

# Plataforma para el Desarrollo de Servicios en el Ámbito de la Telemática de a Bordo en Vehículos

José Santa, Antonio F. G. Skarmeta, Benito Úbeda  
Departamento de Ingeniería de la Información y las Comunicaciones  
Facultad de Informática  
Universidad de Murcia  
Campus de Espinardo, 30071 Murcia, España  
Email: josesanta@dif.um.es|skarmeta@dif.um.es|bubeda@um.es

## Resumen

**Abstract** *Because onboard services are becoming the cornerstone in vehicle equipment, manufacturers are working in attractive and useful solutions to be included in their cars. However, an important drawback of this development arises in the proliferation of devices in the driver compartment and the lack of a shared communication interface with the exterior. Due to these reasons, our work has been directed to develop an open service platform for vehicles with communications capabilities. An embedded computer is used as the on board unit (OBU), and a multiplatform software architecture has been designed and deployed on it to let the implementation of onboard services. The navigation sensors installed in our prototype vehicle are used to implement location based services, and several network devices cover all connectivity requirements. Cellular networks are used in a peer to peer (P2P) based network architecture valid for communications among vehicles and between the car and the road infrastructure. In addition, several services have been implemented and tested to probe the feasibility of the whole system.*

## 1. Introducción

Debido al creciente interés que la sociedad actual tiene en las nuevas tecnologías, nuevos productos en los campos de la información y las comunicaciones están surgiendo en entornos hasta el momento inexplorados. De esta manera, los vehículos se presentan como un marco perfecto para la implantación de una gran cantidad de funcionalidades tradicionalmente disponibles en lugares como el trabajo y el hogar [1]. Sin embargo, esta expansión requiere de un soporte hardware y software adaptado para las necesidades del mercado y del usuario.

Actualmente, los servicios que el conductor y los pasajeros pueden usar en un vehículo requieren de un gran despliegue hardware. Cada nueva funcionalidad se im-

plementa como un nuevo dispositivo que tiene que ser instalado en el habitáculo. Esta estrategia de despliegue no se presenta en absoluto escalable de cara a un aumento en los servicios de a bordo. Desarrollar cada nuevo servicio como software ejecutable en una unidad de a bordo (OBU) basada en un ordenador de propósito general presenta varias ventajas [2, 3]. El modelo de negocio sufre un cambio radical, debido a que la actualización de servicios y la instalación inicial no requieren grandes inversiones en hardware.

Las comunicaciones también son especialmente importantes en el mundo actual, y el vehículo está siendo partícipe de este hecho en los últimos tiempos. Así, la investigación se ha centrado durante los últimos años en dos necesidades de conexión en el mundo del vehículo: comunicaciones con la infraestructura y, sobre todo, comunicaciones entre vehículos. Las redes celulares han sido la tecnología preferida en el ámbito de las comunicaciones con el lado de la carretera, mientras que las redes ad-hoc lo han sido para las comunicaciones intervehiculares [4]. El problema existente actualmente radica en la falta de soluciones que combinen ambos requerimientos.

Acorde con todo esto, nuestro trabajo ha estado centrado en el diseño y desarrollo de una arquitectura extensible para servicios basada en un ordenador embebido de propósito general, adecuado con capacidades de comunicación a nivel de servicio. Nuestros trabajos iniciales sobre un framework de programación para servicios basado en capas [5], que promueven la reusabilidad y el desarrollo modular, han sido extendidos con capacidades de comunicación. Una aproximación peer to peer (P2P), aplicada con éxito en vehículos [6], facilita las tareas de comunicación vehículo a vehículo (V2V) y vehículo a infraestructura (V2I). OSGi (*Open Services Gateway initiative*) es usado para la definición de una arquitectura software en donde se incluye un novedoso módulo de comunicaciones que abstrae al programador de servicios de los detalles sobre la red.

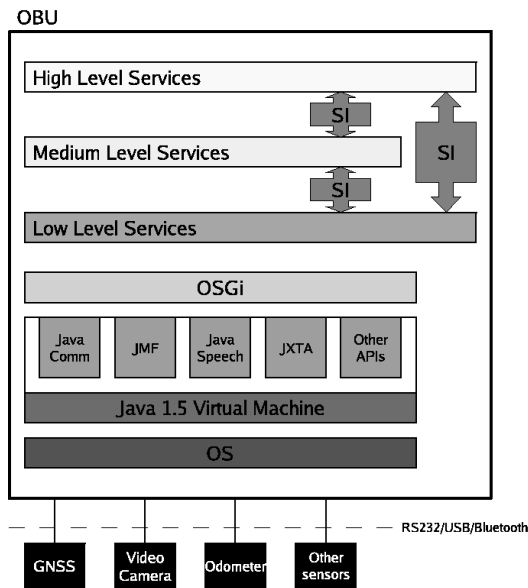


Figura 1: Arquitectura de servicios en la OBU

## 2. Una Arquitectura Multiplataforma Basada en OSGi para el Desarrollo de Servicios

Tal y como se ha dicho, OSGi [7] es usado como contenedor de servicios en el sistema que se propone. Además de ser multiplataforma, permite el despliegue modular de servicios, lo cual hace que se puedan realizar actualizaciones del software de a bordo “en caliente”.

La Fig. 1 ilustra el sistema diseñado para crear servicios sobre el OBU. Todos los sensores incluidos en el vehículo están conectados al ordenador de a bordo mediante enlaces RS232, USB o Bluetooth. El OBU, en su concepción básica, es un PC con un sistema operativo que no está prefijado, gracias a que el software está basado en Java. Sobre la Máquina Virtual Java se sitúan varias APIs de programación que facilitan la implementación de una gran variedad de servicios. Java Speech, por ejemplo, provee de un sintetizador de voz. OSGi, por su parte, está localizado sobre la base Java. Éste actúa como contenedor de los servicios desarrollados. Estos están clasificados de acuerdo con su nivel de abstracción. Los servicios de bajo nivel son, principalmente, software de acceso a los dispositivos físicos del vehículo. Los servicios de nivel medio actúan como middleware entre los servicios de bajo y alto nivel, por lo que aquí se llevan a cabo tareas de transformación y adaptación de información. Finalmente, los servicios de alto nivel son aplicaciones finales con interfaz grá-

fica. Toda esta estructura jerárquica de servicios tiene un doble propósito. Además de la programación modular, se resuelven los problemas de acceso a dispositivos cuando hay problemas de sincronización. Este hecho es realmente importante en sensores ampliamente usados, como el GNSS, por ejemplo.

Las comunicaciones entre capas se llevan a cabo por interfaces de servicio (SI). Cada capa define unas interfaces que indican la funcionalidad disponible. Una SI es, de hecho, un interfaz Java que puede ser implementado por uno o más servicios OSGi, con el objetivo de ofrecer la funcionalidad que especifican. Los servicios de las capas superiores lanzan peticiones al framework OSGi con la SI como parámetro para obtener los servicios disponibles.

## 3. Comunicaciones Vehiculares a Nivel de Servicio

Tal y como se ha descrito antes, las redes celulares mediante GPRS/UMTS tienen un especial interés en este trabajo. Usando una aproximación P2P sobre este tipo de redes, el vehículo puede enviar y recibir información contextual sobre su entorno. La Fig. 2 muestra un diagrama general de la arquitectura de comunicaciones ideada. Las zonas de tráfico están organizadas en áreas de cobertura, cada una usando un grupo de comunicación P2P. Los vehículos pueden moverse desde un grupo P2P a otro, cambiando de zona de cobertura a través de un proceso de roaming. Este proceso está basado en la localización del vehículo, extraída del sensor GNSS. La información sobre áreas es recibida desde la entidad Group Server usando un enlace TCP/IP sobre GPRS/UMTS. Un elemento local a cada zona de cobertura, llamado Environment Server, gestiona los eventos especiales dentro del área. Las notificaciones de eventos son enviadas y recibidas por las implementaciones de servicios localizadas en el vehículo y en el lado de la carretera (Environment Servers). Todos los mensajes emitidos son encapsulados en paquetes P2P. Además, dos técnicas diferentes de emisión han sido desarrolladas, por lo que un mensaje P2P puede ser emitido en modo broadcast en el área, o enviado a un vehículo en concreto.

En la Fig. 2 se pueden observar también los tres escenarios más significativos. De izquierda a derecha, el primero muestra el proceso de roaming. Un vehículo pasa de un área a otra. Group Server provee de los parámetros P2P a utilizar en el siguiente área, para mantener la comunicación en todos los servicios activos. Para ahorrar recursos, el vehículo solamente se comunica con Group Server cuando detecta que ha salido

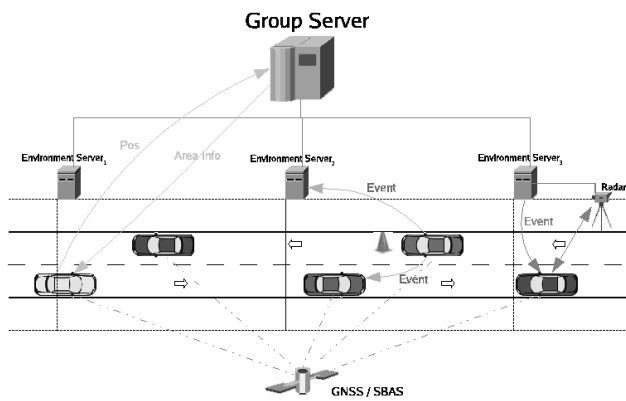


Figura 2: Arquitectura de comunicación basada en P2P

de la zona de cobertura actual. En el segundo escenario, un vehículo envía en broadcast al área un aviso de obras. Puesto que la entidad Environment Server recibe todos los mensajes lanzados, puede procesar cualquier evento que requiera un especial tratamiento, como un accidente, por ejemplo. Los mensajes de servicios críticos, como accidentes, son propagados por Environment Server a las áreas adyacentes, para que los conductores sean conscientes del problema con antelación. El último escenario del diagrama muestra cómo Environment Server es la entidad encargada de conectarse con el resto de los dispositivos al lado de la carretera, como sensores de identificación. Gracias a ésto, los sucesos locales al área pueden ser notificados a los vehículos de la zona y, además, ser enviados a una entidad central.

## 4. Detalles del Prototipo

La plataforma descrita ha sido implementada y testada sobre un sistema real. Además, varios servicios han sido desarrollados con el objetivo de mostrar la viabilidad de la solución propuesta.

### 4.1. Detalles sobre el Hardware

El hardware base usado consta de un prototipo de vehículo ampliamente sensorizado, usado en la Universidad de Murcia [8] en varios proyectos de investigación. Entre los sensores instalados, son de especial relevancia los captadores odométricos para detectar movimiento, y el receptor GNSS para obtener la posición del móvil. El vehículo prototipo se muestra en la Fig. 3. Éste incluye un SBC (*Single Board Computer*) con el sistema operativo Linux Fedora Core 4 y la máquina virtual de Java 1.5. Oscar ha sido la implementación OSGi usada [9]. El vehículo está provisto también de hardware de red Bluetooth, WiFi, y GPRS/UMTS.



Figura 3: Vehículo prototipo usado en los desarrollos

En lo que respecta a la implementación de la arquitectura de comunicaciones descrita anteriormente, la parte concerniente al vehículo ha sido desarrollada en el equipamiento descrito, mientras que las entidades encargadas de la gestión local de eventos (Environment Server) y la que dispone de la información de roaming (Group Server) se han desplegado en servidores PC con sistema operativo Linux Fedora Core 4.

### 4.2. Servicios Implementados

Los servicios que se implementan en la arquitectura de tres capas descrita tienen características especiales en su archivo JAR, distinguiéndose del resto de los instalados en el framework OSGi. Tal y como se describe en [5], diversos servicios han sido implementados en cada una de las capas. En la primera de ellas se ha incluido funcionalidad de acceso al hardware del vehículo, principalmente. En la segunda de las capas se incluyen servicios que actúan como middleware para la capa superior, en la que se incluyen las aplicaciones finales.

Instalado en la segunda capa, se ha creado un servicio llamado *JXTA Communications*. Éste contiene la implementación del middleware de comunicaciones para el vehículo. *JXTA (JuXTApose)* [10] se usa como tecnología P2P para crear el sistema de paso de mensajes basado en grupos P2P descrito. Además, un servicio de alto nivel llamado *Message Console* presenta una aplicación que hace uso de las capacidades de comunicación para notificar alertas en carretera. En la Fig. 4 se observa una captura de pantalla de ésta. El usuario puede suscribirse a diversos servicios de alerta para recibir y poder enviar avisos. La activación de dichos servicios dependerá de la disponibilidad del área de cobertura por la que se circula. La captura muestra-



Figura 4: Message Console service

da corresponde a una de las pruebas llevadas a cabo sobre un circuito a lo largo de la autovía A7 (próxima a la Universidad de Murcia).

## 5. Conclusiones

A lo largo del artículo se ha presentado una arquitectura modular y extensible para la creación de servicios de a bordo. Esta plataforma ha sido concebida para un ordenador de propósito general. El prototipo desarrollado muestra cómo un SBC es usado como aproximación a una OBU embebida. Un vehículo sensorizado se ha usado en dicho prototipo, lo cual permite la creación de servicios dependientes del contexto. Se presenta igualmente un sistema de comunicaciones que auna los requerimientos de conectividad V2V y V2I. Este diseño está basado en la idea de grupos de comunicación P2P para crear áreas de conectividad donde los servicios puedan intercambiar mensajes de una manera sencilla. Esta plataforma ha sido desarrollada y testeada en entornos de circulación real. Los trabajos actuales en esta línea se centran en la provisión de información relativa al contexto, inferida a través del perfil del conductor. También se están realizando estudios de rendimiento en el paso de mensajes con la infraestructura presentada, con el objetivo de estudiar la aplicabilidad de nuestro diseño en sistemas de seguridad.

## 6. Agradecimientos

Los autores desean agradecer al Ministerio de Educación y Ciencia su ayuda en las labores de investigación, a través de la beca AP2005-1437, en el marco del programa FPU, y a la Agencia Espacial Europea, bajo el proyecto GIROADS 332599. Agradecimientos también al Ministerio de Fomento por su continuo apoyo

en las labores de investigación ITS.

## Referencias

- [1] Craig Simonds. "Software for the Next-Generation Automobile". *IEEE IT Professional*, vol. 5, no. 6, pp. 7-11. November 2003.
- [2] E. C. Nelson, K. V. Prasad, V. Rasin, C. J. Simonds. "An embedded architectural framework for interaction between automobiles and consumer devices". *10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'04)*. Mayo 2004.
- [3] Zoltán Benedek. "A Framework Built in .NET for Embedded and Mobile Navigation Systems". *2nd International Workshop on .NET Technologies*, Czech Republic. Mayo 2004.
- [4] C. Sotomayor, R. Toledo, A.F. Skarmeta. "CASHI: Sistema de Evitación de Colisiones en Autovías". *VI Congreso Español sobre Sistemas de Transporte Inteligente*, Vigo. Octubre 2006.
- [5] José Santa, Benito Úbeda, Antonio F.G. Skarmeta. "A Multiplatform OSGi Based Architecture for Developing Road Vehicle Services". *Consumer Communications & Networking Conference 2007 (CCNC 2007)*, Las Vegas. Enero 2007.
- [6] Luciano Baresi, Carlo Ghezzi, Antonio Miele, Matteo Miraz, Andrea Naggi, Filippo Pacifici. "Hybrid service-oriented architectures: a case-study in the automotive domain". *5th International Workshop on Software Engineering and Middleware (SEM'05)*, Lisbon. Septiembre 2005.
- [7] OSGi Alliance. OSGi web site. <http://www.osgi.org>
- [8] Skarmeta A.G., Martínez H., Zamora M.A., Úbeda B., Gómez F.C, Tomás L.M. "MIMICS: Exploiting Satellite Technology for an Autonomous Convoy". *IEEE Intelligent Systems*. vol. 17, no. IV, pp. 85-89. Julio 2002.
- [9] Oscar OSGi web site. <http://oscar.objectweb.org>
- [10] Sun Microsystems. 'JXTA Technology: Creating Connected Communities'. January 2004.