

Una visión de las posibilidades del estándar OpenGIS para la interoperabilidad de aplicaciones GIS sobre CORBA

S. Comella, J. Zarazaga, J. Ezpeleta, D. Infante, R.López, P. R. Muro-Medrano

Departamento de Informática e Ingeniería de Sistemas

Centro Politécnico Superior, Universidad de Zaragoza

María de Luna 3, 50015 Zaragoza

{ javy, prmuro@posta.unizar.es }

Resumen

En este trabajo se aborda el problema de interoperabilidad en aplicaciones con sistemas de información geográfica. La solución se basa en la utilización del estándar CORBA para el procesamiento de objetos distribuidos y un estándar emergente para interoperabilidad en sistemas de información geográfica: OpenGIS, lo que permite resolver el problema de trabajo con fuentes de datos heterogeneas y aplicaciones pobremente integradas.

Más concretamente, se presenta una arquitectura útil para la integración de aplicaciones heterogéneas (opcionalmente también distribuidas) en un entorno en el que la información de carácter geográfico es importante. El sistema se está desarrollando para mejorar los servicios del inventario de puntos de agua que utiliza la Oficina de Planificación Hidrológica de la Confederación Hidrográfica del Ebro. Sin embargo, la utilización de estos estándares no está limitada a este tipo de aplicaciones particulares sino que, es la tendencia de la tecnología de software para abordar la interoperabilidad en GIS.

Palabras Clave

Interoperabilidad, OpenGIS, CORBA.

1. Introducción

Un campo de investigación (I+D) que está adquiriendo un creciente interés dentro de la ingeniería del software es el de la interoperabilidad y los mecanismos y tecnologías para alcanzarla. En [KOTT-95] se define interoperabilidad como la libertad de mezclar elementos software en un sistema de información sin comprometer el éxito global.

Atendiendo a la anterior definición, resulta evidente que cuanto más heterogéneos sean las informaciones y servicios en un sector de la industria del software, tanto más importante resulta la ansiada interoperabilidad y tanto más difícil alcanzarla. Un mercado que caracteriza perfectamente este problema lo constituyen los sistemas de información geográfica (SIG). La gran potencia y disponibilidad de los sistemas PC y software para SIG actuales han provocado grandes expectativas de uso e integración en aplicaciones de estas utilidades. Los sistemas de información geográfica gestionan y emplean datos geográficos o GEODatos, que describen fenómenos o entidades, a partir de ahora *features*, directa o indirectamente asociados con una localización, tiempo y orientación, relativa a la superficie terrestre. La cantidad de GEODatos accesible ha crecido rápidamente a la par que los avances tecnológicos en campos como los satélites de alta resolución y tecnologías de posicionamiento global (GPS), así como la cada vez mayor conciencia de los tecnólogos de que indexar la información por su

posición es muchas veces un modo adecuado de organizar y usar datos digitales [MCKE-96].

En el contexto SIG, interoperabilidad refiere la capacidad de los sistemas digitales para: 1) Intercambiar libremente todo tipo de información sobre la tierra así como los objetos y fenómenos en, sobre o bajo la superficie terrestre; y 2) cooperativamente, a través de redes de datos ejecutar software capaz de manipular tal información [OCG-OG]. Poner en común información cobra sentido por el hecho de que sólo hay una tierra que todos compartimos.

El principal y más conocido esfuerzo por alcanzar la interoperabilidad en el sector lo constituye el proyecto Open Geodata Interoperability Specification (OpenGIS u OGIS). Es una especificación para definiciones orientadas a objeto de geoinformación, que permitirá el desarrollo de geoproceso auténticamente distribuido a través de grandes redes de datos, así como el desarrollo de aplicaciones geográficas realmente interoperables [SMB-96].

OpenGIS es un proyecto muy ambicioso: pretende la interoperabilidad a escala "galáctica" [OHE-97], esto es, interoperabilidad entre TODOS los SIG de la red. Es una tecnología emergente, hasta la fecha ningún software cumple oficialmente con el estándar. El desarrollo expuesto en este artículo emplea conceptos e ideas provenientes de OpenGIS, pero no es ni pretende ser una solución "OGIS compliant". El objetivo que se plantea en este trabajo es algo más limitado: consiste en una propuesta de solución a un problema real, que se ajuste al modelo abstracto de interoperabilidad y lo bastante flexible como para en un futuro extender la solución para hacerla realmente conforme OpenGIS.

A continuación se presenta un problema real de interoperabilidad en un entorno SIG en el que estamos trabajando actualmente: primero se describe el problema, la solución actual (no interoperable) y el resultado pretendido; a continuación se explica la arquitectura de la solución así como las estrategias y tecnologías fundamentales involucradas; finalmente se extraen algunas conclusiones y se compara la solución con otros trabajos análogos.

2. Ejemplo de necesidades de "interoperabilidad" entre aplicaciones con información geográfica.

Un ejemplo prototípico de necesidades de interoperabilidad en el contexto de los SIGs lo constituyen organizaciones como puede ser una Confederación Hidrográfica (CH), que tiene por objeto las actividades relacionadas con la planificación, administración, control y explotación de los aspectos hidrológicos de la cuenca y la elaboración de su Plan Hidrológico. Para poder cumplir dichas funciones, se requiere la utilización de una gran variedad de datos (parte de ellos georeferenciados), equipos informáticos y software de un conjunto variado de aplicaciones. Para ilustrar la problemática que ello acarrea centrémonos como ejemplo en su sistema para Inventario de Puntos de Agua (IPA), que permite gestionar el inventario de puntos de agua superficiales y subterráneas de toda la cuenca (como pozos y manantiales, por ejemplo). Actualmente la CH dispone de una infraestructura informática que le proporciona cierto grado de automatismo en la gestión y tramitación de los expedientes de concesión de aguas. Dicha infraestructura se ve afectada por numerosos problemas debido a serias limitaciones de interoperabilidad de procesamiento y diversidad de formatos de datos.

A continuación se describe de una manera sencilla y esquemática un proceso de tratamiento de datos típico que la CH debe llevar a cabo. Se trata de uno de los múltiples servicios que la CH debe realizar con datos relativos a la cuenca. Supongamos que un nuevo punto de agua debe ser dado de alta en la base del inventario de puntos de agua. En un primer paso, el punto de agua es georeferenciado a través de una aplicación SIG (en este caso ArcView). La inserción del punto requiere que la información asociada al mismo sea relacionada con muchos aspectos diferentes de la información preexistente en el servicio: el punto debe ser situado en la posición correcta de una de las 250 hojas 1:50.000 de imágenes satélite en que se divide la cuenca (70 Gbytes

en total); a su vez, es necesario que se ponga en relación con más de 15 coberturas relativas a términos municipales, cuencas hidrográficas menores, espacios naturales, regadíos o coberturas con aspectos geológicos del terreno, etc. Estas coberturas deben ser creadas y actualizadas por otras aplicaciones de tipo GIS como ArcInfo, de tipo estadístico, bases de datos o aplicaciones de ofimática. Estas imágenes y coberturas deberían poder ser utilizadas también para otras necesidades de la CH. La inserción del punto requiere introducir también informaciones relativas a su localización. Parte de esta información es gráfica, y puede proceder tanto de fotos introducidas con escaner, como de croquis o imágenes provenientes de cámaras de fotos digitales. Actualmente, estos datos gráficos son tratados posteriormente con otra aplicación: Corel. El proceso que se acaba de describir es realizado concurrentemente (puede haber hasta 7 puestos de trabajo simultáneamente en la introducción de datos en el IPA), lo que hace que además de los problemas relativos a un proceso semiautomático (lentitud en la realización de algunos de los pasos del proceso, aumento de la probabilidad de introducción de errores, etc.), aparezcan problemas típicos de disponibilidad de información actualizada y consistente en un sistema concurrente.

En principio el problema presenta dos vertientes, por un lado la heterogeneidad de las fuentes de datos: Coberturas GIS, tablas relacionales, fotos digitalizadas, etc., por otro la baja integración de las aplicaciones involucradas: gestores de bases de datos relacionales, visualizadores GIS, aplicaciones de escaneo, etc. Es importante distinguir entre ambos problemas, que aunque relacionados no son idénticos:

Un ejemplo del problema de las fuentes de datos heterogéneas lo constituyen las coberturas con persistencia en la base de datos relacional (por ejemplo el Inventario de Puntos de Agua), hay diversos motivos por los que almacenar una cobertura en una tabla relacional, en el caso del IPA la decisión estuvo motivada por el interés en reflejar las relaciones entre los Puntos de Agua y otras entidades no georreferenciadas (expedientes administrativos, etc.). A la hora de visualizar estas coberturas mediante una herramienta GIS esta persistencia relacional

obliga a realizar incómodas conversiones "batch" para traducir la tabla relacional a una cobertura en formato GIS, este proceso resta eficiencia y robustez al sistema.

En cuanto al problema de la baja integración entre aplicaciones considérese el siguiente ejemplo de funcionamiento interoperable: se está trabajando en el gestor de la B.D. relacional con la tabla de Puntos de Agua y se desea realizar una consulta de carácter geográfico compleja, sería interesante poder invocar el visualizador geográfico, realizar dicha consulta con las herramientas que nos proporciona e importar la selección al gestor de la B.D. relacional. Este funcionamiento interoperable no afecta a la fuente de datos, sino que involucra, por un lado, a una sesión concreta del gestor de la B.D. relacional y por otro a una sesión del visualizador geográfico; la *Figura 1* ilustra este funcionamiento.

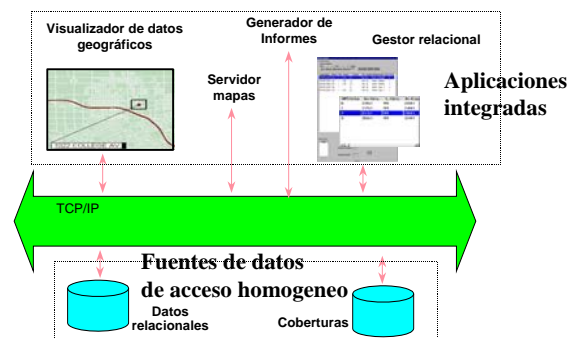


Figura 1. Funcionamiento interoperable

3. Una solución para alcanzar la interoperabilidad.

Esta claro que el problema que nos ocupa requiere un mecanismo de comunicación entre procesos en un entorno distribuido. Actualmente existen varias plataformas de procesamiento distribuido en el mercado, de entre las más relevantes podemos nombrar a DCOM de Microsoft, Java/RMI de SUN y CORBA de la OMG [KOCH-98], DCOM no está lo bastante maduro y RMI limita el desarrollo al mundo, por consiguiente el sistema se está construyendo sobre CORBA, este modelo nos facilita la especificación para un sistema distribuido orientado a objetos mediante una arquitectura cliente servidor, de

un modo independiente del sistema operativo, la plataforma y el lenguaje empleado. Una característica clave en CORBA es que se trata de un modelo de procesamiento distribuido orientado a objetos que permite el trabajo a nivel de componentes¹. En la actualidad se está desarrollando toda una programación orientada a componentes que se centra en los conceptos de encapsulación, extensibilidad y modularidad [Szy-97].

A continuación se presenta una parte de la solución actualmente en desarrollo al problema de interoperabilidad antes descrito. En ella se abordan ambos aspectos: el de homogeneidad de accesos a las fuentes, permitiendo a una herramienta GIS acceder a tablas relacionales, y parcialmente el de integración entre aplicaciones, concretamente posibilitando emplear en una aplicación selecciones realizadas en otra.

En el siguiente diagrama aparecen los distintos componentes y la interrelación entre los mismos, las interrelaciones se especifican mediante los interfaces que un componente exporta² y las referencias a otros interfaces que un componente utiliza

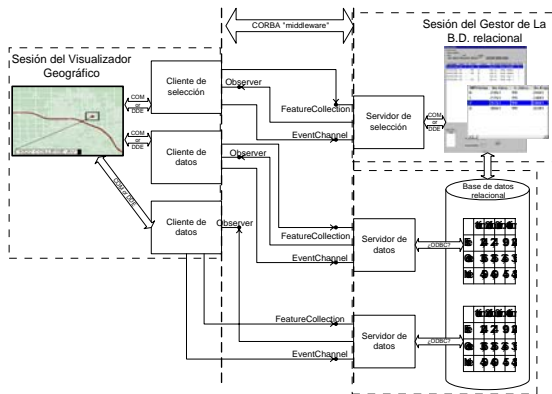


Figura 2: Arquitectura global

A continuación se describen brevemente los distintos componentes, sus responsabilidades y pequeñas notas sobre su implementación:

- El **Servidor de Datos** encapsula fuentes de datos externas, normalmente tablas relacionales

¹ Unidad binaria de producción, adquisición y distribución independiente, que interactúa para formar un sistema funcional

² Hace accesibles al bus CORBA.

a las que accede por ODBC u otro mecanismo similar (puede haber distintos componentes especializados en diferentes fuentes de datos), y las hace accesibles al bus CORBA exportando el interface **FeatureCollection**, interface descrito en OpenGIS para dar acceso a un conjunto de objetos de clase **Feature**, que más adelante se describe. Resaltar que el **Servidor de Datos** se comporta como un Adaptador o “Adaptee” [GHJV-95], esto es, encapsula un recurso que ni siquiera es Orientado a Objetos como es una fuente ODBC y le da el aspecto y comportamiento de un objeto CORBA.

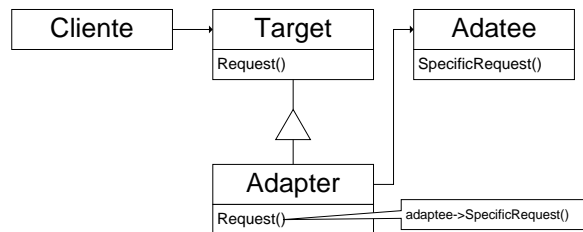


Figura 3: Patrón Adaptador

Otra responsabilidad del **Servidor de Datos** es la de notificar al **Visualizador** de los eventos por los que ha mostrado interés (cambios en los datos, adiciones, borrados, etc.), para ello exporta el interface **EventChannel**, este interface es parte del servicio de eventos de CORBA, este servicio se basa en una variante del patrón sujeto - observador [GHJV-95, BMHP-96] en el que uno o más observadores son notificados de cambios en el sujeto. En este caso, el componente **Cliente de Datos** hace las veces de observador para lo cual exporta el interface **Observer**.

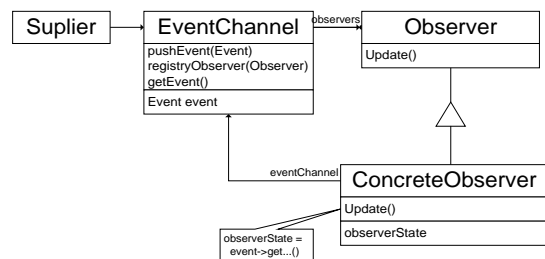


Figura 4: Patrón Event Channel

- El **Cliente de Datos** es el componente que permite a una aplicación tener acceso a los datos ofertados por el **Servidor de Datos**. Para ello establece dos comunicaciones: una local con la aplicación mediante un protocolo que la

aplicación soporte (ejm. DDE) y otra, mediante CORBA y posiblemente remota, con el **Servidor de Datos**. Este comportamiento se ajusta al patrón “bridge” [GHJV-95].

- El **Servidor de Selección** es un componente absolutamente similar al **Servidor de Datos**, ambos comparten los mismos interfaces (un mismo IDL en el modelo CORBA), sin embargo, la semántica del componente difiere y por lo tanto su implementación ya que el Servidor de Selección hace accesible al bus CORBA sólo el conjunto de **Features** seleccionadas en la aplicación a la que esta conectado.

- Análogamente, el **Cliente de Selección** hace uso de los servicios del **Servidor de Selección** para que la aplicación a la que esta conectado pueda emplear de esta selección con los fines que se consideren oportunos.

Ya se ha comentado que los servidores de datos y selección dan acceso a las fuentes que encapsulan en forma de una colección de objetos de la clase **Feature**, clase definida en el estándar OpenGIS (ver siguiente diagrama), hay que resaltar que la clase **Feature** puede contener cualquier tipo de *feature* (ver figura 5), esto es, con la misma clase de objetos podemos almacenar un punto de agua, una carretera, un pueblo o cualquier otro accidente geográfico esto nos aporta una gran potencia: dinámicamente se pueden añadir nuevas clases de *features* al modelo, para ello el **Servidor de Datos** interroga a la fuente externa sobre su estructura y construye dinámicamente un esquema que describe los atributos del nuevo *feature* y su semántica (mediante una clase OpenGIS denominada **FeatureType**), quedando así registrado en el sistema la nueva clase de *features*.

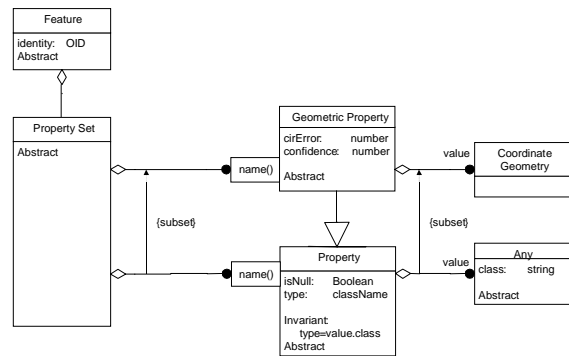


Figura 5: Estructura openGIS para la clase Feature.(Notación UML)

4. Otros trabajos y conclusiones.

Se han realizado diversas aproximaciones al problema del acceso a fuentes de datos geográficas heterogéneas, sin embargo, la gran mayoría son soluciones específicas a problemas concretos, una solución general y aceptada deberá esperar a que el estándar OpenGIS se asiente en la industria. Como ejemplo de este tipo de trabajos citar dos: la agencia americana NIMA (National Imagery and Mapping Agency), principal proveedor de geodatos del departamento de defensa americano, a desarrollado un enorme sistema con ciertas similitudes al nuestro en el que un servidor realizado en SmallTalk sirve de geodatos a clientes Java a través de internet, estos geodatos están modelados a través de una jerarquía denominada OVFP (Object Vector Format Product) desarrollada por la propia NIMA, al usar un modelo propietario y no estándar se limita el acceso al software desarrollado por NIMA o compatible con OVFP [COBB-98]. Otro esfuerzo análogo es GEO, drivers de acceso a repositorios de datos al estilo JDBC, pero específicos para acceso a fuentes geográficas, esta interesante solución, desarrollada en la universidad de Pretoria, está limitada al mundo Java, excepto tal vez empleando mecanismos como JNI (Java Native Interface), además, el modelo tampoco sigue ningún estándar, lo que obligaría a adaptar las aplicaciones específicamente a estos interfaces de acceso.

Con respecto a la integración entre aplicaciones hay muchas menos experiencias documentadas

al respecto, una particularmente conocida es OLE, tecnología de Microsoft sustentada sobre COM, que se centra en permitir a diversas aplicaciones cooperar para la edición de documentos complejos, OLE no está preparado para facilitar el intercambio dinámico de información geográfica. OpenGis está desarrollando un modelo de servicios, mediante el cual los componentes que lo cumplan podrán ofertar operaciones sobre Geodatos, esta parte del estándar está muy poco desarrollada y aún tardará en ser operativa.

Para finalizar una serie de conclusiones sobre los desarrollos presentados en este documento.

Un mismo mecanismo soluciona el problema de heterogeneidad de las fuentes de datos y el de integración entre aplicaciones.

Se evita replicar datos y se mantiene relativamente bajo el acoplamiento entre aplicaciones.

Aunque no implementa en su totalidad el estándar OpenGIS el diseño es perfectamente extensible, de modo que se puede ir añadiendo nuevas capacidades OpenGIS, por ejemplo se pueden añadir al sistema **Traders**, **Catalogs**, etc. y de este modo hacer los datos almacenados accesibles desde el exterior a otras comunidades de información OpenGIS.

Referencias

1. [GHJV-95] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides "Design Patterns" Addison-Wesley" 1996.
2. [BMHP-96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerland, Michael Stal "A system os Patterns" WILEY 1996
3. [Szy-97] Clemens Szypersky "Component Software" Addison-Wesley 1997
4. [SMB-96] David Schell, Lance McKee, and Kurt Buehler "Geodata interoperability – A Key NII Requirement" <http://www.opengis.org/articles/nii2000.html>. 1996
5. [OCG-OG] OCG "The OpenGIS Guide" 1996
6. [KOTT-95] Cliff Kottman "Geodata Interoperability: What Does it Mean for Business Geographics?" 1995
7. [MCKE-96] Lance McKee "Building the GSDI" 1996
8. [KOCH-98] Vivekanand P.Kochikar "The object powered web" IEEE Software 1998.
9. [COBB-98] Maria A. Cobb et al. IEEE Software Mayo/Junio 1998 pag. 22

Agradecimientos

Este trabajo ha parcialmente financiado por CICYT bajo el proyecto TIC98-0587 y la entidad CONSI+D de la Diputación General de Aragón.