5

Wireless in Future Automotive Applications

Volker Schuermann, Aurel Buda, Stefan Jonker,
Norman Palmhof and Joerg F. Wollert
Bochum University of Applied Sciences
Germany

1. Introduction

Wireless technology became a part of the everyday life of many humans. Practically everyone possesses a mobile phone and is mobile attainable over it. Mobile phones became our daily way companions. Thus, and with the in the meantime clearly increased efficiency of these devices a number of new application scenarios are possible. Thereby can be fallen back on the experiences from other areas of application, for example from the mobile phone game market, which brought a quantity of interesting concepts out. Mobile phones increased not only their performance; they also bring along clearly a number of efficient communication interfaces, everything in front Bluetooth.

Also for the automobile industry the integration of mobile devices into vehicles is in the future an interesting market, since here completely new business models can be implemented. The chapter presents the boundary conditions for this. In addition also a concept for the integration of mobile devices in the vehicle belongs to it. Apart from pure aspects of communication, also the development and distribution of mobile applications are more near regarded.

First this chapter gives an overview of in the automotive environment spread communication technologies and their areas of application, the margin is here from short range technologies with ranges from few meters to long range communication over several kilometers away. Whereupon an overview of the key technology Bluetooth follows, whereby the emphasis honor on the application-oriented parts of the specification and the Bluetooth profiles is. Afterwards the Java Micro Edition, for the development of mobile applications, is in the focus of the chapter; here is a special attention, on the communication APIs and security. To the conclusion of the chapter then possibilities of the vehicle integration are described in detail on the basis of an example.

2. Wireless technologies and their areas of application

Wireless communication already belongs to the state of the art in many areas of the automotive environment today. Much works thereby covered off and is not noticed by the user of the end product. First of all a general overview of the assigned wireless technologies and their areas of application is to be given. See also for this Fig. 1.

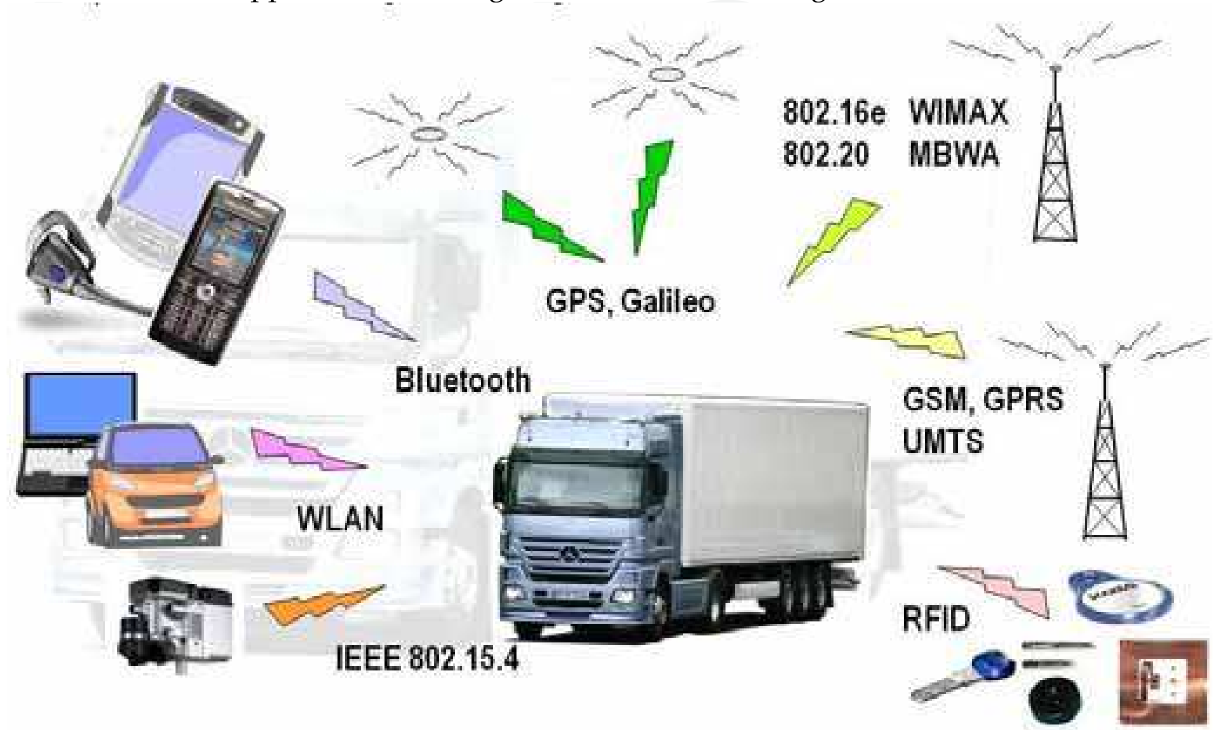

Fig. 1. Wireless Use Cases

IEEE 802.15.4:
IEEE 802.15.4 is a short range radio technology for wireless sensor networks. It forms the lower two protocol layers of a number of, in the automatic control engineering well-known, communication standards for example ZigBee or WirelessHART. The focus lies in the reliable transmission of small data sets if necessary over several hops away. In the automotive environment IEEE 802.15.4 is to be mainly found in production plants.

WLAN:
Main field of application of WLAN is the wireless integration of notebooks into local area networks. It looks similar also within the automobile area. Many of the diagnose tools necessary for modern vehicles, are today PC-based, which means a simple integration of WLAN, since both hardware and protocol stacks are present in large multiplicity at the market. So far these tools are usually connected over cables with the vehicle, whereby the diagnose unit must be in direct proximity to the vehicle. One is at present endeavored to replace these in many cases unpractical cable connections e.g. if the vehicle is on a lifting platform is, by wireless communication. First developments aim at the use of adapters, which are attached to the ODB2 (on board diagnosis) interface. A complete integration of WLAN in the vehicle is not impossible in the future. Further a set of comfort functions can be realized so, for example the transmission of vcard files from the email program of the PC

to the navigation system. In addition there are ambitions to use WLAN for Ad-hoc communication of vehicles among themselves and/or for communication of vehicles with their environment. One speaks in this connection of Car to Car and Car to Roadside communication.

Bluetooth:
Bluetooth is used in vehicles nowadays mainly for the free speech mechanism and the integration of headsets. In addition especially from the comfort and multimedia area a number of further meaningful applications can be realized, for example the playback of, on the mobile phone stored music files on the cars audio system. More to the application possibilities of Bluetooth follows in the further process of the chapter.

GPS, Galileo:
In the today's time almost nobody drives an unknown distance without navigation system. But not only for the comfort of the drivers is the knowledge of an exact position of importance. In logistic processes for example it is important to know, where certain goods or vehicles are at the moment. In times of just in time production a special meaning comes to that. The management of large fleets would not be possible without actual and exact position information. The American GPS represents the state of the art here at present. The European Galileo system up-to-date still is in the planning phase.

WIMAX, MBWA:
WIMAX (Worldwide Interoperability for Microwave Access) and MBWA (Mobile Broadband Wireless Access) are called colloquially frequently also „wireless DSL ".With them „the last mile" to customers is to be bridged to provide them with a DSL equivalent access to the Internet. Operational areas are in special infrastructure-weak regions. Both standards possess besides a mobile component, it permits the transfer of larger data sets over a distance of some kilometers to a moving vehicle.

GSM, GPRS, UMTS:
Beside pure telephony also data communication continues to move into the foreground with these technologies. A similar goal pursued as with WIMAX and MBWA, although with usually smaller data rates. However these technologies in many countries offer a surface covering net cover. In the remaining regions the net still is in the development.

RFID:
Especially in the luxury segment keyless entry and keyless go systems are a firm component of cars. The transponders necessary for it are frequently RFID chips characterized by a very small energy consumption which frequently get along even without battery, since they get the energy from the surrounding electrical field.

3. Bluetooth

The intention behind the development of Bluetooth (Bluetooth SIG, 2009) (IEEE, 2002) was replacing cables between individual devices such as mobile phones, PDA's, PC's, cordless mice, headsets etc. Important aspects thereby were on the one hand the costs of the individual radio modules as well as the energy efficiency of the devices working usually on battery basis. Further it was enormously important to develop robust radio modules which are not damaged in case of transport of the mobile devices. The connection should remain unimpaired of other radio transmitters and be Ad-hoc, thus spontaneously to be developed. All these aspects considered until today with the advancement of the Bluetooth standard, additionally are the requirements to the transmission rate of such radio communications ever more largely. Bluetooth is in the meanwhile a world-wide accepted standard, which is very popular to due to its versatility and fail-safe characteristic also in the industrial area.

The Bluetooth architecture is essentially divided into three parts: the Bluetooth core, a protocol layer and an application profile layer. The Bluetooth core forms the IEEE 802.15.1 standard; it consists of the lower layers, which are necessary for communication. Fig. 2 shows the principle structure of the Bluetooth protocol stack. The subchapter begins with the Bluetooth core and its components, followed by some fundamental protocols, the different application profiles of Bluetooth are more near explained thereafter.

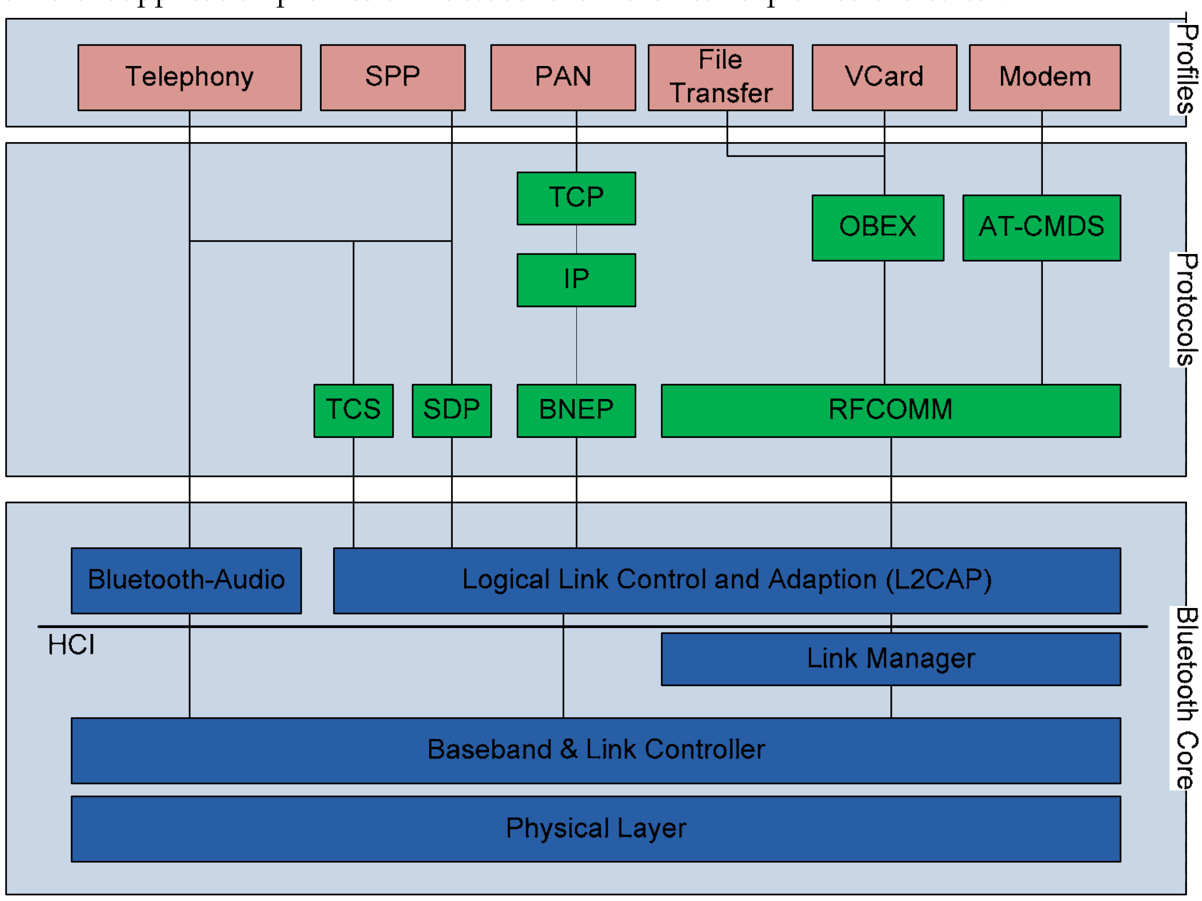

Fig. 2. Bluetooth Protocol Stack

3.1 Bluetooth Core

The Bluetooth core consists of several layers and forms the standard IEEE 802.15.1 in the actual sense. It covers the lower protocol layers beside the radio hardware also for the setting up of connections between devices. It is possible to set up piconets with up to eight active participants; one of them is the master of the piconet. Between the devices both confirmed and unconfirmed communication for example for audio connections, can be established. For applications described in this chapter however the higher protocol layers and the services touching down on them are more interesting, therefore these are described in detail in the following.

3.2 Protocol Layer:

Above the Bluetooth core one finds a further layer, it consists of a multiplicity of different protocols, which represent the connection between Bluetooth core and application.

SDP:

In order to ensure the Ad-Hoc-ability of Bluetooth devices, is it necessarily that the devices between those a connection should be made, can recognize whether the other device supports the desired service. In order to manage this, the Bluetooth standard specifies the Service Discovery Protocol (SDP). Hereby it is possible to query the Service Record of a device. In the Service Records all available services of a Bluetooth device are stored, with a unique ID and service attributes. The inquiry of the Service Record is a Client-Server communication. The device, which would like to establish a connection to a service, sends an SDP Client inquiry to the SDP server of the other device, this sends in the response information about the supported services and it can be begun to establish a connection.

RFCOMM:

One usually used Bluetooth protocol is the RFCOMM (Radio Frequency Communications) - protocol. In principle RFCOMM is used everywhere, where a Bluetooth radio link should replace a physical cable, e.g. for the synchronization between a PDA and a PC. The RFCOMM protocol is able to administer up to 60 virtual serial interfaces at the same time. Other protocols like, the particularly in the mobile phone area spread, OBEX (Object Exchange Protocol) touches down on the RFCOMM protocol, a typical application of OBEX is the exchange of contact information between mobile phones or mobile phone and PC. Bluetooth replaces here with a radio link the device specific data cable. Likewise many Bluetooth profiles use the RFCOMM protocol, in the following with these is more in greater detail dealt.

TCS:

Telephony control Protocol specification (TCS) is the substantial protocol for the controlling of voice connections, all Telephony functions are regulated via this protocol.

BNEP:

The BNEP (Bluetooth Network Encapsulation Protocol) made possible like the name already says the encapsulation of different packets, which arise in a cable-bound network e.g. IP packets. Thus a Bluetooth device which is connected with the network over a Bluetooth Access point can exchange data and thus for example can use network printers. In order to realize this, the network packets are packed within BNEP frames, and passed to the lower protocol layers for transmission.

3.3 Bluetooth Profiles

The different Bluetooth profiles make the interaction of applications on different Bluetooth devices possible. They specify capability characteristics and parameters, which are needed, in order to communicate over certain protocols. They offer vertical access to the protocols. If two devices support the same profile, then they can communicate also spontaneously and problem-free with each another. Bluetooth is the only radio technology, which offers a so various service architecture. The most important fundamental profiles are presented in the following section. Fig. 3 shows the hierarchical layout of the Bluetooth profiles. An explanation of the abbreviations is effected in the following table.

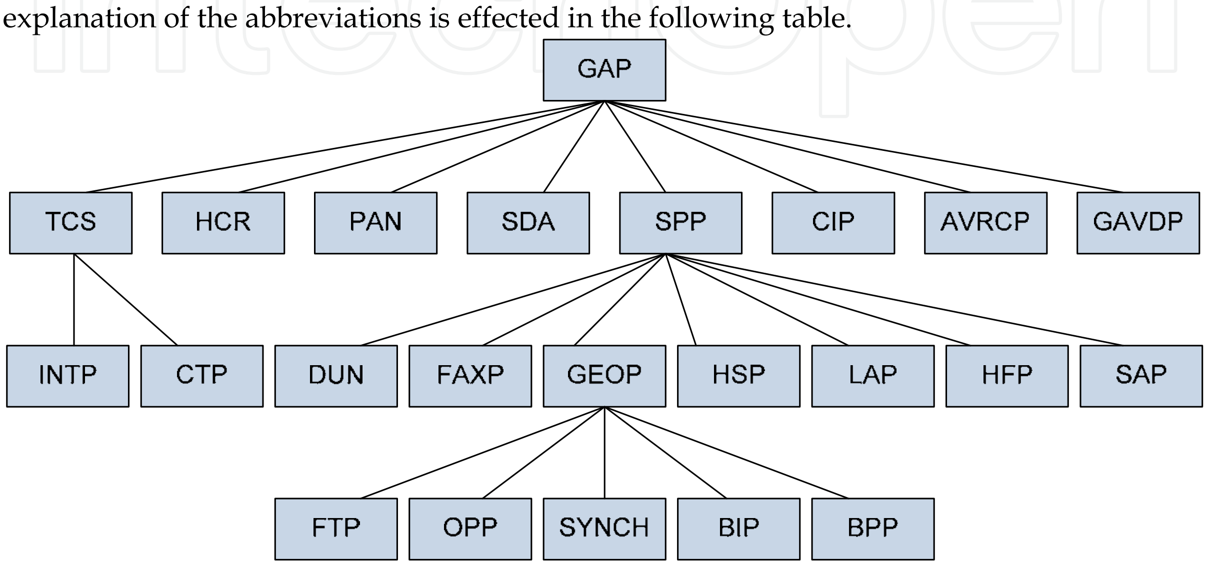

Fig. 3. Hierarchy of the Bluetooth of Profiles

GAP	Generic Access Profile
TCS	Telephony Control Specification Profile
HCR	Hardcopy Cable Replacement Profile
PAN	Personal Area Network Profile
SDA	Service Discovery Application Profile
SPP	Serial Port Profile Profile
CIP	Common ISDN Access Profile
AVRCP	Audio Video Remote Control Profile
GAVDP	Gerneric Audio Video Distribution Profile
INTP	Intercom Profile
CTP	Cordless Telephony Profile
DUN	Dial-up Networking Profile
FAXP	FAX Profile
GEOP	Generic Object Exchange Profile
HSP	Headset Profile
LAP	LAN Access Profile
HFP	Hands Free Profile
SAP	SIM Access Profile
FTP	File Transfer Profile
OPP	Object Push Profile
SYNCH	Synchronisation Profile
BIP	Basic Imaging Profile
BPP	Basic Printing Profile

Table 1. Bluetooth Profiles

GAP:
The Generic Access Profiles (GAP) is the most fundamental profile; here for all profiles fundamental characteristics are specified. To this the device name, the pin code and the Bluetooth address belong. Further functions are described such as connecting administration, operating mode and connecting security in this profile. GAP stands in the hierarchy of the Bluetooth profiles in highest place.

SDAP:
A further profile, which must be supported by all Bluetooth devices, is SDAP (Service Discovery Application Profile). It allows applications to access the already mentioned Service Record, which describes all services the device includes.

SPP:
The SPP is one of the usually-used profiles, because it offers the possibility of making up to 60 virtual serial interfaces available on a device. Each virtual interface possesses the characteristics of the well-known RS232 interface and reaches a data rate of 128kBit/s. The profile actually still serves as basis for further profiles. Fig. 4 describes which profiles SPP as basis for communication use.

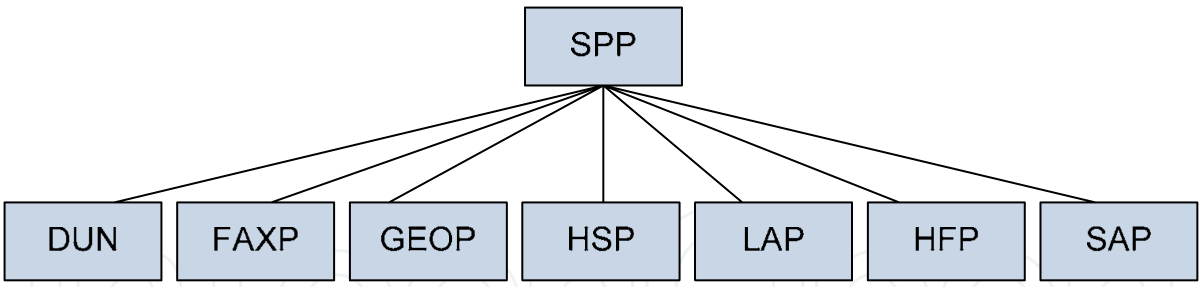

Fig. 4. SPP as basis

PAN:
The Bluetooth PAN profile was developed, in order to cover two concrete use cases. On the one hand it should be possible with the help of the PAN profile to develop a network infrastructure which equals wired LAN's. Further Access Point functionality with Bluetooth should be implemented, whereby the master functions as Access Point.

4. Java Micro Edition

The software for mobile devices is mostly written in C/C++ or Java. The question, which programming language is the best for the development of software on mobile phones could not be answered yet and is often the cause of discussions. But there are some serious reasons to use Java (Sun Microsystems, 2009) as programming language for mobile applications, because there is a Java virtual Machine for all important operating systems which runs the Java Byte Code on almost all devices without or with just very small changes. Thus a port from one system on to another one can be carried out with small expenditure. Restrictions with the port can occur with different hardware, like for example the minimal necessary display size. Java has a security API for security relevant operations like for example authentication and authorization. In the area of the mobile phones Java is very wide spread and has a large acceptance with all considerable hardware manufacturers. The Java Virtual Machine supervises the Java program, so that a crash does not affect other applications, thus the system becomes very robust. Further Java has, contrary to C++, a Garbage Collector which worries about the memory management. Due to nearly all important software enterprises and open source organizations cooperate in the Java Community Process (JCP), at the development and publishing of new specifications, there is a high market acceptance of Java. Because of these reasons Java has become a de facto standard for mobile application development and this will remain for a longer period.
Java Micro Edition (ME) (Breymann & Mosemann, 2008) is a subset of the Java standard Edition (SE) and is adapted on the needs of the mobile devices. Because the libraries of the standard edition are still much too large for the memory of the mobile devices. But in foreseeable time these devices will be efficient enough to use the full function range of the Java SE. Because of Moore's law the memory density on the chips doubles itself every 18 months.
Java ME is a collection of specifications and technologies, which particularly suit to the needs of the mobile devices. The structure of Java ME consists of configurations; these contain the Java Virtual Machine and a small sentence of class libraries from the Java SE. The configurations are extended by profiles, these contain further necessary APIs and optional Packages for special applications like for example Bluetooth communication. The

Java Micro Edition divides itself into two areas, depending upon features of the mobile devices.

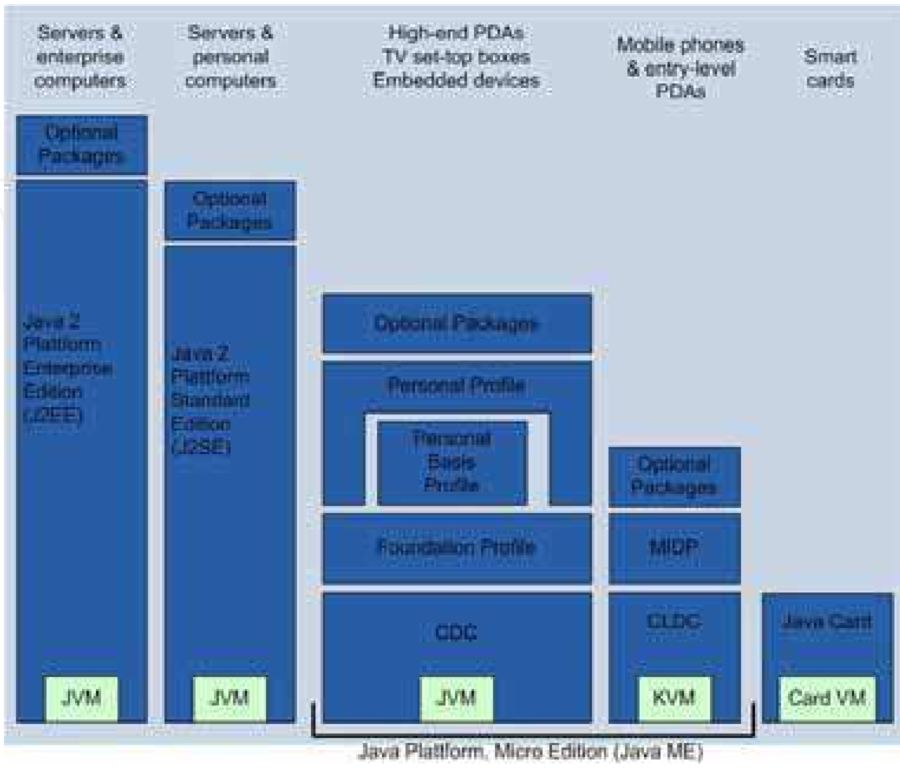

Fig. 5. Java ME in der Java Familie

For simple PDAs and mobile phones the Connected Limited Device Configuration (CLDC) is available. This contains the Kilobyte Virtual Machine (KVM). The configuration is extended by the Mobile Information Device Profile (MIDP). The MIDP builds up on the CLDC and extends this by a quantity of important functions, like for example the controlling of the life cycle of an application. For special applications the optional Packages are still available. The CLDC is designed for slow processors, little memory and unreliable network connections. The typically memory size is between 128 and 512 KByte. The CLDC and MIDP represent together a complete Java run time environment and are everything one needs to run simple Java programs on a mobile phone.

For more efficient devices like for example High end PDAs, set top boxes and embedded devices the Connected Device Configuration (CDC) is available, this contains a Standard Java Virtual Machine. To this run time environment belongs the Foundation Profile too, which makes the basic functions available for embedded systems. As well as the Personal Profile, this extends the Foundation Profile. In addition there are the optional Packages as with the CLDC. Fig. 5 shows the location of the Java Micro Edition in the surrounding field of the Java environment.

Fig. 6. Screenshot Wireless Toolkit: Selection of APIs

4.1 Configurations and Profiles

Since not all characteristics of all devices are known it is difficult to create a run time environment that fits to all characteristics of the different devices. Therefore one pursues the approach of configurations within the Java Micro Edition. A certain number of devices is assigned to a configuration according to their efficiency. Like that those programs are run able on all devices that have this configuration. The partitioning in CLDC and CDC configuration is made as in the previous section. Fig. 7 shows the architecture of a CLDC configuration with the MID Profile like it is used for mobile applications. Because of the rapid development and the short life cycle of such devices it is not possible to specify all variants. Therefore additional native applications, running direct on the operating system, and manufacturer-specific Java classes and applications are used.

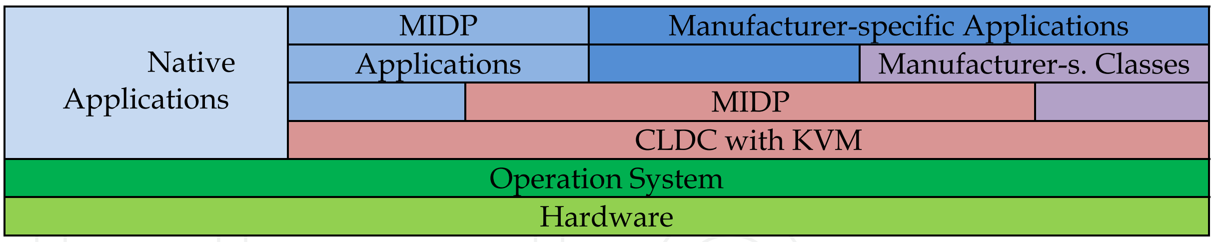

Fig. 7. Architecture of the CLDC

The following table shows an overview of at present available configurations of Java ME. Regarding to the technological development of the mobile devices with the CLDC 1.1 compared to its previous version 1.0 the minimum necessary memory was increased from 160 to 192 KB. The main reason for it was the introduction of the classes `Float` and `Double`. Further smaller errors were corrected and some additional classes were added. It might be only a question of time until all mobile devices support version 1.1 of the configuration, but at the moment one has to consider which version the current hardware supports.

JSR 30	CLDC 1.0	Connected Limited Device Configuration
JSR 139	CLDC 1.1	Connected Limited Device Configuration 1.1
JSR 36	CDC 1.0	Connected Device Configuration 1.0
JSR 218	CDC 1.1	Connected Device Configuration 1.1

Table 2. Java ME Configurations

The CDC configuration contains a substantially larger part of the Standard Edition APIs. And the appropriate Foundation Profile contains the entire Java Abstract Window Toolkit (AWT) with all functions necessary for executing Java Applets. The Personal Basis Profile is a subset of the Personal Profile and makes available nuclear functionality with a minimum graphic support. Here is not dealt with CDC and their profiles, since for the for the mobile application development the CLDC is crucial.

JSR 37	MIDP 1.0	Mobile Information Device Profile
JSR 118	MIDP 2.0	Mobile Information Device Profile 2.0
JSR 75	PDA	PDA Profile
JSR 46	FP	Foundation Profile
JSR 129	PBP	Personal Basis Profile
JSR 62	PP	Personal Profile

Table 3. Java ME Profiles

The MIDP bases on the CLDC and contains many important functions like for example network connections and their protocols, generation of sounds and user interfaces such as screen or keyboard. The Mobile Information Device Profile specifies also a set of minimal requirements to the hardware like for example a screen resolution of 96x54 pixels. Today the version 2.0 is supported by most mobile devices; an overview of the available Java ME profiles gives the above table.

The optional packages can be merged depending upon the needs of the application and the hardware requirements. Following table shows an excerpt of the most important Packages with their JSRs numbers. All JSRs can be looked up under www.jcp.org the homepage of the Java Community Process, under the direction of Sun.

JSR 75	PIM	PDA Optional Packages (PIM und Dateisystem)
JSR 82	BTAPI	Bluetooth APIs
JSR 120	WMA	Wireless Messaging API
JSR 135	MMAPI	Mobile Media API
JSR 172		Web Services
JSR 177	SATSA	Security and Trust Services API
JSR 179		Location API
JSR 180		SIP API
JSR 184		Mobile 3D Graphics API
JSR 205		Messaging
JSR 211		Content Handler
JSR 226		Vector Graphics
JSR 229		Payment
JSR 234		Multimedia Supplements
JSR 238		Internationalization

Table 4. Optional Packages

Following table gives an overview of the spreading specifications. The Mobile Service Architecture specification (MSA) JSR 248 refers like its predecessor JSR 185 to a large extent of already existing specifications. It eliminates ambiguity and gives supplementing data where it is necessary. A goal of these spreading specifications should be to prevent a splintering of the individual APIs and give the different hardware manufacturers a guideline for the smallest common denominator. A device which fulfills the MSA specification must at least fulfill the MSA Subset, which is a subset of the MSA with decreased function range. An overview of the function range and the pertinent JSRs of the MSA and MSA Subset specification give Fig. 8. A minimum requirement to the hardware of the devices is also defined by the MSA specification. At least 1024 KB of volatile memory, a screen size of at least 128x128 pixels with a depth of shade of 16 bits. A multiplicity of further requirements can be inferred from the documentation JSR 248.

JSR 68	Java ME	Java ME Plattform Specification
JSR 185	JTWI	Java Technology for Wireless Industry
JSR 248	MSA	Mobile Service Architectur

Table 5. Package Bundles

JSR 238	Internationalization		**MSA**
JSR 234	MultimediaSupplements		
JSR 229	Payment		
JSR 211	Content Handler		
JSR 180	SIP		
JSR 179	Location		
JSR 177	Security & Trust		
JSR 172	Web Services		
JSR 226	Vector Graphics	**MSA Subset**	
JSR 205	Messaging		
JSR 184	3D Graphics		
JSR 135	Mobile Media		
JSR 82	Bluetooth		
JSR 75	File & PIM		
JSR 118	**MIDP 2.0**		
JSR 139	**CLDC 1.1**		

Fig. 8. MSA and MSA Subset

4.2 JSR 82

The JSR 82 (JCP, 2008) was initiated by the JCP, for the development of Bluetooth based applications of communications and consists of the Java APIs for Bluetooth Wireless Technology (JABWT). This JSR represents no implementation of the general Bluetooth specification, but represents a collection of APIs for the configuration and controlling of the Bluetooth hardware in mobile devices.

The following subsections give beside the requirements of such a device and the structure of API architecture, views into the necessary configuration of services and devices and the general operational sequence of Java ME based Bluetooth communication under consideration of all security aspects.

Requirements:

For the employment of the JSR 82 API on mobile devices at least 512 KB main memory are needed, as well as a complete implementation of the Java ME CLDC version 1.0. In addition the existing Bluetooth hardware must exhibit a qualification of the Bluetooth Qualification Program at least for the profiles GAP, SDAP and SPP. Further the SDP, RFCOMM and the L2CAP profiles must be supported and accessibility for the API of these protocol layers must exist.

The access on the lower hardware and protocol layers is administered of a so-called Bluetooth Control Centre (BCC). Therefore it is not a component of the API, and must be provided by the hardware environment.

If all requirements are fulfilled, the Bluetooth API offers the following features during the application development:

- Registration of services
- Inquiry search of Bluetooth hardware and services
- RFCOMM, L2CAP and OBEX connections between Bluetooth devices

- Transmission of data, excluded voice connections
- Administration and controlling of communication connections
- Security mechanisms for expiration of communication

Here it is pointed out that the presence of Bluetooth and Java on mobile devices does not guarantee the support of the JSR 82 API, since among other things the possibilities of a device configuration are reduced by the Java ME. However this applies only to a part of the mobile phones offered nowadays.

Structure of API architecture:
The JABWT APIs extends the MIDP 2.0 platform with Bluetooth and OBEX support and consists of two packages, the fundamental Bluetooth API `javax.bluetooth` and the OBEX API `javax.obex`. Both are dependent on the package javax.microedition.io, which belongs to the CLDC, and optionally applicable depending upon requirements of the application. Fig. 9 clarifies the position of the Bluetooth API within an CLDC MIDP environment.

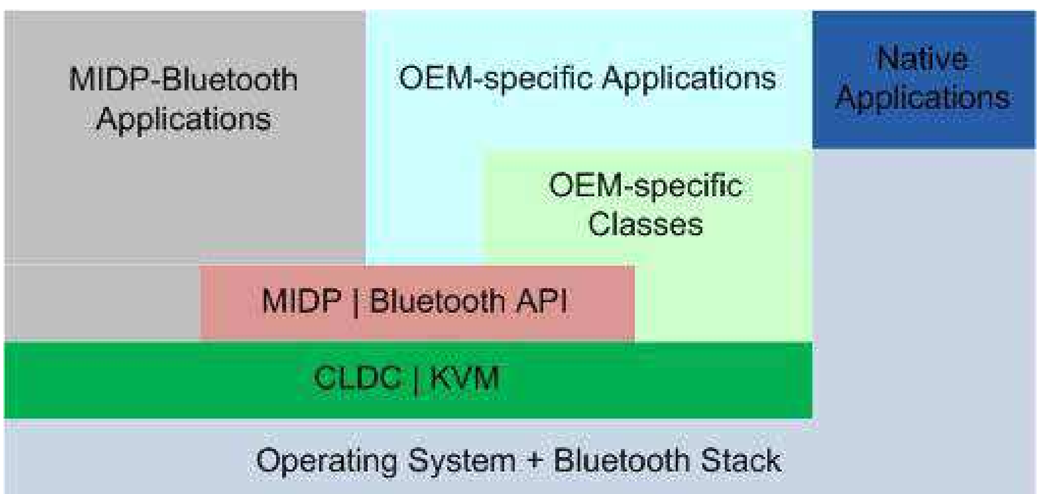

Fig. 9. Bluetooth in the Java Architecture

A Bluetooth application can be divided first into five ranges, which are processed with an implementation in chronological order: Stack initialization, management of devices, finding devices, finding services and communication. All APIs needed for these are part of the `javax.bluetooth` package.
As was already described on the basis the SDP, Bluetooth devices can take the role of a server or a client. This is specified in each case by the application. The activity diagram from following Fig. 10 gives an overview of the individual fields of server and client.

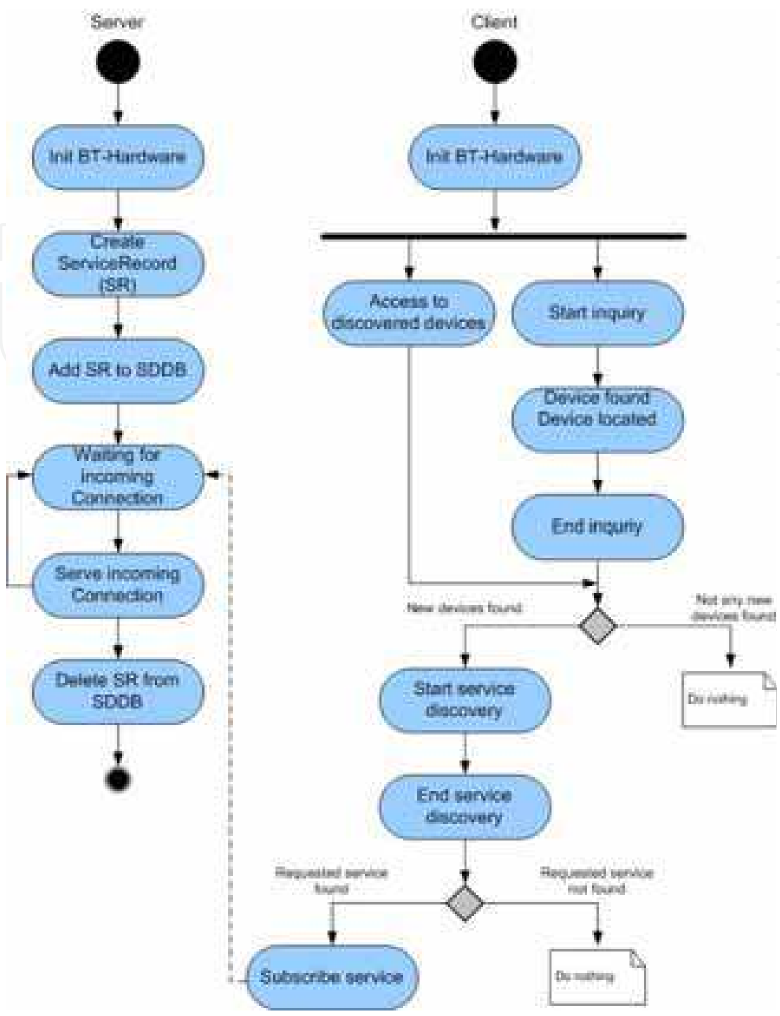

Fig. 10. Client and Server Activities

The initialization of the Bluetooth stack is independently of their operational area necessary for each Bluetooth application. A client application contains the search for devices and services, as well as the connection establishment with devices resulting from it and a following service use. A server application makes services available, administers these and reacts on connecting inquiries.

4.3 JSR 120 and JSR 205

A further Java API for the mobile communication is the Wireless Messaging API (WMA). The versions 1.0 and 1.1 were published in the JSR 120 (JCP, 2003) version 2.0 in the JSR 205 (JCP, 2004). With the Wireless Messaging API a mobile application can react on SMS and MMS messages, which are addressed to a certain port of the mobile phone, to which the application has registered itself, and process the received data. Messages also SMS in a binary format can be processed beside simple text or multimedia messages.
For further data communication in mobile communication networks as for example GPRS or UMTS further APIs are not necessary, since it concerns packet-oriented networks here and

so each mobile phone is IP addressable. The operating system usually makes this connection and administers it. From application view the standard APIs for Socket or HTTP communication can be used. It is the same procedure like in WLAN networks.

4.4 MIDlet

A Java program which was written for the MID profile is called to MIDlet; one or more MIDlets can be combined in a MIDlet Suite. After compiling source code one has a jad and a jar file, which can be loaded on a mobile phone afterwards. Each device on which a MIDlet should be executed must provide an environment which guarantees execution and administration of MIDlets. This environment is called Application Management Software (AMS) and controls the life cycle of the MIDlets. A MIDlet can be like the well-known Java Applet also only in one of three states. Between the two states Paused and Active the MIDlet can change during its runtime. The state Destroyed is however final. The MIDlet can even change its states by the help of special methods, but must notify the AMS about it. The AMS can change the states of the MIDlets at any time. This can happen if the resources of the MIDlets are needed by other processes, for example in case of a incoming telephone call the AMS sets the MIDlet into state Paused and the necessary display is used for the telephone call.
The MIDlet object is generated by the AMS and is first in the state Paused see Fig. 11, thus still no resources are blocked. Afterwards the MIDlet is started by the AMS through a call of the method `startApp()`. Now the MIDlet is in state Active and all needed resources will be requested. From the state Active the MIDlet can change again into the state Paused through the AMS or by itself. If for example a telephone call arrives the AMS sets the MIDlet into state Paused, since it needs some resources like for example the display of the MIDlet. The MIDlet asks periodically with the method `resumeRequest()` if it is allowed to run again, in this case the AMS starts the MIDlet by means of the method `startApp()`.

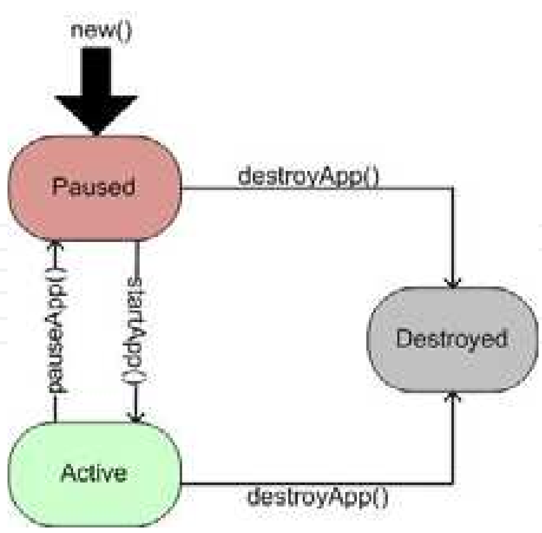

Fig. 11. State diagram of a MIDlet

From the state Active the MIDlet can be set by itself or by the AMS into state Destroyed. It releases then all requested resources and stores if necessary application data for the further use. Afterwards the MIDlet can be eliminated by the Garbage Collector.

4.5 Application Deployment

The occasionally complex installation was a big obstacle in the past which prevented a wide spreading of mobile applications. Usually for this a PC with for the mobile phone suitable configuration software was necessary, with which the mobile phone was connected by a data cable. For mobile Java applications there is another further alternative, which is favored in particular by the mobile games market. Here the installation of new MIDlets is at any time at each place within shortest time possible, always when the user needs certain programs for its mobile phone. This is reached by the download of the desired MIDlet over a UMTS or a GPRS connection. The necessary URL for this receives the user either from the browser of the mobile phone or by SMS. In addition such a call is also directly possible from a MIDlet. The development of specialized Part-MIDlets, for example for different equipment variants of a vehicle, is now possible which are downloaded on demand directly to the user's mobile phone.

The protocol for such a Over The Air (OTA) transmission is HTTP. Communication over HTTP is a firm component of MIDP and thus the standard technique for the data communication of MIDlets. The support of further protocols is however optional. In addition MIDlets offer with the method `platformRequest(string URL)` a standard procedure for the download of new programs over a HTTP connection.

Apart from the MIDP specification the optional content Handler API (JSR 211) contains also this functionality. Duty of the content Handler API is actually to pass certain tasks to other programs. For example playing music at the on the mobile phone installed media player. However it can be also used to download and to install new programs on the device.

With this kind of installation the appropriate jad and jar file must be on a web server reachable for the mobile device. In the jad file thereby to the location of the jar file is referred.

4.6 Security

In MIDP there is an extensive security concept, which on the public key procedure for the verification and authentication of MIDlet Suites is based. This security concept serves the preventing of, the use of sensitive operations, like for example the establishment of a expensive network connection, without preventing the knowledge of the user. So that a signed MIDlet can get access to a sensitive API, the appropriate permission must be set. This permission is indicated in the jad file.

In MIDP there are so-called Protection Domains which MIDlets are assigned to. In the Protection Domains is specified how to deal with the permissions.

There are the following Protection Domains:

- minimum: MIDlets of these Protection Domain, access to all Permissions is refused.
- untrusted: The user must give his agreement with each call to an API proteceted by a Permission of these Protection Domain. This is the default domain for unsigned MIDlets.
- trusted/maximum: The access to all Permissions of this Protection Domain is permitted.

One frequently still differentiates with trusted Protection Domains according to the certification authority:

- manufacturer: Uses certificates of the device manufacturer.
- operator: Uses certificates of the network provider.
- trusted third party: Uses third party certificates.

With the permissions two types are differentiated:

- allowed: The access is permitted without demand of the user.
- user: The user must give his agreement for the call of the associated API.

With user Permissions between the following types one differentiates:

- oneshot: Inquire with each call.
- session: Once inquire, decision remains valid as long as MIDlets of these MIDlet Suite are active.
- blanket: Once inquired, decision remains valid as long as the MIDlet Suite is installed.

If a MIDlet is in the trusted Protection Domain and the type of Permission is allowed, then it can use the associated API without demand of the user.

To which Protection Domain a MIDlet Suite belongs depends on the root certificate existing on the devices. With the installation the signature of the MIDlets is compared with the existing root certificates and accordingly a classification is made.

5. Vehicle integration

Cars are usually products, which come from one hand, from the car manufacturer. The offerers of accessory components so-called off board devices have a not insignificant problem, since usually no standard interfaces for the integration of these devices are present or must be licensed by the vehicle manufacturer. But even if such a license and the necessary installation interfaces are present, still the problem of the user interface remains for the offerer of accessory components. These are frequently goods in short supply and reserved for the OEM (Original Equipment Manufacturer) in the vehicle. From there the accessory offerers mostly offer their own control elements, which are expenditure-stuck or stuck on the instrument panel. Apart from the optical lack that control elements does not fit the design and cables lay partly openly, remains the problem, that these control elements do not fit into the control concept of the vehicle.

There is however one off board device, which is accepted by practically all car manufacturers and for both, interfaces for the integration in the vehicle and a firm place in the instrument panel is present. In addition it is suitable outstanding as universal control element for a multiplicity of devices. Meant here is the mobile phone.

Mobile phones are suitable on the one hand so well, because they possess many communication interfaces, beside the mandatory GSM, GPRS, UMTS support they frequently have Bluetooth and some models even WLAN interfaces. The employment of wireless technologies makes besides the cable to the control elements redundantly. The suitable communication technology can be selected depending upon application. For vehicle-internal communication a short range technology is sufficient as for example Bluetooth. However even if a genuine remote maintenance is to be realized over far distances a UMTS or a GPRS connection offers itself for this.

On the other hand mobile phones can be programmed almost at will, so that control applications for the most diverse devices can be realized. The advantages of the Java Micro edition in this area were stated already in detail.

5.1 Example auxiliary heating

How the integration into a vehicle is in detail realized is to be described in the following by the example of a auxiliary heating. The auxiliary heating is installed in the vehicle and attached to the CAN (Controller Area Network) bus of the car, over which all controllers are interlaced and receive their instructions. The instructions come of one at the instrument panel fastened or into it inserted, control element which is likewise connected with the CAN bus. Instead of this control element or also as addition of it now a mobile phone is to be used.

In principle for this UMTS/GPRS and Bluetooth present themselves as communication technology. Bluetooth for communication within the car and UMTS/GPRS for the remote maintenance from the domestic living room. Since the integration is very similar in both cases and Bluetooth besides brings the standardized communication profiles with it, contains the following example for the sake of simplicity only to Bluetooth. Following Fig. 12 outlines the fundamental structure of such a system.

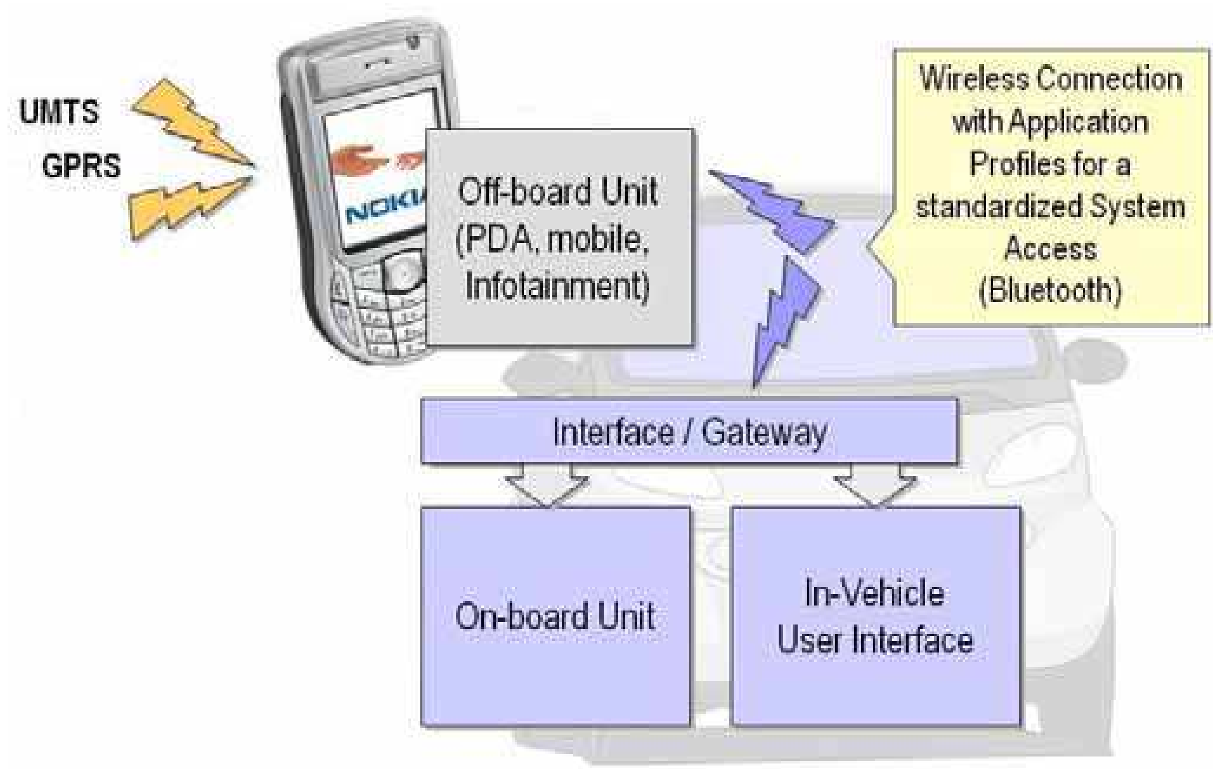

Fig. 12. Vehicle integration

The auxiliary heating and its control elements communicate no longer directly over CAN bus with each another, but over an interface or a gateway. The gateway controls the data transfer in the vehicle and passes the data on to the respective control devices. In the concrete example the gateway has a Bluetooth SPP connection to the mobile phone, over that it transfers the instructions of the remote control unit.

On the mobile phone a Java MIDlet runs, which the user downloaded ideally-proved directly from the Web server of the auxiliary heating manufacturer and installed it afterwards on his device. Security is ensured thereby by an appropriate signature of the MIDlets, which regalements the access to resources of the mobile phone e.g. communication interfaces and memory.

Even the selection of a suitable MIDlet for the vehicle-auxiliary-heating-mobile-phone-combination can be automated to a large extent, if device type and Bluetooth address of the user are deposited on a central server. This is can be done by a service technician for example with the installation.

The scenario to the deployment of the application has the following in Fig. 13 described expiration.

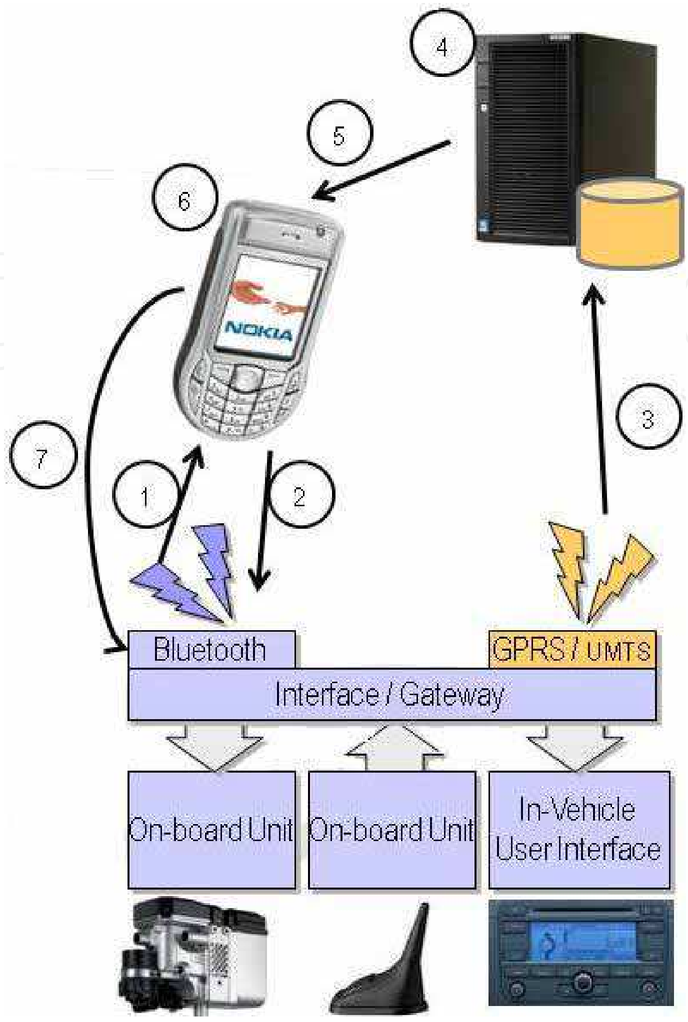

Fig. 13. Automatic Application Deployment

After the auxiliary heating was installed in the vehicle and the service technician has deposited the Bluetooth address, the telephone number and the type of device on the server starts the scenario.

1. The vehicle starts a search for Bluetooth devices in the environment. It acts around a functionality of the Bluetooth standard.
2. The found devices convey their Bluetooth address for later identification.
3. A list of the found devices is sent over a GPRS/UMTS connection to the server.
4. The server examined on the basis the Bluetooth addresses whether it for one of the found devices an order has, and selects on the basis the deposited type information a MIDlet fitting to the device type.
5. The server sends a SMS with the appropriate download link to mobile phone. Subsequently, the user opens the link and downloads the MIDlet to his mobile phone.
6. Subsequently, automatically the installation procedure begins. The user now only has to agree with the installation.

7. After finishing the installation the MIDlet is started directly and can be connected by Bluetooth with the auxiliary heating, in dependence of its signature. The application is ready for use thereby and the user can control his auxiliary heating.

6. Conclusion

The chapter showed that wireless communication already belongs in many areas of the automotive environment to the state of the art. Straight development possibilities further with the integration of mobile phones are present nevertheless. Bluetooth presents itself here as almost ideal communication technology for many applications.

For the development of mobile phone applications the Java Micro Edition is first choice, it offers not only large platform independence, but also detailed concepts to the deployment of applications or for security. In addition APIs are available for all usual communication technologies.

The mobile phone is the only off board device accepted by car manufacturers. That makes it interesting for the manufacturers of accessory components to use these as control elements. A concept for this was explained in the chapter.

7. References

Bluetooth SIG (2009). www.bluetooth.org
Breymann, U. & Mosemann H. (2008). *Java ME. Anwendungsentwicklung für Handys, PDA und Co (Germann)*, Hanser, 3446229973, Munich
JCP (2008). http://jcp.org/en/jsr/detail?id=82
JCP (2003). http://jcp.org/en/jsr/detail?id=120
JCP (2004). http://jcp.org/en/jsr/detail?id=205
IEEE (2002). http://ieee802.org/15/pub/TG1.html
Sun Microsystems (2009). http://java.sun.com/

Mobile and Wireless Communications Network Layer and Circuit Level Design

Edited by Salma Ait Fares and Fumiyuki Adachi

ISBN 978-953-307-042-1
Hard cover, 404 pages
Publisher InTech
Published online 01, January, 2010
Published in print edition January, 2010

Mobile and wireless communications applications have a clear impact on improving the humanity wellbeing. From cell phones to wireless internet to home and office devices, most of the applications are converted from wired into wireless communication. Smart and advanced wireless communication environments represent the future technology and evolutionary development step in homes, hospitals, industrial, vehicular and transportation systems. A very appealing research area in these environments has been the wireless ad hoc, sensor and mesh networks. These networks rely on ultra low powered processing nodes that sense surrounding environment temperature, pressure, humidity, motion or chemical hazards, etc. Moreover, the radio frequency (RF) transceiver nodes of such networks require the design of transmitter and receiver equipped with high performance building blocks including antennas, power and low noise amplifiers, mixers and voltage controlled oscillators. Nowadays, the researchers are facing several challenges to design such building blocks while complying with ultra low power consumption, small area and high performance constraints. CMOS technology represents an excellent candidate to facilitate the integration of the whole transceiver on a single chip. However, several challenges have to be tackled while designing and using nanoscale CMOS technologies and require innovative idea from researchers and circuits designers. While major researchers and applications have been focusing on RF wireless communication, optical wireless communication based system has started to draw some attention from researchers for a terrestrial system as well as for aerial and satellite terminals. This renewed interested in optical wireless communications is driven by several advantages such as no licensing requirements policy, no RF radiation hazards, and no need to dig up roads besides its large bandwidth and low power consumption. This second part of the book, Mobile and Wireless Communications: Key Technologies and Future Applications, covers the recent development in ad hoc and sensor networks, the implementation of state of the art of wireless transceivers building blocks and recent development on optical wireless communication systems. We hope that this book will be useful for students, researchers and practitioners in their research studies.