Realization of Universal HW/SW Module for Integration of Medical Laboratory Devices into Medical Information System

Aleksandar Milenković[1], Dragan Janković[2], Marija Stojković[3], Aleksandar Veljanovski[4] and Petar Rajković[5]

Abstract – **This paper presents a proposal of the universal hardware-software module that enables the integration of laboratory equipment of a health care institution's and medical information system (MIS). Emphasis is particularly given to the connection of the analyzers in biochemical laboratory of a health care institution with MIS, but proposed concept can be used for integration and other laboratory devices in health care facilities with MIS and beyond. The proposed solution is cheap and allows the integration of heterogeneous laboratory devices in MIS in a uniform way. The solution is based on *XML* messages that are generated from the proposed *XSD* schemas. The three *XSD* schemas were proposed: *MessageRequest*, *MessageRequestToResponse* and *MessageResponse*. Communication using *XML* messages, generated on the basis of the proposed schemas, is performed via a hardware proxy, in this case, carried out with minicomputer *Raspberry Pi* model *B*. The communication protocol for data exchange with biochemical analyzers were implemented as set of separated libraries (*DLL*) using *C#.NET* language and *Mono framework*. Adding a *USB-to-Serial* converters and *USB hub*, the *Raspberry Pi* allows connection of a large number of analyzers via serial or *Ethernet* ports and their communication with MIS on the distance over a local network of health care institution.**

Keywords – **Integration, Module, Hardware, Software, Laboratory, Analyzer, Analysis, MIS, *XSD*, *XML*, *Raspberry Pi*, *Mono*, *C# .NET*, *LabIS*.**

I. INTRODUCTION

The great daily flow of patients with a large number of analyses performed on admitted patients in biochemical laboratories requires an enormous expenditure of human resources in the preparation of requests for specific analyses as well as for copying the results of the performed analyses from the analyzer into electronic patient record (EPR) and into a laboratory reports that is given to the each patient. In order to speed up the job by automatically acquisition analyses results from the analyzers, first of all it is necessary to connect biochemical analyzers with the medical information system (MIS). This way it is possible to prevent eventual errors that occurs when analyses results are reading from analyzers and then transfer (retype) to EPR [1, 2]. For example, in the Health Center Nis, daily, hundreds of patients perform several analyses in biochemical laboratory. A large flow of patients and a large number of the performed analyses have had influence to develop a hardware/software module, presented in this paper, in order to enable the automatic acquisition data from laboratory analyzers on the distance. This paper presents the application of the proposed hardware-software module just for connecting biochemical analyzers with MIS, but the concept can be used to connect other medical devices in a health care institution with MIS and beyond.

In the market there are many different types of laboratory analyzers (biochemical, hematological, etc.), from different manufacturers, which implement various communication protocols for data exchange with external host. Communication protocols of the same device model and the same manufacturer often vary among revisions. In order to establish communication between medical information system (MIS) and numerous different analyzers by different manufacturers (as it is often the case) it is necessary to implement the communication protocol for each analyzer. The most analyzers exchange data via serial port, and communication is physically limited to a few meters in the laboratory rooms. On the market there are several wide usage hardware devices (port extenders) which can enable communication between analyzers and MIS on the distance with high price (e.g. few hundreds of dollars). We are tested *LINDY Serial-To-Ethernet* port extender [3]. This paper presents a proposal of the integration of medical information system and the analyzers on the distance by using *XML* messages that are generated on the proposed *XSD* schemas (*MessageRequest*, *MessageRequestToResponse* and *MessageResponse*) and hardware proxy. The *XML* messages are passed through the local area network (LAN) to the hardware proxy, then the hardware proxy parses received messages and translates that to messages "understandable" for the analyzer, collects results from analyzers based on the MIS request, and then creates *XML* messages based on the proposed *XSD* schemas and then sends that to the MIS.

[1]Aleksandar Milenković is with the Faculty of Electronic Engineering at University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: aleksandar.milenkovic@elfak.ni.ac.rs.

[2]Dragan Janković is with the Faculty of Electronic Engineering at University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: dragan.jankovic@elfak.ni.ac.rs.

[3]Marija Stojković is with the Faculty of Electronic Engineering at University f Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: marija.stojkovic@elfak.ni.ac.rs.

[4]Aleksandar Veljanovski is with the Faculty of Electronic Engineering at University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: aleksandar.veljanovski@elfak.ni.ac.rs.

[5]Petar Rajković is with the Faculty of Electronic Engineering at University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: petar.rajkovic@elfak.ni.ac.rs.

Widespread and very popular minicomputer *Raspberry Pi* model *B* [4] was used as hardware proxy. This paper shows an example of the integration of MIS MEDIS.NET (licensed by the Ministry of Health of the Republic of Serbia for using in primary care health institutions) with biochemical analyzers. In order to increase speed of developing and testing and to become more independent from the real laboratory environment *Lab Simulator* was developed to simulate the biochemical analyzer *Abbot Aeroset*. It is enough to connect analyzers with serial ports on implemented hardware proxy, and then connect the hardware proxy to a local area network to be able to communicate with MIS.

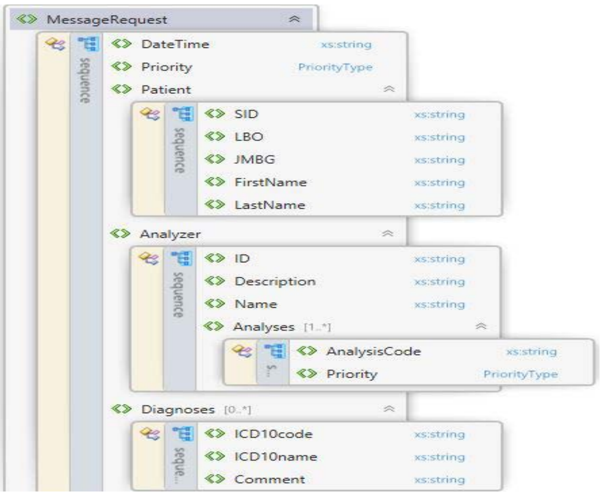

Fig. 1. *XSD* schema *MessageRequest.xsd.*

This paper presents the inexpensive solution which is based on inexpensive hardware proxy and *XML* messages. The proposed solution enable to place only hardware proxy instead of full computer nearby analyzers that can exchange data with analyzers and MIS trough a local area network of the health care institutions. Communication protocol for each connected analyzer to hardware proxy is implemented as separated library (*dll*) and placed on hardware proxy. The similar low priced solution with open communication formats for data exchanging between MIS and analyzers proxy not exists on the market or this solutions are closed and much expensive for biochemical laboratories in the most health centers in the Republic of Serbia. One of them is Siemens solution [5].

After the introduction the proposal message formats for exchanging data between MIS and the analyzers are presented. Below this the integration of biochemical analyzers with a medical information system using HW/SW module is shown after which follows the conclusion.

II. Proposal of Message Formats for Data Exchange Between Analyzers and MIS

The proposed communication between laboratory analyzers and MIS, in this case MEDIS.NET, is performed by sending *XML* messages that are created by using proposed *XSD* schemas [6]. There are three *XSD* schemas on which *XML* messages for data exchange are created: *MessageRequest.xsd* (Fig 1), *MessageRequestToResponse.xsd* (Fig 2) and *MessageResponse.xsd* (Fig 3). The *MessageRequest.xsd* is used to generate *XML* messages that are sent to the MIS while the *MessageResponse.xsd* is used to receive responses (*XML* messages) from MIS. *MessageRequestToResponse.xsd* is used to create requests (*XML* messages) on the basis of which it expects a response (*XML* messages) from analyzer that is created on the basis of the scheme *MessageResponse.xsd*. In order to communication be identical for all analyzers that are need to be connected to the MIS via a hardware proxy, it is necessary to fully comply with the proposed *XSD* schemas on which *XML* messages are generated. On this way data exchange between MIS and *Raspberry Pi* is separated and independent of implementation of communication protocol for each connected analyzer.

A. Description of message format for sending the request to analyzer (MessageRequest.xsd)

Element *Datetime* represents the time when the message was generated. The value of element *Datetime* is needed to save into format (*YYYY-MM-DD HH:MM:SS*). Element *Priority* describes the priority of the message that is sent. Possible values are (*Normal* - the default, *High* and *Low*). Given priority may not always be accepted. Element *Patient* refers to a patient whom analyses are done and he is closer determined with elements *SID* - Sample ID, *LBO* - identification number of the insured, *JMBG* - personal identification number, *FirstName*, and *LastName*. This is followed by an element *Analyzer* that is described with elements (*ID*, *Description*, *Name*, *and Analyses*). Each connected analyzer should have unique *ID* in order to be unambiguously determined in the whole system. Element *Analyses* is the list of analysis that needs to be performed on the certain analyzer for the certain patient. A list of the analyses contains at least one element, a single analysis. Each analysis, that needs to be performed on the analyzer, is uniquely defined by its code (element *AnalysisCode*) and described with priority (*Priority* element - *Normal* is the default value, other possible values are *High* and *Low*). The request that was sent to the analyzer can contain a list of diagnoses. Each diagnoses into diagnoses list is determined by the elements (*ICD10Code*, *ICD10Name* and *Comment*) and on the basis of internationally recognized ICD10 diagnoses code list.

B. Description of message format for initiate response sending (MessageRequestToResponse.xsd)

Elements *DateTime*, *Priority* and *Patient* are already described during describing *XSD* schema *MessageRequest.xsd*. Element *Analyzer* determines from which analyzers is necessary to take over the results of the analyses for the patient which is described with element *Patient* in *XSD* schema. If the list of analyzers is empty, wherein each analyzer are described with elements (*ID*, *Description*, *Name* and *Analyses*), the returning message will include analyses results, that they were performed on all analyzers which are connected to hardware proxy, and for a certain *Patient*. If analyzers are specified in list of analyzers (at least one) then return message that contains the results of analyses from specified analyzers in list. The list of analyses can be empty. If forwarding list of analyses is empty, returning message will contain all analyses results that are performed on specific analyzer for certain patient. If it is specified at least one analysis, than as result, will receive only result of this analysis. It is possible to define a greater number of analyses to get the results from the analyzer in the returning message. The response on the sent request for getting results of analyses is *XML* message that is generated and based on *XSD* schema *MessageResponse.xsd*.

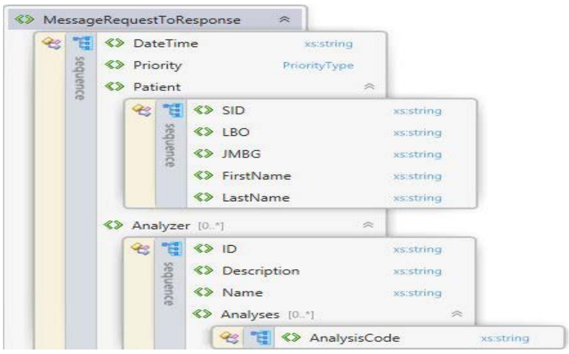

Fig. 2. *XSD* schema *MessageRequestToResponse.xsd.*

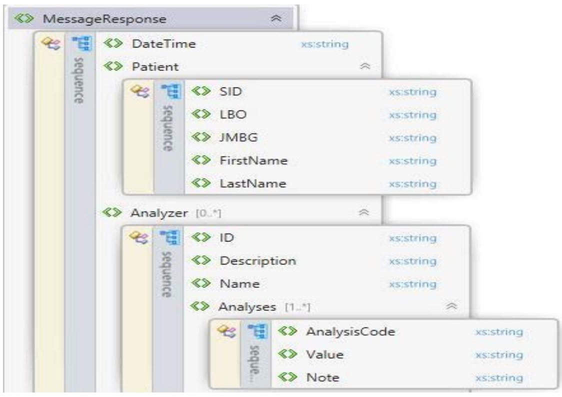

Fig. 3. *XSD* schema *MessageResponse.xsd.*

C. Description of message format for receiving data from analyzer (MessageResponse.xsd)

Elements *DateTime* and *Patient* are already described. Element *Analyzer* is a list of the analyzers from which results are taken by message that is initiated by sending a response (message generated and based on *XSD* schema *MessageRequestToResponse.xsd*). If the list of analyzers is empty that means that there are no analyses that were performed to the *Patient* on the analyzer described by elements (*ID*, *Description*, *Name* and *Analyses*). If there is at least one analyzer in list, in the list of analyses must be at least one analysis with the result. Result of analysis in the list of analyses is described with elements (*AnalysisCode*, *Value* and *Note*). *AnalysisCode* uniquely determines performed analysis on analyzer. Value contains the result of analysis and the type of this element is *string*. Element *Note* contains a description of the format in which result will be presented (the possible values are: *integer*, *decimal* and *string*). If result is a numeric value, the decimal point is presented as a dot (.).

Fig. 4. Sequential diagram of communication between MIS and analyzer.

The Fig 4. shows sequential diagram of communication of one analyzer and MIS that exchange data using minicomputer *Raspberry Pi* and *XML* messages based on proposal *XSD* schemas.

III. Integration of Biochemical Analyzers and MIS Using by HW/SW Module

Below is shown an example of integration biochemical analyzers based on the messages which are generated using the proposed *XSD* schemas. As hardware proxy, between MIS that sends requests and receives a messages with the results of the performed analyses and themselves, was used analyzers minicomputer *Raspberry Pi* model *B*.

The *Raspberry Pi* is minicomputer the size of a credit card that is appeared in the market with the model *B* in February 2012. It was originally developed for educational purposes as well as for connecting to a *TVs* receiver in order to create smart *TVs*. Version of the device labeled with B was extended with another *USB* port, *Ethernet* jack and it has 512 MB of RAM. The price of such devices is US$35. *SD* card is used as data storage. The recommended capacity of *SD* card is 2GB with less time for reading and writing data. On this hardware it is possible to runs different distributions of *Linux*: *Raspbian* "*wheezy*" (a derivative of *Debian* which is optimized for *Raspberry Pi*), *Arch Linux*, *RISC OS* and *Android 4.0*. We chose the first option (*Raspbian* "*wheezy*" *Linux kernel 3.6.11*).

In the market there are more *Serial-to-Ethernet* port extenders that allow you to connect multiple laboratory analyzers via serial ports and then to forward the traffic over the network to the remote computer in order to perform a necessary processing. We have been tried and tested the device *LINDY* [3]. The big problem with this device was its high price. The price of mentioned models on the official website of the manufacturer's is US$325. Variant with 4 serial ports costs US$425, which leads to much more expensive solutions that is for many health care institutions in the Republic of Serbia practically unacceptable.

Fig. 5. The physical appearance of *Raspberry Pi* model *B* [5].

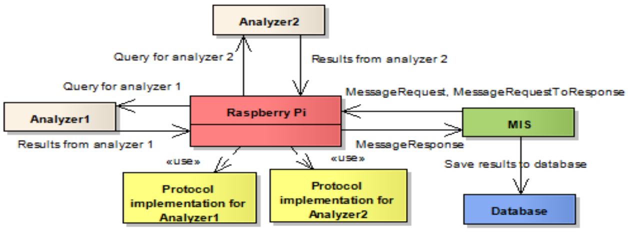

Fig. 6. The communication diagram between analyzers, MIS and *Raspberry Pi*.

In order to connect Raspberry Pi with biochemical analyzer via serial port, it is necessary to connect device with *USB-to-Serial* convertor. To connect multiple analyzers with device it is necessary to get a USB hub (Fig 5.). The driver installation is not required on the *Raspbian* so *USB-to-Serial* converter was ready for use right away. It is only necessary to plug mentioned converter to the device. Converter in the system is identified as */dev/ttyUSB**, where * represents the order number of converter, which is a zero based index. We used a *USB-to-Serial* converter manufacturers *Manhattan* (type *ld_pl2302_v0618*) [7]. The software that executes on the *Raspberry Pi* is written in *C# .NET* wherein the *Mono Framework* is used [8].

The request for ordering analyses from analyzer and for initiate sending results of analyses from analyzer, are sent from the MIS (it sends *XML* messages that are created from *XSD* schemas *MessageRequest* and *MessageRequestToResponse*) to hardware proxy (*Raspberry Pi*). Then, the messages forward to the hardware proxy and to the appropriate *DLL* that implements a specific communication protocol for analyzer for those messages are intended. Then the *XML* messages are translated into "understandable" messages for certain connected analyzer to hardware proxy and then those forward to analyzer. Then analyzer sends results of the analyses to *Raspberry Pi*. The results are parsed and formatted into a standard format (universal regardless of which analyzer is performed based on *XSD* schema *MessageResponse*) and then they are sent to the MIS. The results of the analyses are stored into the MIS database. It is not necessarily to initiates communication by MIS. All messages received from the laboratory analyzers are processed on the *Raspberry Pi* and then analyses results forwarded to the MIS via *XML* messages that correspond to *XSD* schema *MessageResponse.xsd*.

IV. Conclusion

In this paper it is proposed the integration of medical devices and MIS with a universal HW/SW module with a large emphasis on biochemical analyzers. The HW/SW module is based on messages which are described with *XSD* schemas and on a hardware proxy. Messages created on the basis of the proposed *XSD* schemes are *XML* records. As low-cost hardware proxy between the analyzers and the MIS was used minicomputer *Raspberry Pi* model *B*. The *Lab Simulator* was developed for purpose of the rapid development and testing of the proposed message formats and the whole proposed HW/SW module. This paper describes the integration of biochemical analyzers and MIS MEDIS.NET based on the proposed and described module.

The presented solution with a modification of *XSD* schemas on which *XML* messages are created for data exchange between biochemical analyzers and MIS via proxy *Raspberry Pi* can be used for integration of other medical devices with MIS. The described solution can be used outside the field of medicine, wherever is necessary to perform the acquisition of data from the devices and where you need to transfer data from a remote devices to information system over the network while the price of the whole solution is acceptable.

References

[1] D. Vučković, D. Janković, K. Milanović, P. Rajković, A. Milenković, "Data Acquisition in Laboratory Information Systems", First AMA-IEEE Medical Technology Conference on Individualized Healthcare, Washington DC, USA, 22 - 24 March 2010, p. 45-46, Conference Proceedings.

[2] D. Vučković, D. Janković, A. Milenković, T. Stanković, P. Rajković, "Data Acquisition Methods in Laboratory and Medical Information Systems", FACTA UNIVERSITATIS Series: Automatic Control and Robotics Vol. 10, No 1, 2011, pp. 97-103, UDC 681.3.004.14 007:61]:004, ISSN: 1820-6417.

[3] LINDY, http://www.lindy-usa.com/, last time of web site access: 15. April 2014.

[4] Raspberry Pi Foundation, http://www.raspberrypi.org/faqs, last time of web site access: 15. April 2014.

[5] Siemens - Laboratory Automation, http://www.healthcare.siemens.com/laboratory-automation, last time of web site access: 15. April 2014.

[6] P. Walmsley, “Definitive XML Schema”, 2nd Edition, Prentice Hall, 2013, ISBN-13: 978-0-132-88672-7.

[7] A. Milenković, I. Marković, D. Janković, P. Rajković, "Using of Raspberry Pi for Data Acquisition from Biochemical Analyzers", TELSIKS 2013, 16 – 19 October 2013, Faculty of Electronic Engineering Niš, Serbia, October 2013, pp. 389 – 392, ISBN: 978-1-4799-0900-1, IEEE Catalog Number: CFP13488-CDR.

[8] MonoDevelop, http://monodevelop.com/, last time of web site access: 15. April 2014.