

Desarrollo de una Arquitectura Distribuida Orientada a Objeto para la Monitorización Automática de Vehículos¹

P.J. Álvarez, J. Guillo, D. Infante, F.J. Zarazaga, P.R. Muro-Medrano.

Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, 50015. Zaragoza
{alvaper, jguillo, dinfante} @ebro.cps.unizar.es
{javy, prmuro} @posta.unizar.es
<http://diana.cps.unizar.es/iaaa>

Palabras Clave: Sistemas Distribuidos Orientados a Objeto, Componentes, CORBA, Monitorización Automática de Vehículos, GPS, GIS.

Resumen.

Este artículo presenta la experiencia en el desarrollo de software orientado a objeto basado en componentes sobre CORBA en una aplicación para la Monitorización Automática de Vehículos (AVM en inglés). El sistema está arquitecturado en base a componentes de radio (para establecer las comunicaciones con los vehículos) y localización (lleva control en tiempo real de la información enviada por los GPSs instalados en los vehículos) que, en cantidad variable, pueden estar distribuidos por una red de computadores. Por otra parte, componentes de visualización GIS (geographic information systems), utilizan dichos componentes para mostrar las localizaciones o realizar análisis de los recorridos realizados. Se pueden tener varias de estas fuentes de datos GPS distribuidas sobre una red de área local, la arquitectura basada en CORBA proporciona los mecanismos básicos de distribución e integración. El trabajo trata de aportar, desde la perspectiva de una aplicación real, nuestras conclusiones en el uso de componentes con el punto de vista de ingeniería del software.

1 Introducción.

La necesidad de optimizar el coste de los servicios, el incremento de la competencia en el sector de transportes y el interés de las instituciones públicas por promover el transporte público han llevado a muchas empresas a integrar sistemas de seguimiento y control de flotas en sus sistemas de información. Los sistemas de ayuda a la gestión de flotas que incorporan estas tecnologías permiten la recopilación de parámetros reales de funcionamiento. Ofrecen a los usuarios información en tiempo real que permite el control de incidencias, así como información útil para los usuarios finales del transporte público [1].

AVM (*Automatic Vehicle Monitoring*) es un tipo de aplicación que requiere la integración de diferentes tecnologías: sensores para la adquisición de datos en tiempo real, comunicaciones vía radio que proporcionan interacción entre diferentes

¹ **Agradecimientos:** Este trabajo ha sido parcialmente financiado por la *Comisión Interministerial de Ciencia y Tecnología* (CICYT) del gobierno español mediante del proyecto TIC98-0587 y el *CONSYD* del gobierno de Aragón mediante el proyecto P-18/96.

dispositivos, software de comunicación, sistemas de información geográfica (GIS), acceso a base de datos para actualizar regularmente los datos tiempo real, la integración con bases de datos no espaciales, etc [4]. Aunque estas aplicaciones requieran básicamente un conjunto común de servicios, el rápido crecimiento del mercado ha creado un incremento de la necesidad de sistemas en los que los requisitos de usuario son más específicos. El rango de aplicaciones finales se extiende desde un único computador que recibe localizaciones en tiempo real y simplemente las muestra sobre un mapa digital hasta una red de computadores configurada en diferentes áreas de trabajo. Las áreas de trabajo pueden incluir una zona de computadores para recibir datos en tiempo real, computadores que proporcionen la funcionalidad de un GIS completo, servidores que ofrecen información a los usuarios de Internet, bases de datos espaciales y no espaciales, etc.

No obstante, los requisitos y necesidades de cada cliente son muy diferentes en recursos y funcionalidad, por lo que se necesitan soluciones "ad hoc". El trabajo en este campo con nuevas tecnologías de software está en sus inicios [2]. Sin embargo, el coste de estas soluciones "ad hoc" puede ser asumible diseñando una arquitectura flexible que haga uso de nuevas tecnologías de software, como los sistemas distribuidos orientados a objeto, donde características como la interoperabilidad y la reutilización se tratan convenientemente. Las aplicaciones deben ser capaces de transformarse de manera inteligente reutilizando partes necesarias e incorporando nuevas tecnologías para aumentar sus capacidades.

Este artículo muestra las ideas básicas de OODISMAL; un sistema de información distribuido orientado a objeto para la localización automática de móviles. OODISMAL proporciona los componentes básicos para integrar las comunicaciones de radio, los datos de GPS adquiridos en tiempo real y su análisis con componentes GIS. Estos componentes pueden adaptarse fácilmente a cualquier tipo de radio o sensor que capture datos. En primer lugar, se explica la aproximación tecnológica para desarrollar la arquitectura de OODISMAL y sus componentes básicos. En la sección 3, el componente servidor de comunicaciones por radio que proporciona la funcionalidad de una radio trunking a todos los ordenadores de la red; el componente de adquisición de datos que utiliza el componente radio para conseguir las posiciones capturadas por distintos dispositivos GPS remotos; el componente de seguimiento para la localización real o estimada de cada vehículo de la flota que lleva incorporado un dispositivo GPS y el análisis *off-line* de las rutas recorridas; y el componente de visualización para desarrollar aplicaciones finales. Y por último, en la sección 4 se presentan las conclusiones.

2 Arquitectura de OODISMAL.

2.1 Aproximación tecnológica.

La tendencia actual de las aplicaciones cliente/servidor parece indicar que todas las aproximaciones existentes convergerán en un futuro próximo hacia los sistemas de objetos distribuidos [5],[7]. La utilización de esta tecnología está justificada por el deseo de construir software a partir de componentes reusables y la necesidad de integrar componentes que interoperen a través de la red.

El modelo de objetos favorece el trabajo cooperativo y la reutilización a través de la encapsulación. Sin embargo, se tiene la necesidad de hacer compatible las nuevas aplicaciones con los sistemas y aplicaciones anteriores. Si a esto añadimos la rapidez

con que evoluciona la tecnología, la variedad de sistemas operativos y lenguajes de programación, y la necesidad creciente de compartir recursos e información, el desarrollo y mantenimiento de estos sistemas es realmente costoso debido a su heterogeneidad. Esto promueve la obtención de una infraestructura común para el desarrollo de aplicaciones distribuidas orientadas a objeto. Como parte de esta arquitectura surge CORBA (*Common Object Request Broker Architecture*) [6], una arquitectura para la interacción entre las distintas aplicaciones en un entorno distribuido. Las aplicaciones son vistas como objetos, y sus servicios como la interfaz de los mismos, independientemente que estén o no programadas en un lenguaje orientado a objeto [8]. CORBA aporta un modelo estable para sistemas orientados a objeto distribuido que permite abordar la heterogeneidad y el inevitable cambio de la tecnología.

Los lenguajes de programación empleados para la codificación de los componentes básicos son C++ (radio, adquisición, simulación y visualización) y Java (seguimiento). El auge y la gran variedad de librerías disponibles actualmente de este último, han desplazado a C++ como lenguaje utilizado para el desarrollo de los mismos. La gran ventaja de utilizar CORBA es que proporciona transparencia en cuanto al lenguaje usado para codificar los respectivos clientes y servidores.

2.2 Diseño de la arquitectura.

OODISMAL (*Object Oriented Distributed Information System for Automatic Movil Location*) es un sistema de información compuesto por un conjunto de componentes distribuidos en una red de área local. Estos componentes interactúan entre si utilizando la infraestructura CORBA, y proporcionan los servicios básicos para la comunicación vía radio y la adquisición, el seguimiento, almacenamiento y procesamiento de datos de localización provenientes de móviles que incorporen un dispositivo GPS (ver figura 1). Para ello, se definen distintos objetos CORBA, a través de los cuales se realizará la comunicación que sea necesaria entre los componentes. De esta manera la interfaz entre los componentes queda definida claramente, lo que permite reducir el acoplamiento entre ellos. Cada componente funcionará como servidor de los objetos CORBA que contenga y actualizará como cliente de aquellos objetos CORBA que necesite de otros componentes. Un componente puede ser servidor y cliente de tantos tipos de objetos como se requiera, sin ningún tipo de restricción. Las responsabilidades de cada uno de ellos son las siguientes:

- El *componente de comunicación* proporciona servicios para establecer llamadas entre un equipo radio, conectado a un puerto del ordenador sobre el que se está ejecutando, y los instalados en los vehículos.
- El *componente de adquisición de datos* se comporta como un cliente del de comunicaciones para recibir datos de GPS remotos, y como un servidor que ofrece información de localización en tiempo real de los vehículos al bus CORBA.
- El *componente de simulación* permite al desarrollador depurar el sistema simulando la recepción de mensajes con información GPS. Tiene la misma interfaz que el de adquisición de datos. De esta forma, sus clientes no distinguen los datos reales de los datos simulados. La única diferencia es que los datos de localización simulados son tomados de una base de datos a través de un componente de persistencia.
- El *componente de visualización* ofrece recursos para representar posiciones y rutas sobre mapas digitales.

- El *componente de seguimiento* proporciona en cada momento la localización geográfica real o estimada en la que se encuentran los móviles que incorporan un dispositivo GPS y permite el análisis *off-line* de los recorridos descritos por los mismos.

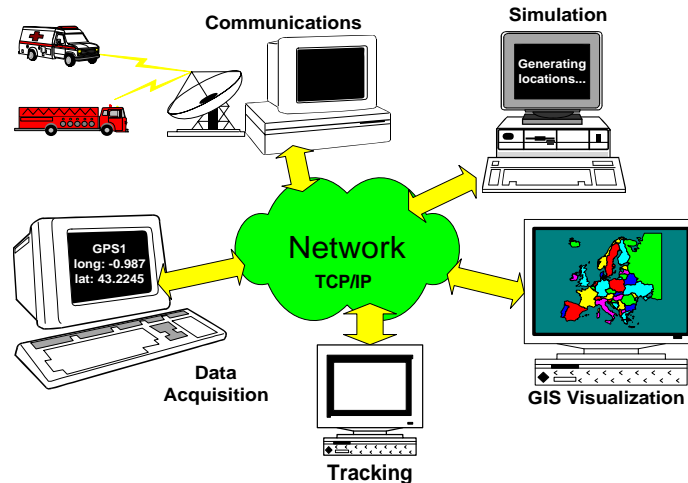


Figura 1 : Sistema Distribuido OODISMAL.

La funcionalidad de estos componentes es ofrecida al bus CORBA. De este modo, cualquier nueva aplicación cliente puede ser introducida en el sistema simplemente registrándose para los eventos producidos por los componentes. CORBA soporta la interoperabilidad, y su estilo cliente/servidor proporciona un fuerte soporte para la reutilización.

Un ejemplo de aplicación típica que integra estos servicios ofrecidos son los AVM, los cuales permiten visualizar en tiempo real sobre mapas digitalizados la localización de cada vehículo proporcionada por el componente de adquisición, el acceso a la radio de los vehículos monitorizados, o la representación gráfica sobre los propios mapas de los resultados del análisis de las rutas grabadas en la base de datos.

Además, los computadores en los cuales residen los componentes pueden ofrecer clientes ligeros Java que pueden ser “descargados” desde cualquier computador conectado a Internet. Esto permite al usuario la supervisión del comportamiento de los componentes. Por ejemplo, es posible acceder al componente de adquisición de datos y mostrar en una ventana los últimos datos recibidos desde cada GPS. Una versión sencilla de seguimiento ha sido desarrollada en Java permitiendo a los usuarios de Internet la visualización sobre mapas digitalizados de localizaciones en tiempo real.

3 Componentes básicos de aplicación.

3.1 Componente Radio.

El componente radio tiene la responsabilidad de manejar y hacer accesibles a múltiples clientes, las diferentes facilidades típicas de un equipo de radio ofreciendo, a través de una interfaz limitada y bien definida, las operaciones permitidas por éste. Entre éstas destaca la posibilidad de establecer diferentes tipos de llamadas usando los canales de comunicación del equipo, tales como el canal de voz, el canal de datos y el canal de control.

El flujo de información entre el componente y la radio física es de dos direcciones. Por un lado, un cliente puede solicitar el envío de un mensaje al *Clients Managing Module*

especificando la dirección destino y la información a enviar. Esta petición es recibida por el *Kernel Module*, quien construye la llamada para su posterior envío al dispositivo de radio a través del puerto COM utilizando un protocolo de comunicación con radio de bajo nivel. El *Low Level Module* implementa el acceso al puerto serie y el protocolo específico para cada tipo de radio. Es posible que en ciertas situaciones de ocupación intensa o desconexión del dispositivo destino no se pueda realizar la llamada de manera inmediata. Para resolver este problema, se ha incorporado en el *Kernel Module* un sistema de *Scheduling* configurable capaz de encolar llamadas no enviadas y proceder a reenvíos periódicos de las mismas (ver figura 2).

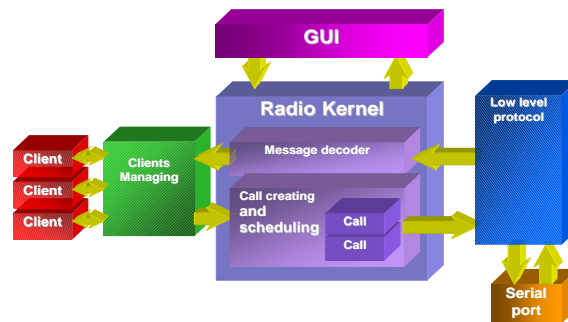


Figura 2 : Módulos y Operaciones del Componente Radio.

Por otro lado, si un cliente desea atender cualquier llamada recibida por el componente radio, debe registrarse en el mismo de acuerdo al comportamiento especificado en el patrón Sujeto-Observador[3]. Este mecanismo describe una relación de suscripción de uno a varios, de forma que cuando el componente cambia de estado, lo notifica a todos aquellos componentes a él suscritos.

Para permitir que los usuarios puedan monitorizar el comportamiento del componente, se proporciona un *GUI* básico. En él, sobre una pantalla de texto, se muestran los mensajes enviados y recibidos, así como otras informaciones operacionales relevantes (progreso de las llamadas enviadas y estado de los datos recibidos).

El diseño flexible del componente radio permite, a partir de un conjunto de servicios comunes, instanciarlo usando diferentes tipos de radios físicas, de forma que la funcionalidad ofrecida sea independiente del dispositivo. Los cambios a efectuar para una nueva instancia son mínimos y están perfectamente acotados en el *Low Level Module*. Esta experiencia está siendo “sufrida” por los miembros del área de comunicaciones, que se están enfrentando a la reprogramación del componente para el caso de una radio convencional (el componente ya puede trabajar con radios *trunking* y de 40Mhz).

Además, este componente ha sido implementado como un servidor CORBA, lo que posibilita el acceso a una misma radio física, conectada a un computador concreto, desde toda la red. Conviene reseñar que el único límite existente al número de radios conectadas a un mismo computador está en el número de puertos serie que éste disponga.

3.2 Componente de Adquisición de Datos.

El componente de adquisición de datos de GPS tiene la responsabilidad de ofrecer la localización de diversos vehículos, en los cuales ha sido instalada una radio con GPS incorporado. Para ello utiliza un mecanismo de comunicación en tiempo real, empleando el componente radio descrito previamente. Las llamadas de radio son transformadas en información sobre la localización, que consiste en la longitud, la

latitud y el instante de tiempo en el que se recibió dicha posición. Esta información puede ser empleada por el sistema para la visualización sobre mapas digitalizados de la posición de cada vehículo o para tareas de seguimiento de los mismos. Además, puede consultarse información adicional, incluyendo la velocidad, el rumbo o el estado de la señal GPS. El componente ha sido construido como un servidor CORBA lo cual permite que diversas aplicaciones puedan acceder al mismo con diferente propósitos.

Todo dispositivo GPS instalado en un vehículo que deba ser monitorizado tiene un objeto GPS correspondiente en el componente de adquisición (ver figura 3). A través de este objeto se puede acceder a toda la funcionalidad del dispositivo. Los dispositivos GPS pueden ser configurados para que emitan posiciones de manera periódica según un lapso de tiempo o una distancia. Pero además, se ha implementado un *Polling Process* para obtener nuevas localizaciones de cada GPS por encuesta, es decir, solicitándolas a cada uno por turno. De esta forma no se saturan las comunicaciones y se reducen las colisiones de llamadas de radio. Un cliente puede suscribirse a un objeto GPS para que cuando reciba una nueva localización sea notificado siguiendo el patrón Sujeto-Observador.

El *Message Manager* se emplea para construir y enviar cualquier mensaje que el componente necesite hacer llegar a los dispositivos GPS para configurarlos o solicitarles localizaciones. Para ello, solicita al *Coding/Decoding Module* que construya el mensaje y luego utiliza el componente radio para enviarlo al dispositivo adecuado.

Además, el componente ofrece otras características adicionales que le proporcionan una mayor funcionalidad: *Automatic Storage*, que almacena en una base de datos relacional todas las localizaciones recibidas por cada GPS, y *Stop Detection*, que activa un flag de estado asociado a cada GPS cuando detecta que éste no se ha movido durante un cierto tiempo.

Un *GIS Module* permite a un usuario externo monitorizar y configurar las diferentes características del componente a través de una interfaz gráfica, implementado como un cliente ligero Java que puede ser "*descargado*" desde cualquier computador conectado a Internet.

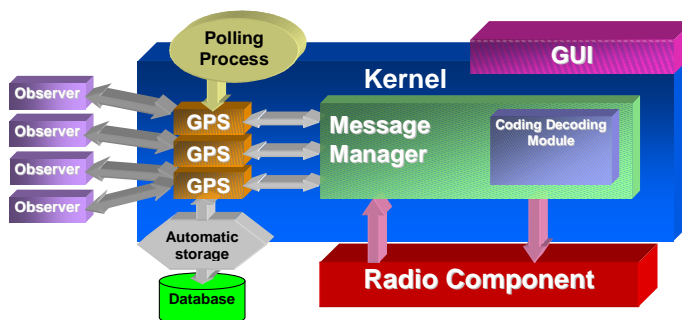


Figura 3: Módulos y Operaciones del Componente de Adquisición de Datos.

Como ayuda a la realización de pruebas y demostraciones de las aplicaciones que trabajan con el componente de adquisición, se ha desarrollado un componente de simulación que recupera posiciones de una base de datos y las ofrece al bus CORBA. Ambos componentes han sido diseñados con la misma interfaz de forma que puedan ser intercambiados sin afectar a sus clientes.

Por último, y como algo destacable pero dependiente del tipo de radio empleado, comentar que el diseño del componente permite el desarrollo de extensiones del mismo. Por ejemplo, junto con la información de localización se puede enviar información

sobre el estado de ciertos sensores o sistemas de entrada/salida conectados a la radio. Una de estas extensiones ha sido implementada para el tratamiento de entradas externas empleando un terminal de voz de tecnología *trunking* T500 con GPS incorporado, desarrollado por la empresa *Teltronic*.

3.3 Componente de visualización.

Este componente permite al usuario realizar operaciones típicas de un sistema GIS, visualizar en tiempo real los vehículos de una flota sobre uno o más mapas cartográficos, detectar informaciones provenientes de equipos de radio con entradas digitales a las que se les conecta sensores y representar sobre mapas rutas de vehículos previamente recorridas y almacenadas en base de datos.

En cuanto a capacidades propias de GIS, el componente muestra por pantalla un conjunto de mapas que cubren una superficie topográfica, aceptando, principalmente, dos tipos de cartografías digitales; imágenes digitalizadas y referenciadas (ficheros tiff, bip, ERDAS...) y mapas vectorizados que proporcionan una mayor definición de detalle, como por ejemplo los Shapefiles de ESRI². Permite al usuario elegir las propiedades gráficas de los mismos, como colores, tipos de línea, rellenos de figuras, etc. Sobre estos mapas se pueden realizar funciones GIS típicas como zooms y desplazamientos, siempre de una forma limitada y accesible para el usuario. Además, el componente tiene servicios específicos para crear puntos y moverlos sobre los mapas, facilitando el seguimiento en tiempo real de las localizaciones de los vehículos de la flota.

Tal como se ha indicado en el punto anterior, algunos equipos de radio tienen entradas digitales a las que se les puede conectar sensores físicos que pueden indicar diversas situaciones y estado, como la pulsación de un botón o la apertura de una puerta. Estas informaciones pueden ser detectadas por el componente, y el usuario puede asociar símbolos distintos según el tipo de entrada que se produzca, reflejando el estado de los vehículos de la flota gráficamente. Como en el caso del componente de adquisición esto es una especialización del componente GIS.

El componente de adquisición permite el almacenamiento en una base de datos del conjunto de localizaciones que describen el recorrido realizado por cada móvil. El usuario puede seleccionar qué porción de este recorrido desea visualizar, especificando la hora de inicio y de fin del trayecto a representar, así como las características gráficas de la misma: su tipo de línea, color y grosor.

Como aspectos destacables en el diseño del componente se puede reseñar que éste utiliza dos modos básicos de interacción con el resto de componentes, uno el típico método de invocación de los servicios que componen su interface y otro basado en eventos de acuerdo al patrón Sujeto-Observador.

3.4 Componente de Seguimiento.

La principal responsabilidad del componente de seguimiento, dado un conjunto de vehículos que interesa tener localizados, es la de poder proporcionar en cualquier momento la posición real o estimada en la que se encuentra cada uno de ellos. Para poder tener localizaciones reales, debe interactuar como cliente del componente de

² ESRI es una de las empresas con mayor implantación en el mundo de los GIS. Sus productos estrella son ARC/INFO y ARC/VIEW.

adquisición de datos, pero esto no siempre será posible por pérdidas de cobertura radio o GPS. En estos casos, él mismo deberá estimar en que posición se encuentra en cada instante de tiempo, siempre y cuando se conozca la ruta que tiene previsto recorrer el vehículo. Además, en ciertas aplicaciones de control de flotas, interesa almacenar en base de datos las rutas recorridas por los vehículos con el propósito de su posterior análisis. Con este fin, consta de un módulo de análisis *off-line* capaz de evaluar lo sucedido a lo largo del recorrido del móvil y de contrastarlo con lo planificado o esperado. Los resultados de este análisis pueden ser visualizados gráficamente sobre los mapas digitalizados por el componente de visualización.

Como se representa en la figura 4, a través del *Tracking Clients Managing* se especifican los vehículos a seguir y se accede individual o colectivamente a la información de seguimiento de los mismos. Cada uno de ellos está representado por un objeto *Tracking*, cuya responsabilidad es la de disponer en cada momento de información sobre la localización actual del móvil, la ruta real descrita a lo largo de su recorrido e incidencias y eventos surgidos a lo largo del mismo. Para poder disponer de toda ella, los objetos *Tracking* actúan como clientes de diversos componentes de adquisición; algunos ya desarrollados, como el de datos GPS, y otros en vías de desarrollo, como el de incidencias, alarmas y eventos.

Módulos para tareas concretas han sido diseñados como parte del componente de seguimiento. El *Improved Position Module*, dada una localización real y conocida la ruta que prevé recorrer el vehículo, ajusta la posición a dicha ruta mitigando el posible error introducido por el GPS. El *Analisis Module* aporta la funcionalidad necesaria para el estudio *off-line* de lo acontecido durante el recorrido descrito, útil por ejemplo para empresas de transporte de viajeros con el fin de tomar decisiones que mejoren el servicio (tramos conflictivos, identificación de horas punta, etc).

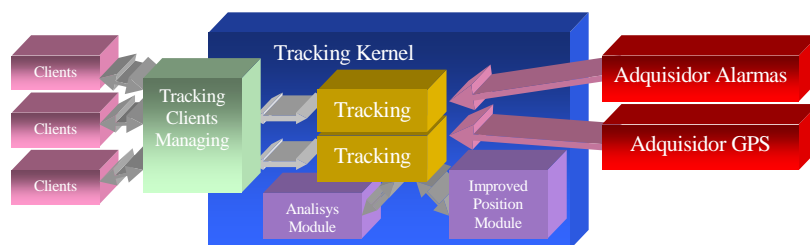


Figura 4: Módulos y Operaciones del Componente de Seguimiento.

El componente de seguimiento ha sido implementado como un servidor CORBA, ofreciendo información actualizada sobre las rutas seguidas por un conjunto de vehículos a todas aquellas aplicaciones interesadas que se ejecutan en computadores de la red de área local. Podemos encontrar desde sistemas para el control de flotas, hasta visualizadores para la gestión de llegadas y salidas de autobuses interurbanos, compartiendo una única instancia del citado componente.

Además, el diseño está abierto a una inmediata integración de nuevas fuentes de información y al incremento de la funcionalidad específica aplicable al seguimiento. Algunos ejemplos son el futuro componente de adquisición de alarmas y un módulo para la evaluación de las mismas respectivamente.

4 Conclusiones.

En este artículo, hemos presentado una arquitectura que proporciona componentes básicos para integrar la funcionalidad de los sistemas de información geográfica (GIS), con la de las telecomunicaciones con radio, y seguimiento de vehículos. Sobre el componente de radio es posible construir diferentes componentes para capturar datos, tales como el de adquisición de localizaciones de vehículos en tiempo real usando unidades GPS. Este último a su vez es empleado para el desarrollo de componentes más específicos dedicados al control de flotas, como es el caso del componente de seguimiento. La integración de las funcionalidades de estos componentes con las capacidades GIS ofrecidas por el de visualización, proporcionan un mecanismo para acceder a la información de la flota, manejarla y explotar sus capacidades al máximo. El resultado es una infraestructura flexible, que soporta las funcionalidades básicas de los sistemas de seguimiento y control de flotas, y que además posibilita su fácil integración. Esta serie de servicios comunes son proporcionados por los componentes previamente presentados.

Son varias las ventajas que ofrece la arquitectura propuesta:

- La ventana de visualización del GIS proporciona el marco perfecto para ofrecer la funcionalidad requerida por el sistema. De esta forma, la funcionalidad de otros componentes puede ser gráficamente integrada con un mapa digitalizado.
- Permite un desarrollo flexible. Todos los componentes pueden ser fácilmente intercambiados por nuevos componentes con funcionalidad mejoradas, o bien cohabitar con ellos. De esta forma es posible incrementar los servicios sin que las aplicaciones que los emplean se den cuenta de ello.
- Los componentes, debido a su diseño, pueden ser reusados en diversas aplicaciones. Por ejemplo, el componente de radio además de ser usado por el de adquisición de GPS para la recepción de información de localización, es reutilizado en proyectos relacionados con el desarrollo de centros de control integrados para la gestión y comunicación con distintos tipos de radios y de teléfonos.
- Las aplicaciones pueden ser configuradas de acuerdo a los requerimientos del cliente. No es necesario realizar ninguna modificación en los componentes para configurar diferentes aplicaciones. La arquitectura propuesta es escalable según los requerimientos solicitados.

CORBA ha sido empleado como infraestructura de red para proporcionar interoperabilidad y distribuir la funcionalidad. Cualquier aplicación puede buscar servidores en la LAN que proporcionen in interfaz CORBA predeterminado, y usarlos. Además, puede ser notificado de cualquier cambio de estado en los servidores simplemente registrándose en ellos para aquellos eventos que generan. Por ejemplo, los componentes de adquisición GPS y de simulación pueden ser intercambiados sin afectar a sus clientes; y el de visualización puede representar localizaciones sobre los mapas provenientes de diferentes componentes de adquisición.

En la actualidad, versiones de OODISMAL están prestando servicio en diferentes lugares de la geografía francesa: Toulouse, Ayuntamiento de Perigeux y los departamentos de Grenoble, Laon y Melon. En todos ellos, el sistema está dedicado al control de flotas de vehículos municipales.

Recientemente, ha comenzado el diseño de una nueva versión de OODISMAL, cuyo trabajo se enmarca dentro de un proyecto más ambicioso desarrollado para una empresa

dedicada al transporte interurbano de viajeros. Esta futura versión ofrecerá una más amplia gama de servicios en todos sus componentes básicos y se integrará con otras aplicaciones, más específicas dedicadas a la venta distribuida de billetes, a la gestión del sistema de información de la propia empresa, así como un sistema para la gestión de información horaria y de las próximas llegadas y salidas de autobuses.

Finalmente, en respuesta a la amplia aceptación social y comercial que experimenta Internet, recientemente se ha desarrollado un componente Web que ofrece clientes ligeros Java para la visualización en tiempo real de la localización de los vehículos de la flota en mapas digitalizados. Es una versión sencilla del componente de visualización que no incorpora ninguna de las funcionalidades complejas GIS previamente presentadas. Simplemente se trata de un *applet* que puede ser cargado por cualquier *browser* de Internet con un Java *plug-in*.

Hoy en día, OODISMAL es una realidad y sus componentes no sólo se emplean en el desarrollo de aplicaciones AVM, sino que su versatilidad permite su reutilización en otros campos de trabajo. El futuro alberga el desarrollo de nuevos componentes más especializados.

Referencias

- [1] B. De Saint-Laurent and K. Bourée. "Eurobus Project. Final Report". DRIVE PROGRAMME, Eurobus Project V2025.
- [2] Ted Foster and Liping Zhao. "Structuring the Network Model to Service More Functions". Report to assess the impact on Transmodel Version 4.1 of some of the change request raised within the Titan Data Model Management Grupo. 1998
- [3] E. Gamma, R. Helm, r. Johnson, J. Vlissides. *Desing Patterns. Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company. 1994.
- [4] E.J. Krakiwsky. "Tracking the Worldwide Development of IVHS Navigation Systems". *GPS World*, V4, N 10, pp. 40-47. October, 1993.
- [5] R. Orfail, D. Harkey and J. Edwards. *The Essential Client/server Survival Guide CORBA*. Wiley Computer Publishing. 1997.
- [6] Object Management Group. Object Management Architecture Guide, Version 3.0. Framingham, MA: Object Management Group, 1995.
- [7] C. Szyperski. *Componente Software. Beyond Object-Oriented Programming*. Addison-Wesley. 1997.
- [8] S. Vinoski. *CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments*. IEEE Communications Magazine, pp 46-55. Feb. 1997.