

Structure of Java Applications for Telemedicine

Blanka Filipová¹, Marek Penhaker¹, Ondrej Vlcek¹ and Martin Cerny¹

1 VSB – Technical university of Ostrava, 17. listopadu 15, 70833 Ostrava, Czech Republic

Abstract. This article gives insights into the technologies needed to develop Java Bluetooth applications for mobile devices. This technology is very useful now, because the implementation of Java into mobile devices is more than natural. Bluetooth communication interface and the structure of Java 2 Micro Edition are described. The standard scheme of Java application and an example of application is described.

Keywords: Bluetooth, Java, Java Micro Edition, Mobile Phone, MIDlet

1. Introduction

Wireless technology is enjoyed by people becoming more popular. There are several reasons. Contribute to the greater mobility devices, eliminate clutter of cables, get to the places where the concentration is another way impossible or inappropriate, and last but not least, just now "wears" an electronic gizmo that can wirelessly communicate with the outside world.

One of the many available technologies as Bluetooth is. The development of relatively old, approximately 20 years on the technology quite a long time; culminating in the last few years. This is due to increased compatibility of different manufacturers, but also a substantial reduction in prices of chipsets and the massive expansion in consumer electronics. Today, Bluetooth is used by hundreds of different devices, from PCs, through personal assistants, navigation systems, mobile phones, to home appliances. The total number of chipsets has produced over 100 million per year.

For mobile users are not limited to software that will give them the instrument manufacturer, it was necessary to establish a universally applicable phones programming language that would not require heavy demands on performance and main memory devices. As a candidate, showed adequate programming language Java in, its simplified version of J2ME. Since Java is undergoing constant development and programmers can then enjoy the news, users in the form of improved controls, graphics and sound for your application.

The work benefits from the combination of these two technologies and delivering new resources for physicians and patients to monitor vital functions such as pressure, heart rate, etc. Possible applications are very wide and depend only on the imagination of the programmer. Currently available with a pressure gauge installed Bluetooth module, which can wirelessly communicate with applications on your PC or mobile phone.

There is another possibility that the development and miniaturization of measurement devices with no built-in BT module, which the patient will be easily worn on the body, and his cell phone to check as heart function in an emergency situation automatically call the doctor.

2. Bluetooth

Bluetooth technology was designed by Ericsson in 1994. Its primary purpose was to replace cables between mobile phones and accessories. Another use for the connection of several mobile phones, PDAs,

and ultimately personal and portable computers did not take long. The first commercial products saw the light of the full five years later in 1999.

BT architecture consisting of several layers. HCI layer is usually a guidepost between hardware and software parts of the instrument. Exceptions are simpler devices (eg hands-free kit), mostly composed of layers of hardware of the device.

Although not a programmer about the layers complete or almost any survey must be at least understand the basic hardware features. In the same frequency band is working as microwave ovens or WiFi transmitter and is therefore a need to allow for any interference.

3. Java

Java is a language with a very wide range of application from small handheld devices to complex network and database applications. To be able to successfully use the equipment are different machines, it was necessary to specify the number of categories of language - editions.

- J2EE - Enterprise Edition, designed for servers and large desktop applications
- J2SE - Standard Edition for normal desktop use

• J2ME - Micro Edition for embedded devices, PDAs, mobile phones, fax machines, cash registers, navigation systems, hi-fi equipment, home appliances. To carry out this work was given to the J2ME application selected editions. To zoom in and should be divided into two parts J2ME as technically possible facilities which will run Java. Java has a layer which is called configuration. Configuration we have for J2ME two.

Configuration	Virtual Machine	Processor	Memory for Java	Power
CDC	JVM	32 bit	2 MB	Usually from network
CLDC	KVM	16 bit 16 MHz	160-512 kB	Usually from battery, from network

Fig. 2: Configuration Requirements

Configuring therefore comprises the requirements for host system. Then determines the type of VM and a list of basic library owned by the default configuration must support.

CDC (Connected Device Configuration) configuration, which aims to segment more complex devices, usually powered from the network. As an example, television shows, navigation systems, "white" appliances, as well as some mobile phones (from Nokia communicators).

On the contrary, simpler and less demanding on the hardware configuration CLDC (Connected Limited Device Configuration), almost all the Java phones, including smart fellow - Smart Phone or PDA. CLDC does not already standard JVM, but simpler KVM (Kilo Virtual Machine). It got its name because the application processes in the sizes of units or tens of kilobytes. The important fact is that the CLDC does not use taking other libraries to ensure functionality at all CLDC devices.

3.1. CLDC Libraries

Configuration CLDC has undergone since its inception, the next evolution from version 1.0 to version 1.1. Differences should not be too large to be as much as possible to maintain backward compatibility. The only noticeable change is the addition of a data type with a floating decimal point. This raised the minimum memory requirement for the Java environment from 160 to 192 KB. At the present time, but still the majority of mobile phone configuration only supports 1.0.

CLDC consists of a set of packages listed in the table. Apart from the last of them are subgroups of the classical library of known standard edition J2SE. Java.io does not, as in J2SE, all libraries taking care of input / output data streams. Working with files and data permanently stored cater just javax.microedition.io library package.

package	functionality
java.io	Classes for in/out data streams
java.lang	Basic classes of Java in J2ME without any classes and functions
java.util	Classes and collections for time and date processing času
javax.microedition.io	classes GCF

Fig. 3: Configuration CLDC Packages

3.2. Mobile Information Device Profile (MIDP)

The last link in the chain of Figure 3-1 shows a profile. A profile is a set of API commands created for the configuration used to simplify the use of the functions of individual devices and their functions. I will use in their work profile MIDP (Mobile Information Device Profile).

Currently there are defined two versions of MIDP profile. Older MIDP 1.0 [11] and the newer MIDP 2.0. The only phone that can use Java to control the BT while only supports MIDP 1.0 was at the time of the Siemens SX1. All others are equipped with a newer version of the profile.

MIDP 1.0: This profile marked by its development team (Java Community Process Expert Group) as JSR-37 API contains additional libraries that form the superstructure of the CLDC configuration. Applications built for this profile is called a MIDlet. To this device can run MIDlets must meet the minimum configuration

MIDP 1.0 CLDC added to the library packages: Description of the package with javax.microedition.lcdui contains user interface classes and javax.microedition.midlet Defines the MIDlet and the interaction between the application and its runtime environment. Also it has javax.microedition.rms Records Management System provides a permanent data storage.

Application Management, then uploading them to the device, installing, executing, and deleting is not part of the MIDP 1.0 specification. Care for the operating system device. Therefore, it can vary from manufacturer to manufacturer. Must be maintained, but supporting all the standards.

MIDP 2.0: The new version of the MIDP has been developed as JSR-118th Backward compatibility is maintained. All MIDP 1.0 MIDlets then work on MIDP 2.0. Conversely, of course, does not. Change from MIDP 1.0 is a greater need for memory and runtime environment for the application. Further, the requirement for sound reproduction.

Some packages were compared with an earlier version of the amended or supplemented by additional features to improve control applications (lcdui). He added game package that includes instructions for easier handling and display graphics and simple game engine Teke. ABB offers a package of media (audio building block), through which the device can handle audio files.

Managing the MIDlet version 2.0 specification moved to profile. First, because the phone manufacturers to make work easier, but also improves compatibility.

MIDP 2.0 also provides support such as HTTPS (web services), communication via serial port, the push architecture, the MIDlet can be run remotely using a configured server, and many other interesting features.

3.3. MIDlets

As already mentioned in the previous chapter, the MIDlet is a Java application using CLDC configuration and MIDP profile extension.

When we look at the architecture of the mobile phone site for software, its core operating system layer, of course. It's either a firmware developed for the device or a standard operating system such as the popular Symbian. Each phone is programmed so the suggested equipment for the operating system. To zoom in, it's all the alarm clock, stopwatch, image viewer and other features of the type of phone.

Other applications are those which, while running Java (CLDC are valid applications), but use other non-standard libraries developed by the OEM manufacturer of the phone. In this case then becomes Java applications such as Nokia's broken in your Sony Ericsson. In a typical Nokia API library. These applications are sometimes referred to as MIDlets, even though they really are.

The last type of standard MIDlet applications are fully supporting the CLDC configuration, or using other API libraries MIDP profile. These applications must already be compatible with all OEM manufacturers.

3.4. MIDlet suite

When compiling MIDlet application is to create a so-called MIDlet suite, two related files together. Jar and. Jad. First, we are already well-known compressed package one or more classes and complete Java Archive. Jad (Java Application Descriptor) is a small file that contains all the necessary information about the MIDlet. These include the type of configuration and profile, the size of jar archive version, the High Street address and spring. Usually the device before downloading the JAR archive loading only information JAD file and check whether the application is valid for the version of the runtime environment and also has enough memory.

Java APIs for Bluetooth Wireless Technology (JABWT)

If we wanted to include JABWT, would be placed in the same field with the MIDP profile. It is the extension of the API library controlling BT interface and given the designation JSR-82. To use your communication streams as defined in the CLDC GCF class.

Javax.bluetooth provides functions such as device inquiry and service discovery. It also creates a connection-based BT L2CAP and RFCOMM layers.

Javax.obex OBject Exchange is an implementation of orders. Used to exchange contacts, multimedia, software, and synchronization between devices. While these libraries are no cell phones do not. This service is obviously working as a Bluetooth, infrared and most appliances. But it is thanks to the implementation in the operating system phones and Java can not access them.

Most novice programmers do not understand why the device can not work with a written application, although supports Java and has the BT interface. The reason is similar to the obex. It is not enough alone to support Java (MIDP profile accurately) and BT's own interface. The device must also support JSR-82 Now, a library javax.bluetooth. Otherwise, using the Java interface is unable to proceed.

4. Application in telemetry

For the development of telemedicine applications are made for two basic requirements. Use the resources available to the Java implementation of all relevant functions such as databases of patients with a history of measured data, text and graphical evaluation of measurements, etc. Another equally important requirement was to design an application so that in the future as easily upgradeable to the possibility of working with other devices .

Thanks to Java architecture can meet the second requirement gracefully. The core application consists of classes that are responsible for controlling, setting and communicating via the Bluetooth interface. Each device has a separate class that uses the core of the program to communicate with others and care only for evaluation of the data, and graphical user interface to control the medical device.

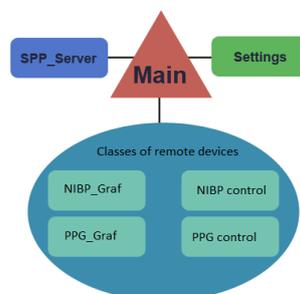


Fig. 3: Structure of model application

Main is the base class of applications. He is responsible for creating the associated object classes. The report also contains a graphical environment for application instructions.

SPP_Seiver is a class that provides the actual communication over the BT interface. Handles the connection to the surrounding devices. It sends and reads data. Index data for the mission, as well as processing incoming data, providing classes nearby devices. SPP_Server is only a facilitator of communication.

Settings is a class with which the user can change settings.

- NIBP is a class with a graphical interface to control the non invasive blood pressure monitor. Managed database of measured data for each patient and store them in memory.
- BPM_Graf plots of the data stored curve measurement history
- PPG class is working with Pulse oximetry device, a device measuring pulse, blood oxygen saturation and pletysmo curve.
- PPG_Graf is a visualization of the interface for working with Pulse oximetry design.
- This design could be used for application of Java in Telemetry application in mobile devices as a model structure for Java applications.

5. Conclusion

The article deals about applications in telemetry supported by Java language for implementation in the mobile devices. The Bluetooth technology has been chosen for data transmission, because it is very often used technology in all mobile devices. The possibilities of Java has been described. It is very useful language, which is often used for mobile applications. The application for telemetry is not so often. On the basics of Java this article presented model structure for Java application for mobile phone telemetry with several medical devices.

6. Acknowledgements

The work and the contribution were supported by the project: Ministry of Education of the Czech Republic under Project 1M0567 “Centre of applied Cybernetics”, student grant agency SV 4501141 “Biomedical engineering systems VII” and TACR TA01010632 “SCADA system for control and measurement of process in real time”.

7. References

- [1] R. Dewri, and N. Chakraborti. Simulating recrystallization through cellular automata and genetic algorithms. *Modelling Simul. Mater. Sci. Eng.* 2005, **13** (3): 173-183.
- [2] A. Gray. *Modern Differential Geometry*. CRE Press, 1998.
- [3] O. Krejcar, Problem Solving of Low Data Throughput on Mobile Devices by Artefacts Prebuffering. *EURASIP Journal on Wireless Communications and Networking*, 2009, Article ID 802523, 8 pages. Hindawi publishing corp., New York, USA, DOI 10.1155/2009/802523
- [4] Kasik, V.: Acceleration of Backtracking Algorithm with FPGA. In 2010 International Conference on Applied Electronics, pp. 149-152, Pilsen, Czech Republic, 2010, ISBN 978-80-7043-865-7, ISSN 1803-7232
- [5] O. Krejcar, R. Frischer, Detection of Internal Defects of Material on the Basis of Performance Spectral Density Analysis, *Journal of Vibroengineering*, 2010 – Vol. 12, No. 4 - pp. 541-551.
- [6] O. Krejcar, D. Janckulik, L. Motalova, Complex Biomedical System with Biotelemetric Monitoring of Life Functions. In *Proceedings of the IEEE Eurocon 2009*, May 18-23, 2009, St. Petersburg, Russia. pp. 138-141. DOI 10.1109/EURCON.2009.5167618
- [7] Kasik, V.: FPGA-powered Embedded Vector Graphics. In conference proceedings UBICOMM 2008, Valencia, Spain, ISBN 978-0-7695-3367-4, DOI: 10.1109/UBICOMM.2008.46.
- [8] Kasik, V., Stankus, M.: Graphical development system design for creating the FPGA-based applications in biomedicine. *XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010*. IFMBE Proceedings, 2010, Volume 29, Part 5, pp. 879-882, DOI: 10.1007/978-3-642-13039-7_222.