34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Integration of DASH and ICS 3000 Devices with Hospital Information System and REPACE Central Registry*

K. Vlach, J. Jirka, J. Černohorský

Abstract— **Paper describes the system that is developed by the authors and which allows interconnection of vital signs monitor DASH, implant control system ICS 3000, HIS (Hospital Information System) used at Městská nemocnice Ostrava and REPACE the central registry of pacemaker implantations. This connection allows users to effectively create, store and visualize operation reports with patient's data. The new system is being developed using C# programming language within .NET environment. The developed application's title is ImplantSys.**

I. Introduction

The main goal of this project is to create a system that helps physicians at the cardio stimulation department of the hospital in Ostrava (*Městská nemocnice Ostrava*). After the operation, which is most often the implantation of a pacemaker, there is a need for physicians to make a report. This report contains various information like the patient's identification data or the measured signals from devices. Then it has to be stored in the nationwide central registry named **REPACE**.

There are four sources of data that the new developed system **ImplantSys** has to combine for the most effective workflow of the operational team. The hospital **HIS** (*Hospital Information System* aka *EHR – Electronic Health Records*) which contains patient's administration data, **DASH** vital signs monitor used mainly for measuring ECG *(Electrocardiogram*), implant control system **ICS 3000** mainly used for measuring IEGM (*Intracardiac Electrogram*) and programming the pacemaker, and the **REPACE** central registry.

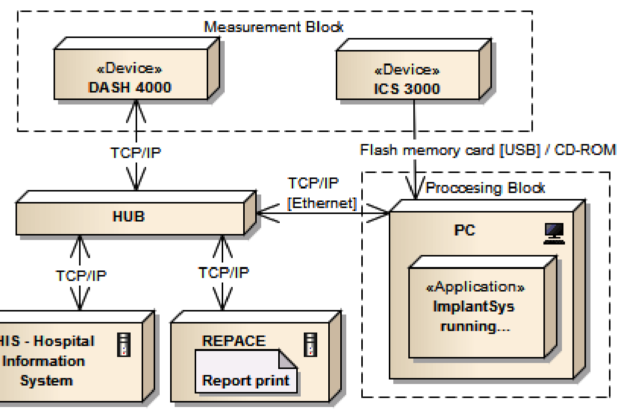
From this the **ImplantSys** allows the physicians to work faster and more effectively with the patient's records in a user-friendly way. It will be the first system that allows the mentioned features.

II. The New System

The **ImplantSys** application is running on the PC in the block of processing (see Fig. 1). There is a HUB that connects HIS, **DASH** and **REPACE** via an Ethernet interface. The **ICS3000** does not support online connection in real-time. Therefore the data from the **ICS3000** is transferred via one of the supported media (USB flash drive or CD) in XML format. Communication with the **REPACE** registry is via the internet.

Figure 1. Physical interconnection architecture

The data from devices and other connected systems can then be imported into the application safely, reliably and



above all automatically. This is much more efficient compared to the current method of manual rewriting.

III. Hospital Information System

There is important data in **HIS** which needs to be imported to the **ImplantSys**. Such data includes patient's name, birthdate, gender, weight, height, diagnosis, address, insurance and other relevant information. This will help the physician to automatically fill in the form of the report and prevent them from making a potential mistake, which can happen while rewriting the data manually. The data can be also inserted manually. The user can add, edit or delete the records and sees all the records in the sortable table as shown on Fig. 2. Therefore the application can be used as a patients management system – database.

The name of the in service HIS is **CLINICOM**. **CLINICOM** is a robust hospital information system with a modular design and an object-relational database *Caché* from the *InterSystems* company [1]. It is used by several hospitals in Czech Republic [2].

Communication between the **ImplantSys** and the **CLINICOM** database is solved by the interface implemented as a dynamic-link library in C#. This interface contains several methods for getting the relevant data from

*Research supported by FRVS of Ministry of Education of the Czech Republic under project 540/2012 and by Ministry of Education of the Czech Republic under project 1M0567 and by SGS SP2012/75.

K. Vlach is with the Department of Cybernetics and Biomedical Engineering, VŠB-Technical University of Ostrava, 17. listopadu 15, Ostrava, 70800, Czech Republic (e-mail: karel.vlach@vsb.cz; karel1.vlach@gmail.com).
J. Jirka is with the Department of Cybernetics and Biomedical Engineering, VŠB-Technical University of Ostrava, 17. listopadu 15, Ostrava, 70800, Czech Republic (e-mail: jakub.jirka1@vsb.cz).
J. Černohorský is with the Department of Cybernetics and Biomedical Engineering, VŠB-Technical University of Ostrava, 17. listopadu 15, Ostrava, 70800, Czech Republic (e-mail: jindrich.cernohorsky@vsb.cz).

the database to the **ImplantSys** application. The form for the report creation is filled in with the information and can then be edited. Relevant data is picked from the database by a method in which the personal identification number serves as the primary key for selection. If there is more than one record associated with the patient another key for selection is used. Date/time is the secondary key.

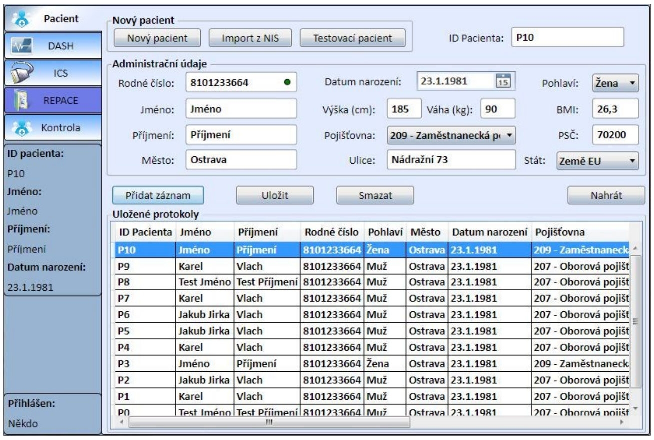


Figure 2. ImplantSys screen with patients management

IV. Vital Signs Monitor DASH

Various vital and electric parameters are measured during the pacemaker implantation. External and internal ECGs are the primary parameters of those vital signs. Other parameters such as respiration, O_2 saturation, blood pressure and others are observed only in more severe cases of a patient's condition. The vital signs monitor **DASH 4000** is used for the measurement.

A. Communication Protocol

The **DASH** devices (x000 series, where x is 1, 2, 3, 4, 5) and also the vital signs monitor **EAGLE** uses a closed protocol of the "Unity Network" [3]. There are three known types of UDP messages: *RWHAT*, *BEDSIDE* and *WAVEFORM*.

1) RWHAT message

This message serves to identify all devices on the Marquette network. Each device sends this message typically every 10 seconds. Each device involved maintains a list of these devices.

2) BEDSIDE message

This message serves to call the functions of the devices and to change the parameters. There are two types of the message - a request ("request" packet) and answer ("response" packet). The requests are used for reading and writing of the parameters, or to request a subscription ("subscribe") of the parameters and waveforms in real-time. The answer then contains the specific parameters or results of operations.

3) WAVEFORM message

This message serves to transmit the monitored waveforms in real-time. The specification of the message is not complete yet. However, there is a functional way to handle such messages and process them.

The vital signs monitor **DASH** filters out these waveforms and computes another three (1) from them automatically. But the **DASH** only displays them and does not send them to the PC so the **ImplantSys** must compute these as well. These three computed waveforms are: aVR, aVL, and aVF, which are augmented limb leads [4].

$$aVR = -\frac{I + II}{2},\ aVL = I - \frac{II}{2},\quad aVF = II - \frac{I}{2}. \tag{1}$$

For long-term remote monitoring by the application, it was necessary to create a driver that allows a reliable connection to the **DASH** device in real-time. The **ImplantSys** uses advanced design technology with regard to the real-time data processing [5]. Based on the described specification, the driver for patient monitor **DASH** was developed in the .NET framework [6].

Developed ***TMqExplorer*** component is used to detect devices on the network. The main component is ***TMqDevice*** which contains information about supported services ***TMqService*** and provides an interface for the collection of the waveforms of the measured signals ***TMqWaveform*** and for update of the parameters ***TMqParameters***. ***TMqWaveform*** component organizes the measured signals into the individual channels ***TMqChannel*** and each channel is then formed by samples ***TMqSample***.

The direct usage of these classes is too complex so that is why another component was created - ***TSimpleDASHReader***. This component is specifically designed to collect the waveforms from the **DASH** monitor at the constant sampling rate.

The class allows it to:

1. Connect to the target monitor based on its IP address.
2. Get a list of all currently measured channels from the monitor.
3. Continuously measure the selected channels. Measured waveforms are returned as a table of values (channel, time). Possible error conditions and failures in the measurement are replaced by the constant value NaN.

In **ImplantSys** many ECG recordings can be done for every patient (see Fig. 3). The user can see in real-time the exact size, actual length and filename where the data are saved. It is possible to select the leads to show, add comment, take screenshots, zoom, etc. After the recording user can load and show or delete existing records. Every record is automatically compressed with GZip technology to reduce the amount of bytes which every file has.

Figure 3. ImplantSys screen with patient's ECG records

V. Implant Control System ICS 3000

Biotronik **ICS3000** serves as a mobile implant control system that helps surgeons to fully control an implanted pacemaker. Also it is used for the measurement of electrical parameters and as the external pacemaker. As for the measured electrical variables, there are, for example IECG, voltage threshold, impedance of the electrodes and others.

In order to integrate the **ICS3000** device and the patient data into the **ImplantSys** application, an interface component needs to be implemented. Once this component is used the software is able to extract, transform and translate the internationally standardized *Biotronik IEEE 11073-10103 XML* data and programmer data into the **ImplantSys** software or hospital's **EHR** system.

The interface component is programmed in .NET 3.5, .NET 4.0 in a mixed assembly manner and compiled as a dynamic library with a public API (*Application Programming Interface*) that can be used by any .NET language.

Data export process and integration into the **ImplantSys** application is done with respect to the specification provided by *Biotronik* [7]. In order to read **ICS3000** exported data, transformation of this data into the internationally standardized format needs to be performed. This transformation from *Biotronik* programmer XML file format to internationally standardized *Biotronik IEEE 11073-10103 XML* is done via a command line application called Programmer Adapter which is provided freely from the *Biotronik* website [8].

A. Customized Adapter Implementation

The customized adapter is implemented in C# .NET 3.5 and .NET 4.0 mixed assemblies and compiled as a dynamic library included into **ImplantSys** application.

The customized adapter's XML parser expects either *IEEE 11073-10103 XML* format or ICS programmer XML format. In the case of ICS programmer XML format, XML is transformed into IEEE format by a *Biotronik* Programmer Adapter command line utility [8]. Encoded backup zip file reading is not implemented yet as *chilkat* component is not freely available and the password protection of these backups is therefore unknown.

Customized adapter's XML parser is using a language-integrated query to parse and process XML stored data. All data stored in the XML format is described in the document provided by *Biotronik* [9]. Parsed XML data is mapped into object structures. Customized adapter is working as a *Biotronik* XML to object mapper.

VI. Central Registry REPACE

REPACE is the name of the nationwide central registry developed by the Institute of Biostatistics and Analyses at the Faculty of Medicine and the Faculty of Science of the Masaryk University, Brno, Czech Republic. This registry contains reports from the implantations of pacemakers and other relevant operations such as re-implantations.

Reports contain information about the operation such as patient's data, implanted devices, electrodes, complications, measured data and the names of the operational personnel. All this data is inserted via the screen with forms in **ImplantSys** as shown on Fig. 4.

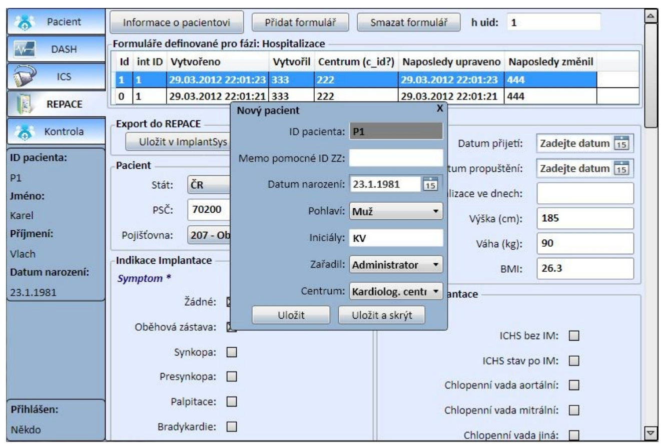


Figure 4. ImplantSys screen with REPACE forms

Most of this data **ImplantSys** can obtain automatically – from the **CLINICOM**, **DASH** and **ICS3000**. An option to export the data to the **REPACE** is another one of the developed main features. The implementation lies in .NET interface which enables creation of XML file with patient's data. This file is then sent to the **REPACE** system online where it is imported to the *ORACLE* database running under the **REPACE**.

There is an ongoing development of the API which enables real-time administration of records.

VII. Conclusion

ImplantSys is very helpful as it connects four sources of information and combines it into one framework where the user – the physician – can effectively work with it in the modern GUI (*Graphical User Interface*).

Every part of the system can be used alone for developing other applications. For example there can be a simple application used in ambulances car to store important vital signs during patient transfer where the **DASH** is used. This data can later be used for analysis. Another software application can evaluate the length of the interval between two pulses from the pacemaker. Longer intervals mean that the battery is discharging. This software can be used to determine the status of the battery. This information is very important not only in the case that the patient is pacemaker dependent.

There is also the possibility of using the concepts of the **ImplantSys** for making ECG, SpO_2, and blood pressure classifiers with neuro-fuzzy or other methods approach. Also getting data from devices that are **DASH** communication protocol compatible (vital signs monitors) allows for usage as the central where every device and its data can be visualized, stored and prepared for the next processing step.

Other applications can be developed with the **ICS300** dynamic library or the **REPACE** interface anywhere it is needed. From this point of view this is the most valuable contribution of this whole project and its main system the **ImplantSys**.

ACKNOWLEDGMENT

Authors would like to thank funding agencies which are stated in the unnumbered footnote on the first page.

REFERENCES

[1] CompuGroup, "CareCenter G3 NIS CLINICOM," [online], c2010, [cit. 2011-30-03]. <http://www.compugroup.cz/underwood/download/files/cc_g3_cz_cg_newdesign_v01.pdf>

[2] SMS spol s.r.o., "Reference SMS system implementation," [online], c2005, [cit. 2011-30-03]. <http://test.nlogy.cz/smsbrno/public/reference.html>

[3] V. Michna, K. Vlach, "System for monitoring pacemaker implantation," in *Proceedings of 8th International PhD Student's Workshop on Control and Information Technology – IWCIT 2009,* Brno: Brno University of Technology, 2009. pp. 97-102, (ISBN 978-80-214-3949-8).

[4] Wikipedia, "Electrocardiography," [online], c2007, last revision 4th of April 2011 [cit. 2011-30-03]. <https://secure.wikimedia.org/wikipedia/en/wiki/Electrocardiography>

[5] J. Cernohorsky, J. Koziorek, G. Hrudka, "The examples of implementation of two specialized case tools as extensions of a general framework," In *IECON 2006 – 32nd Annual Conference on IEEE Industrial Electronics 2006,* New York: IEEE, 2006, Vols 1-11., pp. 4909-4914, (ISSN: 1553-572X, ISBN: 978-1-4244-0135-2).

[6] V. Michna, B. Babusiak, "Zobrazování a analýza EKG z pacientského monitoru DASH," *Mezinárodní konference Technical Computing Prague 2009*, HUMUSOFT s.r.o., 2009, pp. 72, (ISBN 978-80-7080-733-0). <http://www.humusoft.cz/akce/matlab09/sbornik/>

[7] Biotronik SE & Co. KG, "IT-Specialist Guide for integrating BIOTRONIK Home Monitoring® and programmer data into hospital Electronic Health Records system," [online], c2009, [cit. 2011-30-03]. <http://91.204.11.72/sixcms/media.php/211/EHR%20Tech%20Guide.pdf>

[8] Biotronik SE & Co. KG. "Technical documentation for EHR Integration of BIOTRONIK programmer data," [online], c2009, [cit. 2011-30-03]. <http://www.biotronik.com/biohm/technical-documentation-for-ehr-integration-of-biotronik-programmer-data/28769>

[9] Biotronik SE & Co. KG, "Documentation of the BIOTRONIK IEEE 11073-10103 XML Structure," [online], c2009, last revision 7th of February 2010 [cit. 2011-30-03]. <http://www.biotronik.com/sixcms/media.php/211/BIOTRONIK_IEEE_11073-10103_XML_Structure.pdf>