

Trabajo con Tecnologías de Componentes Distribuidos, experiencias del grupo de Sistemas de Información Avanzados

P.Álvarez, J.A.Bañares, P.Fernández, P.R.Muro-Medrano, J.Nogueras,
F.J.Zarazaga,

Grupo de Sistemas de Información Avanzados
Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
María de Luna 3, 50015. Zaragoza
alvaper,pedrofb,jnog@ebro.cps.unizar.es
banares,prmuro,javy@posta.unizar.es
<http://iaaa.cps.unizar.es>

Abstract. Desde hace más de cuatro años en el grupo IAAA venimos trabajando con tecnologías de componentes distribuidos como herramienta básica para llevar a cabo la integración de sistemas (algunos de ellos ya desarrollados), proponer y construir arquitecturas software flexible, y posibilitar un desarrollo de software que resulte económico o de bajo coste. En el presente trabajo se hace un repaso de los trabajos realizados y las experiencias adquiridas por el grupo.

Introducción

El grupo de Sistemas de Información Avanzados (IAAA) es un grupo de I+D con sede en el Centro Politécnico Superior de la Universidad de Zaragoza. El personal que trabaja a tiempo completo en el grupo está integrado por ingenieros y profesores adscritos al Departamento de Informática e Ingeniería de Sistemas de esta Universidad. El IAAA tiene personal desplazado en el Software Engineering Institute de la Universidad de Carnegie Mellon, donde colaboran en proyectos de COTS y proceso de software. Con dedicación parcial colabora también con el IAAA personal con formación multidisciplinar, ingeniería informática, industrial y de telecomunicaciones, matemáticas, geografía y filología.

Desde hace más de cuatro años, en el grupo de IAAA se viene trabajando con tecnologías de componentes distribuidos. En un principio, el trabajo se basó en el uso de CORBA desde diferentes lenguajes de programación, especialmente C++ y Java. La popularización de Java como lenguaje de programación dentro del trabajo del grupo promovió el uso RMI como tecnología de objetos distribuidos al resultar “más natural” su incorporación a sistemas desarrollados con este lenguaje.

Desde el principio, nuestro trabajo con las diferentes tecnologías de componentes distribuidos ha estado dirigido al aprovechamiento de la funcionalidad que las mismas proveen. Nosotros vemos estas tecnologías como unas potentes herramientas que nos

permiten atacar otros problemas que centran nuestras áreas de interés. Los tres campos de actuación en los que se trabaja con las distintas tecnologías son la integración de sistemas, la construcción de arquitecturas flexibles y el desarrollo de software económico o de bajo coste. Seguidamente realiza un repaso del trabajo que se ha llevado a cabo dentro de estos tres campos de actuación, y las experiencias que el grupo ha adquirido. En este repaso no se va a profundizar en los diferentes trabajos abordados discutiendo cómo se ha hecho uso de las distintas tecnologías de componentes y las ventajas y problemas surgidos, para ello se ofrecen referencias bibliográficas del propio grupo en las que se puede encontrar mayor nivel de detalle sobre los mismos, y en los que se estudian los trabajos que otros grupos han realizado y que están relacionados.

Integración de Sistemas

Una de las primeras utilidades que se le dio al uso de este tipo de tecnología de componentes distribuidos fue la integración de software desarrollado haciendo uso de diferentes lenguajes. Así resulta posible la utilización de aquél que sea más apropiado para cada una de las partes del problema. Por ejemplo, en [1] se muestra la utilización de CORBA para integrar módulos desarrollados en C++, Java y CLOS.

Otro de los campos de actuación en los que se trabaja es el desarrollo de servidores de una determinada tecnología (generalmente RMI o CORBA) que actúan como wrappers (o envoltorios) encargados de ofrecer el acceso a componentes en otra tecnología (generalmente COM o CORBA) que son productos comerciales accesibles mediante esta otra tecnología de componentes distribuidos. Uno de los primeros trabajos en este tipo consistió en la construcción de componentes de visualización geográfica utilizando tecnología ActiveX, concretamente el producto MapObjects de ESRI, embebidos en una arquitectura basada en componentes CORBA (ver [2] y [3]). Otro ejemplo de construcción de este tipo de envoltorios puede verse en [4] donde se presenta un wrapper RMI para el acceso a los servicios proporcionados por producto ArcInfo de ESRI a través de una interfaz COM suministrada con dicho producto. En ocasiones, este producto que se está “envolviendo” ha sido desarrollado dentro del grupo mediante una tecnología para proyectos anteriores, o por razones de facilidad de construcción. Así, por ejemplo, en [5] y [6] se presentan diferentes aspectos de un catálogo de información geográfica construido ajustado al estándar propuesto por un consorcio internacional denominado OpenGIS (<http://www.opengis.org>). El núcleo de este catálogo es un servidor RMI con un amplio abanico de servicios que es posible acceder mediante el modo de trabajo clásico de componentes distribuidos o, con mínimas modificaciones en el código fuente, incorporarlos a una única aplicación local. Sobre este núcleo se está construyendo un servidor encargado de proporcionar la interfaz CORBA ajustada a la especificación de estándar, y cuyo conjunto de servicios es más reducido que la del servidor RMI.

Finalmente, se han utilizado los componentes para acceder a aplicaciones independientes, que ofrecen unos sistemas de comunicación muy rudimentarios, para ofrecerlas al resto del sistema de información como componentes distribuidos. Por ejemplo, en [4] se muestra el acceso mediante DDE a aplicaciones independientes,

concretamente a la herramienta ArcView de ESRI, desde un servidor RMI que lo envuelve para poder ser utilizada desde el resto del sistema de información.

Arquitecturas flexibles

El uso de tecnologías de objetos distribuidos también ha permitido el llevar a cabo la construcción de arquitecturas con un alto grado de flexibilidad. En este sentido, quizás el trabajo más interesante que se ha abordado ha sido el desarrollo de OODISMAL (ver Fig. 1) una arquitectura distribuida orientada a objeto para la monitorización automática de vehículos.

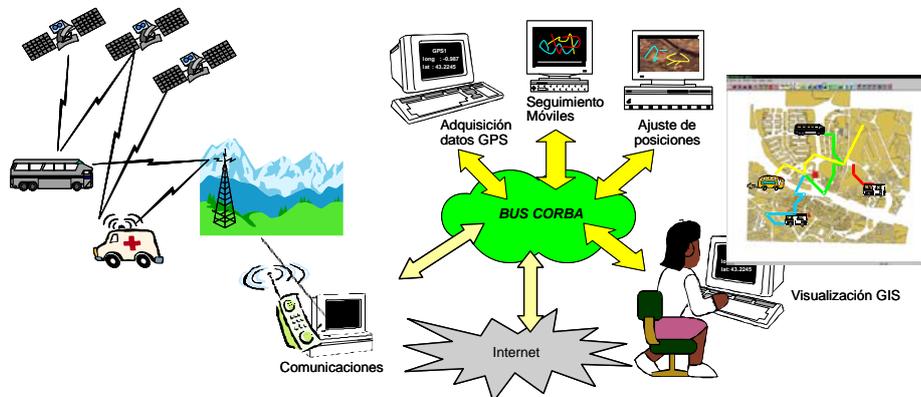


Fig. 1. Esquema simplificado de la arquitectura de OODISMAL

OODISMAL (Object Oriented Distributed Information System for Mobil Automatic Location) es, básicamente, un sistema de información integrado por un conjunto de componentes distribuidos sobre una red de área local. Estos ínter operan entre ellos usando la infraestructura CORBA y proporcionan los servicios para la comunicación vía radio, la adquisición, seguimiento, almacenamiento y procesamiento de datos de localización provenientes de móviles que incorporen un dispositivo GPS, y la visualización de la misma sobre mapas digitalizados. Esto se ha conseguido construyendo cada componente como un componente CORBA cuya interfaz es utilizada, actuando como cliente, por otros componentes o la aplicación principal. De esta forma se consigue una dualidad de comportamiento cliente/servidor CORBA de cada uno de ellos. Información más detallada sobre este sistema puede encontrarse en [7] y [8].

Desarrollo de software de bajo coste

Otra de las posibilidades que proporciona el manejo de componentes distribuidos es el diseño de arquitecturas que permiten reducir los costes finales de los sistemas de

información mediante el reaprovechamiento de componentes ya desarrollados o la encapsulación del acceso a productos comerciales con el fin de minimizar las licencias de run-time necesarias. Lo primero encaja con la filosofía propuesta por las arquitecturas COTS (Commercial Off-The-Shelf) cuya base tecnológica más representativa son los EJBs. En este campo ya hemos hecho pequeños desarrollos utilizando EJBs, y hemos comenzado a dar los primeros pasos para la construcción de un producto industrial basado en esta tecnología.

Respecto a la encapsulación del acceso a productos comerciales, posiblemente los ejemplos más representativos sean los wrappers de base de datos. Cuando no es necesario una funcionalidad muy compleja de la base de datos no es demasiado complicado realizar la construcción de envoltorios alrededor del motor de base de datos. Esto nos aporta dos posibilidades de ahorro. Una de ellas consiste en reducir la necesidad de múltiples licencias de sistemas gestores robustos (como por ejemplo Oracle) al canalizar todos los accesos por un único componente. Por otro lado, también resulta posible optar por la utilización de sistemas gestores muy básicos que no dan un adecuado manejo a los temas de concurrencia o acceso vía Web como puede ser el caso de Access. Ejemplos de desarrollos industriales de bajo coste con estas arquitecturas los podemos ver en [9] y [10]. También en estas líneas se integrarían algunos de los trabajos sobre la integración de componentes COM expuestos en el apartado de integración de sistemas

Otros campos de aplicación de tecnologías de objetos distribuidos

Al margen de las líneas ya explicadas, y que suponen los tres grandes ejes con los que trabajamos en nuestro grupo dentro de las tecnologías de componentes distribuidos, nos gustaría hacer mención de dos áreas más en las que estamos operando.

Por una parte, estamos estructurando todos nuestros desarrollos sobre la base de la utilización de componentes, generalmente servidores RMI, como herramienta para proporcionar funcionalidad en aplicaciones a diferentes niveles que se ejecutan en Internet/Intranet. Algunos ejemplos representativos son [11] y los ya nombrados [5] y [10].

La otra área es la explotación de los Java Beans como herramientas para la construcción de software reutilizable y que reduzca los tiempos de desarrollo. El ejemplo más notorio que podemos presentar es la exploración del uso de Java Beans para la visualización de información geográfica [12], que ha llevado al desarrollo de un producto comercial basado en esta tecnología [13].

Conclusiones

El uso de tecnologías de componentes, y las posibilidades de distribución que éstos ofrecen, aportan unas herramientas de gran flexibilidad a la hora de proceder a la construcción de sistemas de información industriales. No obstante, no es oro todo lo que reluce. Desde el punto de vista que nos ha dado nuestra experiencia dos son los

grandes problemas que vemos en el trabajo con estas tecnologías. Por un lado se tienen la gran cantidad de “estándares” que existen, cada uno de ellos con sus propias limitaciones y beneficios. Esto lleva a la necesidad de ir creando envoltorios que permitan el acceso a unos mediante la tecnología de otros.

El otro gran problema que vemos es la necesidad de que los diseños de los sistemas tengan presentes las diferencias que deben plantearse a la hora de realizar la construcción de los mismos según se utilicen componentes distribuidos o no. No se debe caer en el error de llevar a cabo la modelización de un sistema de información independientemente de su implementación distribuida y luego convertir las 50 clases del modelo en 50 servidores RMI, o 50 servidores CORBA, o 50 EJBs. La proliferación de componentes acarrea tres tipos de problemáticas operacionales: la ralentización del sistema al incrementarse las comunicaciones remotas, el desaprovechamiento de recursos hardware al convertir cada componente en un “ente con vida propia” ya que dos componentes necesitan más recursos que uno solo que sea la suma de los dos, y, por último, la problemática derivada de la necesidad de instalar y lanzar los 50 servidores. Afortunadamente, el patrón facade (ver 14) puede acudir en nuestra ayuda para proceder a la unificación de componentes de manera sencilla. En [15] se presentan una serie de lecciones aprendidas en el trabajo con componentes algunas de las cuales son perfectamente extrapolables al caso de los componentes distribuidos.

Referencias

1. M.A.Latre, J.A.Bañares, P.J.Álvarez, F.J.Zarazaga, F.J.Nogueras, P.R.Muro-Medrano. “Una experiencia en la programación de componentes CORBA en CLOS para el desarrollo de aplicaciones para la gestión y explotación de flotas de vehículos”. Research Report 99-13. Departamento de Informática e Ingeniería de Sistemas. Universidad de Zaragoza, 1999.
2. P.Álvarez, J.A.Bañares, R.López, F.J.Nogueras, P.R.Muro-Medrano. “Incorporación de componentes de visualización SIG en entornos distribuidos con tecnologías COM/CORBA, aplicación a un sistema de monitorización de flotas”. Taller sobre la aplicación de la ingeniería del software y las bases de datos a los sistemas de información geográfica. Aceptado para su publicación en las actas de las V Jornadas de Ingeniería del Software y las Bases de Datos. Valladolid, España, 8 - 10 Noviembre de 2000.
3. S.Comella, R.López, P.Fernández, J.Nogueras, J.Zarazaga, P.R.Muro-Medrano. “Utilización de tecnologías de componentes COM/CORBA para la integración de visualizadores interoperables sobre MapObjects”. Actas de la VIII Conferencia Nacional de Usuarios de ESRI. Madrid, España, 20-21 de octubre de 1999.
4. J.Nogueras, O.Cantán, J.M.Bobadilla, M.Casanovas, F.J.Zarazaga. “CatArcConnector, un componente para el acceso a servicios de catálogo distribuido compatibles con la OpenGIS Catalog Interface Specification desde ArcView y ArcInfo”. Aceptado para su publicación en las actas de la IX Conferencia Nacional de Usuarios de ESRI. Madrid, España, 25-26 Octubre 2000 .
5. F.J.Zarazaga, R.López, J.Nogueras, O.Cantán, P.Álvarez, P.R.Muro-Medrano. “First Steps to Set Up Java Components for the OpenGIS Catalog Services and its Software Infrastructure”. 3rd AGILE Conference on Geographic Information Science, Helsinki/Espoo, Finlandia, 25 – 27 Mayo de 2000.
6. F.J.Zarazaga, R.López, J.Nogueras, O.Cantán, P.Álvarez, P.R.Muro-Medrano. “Cataloguing and recovering distributed geospatial data, a Java approach to build the OpenGIS Catalog

- Services". EC-GIS 2000, 6th European Commission GI and GIS Workshop. Lyon, Francia, 28 – 30 Junio de 2000.
7. F.J.Zarazaga, P.Álvarez, R.López, J.Nogueras, J.Valiño, P.R.Muro-Medrano. "Use Cases of vehicle location systems using CORBA based distributed real-time GPS data and services". Computers, Environment and Urban Systems. Aceptado para su publicación.
 8. P.Muro-Medrano, D.Infante, J.Guillo, J.Zarazaga, J.Bañares. "A CORBA Infrastructure to Provide Distributed GPS Data in Real Time to GIS Applications". Computers, Environment and Urban Systems. Vol. 23, 1999, pp 271 – 285.
 9. M.A.Latre, P.Fernández, R.López, J.Valiño, P.Muro-Medrano. "Integración de funcionalidad SIG en el software de Gestión de proyectos LEADER, interesantes posibilidades a bajo coste". Actas IX Conferencia Nacional de Usuarios de ESRI. Madrid, España, 25-26 Octubre 2000.
 10. M.A.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". Proceedings of the Second International Symposium on Telegeoprocessing, pp 93 -- 101. Sophia-Antipolis, Francia, 10 – 12 Mayo de 2000.
 11. 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. Lyon, Francia, 28 – 30 Junio de 2000.
 12. J.Valiño, R.Béjar, J.A.Bañares, P.Fernández. "Integración de visualización SIG en aplicaciones de sistemas de información a partir de JavaBeans". Taller sobre la aplicación de la ingeniería del software y las bases de datos a los sistemas de información geográfica. Actas de las V Jornadas de Ingeniería del Software y las Bases de Datos. Valladolid, España, 8 - 10 Noviembre de 2000.
 13. R.Béjar, P.Fernández, P.Álvarez, J.Valiño, P.R.Muro-Medrano. "JGISView: JavaBeans para la integración de utilidades de visualización SIG en aplicaciones y applets". Taller sobre la aplicación de la ingeniería del software y las bases de datos a los sistemas de información geográfica. Actas de las V Jornadas de Ingeniería del Software y las Bases de Datos. Valladolid, España, 8 - 10 Noviembre de 2000.
 14. E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley 1995.
 15. M. Sparling. "Lessons Learned through Six Years of Component-based Development" Communications of the ACM, vol. 43, n. 10, 2000 pp: 39 - 46