

Incorporación de componentes de visualización SIG en entornos distribuidos con tecnologías COM/CORBA, aplicación a un sistema de monitorización de flotas

J. A. Bañares, P. Álvarez, R. López y P.R. Muro-Medrano

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3. 50014 Zaragoza

RESUMEN

De todas las capacidades SIG, las funcionalidades de visualización de información geográfica son las más requeridas en el desarrollo de sistemas de información que tratan con datos georeferenciados. Por su parte, las aplicaciones de seguimiento de flotas de vehículos son un ejemplo prototípico en el que, un desarrollo basado en componentes, permite abordar las necesidades de una alta reusabilidad, interoperatividad y escalabilidad del software desarrollado con el objeto de acelerar su proceso de desarrollo, y de evitar soluciones ad hoc. El presente trabajo presenta una aproximación tecnológica para este tipo de sistemas basada en una arquitectura de componentes distribuidos sobre CORBA. Los componentes que requieren la visualización de información geográfica se han desarrollado a partir de MapObjects, una librería de objetos ActiveX. El trabajo ilustra los aspectos tecnológicos de la integración de componentes SIG donde están involucradas distintas plataformas de distribución de objetos, así como adaptaciones requeridas, y posibilitadas por esta tecnología, para poder integrar componentes SIG comerciales.

1 Introducción

Los avances en los sistemas de seguimiento y control han revolucionado las posibilidades de los sistemas de Ayuda a la Explotación (SAE) de flotas de vehículos, posibilitando la recopilación de parámetros reales del funcionamiento de la explotación y ofreciendo información en tiempo real que permite el control de incidencias y una información valiosa a los usuarios de estas aplicaciones. Este avance se debe en gran medida al abaratamiento de los costes de las radiocomunicaciones y a la tecnología GPS [1] junto al avance de las tecnologías de la información.

La infraestructura necesaria para el desarrollo de aplicaciones de seguimiento y control de flotas requiere una serie de servicios básicos comunes (adquisición, análisis, visualización, comunicaciones, etc.) que exigen al equipo de desarrollo conocer e integrar un gran número de tecnologías, como tarjetas con microprocesadores, comunicaciones por radio, comunicaciones en una red de computadores, desarrollo de software de control, etc. Para disminuir el alto coste que suponen las soluciones "ad hoc" y dada la necesidad de integrar distintas tecnologías se ha buscado una alta reutilización, interoperatividad y escalabilidad, para lo cual se ha diseñado un sistema basado en componentes distribuidos que permite un alto grado de intercomunicación entre componentes que ofertan servicios básicos y aplicaciones más finalistas que los utilizan.

Junto con los componentes que resuelven las comunicaciones y la distribución de posiciones GPS, un aspecto fundamental del sistema de localización es la posibilidad de visualizar las localizaciones en tiempo real de los móviles sobre mapas o planos. El módulo de visualización desarrollado se conecta al componente GPS y muestra las posiciones recibidas en un entorno de sistema de información geográfica (SIG). Los componentes de comunicaciones por radio y de adquisición y distribución de posiciones GPS fueron realizadas en CORBA por ser un modelo estable para sistemas orientados a objeto distribuidos que permite abordar la heterogeneidad y el inevitable cambio de las tecnologías[2].

Para la implementación del componente de visualización, se optó por el uso de MapObjects, una librería de objetos ActiveX, rápidamente integrable en cualquier lenguaje de programación de propósito general (Visual C++, Visual Basic, ...) y que proporcionan a una aplicación todas las funcionalidades SIG de forma rápida y sencilla.

MapObjects proporciona una potente herramienta para integrar visualización de información geográfica en el desarrollo de aplicaciones. En la concepción de MapObjects, al igual que en todo sistema software, actúan varias fuerzas contrapuestas: por un lado el deseo de dotar al componente de la máxima funcionalidad, que aporte al usuario el mayor valor añadido con el mínimo tiempo de integración posible, y por otro lado la flexibilidad/generalidad, que permita cubrir el amplísimo espectro de necesidades de visualización de datos geográficos en la integración de sistemas de información sin sobrecargar el componente con funcionalidades que sólo una parte del mercado va a emplear. Cuando no consideramos el mercado globalmente sino que nos fijamos en un usuario concreto, el equilibrio entre estas dos fuerzas se desplaza, esto es, según el tipo de aplicaciones que el usuario desarrolle precisará un componente de visualización que presente unas determinadas características. Posiblemente no le importe que su componente de visualización pierda cierta generalidad siempre y cuando gane potencia y velocidad de integración en su campo de trabajo. Esta filosofía nos ha impulsado a elaborar un componente sobre MapObjects, que extiende sus funcionalidades para ajustarse a las necesidades de un amplio abanico de aplicaciones desarrolladas por nuestro grupo de trabajo en los que es preciso un componente de visualización SIG.

Esta filosofía de desarrollo basado en componentes se adopta por el hecho de que las aplicaciones SIG desarrolladas en nuestro grupo integran diversos subsistemas y fuentes de datos, algunos de los cuales han sido desarrollados previamente por nosotros o por agentes externos, siendo necesario que los distintos subsistemas interoperen entre sí. Ejemplos de este tipo de aplicaciones, además de las aplicaciones de seguimiento de móviles en tiempo real, son los sistemas de información en que sólo una parte de los datos tienen carácter geográfico, lo que permite su visualización en mapa o el acceso a la información utilizando fundamentalmente la localización través de la funcionalidad de un SIG. Estos sistemas necesitan un visualizador geográfico que, además de permitir las acciones típicas (apertura de capas, zoom, pan...), aportan capacidades extras como el acceso a fuentes de datos heterogéneas, capacidad de interoperar con aplicaciones externas y soporte a los grandes modelos de procesamiento distribuido. Los siguientes ejemplos ilustran estas necesidades:

1. Se quiere visualizar las rutas previstas para unos móviles, estas rutas son editadas por otro sistema, posiblemente "legacy" y almacenadas como secuencias de puntos en tablas de una base de datos relacional.
2. Se precisa visualizar la posición en tiempo real de un móvil, por motivos de diseño esta localización es accesible como un objeto remoto a través del estándar CORBA.
3. Una parte del sistema esta implementada haciendo uso de una herramienta de gestión de bases de datos relacionales (Access, Paradox o cualquier otra). Estas herramientas no tienen la posibilidad de realizar selecciones siguiendo criterios espaciales. Para realizar estas selecciones se desea invocar al visualizador geográfico, realizar en el mismo las selecciones que precisemos, ya sean con el ratón o con una herramienta de análisis espacial y exportar esta selección nuevamente al gestor de la base de datos relacional mediante COM.

Montar estas funcionalidades sobre MapObjects cada vez que se integra una aplicación es costoso, justificándose el desarrollo de un componente que extienda las capacidades de MapObjects para ajustarse a los nuevos requisitos. Este componente complementa MapObjects con nuevos *Automation Objects*. Estos nuevos objetos son expertos en diferentes tareas como por ejemplo la selección de elementos de una capa, la creación de capas a partir de fuentes de datos externas (ODBC, OpenGis sobre CORBA, etc.), la edición gráfica de *symbols*, *renderers* y consultas, la interacción con aplicaciones externas, etc.

Este trabajo presenta los patrones de diseño y las técnicas utilizadas para el desarrollo de un componente de visualización geográfico a partir de MapObject que permite rápidamente su integración y adaptación para que se ajuste a las fuentes de datos, entorno de aplicaciones y funcionalidades propias del sistema.

