relación con el componente de visualización. Se puede observar que el servidor de mapas incluye algunas clases adicionales que van a facilitar la construcción y distribución del servidor. La clase JMapServer está codificada como una clase Java normal. No dispone de un interfaz al exterior que permita a aplicaciones externas solicitar sus servicios. La forma de utilizar esta clase es integrándola como librería dentro del contexto de otra aplicación. Para dotar a la clase de un mecanismo que permita la invocación desde aplicaciones remotas se ha utilizado RMI, el mecanismo proporcionado por Java para la distribución, y que permite solicitar servicios de instancias de objetos remotos. La funcionalidad y uso de RMI es similar a la que ofrece CORBA. La clase RemoteMapServer proporciona un interfaz RMI con los mismos servicios que el servidor de mapas, y cuyos servicios pueden ser invocados por objetos remotos. Su misión es traducir las peticiones a un objeto MapServer en concreto.

Sin embargo, utilizando esta jerarquía de objetos no se consigue construir completamente un servidor de mapas conforme con la especificación de OpenGIS. La especificación exige recibir y contestar peticiones a través de un servidor web, utilizando el protocolo HTTP. La misión de la clase HTTPMapServer es realizar la

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

función de traducción. Esta clase está implementada como un Servlet, una clase especial de Java que se ejecuta dentro del entorno de un servidor Web. El servlet recibe las peticiones de mapas a través del servidor web, traduce la petición a su equivalente RMI y se la envía al servidor de mapas remoto codificado en el objeto *RemoteMapServer*. Finalmente devuelve al cliente, a través del mismo protocolo HTTP una referencia al mapa generado por el servidor. La división del servidor de mapas en dos zonas o niveles, el nivel del servidor web, y el nivel de aplicación remota tipo RMI, permite por una parte dividir la complejidad del problema, ya que el funcionamiento del servidor tiene su propia política de inicialización y cacheado de información, que puede interferir con la deseada para el sistema, y por otra parte la localización de un servidor de mapas accesible desde RMI permite una utilización mucho más sencilla y potente de los servicios de mapas por parte de otras aplicaciones hechas a medida localizadas en el propio servidor o en cualquier otra máquina.

La última clase de diagrama es el *MultiMapServer*. Esta clase no añade nueva funcionalidad al servidor pero incrementa de manera notable sus prestaciones. Codifica bajo el mismo interfaz de servicio de mapas el acceso a varias instancias de objetos *JMapServer*. El *MultiMapServer* va a tener referencias a varios *MapServer* con la misma información y dotados de una capacidad de servicio similar. Su misión es distribuir las peticiones entre los objetos de servicio de mapas, maximizando la utilización de recursos del sistema. Cuando se reciba una petición, el "dispatcher" va a decidir cual es el servidor al que se le debe enviar la petición para que la procese. En el caso de que haya algún servidor disponible se enviará aleatoriamente a uno de estos, y si todos están ocupados se estimará cual de los servidores va a quedar libre en menos tiempo y se encolará la petición. De esta manera se consigue minimizar el tiempo de respuesta, aunque la existencia de varios procesos simultáneos obligatoriamente exige un aumento en el tiempo medio de servicio.

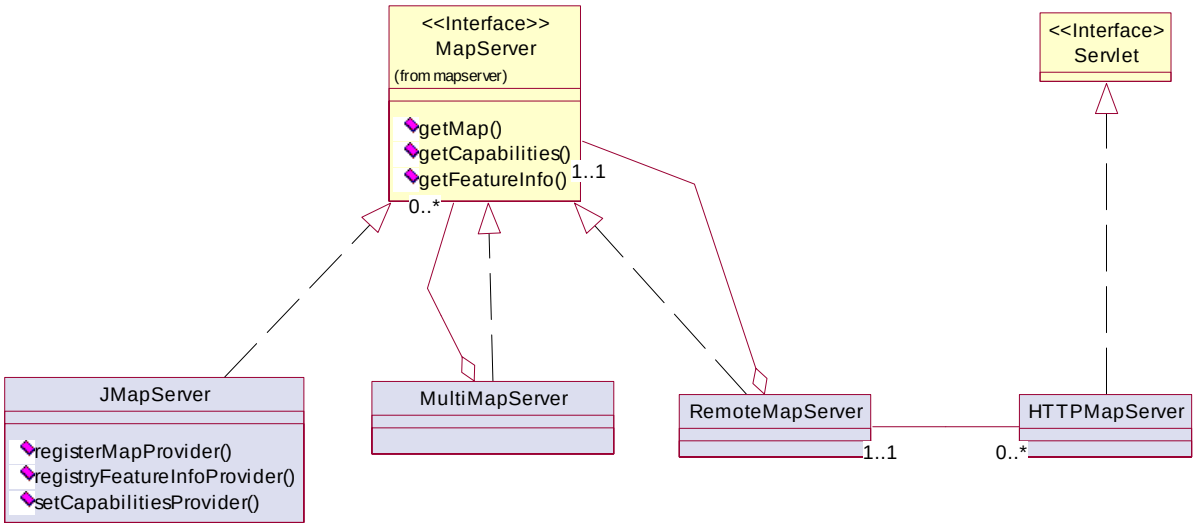


Figura X: Jerarquía de objetos del servidor de mapas

Ampliación del componente de visualización para la producción de mapas

En este punto se va a abordar la estructura de la pieza fundamental del sistema, el productor de mapas. Como se ha indicado en el punto anterior el *JMapServer* es el encargado de procesar y responder finalmente todas las peticiones que llegan al servidor. Con el objetivo de poder incrustar diversas fuentes de datos con características de servicio distintas, el *JMapServer* divide la complejidad del interfaz MapServer de OpenGIS en tres interfaces complementarios: *MapProvider*, *FeatureInfoProvider* y *CapabilitiesProvider*. Cada uno de estos subinterfaces será implementado por un objeto especializado en realizar el servicio. La división permite a su vez la especialización y colaboración de servidores adaptados al trabajo con determinadas fuentes de datos. Dentro del *JMapServer*, se puede indicar que *Provider* debe procesar las peticiones que incluyan una capa o fuente de datos específicas. De esta manera mapas generados con el componente de visualización pueden tener un *MapProvider* especializado, así como pueden ser integradas en el futuro otras fuentes de datos con características particulares implementando un *MapProvider* especializado, por ejemplo para acceder a información en una base de datos, o en otro servidor de mapas conforme con OpenGIS.

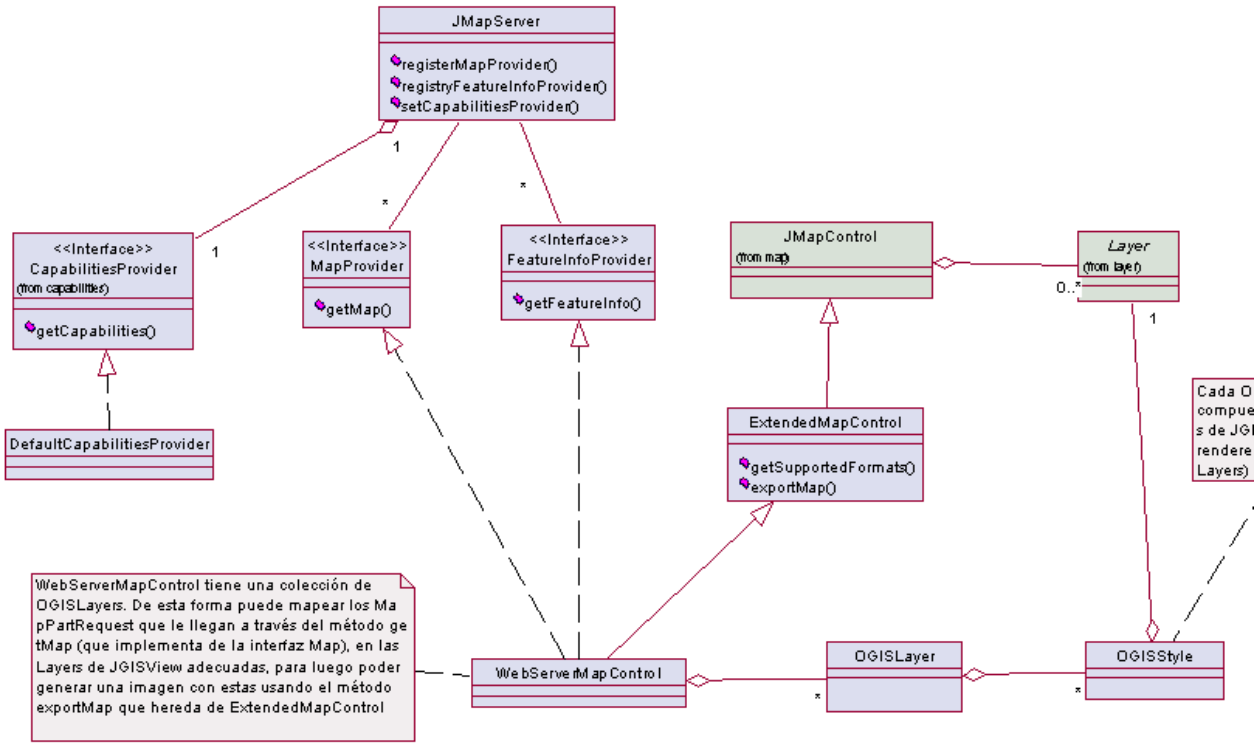
Las capacidades de visualización de información geográfica con las que dispone el servidor van a estar soportadas por el componente de visualización. A su vez los mapas que va a entender nuestro servidor van a estar generados a partir de las utilidades que ofrece el componente de visualización genérico. Estos mapas son

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

almacenados y cargados de disco utilizando las herramientas de gestión de configuraciones del componente. A través de la herencia, que ofrecen los sistemas de orientación a objeto se van a poder reutilizar fácilmente los servicios del componente de visualización, extendiéndolos con la funcionalidad adicional requerida por el servidor de mapas.

La extensión del componente de visualización se debe realizar en dos partes diferenciadas. Por una parte se necesita dar persistencia a los mapas que el componente de visualización es capaz de generar, utilidad que puede llegar a ser útil en otros contextos y que se incluirá en una siguiente versión del componente de visualización. A su vez el procesamiento y configuración de un mapa a partir de una petición concreta tiene una problemática particular que solo se tiene lugar en el contexto de los servidores de mapas en Internet. Como resultado de este análisis se ha decidido incrementar las capacidades del objeto genérico de visualización geográfica *JMapControl* en dos niveles adicionales de herencia. El primero, denominado *ExtendedMapControl*, aporta las capacidades de exportación de un mapa concreto usando diferentes formatos (JPEG, GIF, PPM, PNG, SVG y GML [4]). Este nivel todavía en pruebas, es el que se incluirá por su genericidad en la siguiente versión estable del componente de visualización. El siguiente nivel, encargado de la problemática del procesamiento de peticiones según el interfaz de OpenGIS se codifica en la clase denominada *WebServerMapControl*, y que también implementa el interfaz *MapProvider*, lo que permite que sea usada como fuente de servicio de datos dentro del servidor de mapas.

El último punto que queda por tratar es la conversión de las estructuras de datos especificadas por OpenGIS a las soportadas por el componente de visualización. La estructura de ambos sistemas no es equivalente, mientras en este último un mapa esta formado por un lista de capas de información consecutivas, en la especificación de OpenGIS se define una estructura de datos formada por capas, donde cada capa puede estar representada por varios estilos de visualización. Por ejemplo un mapa aceptado por el componente de visualización podría llamarse Zaragoza y contener una capa de ríos, una de manzanas y otra de calles. En el servidor de mapas podría almacenarse de manera equivalente, con tres capas: ríos, manzanas y calles, con la salvedad de que una capa puede estar representada por varios estilos. Así la capa de calles podría estar dividida en los estilos tramos_de_calles y nombres_de_calles. La solución que se ha tomado ha sido crear dos estructuras de datos paralelas a las existentes en el componente de visualización y que codifican la relación entre capas y estilos de la especificación de OpenGIS (ver objetos *OGISLayer* y *OGISStyle*). Un estilo se hace corresponder a un mapa del componente de visualización por lo que una capa OpenGIS, formada por un conjunto de estilos, representa realmente un conjunto de mapas del componente de visualización.. Dada una petición, la estructura de datos basada en capas y estilos, y la relación existente entre un *OGISStyle* y un conjunto de capas. El *WebServerMapControl*, debe incrustar en el mapa que representa todas las capas contenidas en todos los *OGISStyles* que estén implicados en la petición. Una vez que el mapa esté construido, se utilizarán los servicios que proporciona el objeto padre *ExtendedMapControl* para exportar el contenido del mapa a disco y se devolverá la referencia al servidor, que se encargará de transmitirla al cliente.



(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

Gestión de capacidades e información sobre elementos

De forma equivalente al servicio de producción de mapas, los servicios de capacidades e información sobre elementos se gestiona desde el *JMapServer*, utilizando respectivamente las funciones ofertadas por el *CapabilitiesProvider* y por el/los *FeatureInfoProvider* registrados. El procesamiento de peticiones que involucran información tabular sobre los elementos situados bajo un pixel del mapa se realiza también por el *WebServerMapControl*, que aprovechando las capacidades de acceso a la información tabular de los elementos vectoriales de una capa que ofrece el componente de visualización, implementa el servicio *getFeatureInfo()* de manera similar al de producción de mapas. La devolución de los resultados es en este caso a través de un fichero XML que contiene la descripción de los elementos. La gestión de capacidades está relacionada con un único servidor, por lo que no es necesario crear una infraestructura parecida a la anterior para relacionar capas con proveedores de servicios. Las capacidades son generales para el conjunto del servidor y vienen descritas en un fichero XML asociado al servidor y que sigue la especificación de formato y contenidos que define OpenGIS. El proceso de translación de la información del fichero XML a datos sobre objetos Java se localiza en la clase DefaultCapabilitiesProvider, también utilizada desde una aplicación auxiliar de gestión de capacidades para dar persistencia los cambios que se produzcan sobre las capacidades. Esta clase utiliza internamente el parser Java de Sun [13] para la interpretación del fichero XML

Clientes y aplicaciones adicionales.

El software desarrollado permite distribuir mapas en Internet cumpliendo con la especificación de servicios que define OpenGIS. Junto con el software principal se han desarrollado varias aplicaciones complementarias que facilitan al usuario final la tarea de configurar el servidor y visualizar los resultados. La aplicación de gestión de capacidades presenta un interfaz gráfico que facilita al usuario la tarea de configuración y mantenimiento de los servicios e información disponible en el servidor. La aplicación hace uso de las librerías de clases desarrolladas en el servidor para la configuración del fichero XML de capacidades. La segunda aplicación en el servidor es el generador de mapas, que básicamente es el mismo componente de visualización en si. Permite detallar gráficamente la información que va a contener un mapa concreto, así como su aspecto gráfico.
En la parte del cliente, se han implementado tres tipos de aplicaciones que permiten acceder al servidor de mapas, mostrando una vista de la información geográfica que el servidor contiene. El primero es una simple página HTML que codifica el acceso al servidor de mapas. En su parte central tiene un mapa, que corresponde a la selección de algunas de las capas disponibles del servidor, y sobre el que se pueden realizar operaciones de zoom, pan, o de visualización de la información de los elementos situados en un determinado pixel. El segundo cliente es un Applet de Java, que es una pequeña aplicación incrustada en una página HTML y que básicamente puede realizar las mismas operaciones que el documento HTML. La diferencia es que realiza un procesado dinámico de las capacidades del servidor actualizando la lista de capas y estilos de que dispone el servidor. También codifica algunas utilidades adicionales de visualización, como medición de áreas y distancias, barras de escala, coordenadas o estado, y modificación del orden de visualización de capas. El último cliente es la propia aplicación de generación de mapas instalada en el cliente, con la salvedad de que permite acceder a la información del servidor de mapas. Permite también combinar coberturas locales con mapas provenientes de uno o de varios servidores conformes con la especificación de OpenGIS.
La principal característica de estos clientes es que están construidos utilizando los javabeans del componente de visualización, lo que permite incrementar sus capacidades de una forma sencilla, a la vez que simplifica la tarea de mantenimiento, ya que una modificación en el comportamiento de un bean, se traslada directamente a cada una de las tres versiones.

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

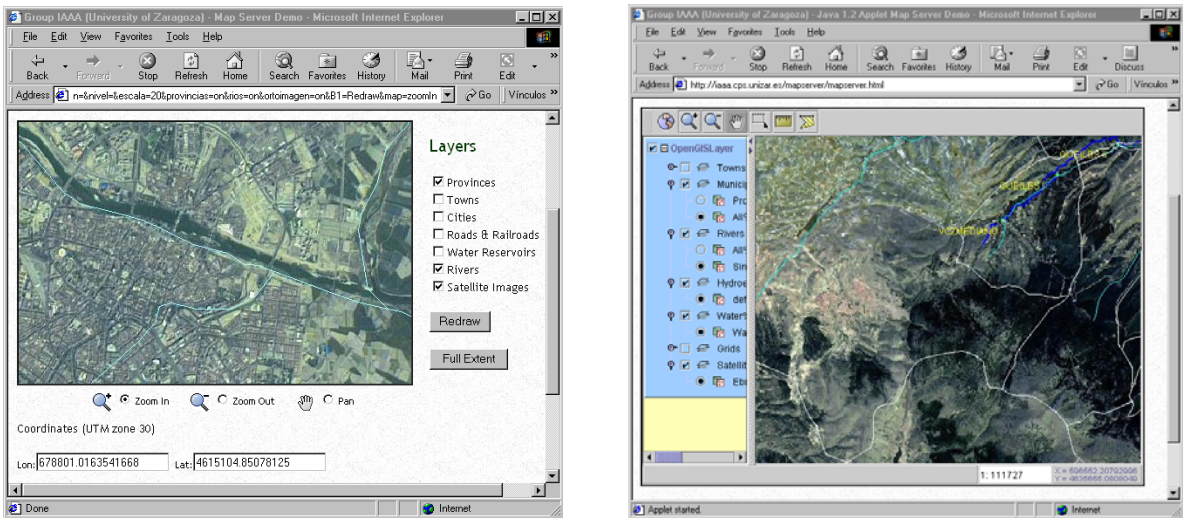


Figura 2: Clientes ligeros (HTML a la izquierda y applet de Java a la derecha)

Conclusiones

El desarrollo del módulo ha sido cofinanciado por varios proyectos GIS que necesitaban un soporte genérico para la visualización GIS, y que por necesidades económicas debía ser de bajo coste. La existencia de varios proyectos relacionados usando la misma herramienta GIS ha permitido recuperar la inversión inicial y la exoneración de los proyectos del pago de licencias de ejecución.

[MAS]

Bibliografía

[1] OpenGIS Project Document 99-077r4, OpenGIS Consortium 2000 "*OpenGIS Web Map Server Interface Specification (version 1.0)*".

[2] OpenGIS Project Document 99-112, OpenGIS Consortium 1999. *The OpenGIS Specification Model. Topic 12: The OpenGIS Service Architecture (version 32)*.

[3] Homepage del OpenGIS Consortium. http://www.opengis.org

[4] P. Fernández, R. Béjar, M.A. Latre, J. Valiño, J.A. Bañares, P.R. Muro-Medrano " Web mapping interoperability in practice, a Java approach guided by the OpenGis Web Map Server Interface Specification." EC-GIS 2000, 6th European Commission GI and GIS Workshop

[5] P. Fernández, J. Nogueras, O. Cantán, J. Zarazaga, P.R. Muro-Medrano. "*Java Application Architectures to Facilitate Public Access to Large Remote Sensed and Vector Geographic Data*". Telegeo '2000. Second International Symposium on Telegeoprocessing. Sophia Antipolis, pp. 81-92.France, 10 –12 May, 2000.

[6] M. Á. Latre, R. Béjar, P. Fernández, P. Álvarez, P. R. Muro-Medrano. "*Trying Java technology in a Geologic-Mining Information System distributed over an inter/intranet environment*". Telegeo '2000. Second International Symposium on Telegeoprocessing. Sophia Antipolis, pp. 93-102. France, 10 –12 May, 2000.

[7] F.J.Zarazaga, P. Álvarez, J. Guillo, R. López, J. Valiño, P.R. Muro-Medrano. "*Use Cases of vehicle location systems based on distributed real-time GPS data*". Telegeo '2000. Second International Symposium on Telegeoprocessing. Sophia Antipolis, pp. 53-62. France, 10 –12 May 2000.

[8] W.F. Limp, "*WEB MAPPING*". GEOEurope. N. 8, pp. 18—22. Dec. 1999.

[9] A. Sorokine, I. Merzliakova. "*Interactive map applet for illustrative purposes*". Proceedings of the 6th International Symposium on Advances in Geographic Information Systems. pp. 46—51. 1998.

[10] Yafang Su, Joan Slottow, Avi Mozes. "*Distributing proprietary geographic data on the World Wide Web—UCLA GIS Database and Map Server*", Computer&Geosciences N.26 pp.741-749, 2000

[11] Harder C, "*Serving maps on the Internet*" Environmental Systems Research Institute Inc, Redlands, California, 130 pp. 1998

[12] Rober Orfali, Dan Harkey, "*Client/Server Programming with Java and Corba*", second edition, Wiley Computer Publishing, 1998

[13] Homepage of Java, from Sun Microsystems. http://www.java.sun.com

[14] Bruce Eckel, "*Thinking in Java*", Prentice Hall, 1998

(Draft) Sistemas de información geográfica: una aproximación desde la ingeniería del software y las bases de datos.
Monografías y publicaciones. Colección ingeniería informática.
Madrid: Fundación Dintel, 2001, p.179-184. ISBN 84-931933-1-3.

[15] Tom Barclay, Jim Gray, Don Slutz. "Microsoft TerraServer: A Spatial Data Warehouse". 25th VLDB Conference 31-May-1999.

[16] O.T. Balovnev, A. Bergmann, M. Breunig, A.B. Cremers, S. Shumilov. "Remote Access to Active Spatial Data Repositories". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 125--130. Lyon, France. May, 1999.

[17] B. Cambray, C. Leclerc, J.R. Houllier. "Software Architectures Based on Cartographical Products". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 31--39 Lyon, France. May, 1999.

[18] "ER Mapper Image Web Server". Earth Resource Mapping Pty Ltd. 1999

[19] P. Fernández, P. Álvarez, M.A. Latre, R. Béjar, J. Zarazaga, Muro-Medrano "*Sistema de Información Geológico-Minero con capacidad de visualización SIG*" VIII Conferencia Nacional de Usuarios de ESRI 21-Oct-1999

[20] P.P. Gonzalves, M. Costa. "Local and Remote Geoprocessing Applications". *TeleGeo'99, First International Workshop on Telegeoprocessing*. pp. 53--60. Lyon, France. May, 1999.

[21] *"A client/server JDBC Driver based on Java RMI"* http://dyade.inrialpes.fr/mediation/download/RmiJdbc/RmiJdbc.html

[22] M. Morrison, R. Weems, P. Coffe, J. Leong "How to Program JavaBeans" Macmillan Computer Publishing 1997.