

ARCHITECTURE OF THE INTERNET REAL TIME COMMUNICATION WITH THE VEHICLE ELECTRONIC CONTROL UNITS SYSTEM

Georg-Otto WAGNER,
Continental Automotive GmbH
Philipsstr. 1, 35576 Wetzlar, Germany
and PhD at University of Pitești, Romania
Faculty of Electronics Communications and Computers
georg.otto.wagner@continental-corporation.com

Keywords: remotely real-time communication, electronic control units (ECU), Controller Area Network (CAN), Media Oriented Systems Transport (MOST), Open Services Gateway initiative (OSGi)

Abstract: This article presents an architecture of the Internet real-time communication with the vehicle electronic control units system. In order to communicate remotely with electronic control units connected to the MOST and CAN buses inside the vehicle, an OSGi-based architecture was proposed, whereby the MOST protocol was used for the internal client-server communication protocol.

1. INTRODUCTION

A whole range of electronic functions, such as navigation, adaptive control, traffic information, traction control, stabilization control, and active safety systems, are implemented in today's vehicles. Many of these new functions are not stand-alone in the sense that they need to exchange information—and sometimes with stringent time constraints—with other functions. For example, the vehicle speed estimated by the engine controller or by wheel rotation sensors needs to be known in order to adapt the steering effort, to control the suspension, or simply to choose the right wiper speed. The complexity of the embedded architecture is continually increasing.

Nowadays, up to 2500 signals are exchanged through up to 70 electronic control units (ECUs) on different types of networks. [1]. The main bus systems and protocols used for interconnecting the ECUs and hence to serve to the real-time communications in vehicles are CAN (Controller Area Network) and MOST (Media Oriented Systems Transport). These CAN and MOST ECUs can be mostly diagnosed and maintained through

connecting directly to the CAN or MOST busses. But it would be interesting to propose a system which can establish a real-time communication from the outside world (e.g. Internet) to the ECUs connected to the CAN bus or MOST bus. Therefore in this paper is proposed architecture of the Internet real-time communication with the vehicle electronic control units system.

2. OSGI AS REAL-TIME COMMUNICATION FRAMEWORK

OSGi[4] appeared on the market in 1999 to define specifications for the providing of services to enable local network interconnection devices within vehicles, home or other environments.

OSGi specification attempts to standardize how to provide secure and reliable service to their remote management, reuse of services and interconnection of different network standards.

The OSGi specifications define a platform that provides a framework for

implementing the various services. The framework includes a Java framework for extended service life-cycle management, persistent data storage, version management and a service registry. The life-cycle management provided by the framework allows developers to divide the running applications in small parts, providing mutual services, called bundle. These software components implemented in Java can be loaded, installed and activated in a framework. When activated, the bundle can register services in the registry service to provide services to other bundles in the framework. When a service is no longer needed anymore, it can be removed from the framework without to enable other bundles.

The bundle can be installed and updated by the framework in a dynamic and accessible way. The new bundle can be installed for added features and the bundle existing can be modified and updated without having to reboot the whole system. The services registration is used to find and use other services in a secure and controlled manner. The service gateway operator has complete control over the platform and it can decide which services are allowed to be used. Therefore due to its capabilities, in this paper it was used the OSGi platform as real-time communication framework.

3. VIRTUAL “MOST” USED AS COMMUNICATION PROTOCOL FOR MANAGING THE ELECTRONIC CONTROL UNITS INSIDE VEHICLES

MOST is a function oriented high-speed multimedia technology to network a variety of devices (namely the MOST nodes). MOST defines mechanisms for sending streaming data and packet-based data, and provides a complete application framework to control interaction between devices in a clearly structured way [5].

The MOST specification defines not only the lower layers of a MOST network, which provide the basics for the transmission of data and for the network management, but

also the protocols and mechanisms for implementing applications on top of those [6]. This protocol determines how the services offered by an MOST application can be accessed, describing the data format for the messages, the operational sequences and conditions for read/write acces to the properties and methods of an FBlock, and the notification mechanism [8].

In the MOST specification, a MOST device contains multiple components that are the interface of an application and are called function blocks (FBlock) [2].

It is also possible that there are multiple FBlocks in a single MOST device connected to the MOST network via a common MOST Network Interface Controller. Between the FBlocks and the MOST Network Interface Controller, the Network Service forms an intermediate layer providing routines to simplify the handling of the MOST Network Interface Controller.

Figure 1 illustrates the model of a MOST device as defined in the MOST specification. On the hardware level, a device has acces to the physical transmission layer via the MOST Network Interface Controller (NIC).

The Network Service implements a driver layer that controls the access to the interface controller chip and enables applications to carry out basic functions such as sending and receiving messages [6].

Each MOST device implements the Fblock NetBlock, which is required for administrative tasks within the MOST system. Additionally, each MOST device can implement one or more FBlocks for the functionalities of its corresponding applications. According to the MOST specification, an MOST FBlock, whose functions are used and controlled by an application, is referred to as a Slave [2].

In the figure 2 it is shown how the application, which uses the FBlock and is referred to as its Controller, is setting/reading the properties of the FBlock, is receiving the notification of changes sent by the FBlock, or is calling the methods implemented by the controlled FBlock[6].

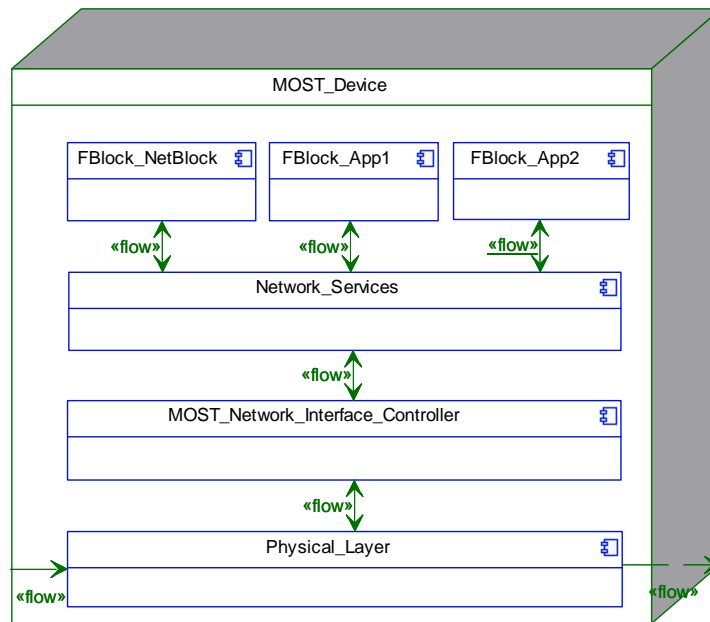


Fig. 1 Model of MOST devices

It should be done a differentiation between two types of functions implemented by the FBlock [2]:

- Properties that describe a specific attribute of the FBlock (Slave) and its values can be read by the Application (Controller) and, if this is supported by the respective properties, modify them. Additionally, a notification mechanism is defined by the MOST specification – this notification mechanism allows the Application (Controller) to register its interest in particular properties of the FBlock (Slave) and to be informed by the FBlock (Slave) about any changes

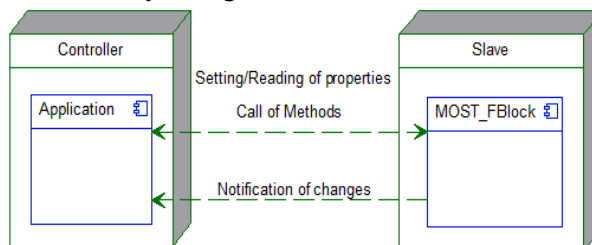


Fig. 2 Interacting with an FBlock (Slave)

- Methods that trigger a certain action within the FBlock (Slave) when they are called by the Application (Controller); after performing the method, the FBlock (Slave) returns the results of the execution to the Application (Controller).

4. PROPOSED ARCHITECTURE

In order to illustrate the Internet real-time communication with the electronic control units in vehicles, in this paper I proposed architecture for online diagnosis, maintenance and monitoring applications. The figure below illustrates the diagram of the systems which are located in a vehicle and which communicate in real-time via the Internet:

- **The diagnosis, maintenance and online monitoring system** – this system embodies the remote monitoring, diagnostics and maintenance center and it includes the server which the damaged vehicle’s driver will register and will request online diagnosis from;

- **The vehicle infotainment and telematics system** - this system represents the head unit of the vehicle which in addition to the function of providing infotainment and telematics services also provides diagnostics functions. It communicates online with the diagnostic, maintenance and monitoring center and as well as with the CAN and MOST electronic control units (ECUs) vehicle to be diagnosed;

- **The CAN ECU system** - this system represents the electronic control unit (ECU)

which is connected to the CAN bus and communicates in real-time with the infotainment and telematics system;

- **The MOST ECU system** - this system represents the electronic control unit (ECU) which is connected to the MOST bus and communicates in real-time with the infotainment and telematics system.

As shown in the figure below, the infotainment and telematics system in the vehicle contains the following sub-systems / interfaces:

- **the CAN Driver** - this is the real-time communication interface with the CAN system connected to the CAN data bus;

- **The MOST Driver** - this is the real-time communication interface with the MOST system connected to the MOST data bus;

- **The CAN / MOST Gateway** – its role is to convert the CAN messages into the corresponding MOST messages and vice versa by converting messages coming from the corresponding MOST devices into CAN messages. This translation is done according to a predefined table CAN-MOST;

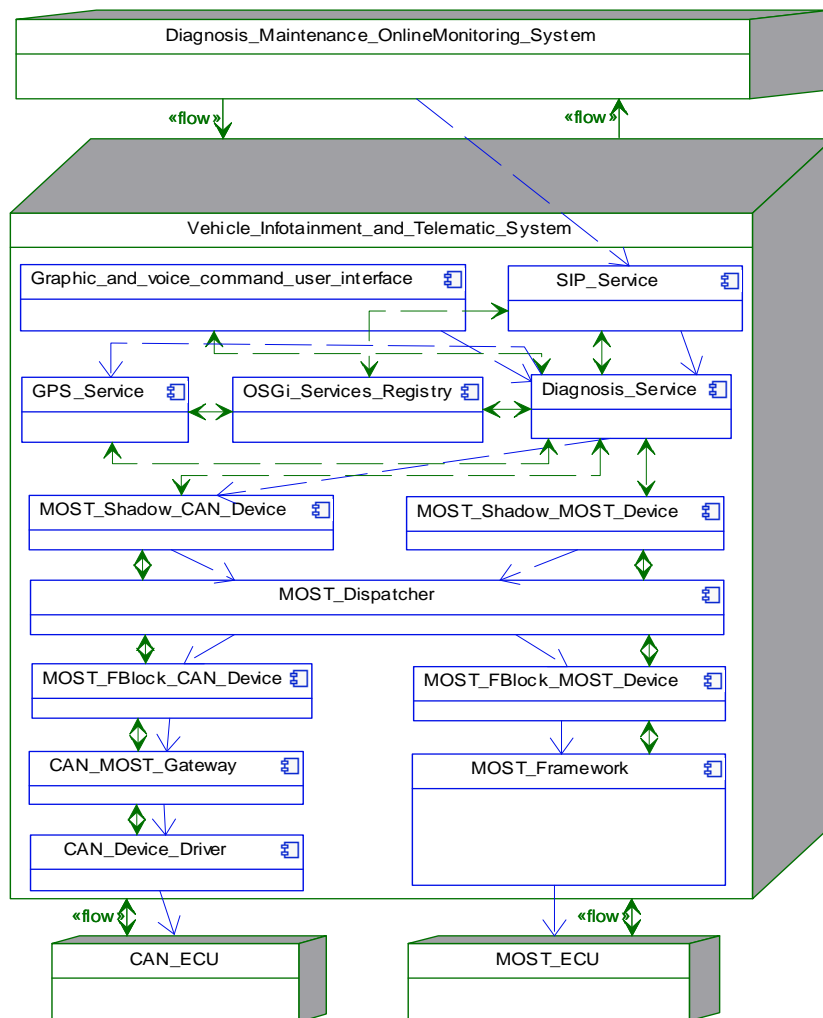


Fig. 3 Architecture of remotely real-time communication with ECUs in vehicles

- **The MOST FBlock for CAN device** - this component is the MOST functional block associated to the CAN ECU;

- **The MOST FBlock for MOST device** - this component is the MOST

functional block associated to the MOST ECU;

- **The MOST Dispatcher** – its role is to route the messages between the MOST

functional blocks and their associated MOST shadows;

- **The MOST Shadow of the CAN ECU** – this is the MOST FBlock shadow designated to the associated CAN device;

- **The MOST Shadow of the MOST ECU** - this is the shadow associated to the functional block of the MOST device;

- **The diagnostics service** - on the one hand, this service receives the requests from the online diagnostic, maintenance and monitoring center and on the other, it sends back commands in the MOST format to the electronic devices that are being diagnosed in the vehicle. For devices connected to the CAN data bus, the above mentioned request commands are properly converted into CAN messages. Simultaneously, this service is also responsible for capturing the location information from the GPS service and sending it further to the online monitoring, diagnostic and maintenance center;

- **The GPS service** - it provides positioning information to the diagnostic service so that the latter may forward it through the HTTP service to the online monitoring, diagnosis and maintenance center;

- **The SIP service** – this represents the SIP service in the Java Embedded Server which remotely delivers the OSGi online services to the monitoring, diagnostics and maintenance center thanks to the user's web browser. This service accepts SIP messages from users logged onto the Internet and converts SIP message from the DMP (Device Messaging Protocol) content into the currently available OSGi converting methods. This software component also sends back the OSGi generated messages to the SIP operating device which are located on the Internet;

- **The graphic and voice command user interface** – as its name suggests it, this interface gives the local vehicle user the possibility to interact with the infotainment and telematics system.

Thanks to the architecture illustrated in the figure 3, it is possible to establish remotely a real-time communication with the MOST and CAN electronic control units connected in vehicles through the infotainment and

telematic system. For the Internet real-time communication, the SIP[7] protocol will be used. The module service interface to the Internet is assured by the "SIP Service" which communicates internally with the "Diagnosis Service" application within the infotainment and telematic system. The "Diagnosis Service" application is responsible for the controlling of the MOST and CAN units according the requests send via Internet by the online diagnostic, maintenance and monitoring center. For the internal communication inside the infotainment and telematic system will be used the virtual MOST protocol and in order to exchange data information between the software modules will be used the services provided by the OSGi framework.

Below are presented some of the main advantages of the proposed architecture presented in this paper:

- by using the OSGi framework within the infotainment and telematic system, the software modules:
 - are running as secure and reliable services, which can be remotely managed,
 - can be reused,
 - can assure interconnection of different network standards;
- by using the MOST virtual protocol as communication protocol within the infotainment and telematic system:
 - the client-server concept is implemented as a standardized communication protocol,
 - the software MOST modules (the functional blocks and their associated shadows) can be reused later in other products.
 - The MOST devices builded inside the vehicles can be accesed directly using their "native" language, without any needs to implement additional modules/devices to translating the communication protocols into the MOST protocol.

5. SUMMARY

In this paper was proposed the architecture of the Internet real-time

communication with the vehicle electronic control units system.

In order to communicate remotely with electronic control units connected to the MOST and CAN buses inside the vehicle, an OSGi-based architecture was proposed, whereby the MOST protocol was used for the internal client-server communication protocol.

6. REFERENCES

- [1] N. Navet, F. Simonit-Lion, Automotive Embedded Systems Handbook, 2009, ISBN-10: 0-8493-8026-X
- [2] MOST Cooperation, 2011, Online: <http://www.mostcooperation.com>
- [3] Controller Area Network, 2011, Online: <http://www.controllerareanetwork.de>
- [4] OSGI Alliance, Online : <http://www.osgi.org>
- [5] MOST Cooperation, “MOST – Media Oriented Systems Transport” MOST Specification Rev. 3.0 E2, 07/2010
- [6] A. GRZEMBA, “MOST – The automotive multimedia network”, ISBN 978-3-645-65061-8, 2011
- [7] Wikipedia, the free encyclopedia, “E-learning”, accessed 01.06.2010: http://de.wikipedia.org/wiki/Session_Initiation_Protocol
- [8] Georg-Otto, Wagner, „Implementing a virtual MOST communication protocol for managing the electronic control units within the automotive systems”, University of Pitești: International Conference – 4th Edition - Electronics, Computers and Artificial Intelligence, 30 June -2 July, Pitești, Romania, 2011.