Standardization in automotive industry

Thilo Schumann, CAN in Automation (CiA)

More than 25 years ago Robert Bosch developed CAN with the needs and requirements of the automotive industry. CAN is now in use for over 20 years in the automotive industry and every major international carmaker has adopted it for most of their product lines. Almost any new introduced car makes use of CAN. But there is no standardized higher-layer protocol in use by these manufacturers. There are several attempts in standardization like namely OSEK, but they are not very widely adopted. Together with different carmakers CAN in Automation has developed one defined interface based on the international standardized higher-layer protocol CANopen. It is intended to be used in special purpose cars like for example taxi, police, emergency response vehicles, and cars for disabled persons.

Introduction

Historically, there are different standardization efforts in the automotive industry. The first well-known approach was OSEK/VDX (Open Systems and the Corresponding Interfaces for Automotive Electronics/Vehicle Distributed eXecutive). There is Autosar, the currently well-known standardization effort in terms of the development process. Then there was the standardization effort regarding the diagnostic interfaces as requested by law in the European Union and mandated by the US.
Contrary to these standardization efforts during the last years the automotive industry started a different approach with the help of CiA. This standardization effort is regarding cars for special purposes, like for taxi, police, emergency response vehicles, and cars for disabled persons. Why and how does that happen?

OSEK/VDX

OSEK was founded as a joint project by the German automotive industry. The purpose was to develop an open-ended architecture for distributed control units. The standardization effort was aimed at mainly the following three topics:

- Communication (data exchange within and between control units)
- Operating system (real-time execution of ECU software and base for other OSEK/VDX modules)
- Network management (configuration determination and monitoring)

The goal of the initiative was defined according to their website as to support the portability and reusability of the application software by:

- specification of interfaces which are abstract and as application-independent as possible
- specification of an user interface independent of hardware and network
- efficient design of architecture: the functionalities shall be configurable and scalable to enable optimal adjustment of the architecture to the application in question
- verification of functionality and implementation of prototypes in selected pilot projects

Several specifications were developed as part of this standardization effort. The specifications cover those three topics and are named accordingly OSEK COM, OSEK OS, and OSEK NM. Other specifications are developed in addition to support the architecture design process and portability, namely OSEK OIL and OSEK RTI. Most of the specifications are now internationally standardized at ISO and are available since 2005 as a set of specification with then name ISO 17356: Road vehicles – open interface for embedded automotive applications.
In terms of standardization this was not an approach to standardize an automotive application but the design process of automotive applications. The real use as of today is limited: widely adopted is OSEK OS, which means there are several proprietary real-time operating systems,

which provide APIs according to the requirements of OSEK OS. OSEK NM is partly used. OSEK COM is not used at all and being replaced with OEM proprietary solutions. But even OSEK COM in terms of specification is very loosely defined, e.g. both existing byte-orders are allowed to exist and they are allowed to even exist in one message. The reason is, that OSEK COM focuses on the specification of the services, but does not include any protocol definitions. There are no protocol definitions of OSEK/COM on CAN, on Flexray, and LIN. According to the current state the protocol definition has to be manufacturer specific.

AUTOSAR

Following the approach of OSEK/VDX, Autosar (Automotive Open System Architecture) is another try of standardization. Autosar is a different initiative than OSEK/VDX in terms of its objectives and goals, but setup by almost the same market players. Autosar describes itself as an open and standardized automotive software architecture, jointly developed by automobile manufacturers, suppliers and tool developers.

As shown in Figure 1 Autosar has the approach of a middleware, of a framework. Autosar as a framework defines basically an abstraction of hardware to allow independent hard- and software development. An input function, an output function, a control function are an Autosar software component that basically can run on any hardware and does not know anything about the hardware, because it is encapsulated by the Autosar interface definition of the Autosar Runtime Environment. But Autosar also defines the interfaces for different services within the basic software. There is a big advantage for his approach it allows to encapsulate the runtime environment from the hardware itself. This means the supplier of the runtime environment has clearly defined interfaces he can rely on. Also the basic software itself is designed with a modular concept in mind. The advantage is the well-defined separation of each module.

Autosar as a middleware means, that there could develop an ecosystem of software vendors that provide standardized software components that can be easily replaced by other vendors. That means,

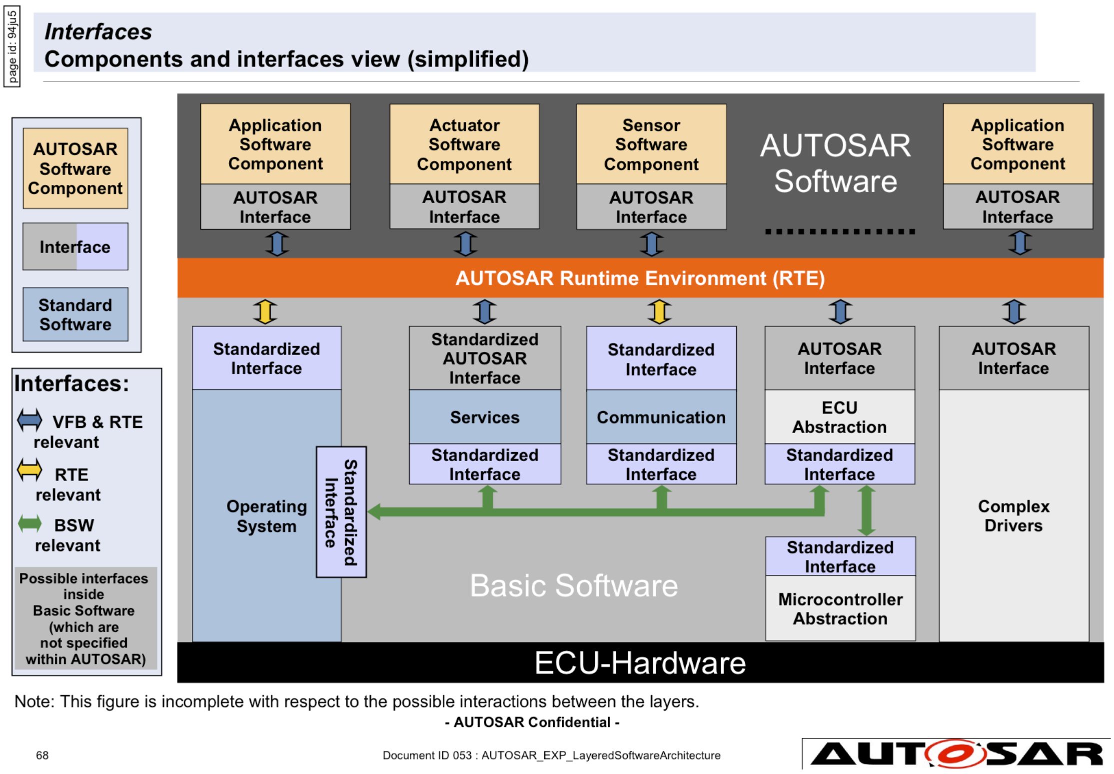

Figure 1 — Autosar simplified software structure

that a carmaker does not have to buy in one supplier.
The general approach of this concept is not new to information technology, it is known as POSIX. POSIX has the same ideas in the computer world and is realized in many operating systems these days, including but not limited to GNU/Linux, Windows, and the different BSD variants including Mac OS X.
The basic idea behind this concept is clearly to have re-usable software components, to reduce software development in future car generations. But Autosar does not standardize the car itself or any of the functions and signals within a car. This is still completely manufacturer specific.

Diagnostics

The standardization of the diagnostics interface of a car has a completely different approach. It was mandated by the European Union and the US for two different reasons: first as cars becoming more and more complex, it is almost impossible to have a third party servicing a car. Second, as cars having sensors in place for measuring the exhaust gases for control purposes and cars have to fulfill the more and more demanding environmental requirements there should be an easy access to these data. With these requirements in mind the diagnostic interface has been defined. There are standardized services and protocols to read out diagnostic data from the different devices in the car. The basic diagnostic data is also standardized to enable basic servicing and to read out environmental data collected by the car itself. But the standardized diagnostic data is still very limited even as the manufacturers have extended the data sets very largely. The extension is made, because the same diagnostic interface is used by the carmakers to allow detailed servicing, but only for the manufacturer specific contractors.
The standardization of the diagnostic data is still the first time that automotive specific data sets have been defined. It is available as ISO 15765: Road vehicles – Diagnostic communication over Controller Area Network (DoCAN).

CiA447

The standardization of CiA 447 follows somehow into the same footsteps of ISO 15765. It defines the automotive application for a specific purpose. The purpose of CiA 447 for now is special purpose cars.
It started in 2006 that the BZP (German nonprofit organization for Taxi and Cars for hire) organized an event on CAN networked taxi-specific devices. As a result the Special Interest Group (SIG) car add-on devices within CiA was born. The purpose was and is to specify the CAN networked devices within special purpose cars. It started with taxi, evolved for police and the next steps will be emergency response vehicles and car for disabled persons.
CiA 447 is the specification defined by that group. It is currently available in version 2.0 and partly implemented by some carmakers as of now. Carmakers are currently implementing it for their next generations' product lines.

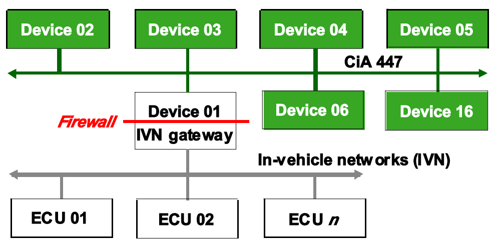

Figure 2 — CiA 447 device structure

The standardization in CiA 447 is different to Autosar in a way that it is automotive specific for that purpose only. The standardization is regarding devices and data signals (see Figure 2 and Figure 3). The expert group identified the relevant devices for the specific purpose and defined their data signals and behavior. The basic idea behind that approach is, to allow an easy, plug and play connection of those devices. Not defined is the application behavior. The application behavior is left open to be manufacturer specific. This is by intention, because the behavior of the taxi application or the police application highly depends on the requirements defined by country, by governmental organization, by state level, or sometimes even by district.

The most basic function in all fleet management systems, is vehicle tracking. It is usually GPS-based.

The CiA 447 provides several GPS-related and navigation system parameters:

- 60B0$_h$: GPS current position
- 60B1$_h$: GPS satellites
- 60B2$_h$: GPS status
- 60B3$_h$: GPS date
- 60B4$_h$: GPS UTC time
- 60B5$_h$: GPS velocity and heading
- 60B6$_h$: GPS altitude

- 60C0$_h$: Distance to selected destination
- 60C1$_h$: Position description request
- 60C2$_h$: Position description
- 60C3$_h$: Start route guidance
- 60C4$_h$: Current position request
- 60C5$_h$: Current position

Figure 3 — CiA447 data signal example

In the thinking of CiA 447 even the car itself is a simple device. It provides some input and output functionality, but no application behavior. That means, the lights of a car can be switched off and on, and the status can be inquired. The current GPS position can be inquired. In the thinking of CiA 447 it does not matter which device will provide the GPS information. It could be the car itself, as it has a navigation system available anyhow. But it could also be a separate GPS device providing this information. The advantage of that approach is, that the input and output devices are very well defined. There is no re-implementation needed. But the application behavior can be fitted to the current requirements very easily.
The carmakers are very interest in this standardization, because as described there is no one fits all solution possible. The carmakers are now in position in which normally machine builders are. They have to build a device and must be able to partly configure their device to the specific needs afterwards, after it is build.

Conclusion

Autosar and CiA 447 are two approaches of standardization, which do not conflict with each other, but will build a complete picture. Device functionality and data signals are well defined in terms of the application, like is being done in the automation domain. The application functionality is application specific and there is a standardized development process with well-defined interfaces. This allows portability and reusability of the control functionality.
Also other applications and specifically OEM-independent applications like the eCall initiative may be able to profit from the experiences made from Autosar and CiA 447.

References

OSEK/VDX: http://www.osek-vdx.org
Autosar: http://www.autosar.org

Thilo Schumann
CAN in Automation (CiA)
Kontumazgarten 3
90429 Nuremberg
Germany
Phone: +49 – 911 – 928819 – 0
Fax: +49 – 911 – 928819 – 79
E-mail: headquarters@can-cia.org
Website: www.can-cia.org