

A Component Based Framework to Enable Medical Devices Communication

Imad Eddine Touahria*, Abdallah Khababa

Computer Science Department, Faculty of Science, University Ferhat Abbas Setif 1, Sétif 19000, Algeria

Corresponding Author Email: imad.touahria@univ-setif.dz



<https://doi.org/10.18280/isi.260306>

ABSTRACT

Received: 24 March 2021

Accepted: 27 May 2021

Keywords:

medical device, integrated clinical environment, software, component based system, safety, heterogeneity

The interconnection of medical devices is emerging as a new requirement in modern medicine. The final goal of connecting heterogeneous medical devices in a wider network of computational servers is to monitor and improve patient safety, where it also constitutes a major goal in the Integrated Clinical Environment (ICE) framework. The heterogeneity of medical devices provided by different suppliers is a key challenge in ICE-based systems, where interoperability and data communication across devices is still under study and specification. ICE aims to create a standard interface that covers medical devices heterogeneity, hence, achieving interoperability in a safe way. It focuses on defining an interoperable bus between the patient, medical devices, software applications, and the clinician. Given the lack of realization of ICE standard, this paper presents a component-based framework for making ICE usable for medical applications. This work illustrates the component model in detail and validates it with a prototype implementation that focuses on the integration of heterogeneous medical devices as the most relevant requirements faced by ICE.

1. INTRODUCTION

The integration of medical devices in a high acuity patient environment is taking an important trend in modern medicine where these devices may be implanted in the patient house, hospital or ambulance, and therefore their use become particularly challenging for many reasons. At first, medical devices are increasingly complex [1], and this is due to the complexity of the human body, and second, the need to integrate these devices in the network, also, because they are exclusively safety-critical systems [2] and directly connected to the patient body. The need to communicate medical devices is in the rise [3-5], and this is to simplify the clinical workflow and avoid the deterioration of the patient health because of the lack of coordination between the devices.

A number of projects as ICE [6], OR.NET [7], OpenPCA [8], US FDA (Food and Drug Administration) medical devices interoperability project [9] and NIST (National Institute of Standards and Technology) medical device communications testing project [10] have appeared to provide safe solutions to integrate medical devices in the same OR (Operating Room) or clinical situation. These projects took into consideration lower possibilities of hazards generated by the lack of data formatting, transfer and treatment mechanisms between medical devices but no solution was widely adopted by researches and tried to approach PnP (Plug and Play) concept into ICE.

The Integrated Clinical Environment (ICE) [6] aims to interconnect medical devices in a safe way by realizing interoperability between them in a high acuity patient situation, it's a solution to solve the problems generated by the heterogeneity of medical devices where every supplier has its own policy for devices interconnection and interfacing. ICE is simply a generic architecture to design medical devices and

services that can be easily interconnected [11], consequently, ICE does not require the integration of a specific software technology design to define the behavior of the components of the architecture and how data is transferred between those components.

The initial idea of ICE is to allow medical devices that are compatible with ICE standards to communicate with other devices that are also ICE-compliant and coordinate their activities, therefore, medical devices suppliers have to integrate ICE protocols and standards. Modern medical devices are more integrated in the network and connected to electronic health records, common data displays and cooperate with other devices to realize closed loop scenarios [12], which makes the integration of ICE standards during the production process more possible to realize for manufacturers and interoperability will be easier to achieve where developers will concentrate more on application level rather than hardware incompatibilities.

On the one hand, medical devices manufacturers have some particular solutions for realizing interoperability across heterogeneous devices, but the problem with those solutions is that they bound to the devices of the same manufacturer which is the case of MEDIBUS protocol of Drager [13] and Instrument Telemetry System (ITS) of Philips [14] and cannot support the heterogeneity of different devices. Heterogeneity in ICE-based systems is described by differences in the running software, data output and encoding formats, the hardware characteristics of each device and network connection modes and interfaces, so, heterogeneity is related to many aspects of a medical device and this defines the importance of studying it in ICE development process, which is a way to achieve the final goal of ICE-based systems and that is safety.

The contributions presented in this paper are the following:

1. A component model for ICE-based systems. Considering that ICE architectural concept is still under specification, we present a model that identifies the different functional entities that are needed in an ICE systems.
2. Handling the heterogeneity of medical devices. As a primary objective of ICE, heterogeneity is realized by designing an additional component, the mediator that supports plug-and-play of medical devices.

The rest of the paper is structured as follows. Section 2 describes ICE architecture and after that section 3 shows the proposed component model and the modules that are used to cover heterogeneity. Section 4 details the solution to communicate heterogeneous medical devices and the tools to program and section 5 illustrates the implemented prototype and some test results. Section 6 presents relevant work on component technology mostly relevant for medical systems. Eventually, and finally, section 7 draws some conclusions and presents the future work lines.

2. THE INTEGRATED CLINICAL ENVIRONMENT

A survey realized in 2015 by the West Health Organization on the need to implement interoperability to avoid medical errors illustrates that 93% of nurses strongly agreed that medical devices should be able to seamlessly share data with one another automatically [15]. In fact, 50% of nurses stated that they witnessed a medical error caused by the lack of coordination between medical devices. These numbers show the importance of medical devices coordination and communication which can improve patient safety and avoid hazards.

ICE was proposed by the Center for Integration of Medicine and Innovative Technology (CIMIT) [6]. After a draft prepared by Goldman [16] in 2008, ICE functional model was presented by the American Society for Testing and Materials (ASTM) in what is now known as a reference paper for integrating medical devices in a high acuity clinical situation ASTM-F2761 [17]. Goldman [16] defines ICE as "a medical system that aims to support the care of a single high acuity patient by realizing interoperability between medical devices and other equipment in order to improve patient safety". The reference architecture model of an ICE-based system is shown in Figure 1.

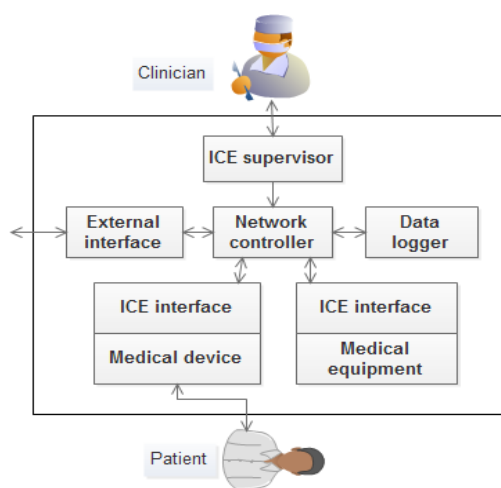


Figure 1. Functional model of the integrated Clinical environment

Medical device. Measure vital signs and monitor the state of the patient and in some cases can be used to inject drugs in patient body, it's provided with data input/output ports to analyze data for a later use. There is a variety of medical devices that can be used in different situations like: monitors, mobile sensors, bedside devices. etc. An infusion pump is an example of a medical device.

Medical equipment. These are more complex and sophisticated instruments than medical devices; this equipment may be needed for some clinical situations, e.g., a tomography scanner that integrates sensors and presentation system.

Network controller. It is a unit defined in ICE standard to connect each medical device and equipment to the network. The network controller establishes a common time base for medical devices data.

ICE interface. A data communication port that transports data between medical devices, medical equipment and the network controller.

External interface. This interface is used as the public access port to medical data flowing in ICE, the access to medical data should respect ICE security and privacy policies.

Data logger. This entity is used to detect, predict, and record errors that are flowing in the system. The data saved by the data logger present the technical issues that describe abnormal activities of medical devices, equipment, and applications.

ICE supervisor. It is a core component in the architecture that provides medical support for clinicians, allowing them to remotely access and control the medical devices. Also, it processes and presents the data provided by medical devices by means of a set of reports and data analysis diagrams.

3. COMPONENT MODEL FOR MEDICAL SYSTEMS DEVELOPMENT

3.1 General framework

The design of medical applications is enabled through the definition of a component model that is compliant with the Integrated Clinical Environment framework. The component model supports data processing operations, including the sampling and collection of data produced by medical devices to its transmission and presentation to the clinician. Components yield to modular designs and provide a clean separation between code and interfaces.

The component model is based on exporting the basic hardware/software low-level interface of medical devices through the appropriate use of services and precisely web services. In this way, medical devices providing common networking interfaces and processing modules will take commands from external medical devices or clinicians and will provide data to either other computation servers or clinicians.

The component model supports the real time data acquisition, processing and transfer between the different components and is divided into three layers (see Figure 2) defined with proper interrelation. Each layer is related to the actors that interact with the applications that it facilitates.

- Data Control and Transmission Layer DCTL (data and devices layer).
- Diagnostic and Decision Layer DDL (business layer).
- Control Layer CL (architecture self-assessment layer).

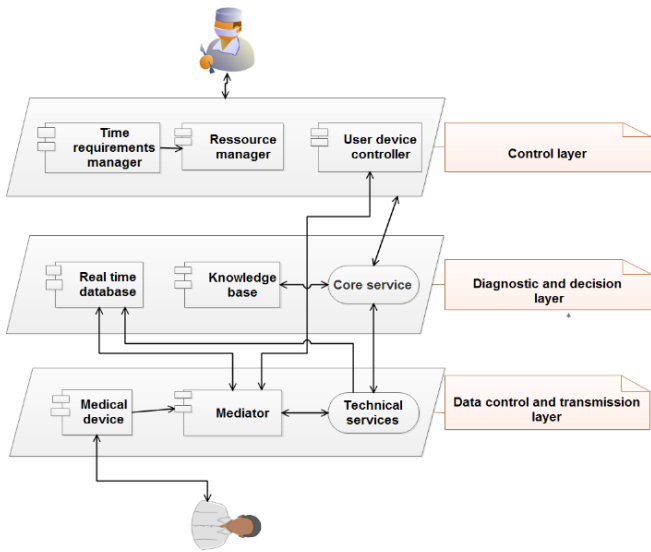


Figure 2. A general overview of the component-based model for the Integrated Clinical Environment

3.2 Detailed design

The component model follows the back box approach for more modularity where the code executed by each component is invisible for others. Each component communicates with proper ports which by their turn define provided and required interfaces, each interface provides or requires services from other components. Figure 3 shows a detailed design of the component model using UML representation. Each layer and component is described in what follows.

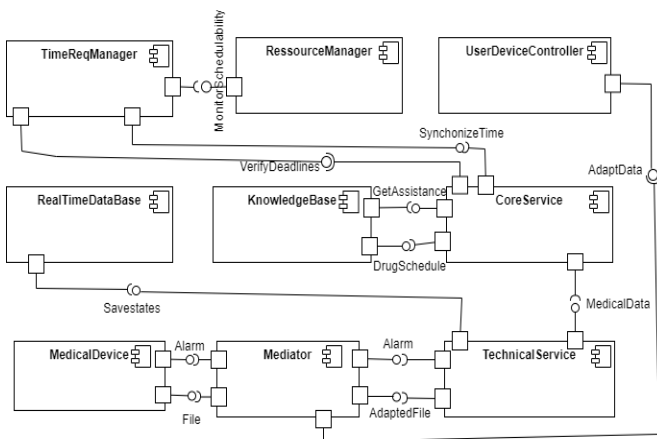


Figure 3. Detailed design of the component model

3.2.1 Data Control and Transmission Layer (DCTL)

DCTL is in direct contact with medical devices that are connected to the patient. It is the layer where medical devices and data are integrated into the medical system via the following components:

Medical devices. Send the vital signs data to the mediator in real time. In this context, the delay of data transmission varies around milliseconds units (depending on the specific medical device manufacturer) and it must have to be accounted for.

Mediator. The structure and the behavior of the mediator is explained in detail in section 4.

Technical services. Are defined for each medical device connected to the network. A technical service is associated to one medical device (as shown in Figure 4) such that it

encapsulates its functionality and data. The choice of this programming paradigm was made to provide a high level of interoperability. An alarm service is also deployed as a technical service where it detects abnormal vital signs or the disconnection of a medical device of the network or when the call button is pressed.

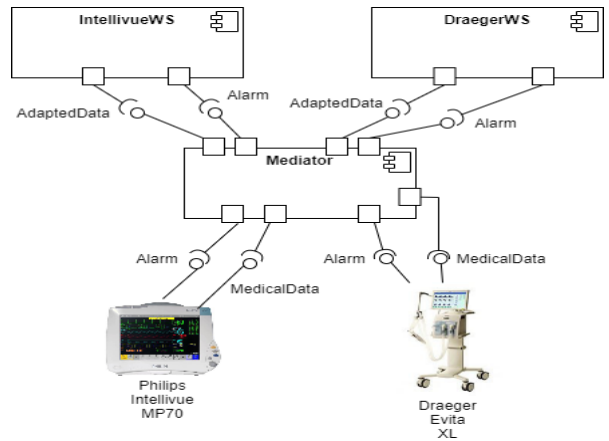


Figure 4. Attachment of a Web service to a medical device

3.2.2 Diagnostic and Decision Layer (DDL)

Data flowing in the system and decisions are made in this layer. The time intervals of the architecture are the longest in this layer because of the time that the core service take to provide self-decisions. DDL is composed of a set of components and that are presented as following.

Core service. Is a composed service that provides medical decisions such as disease analysis and drug schedule that constitute a support for clinicians. The core service has as input data provided by the technical services and produces as output data that is used by different components.

Real time database (RT-DB). Medical systems are characterized by the continuous amount of data flowing between its components, and due to the importance of time response the real time database emphasis deadlines on time to respond queries. Data access simultaneity is also provided where components can interrogate the database at the same time.

Knowledge base. Constitutes the brain of the architecture where every medical decision that is related to the diagnosis process is taken. It is a solution to ensure a certain level of the architecture autonomy where self-decisions are taken. The knowledge database communicates with core service in order to provide decisions.

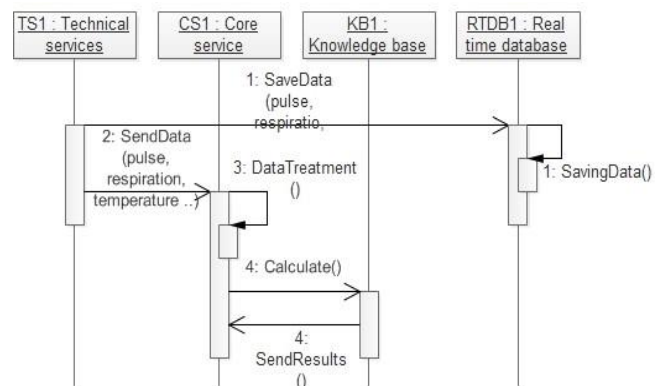


Figure 5. Sequence diagram of data flow in DDL

Figure 5 defines how data is transferred in DDL. Data Treatment () defines the set of operations that the core service executes such as: disease diagnosis, treatment proposition and disease criticality level. The core service and the knowledge base communicate with each other to analyze data and provide answers to questions such as visits calendar proposition.

3.2.3 Control Layer (CL)

The control layer checks in a continuous way the behavior of the components that are part of the architecture. It verifies the execution of the running components operation and the status of the communication to the user device. It also constitutes a mean to support autonomous behavior, CL is composed of:

Time requirements manager. This component verifies the compliance with the specified time constraints of the functions to execute. It also synchronizes the clocks of the different actors like: servers, medical devices, and user devices (smartphone, tablet or PC).

User device controller. Checks the hardware and the software characteristics of the user device. It sends queries requesting information about: screen size, battery level, and processor power. Collected data is sent to the mediator where the adaptation process is started according to the device characteristics.

Resource manager. Monitors the fulfillment of the operation deadlines for those cases where real-time operation is required. Some applications and service functions have execution priorities and associated finalization deadlines. The resource manager component monitors the schedulability of the different functions of the system.

3.3 Specific program design of medical devices

The proposed component model supports medical devices with USB and Bluetooth as data input/output technologies. Each manufacturer provides two types of software versions for each medical device, the first one is installed on the device in order to run it and the second is generally provided to be installed on a PC or phone in order to connect the medical device to other equipment. The software installed on the device is proprietary and cannot be used for development purposes, the proposed component model uses the data generated by the software installed on a PC.

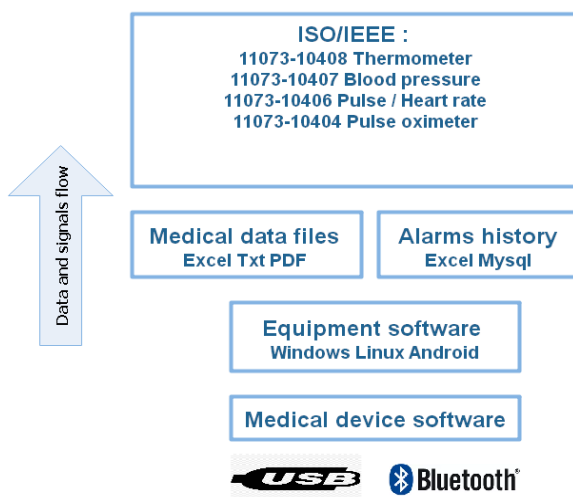


Figure 6. Specific program design of supported medical devices

Each medical device generates data in various formats, the devices supported by the architecture are those providing data in Excel, text or PDF formats, these data files may be small or big and generated in a continuous or separated ways. Alarms are also saved in Excel or MySQL database and that is the real time database.

A set of standards are supported by the component model and that are defined by ISO/IEEE. Each standard of 11073- {10408, 10407, 10406, 10404} family is associated to a specific type of medical devices like oximeters and the way to connect each type of devices to personal computers or smartphones in order to realize PnP (Plus and Play) interoperability.

Figure 6 defines the specific program design of medical devices that are supported by the proposed component model.

4. MEDICAL DEVICES CONNECTOR AND DATA ADAPTATION INTERFACE

4.1 Overview

The mediator is introduced in order to connect heterogeneous medical devices into the same network considering their differences and it also serves as an adapter of various types of data coming from medical devices where this data can vary in format, size, encoding, and accuracy. The mediator is a hardware and a software component that has as input data coming from medical devices and as output data provided to the technical services. The mediator is composed of three components each one of them performs operations that are explained in what follows:

1. The XML files processing incurred by the parsing and additional message headers leads to high delays [18], so it has to be compressed in order to speed up all the operations that it will undergo, which is the work of the data compressor component (shown in Figure 7) that reduces the files size without changing their type or quality.

2. Data has to be converted from its primary format to another for printing, visualization and processing, and here comes the data converter component (shown in Figure 7), it converts data types based on a set of parameters such as network load, user device type or battery charge. For example, a text file will be transmitted faster than an Excel file.

3. Due to the criticality of ICE it must provide accurate results and in time, therefore, the data must undergo an accuracy check performed by the component data correctness (shown in Figure 7, It is classified as the last component of the mediator so it works as a data checker and validator.

4.2 API

The mediator is a composed component that plays an essential role on the presented component model in this paper, it creates communication between medical devices and adapts data according to the clinician device hardware/software characteristics. Figure 7 shows the internal structure of the mediator.

The mediator communicates with other components via predefined ports and interfaces while components communicate via interfaces. The mediator has as inputs: files (contain technical and medical data) and alarms, it produces as outputs data (adapted according to the user device and Internet speed) and alarms.

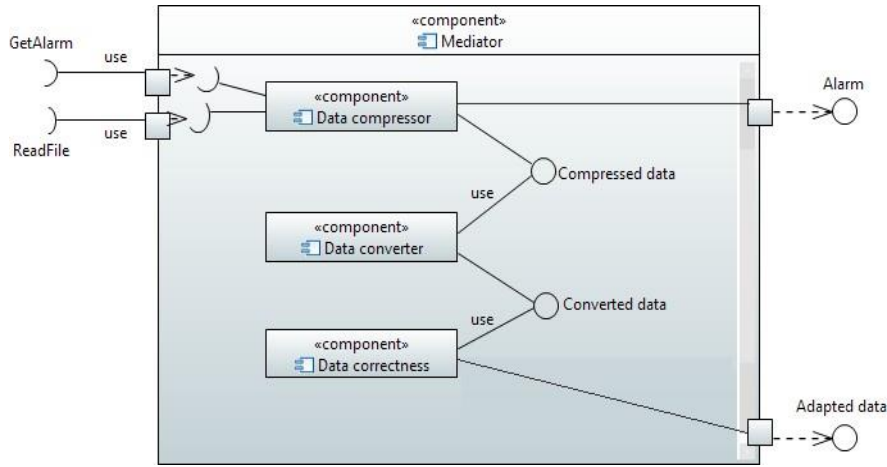


Figure 7. Component model of the mediator

Table 1. Mediator internal components inputs and outputs

Component	Input	Output
DataCompressor	Text, excel and binary files Boolean	Text, excel and binary files Boolean
DataConverter	Output of DataCompressor	Text, excel and binary files
DataCorrectness	Output of DataConverter	Pulse: integer, respiration: integer, temperature: float

Table 2. Mediator interfaces and data values

Interface	Input parameters	Output parameters
GetAlarm	IsAlarmRaised: boolean, GetAlarmTime: Date, GetAlarmText: String	AlarmRaised1: boolean, AlarmTime1: Date, AlarmText1: String
ReadFile	GetFileLocation: String, GetFileSize: float, GetFileFormat: String	FileLocation: String, FileSize: float, FileFormat: String
Alarm	AlarmRaised1: boolean, AlarmTime1: Date, AlarmText1: String	AlarmRaised2: boolean, AlarmTime2: Date, AlarmText2: String
CompressedData	GetNewFileLocation: String, GetNewFileFormat: String	FileLocation2: String, FileFormat2: String
ConvertedData	GetNewFileLocation2: String	FileLocation3: String
AdaptedData	GetVitalSignName: String, GetVitalSign Value: String or Integer or Float, GetVitalSignTime: Date	VitalSignName: String, Value: String or Integer or Float, MeasurTime: Date

provided by the DataCompressor, it converts the files to a unique file format that is supported by other components. DataCorrectness has as input a unique file format, it generates variables for each measured value.

Table 2 shows the interfaces implemented on the component model and data values for each provided and required interface. GetAlarm interface transfers data from medical devices to DataCompressor component while Alarm interface transfers data from DataCompressor component to user device and this is due to the criticality of the alarm. ReadFile interface is responsible of transferring data included in files provided by medical devices into DataCompressor component, CompressedData interface transfers data from DataCompressor component to DataConverter while ConvertedData interface transfers data from DataConverter component to DataCorrectness component. The AdaptedData interface is the link between the mediator internal treatment and DDL.

4.3 Mediator operation process and program design

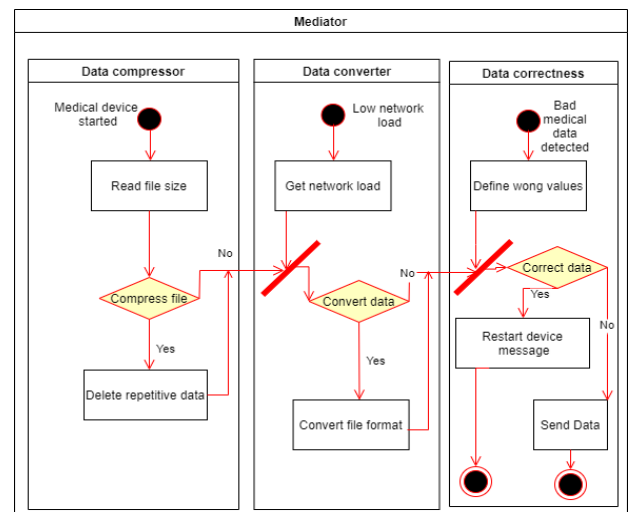


Figure 8. Mediator operation process

Table 1 shows input and output parameters of each component of the mediator. DataCompressor has input files with different formats and sizes generated by the medical devices, it compresses the size of the files to smaller files and treats excel and binary files, DataConverter has as inputs files

The mediator is a first class citizen of the component model and therefore its behavior has to be studied. The operation process of the mediator is shown in Figure 8, it starts when the medical device is connected to the network, if the file size is big then it's converted to a smaller one by deleting repetitive lines of the file then sent to the data converter component. This

last, receives internet load and convert file type if needed to text for example for faster transmission.

The data correctness component receives a message from the core service if some bad medical data is detected, for example, if some vital signs data are wide out of predefined range. In this case, a restart device message is sent to the clinician or just to verify if electrodes are installed correctly.

5. VALIDATION

The development of the component model is based on Python 2.7.18, the runtime environment is an Intel core i5-7300 at 2.20 Ghz with 8 GB of RAM.

Table 3. Used elements for the application

Medical devices	Data output types	Mediator	Web services
2	2	1	2

Table 3 shows that two medical devices are used and which are the Microlife blood pressure monitor BP A200 AFIB and Onyx Nonin 3231 USB model, the two data output files are an Excel and a text files and there is one mediator (mediator1) to connect medical devices and extract data from them, also, two web services are used (MicroLifeWS and NoninWS) in this case study for each medical device is affected a web service that defines it behavior and the generated data.

Table 4. Medical devices connectivity and data exchange parameters

Execution time	Exchanged messages	Alarms	Generated files
1 minute	19	1	2
10 minutes	155	1	2
60 minutes	917	2	2

Table 4 shows some results about data exchange between medical devices and the mediator for a given laps of time, the number of generated files is fix during all the execution process and this because each device generates a file when the execution is started this file is used as a data buffer, in this case there are two files formats a text and an Excel files. After 60 minutes of execution 917 messages were exchanged this illustrates that the component models supports the heterogeneity of medical devices and that data is transferred without problems. Alarms are an important mean to guarantee patient safety in this case two alarms were generated during 60 minutes of the execution cycle, these alarms were caused by a degradation in the patient vital signs: heart rate registered between 59 and 50 beats per minute for 1 minute and respiration rate between 9 and 11 breaths per minute.

This example shows the efficiency of the component model to guarantee heterogeneity of medical devices by coordinating medical devices actions through the mediator and patient safety monitoring by generating alarms when one or more vital signs are out of the regular range.

6. RELATED WORKS

Medical devices interoperability projects are under

specification and development by both academia and industry, OR.NET [7] is a project aiming to realize a safe and dynamic platform that coordinates medical devices and IT solutions in the Operating Room (OR). OpenPCA pump project [8] presents development and assurance artifacts that simulate the result of domain experts working with systems engineers to define function that will be safe for patients Patient Control Analgesia (PCA) Pump. US FDA interoperability project [9] defines interoperability as an objective to achieve for devices manufacturers and specifies rules for devices verification, validation and risk management activities. NIST (National Institute of Standards and Technology) [10] presents the medical device communications testing project that aims to facilitate the development and adoption of standards for medical device communications throughout the healthcare enterprise.

ICE is becoming a standard solution that supports MD interoperability in order to improve patient safety [19], ICE standards is supported by many organizations CIMIT [6], ASTM [17] and MDPNP [20]. Some works concentrate on the study of specific medical devices and the possibility to improve the interoperability among the use of these medical devices within it infrastructure, Arney et al. [21] with the X-Ray/Ventilator, and in [22] with the patient-controlled analgesia (PCA) pumps. OpenICE [23] is the open source implementation of ICE standards that aims to provide a set of tools for developers to create applications related to medical devices connectivity and interoperability. HITSP [24], HL7 [25] are two standards solutions aiming to create interoperability between different health providers and define standards exchange formats used by all stakeholders.

The use of software and component models facilitates the connection between the separated parts of a medical system and provides a good solution to overcome the heterogeneity of different components. Beyer et al. [26] defined a component-based framework that supports the process oriented approach to study requirements of a medical system and deployment of processes supporting applications within a healthcare network. Gaynor et al. [27] proposed a framework that supports interoperability between heterogeneous components of a system and implemented it in the medical domain. Jung et al. [28] proposed a component-based layered framework that is oriented to reuse the experiences and knowledge on patient safety, experiences are generated from robots hazards operations. Wu et al. [29] proposed a component-based software for blood flow simulation from the human body.

Medical applications have shown efficiency in improving patient safety and the creation of a link between the clinician and it patient, therefore, ICT is more integrated in healthcare. Archip et al. [30] integrated a set of devices and wearable mobile technologies that are connected to the patient and health facilities via IoT (Internet of Things) solutions. Corno et al. [31] also used IoT to communicate data in real-time to clinicians and notify them about patients health status. CPS (Cyber Physical System) is an emerging technology that is widely used in healthcare [1]. Sawand et al. [32] presented a layered CPS-based framework to ensure communication between clinician and patients and studies security issues of the proposed architecture over the CPS system. King et al. [33] presented a modeling language that describes medical devices and clinical scenarios in CPS-based systems, the language is formal, modular and allow the description of medical devices behavior.

7. CONCLUSION

The integrated clinical environment aims to create communication and coordination of medical devices installed on the same high acuity patient environment. ICE-based systems define safety as the ability of the system to avoid the patient health status deterioration because medical devices couldn't operate in the same network, it also covers the heterogeneity of medical devices where each device has a specific hardware/software configuration. The paper presented a component model for ICE-based systems where it emphasis on safety and heterogeneity as two essential concepts for the design of medical systems. The paper studied the component model in details and ICE-based systems in particular. The mediator was integrated as a solution for heterogeneity, notification, clinician support and data accuracy as solutions for safety.

REFERENCES

- [1] Dingler, M.E., Ortiz, D., Dietz, C., Lueth, T.C. (2016). An approach towards compositional behavior specification of medical device networks. In 2016 IEEE/SICE International Symposium on System Integration (SII), Sapporo, Japan, 483-489. <http://doi.org/10.1109/SII.2016.7844045>
- [2] Lindvall, M., Diep, M., Klein, M., Jones, P., Zhang, Y., Vasserman, E. (2017). Safety-focused security requirements elicitation for medical device software. In 2017 IEEE 25th International Requirements Engineering Conference (RE), Lisbon, Portugal, 134-143. <http://doi.org/10.1109/RE.2017.21>
- [3] Barrón-González, H.G., Martínez-Espronedada, M., Led, S., Serrano, L., Fischer, C., Clarke, M. (2013). New use cases for remote control and configuration of interoperable medical devices. In 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 4787-4790. <http://doi.org/10.1109/EMBC.2013.6610618>
- [4] Larson, B.R., Zhang, Y., Barrett, S.C., Hatcliff, J., Jones, P.L. (2015). Enabling safe interoperation by medical device virtual integration. *IEEE Design & Test*, 32(5): 74-88. <http://doi.org/10.1109/MDAT.2015.2464813>
- [5] Robkin, M., Weininger, S., Preciado, B., Goldman, J. (2015). Levels of conceptual interoperability model for healthcare framework for safe medical device interoperability. In 2015 IEEE Symposium on Product Compliance Engineering (ISPCE), Chicago, IL, USA, 1-8. <http://doi.org/10.1109/ISPCE.2015.7138703>
- [6] Parrish, J.A. (2002). Center for Integration of Medicine and Innovative Technology (CIMIT). Massachusetts General Hospital Boston.
- [7] OR.NET - Operating Room project, <http://ornet.org/>.
- [8] OpenPCA pump project - Open Patient Controlled Analgesy pump project. <http://openpcapump.santoslabs.org/>.
- [9] FDA - US Food and Drug Administration, Medical Devices Interoperability Project. <https://www.fda.gov/MedicalDevices/ScienceandResearch/ResearchPrograms/ucm477390.htm>.
- [10] NIST - National Institute of Standards and Technology, Medical Device Communications Testing Project. <https://www.nist.gov/itl/ssd/systems-interoperability-group/medical-device-communications-testing-project>, accessed on Jan. 13, 2019.
- [11] García-Valls, M., Touahria, I.E. (2017). On line service composition in the integrated clinical environment for ehealth and medical systems. *Sensors*, 17(6): 1333. <https://doi.org/10.3390/s17061333>
- [12] King, A., Procter, S., Andresen, D., Hatcliff, J., Warren, S., Spees, W., Weininger, S. (2009). Demonstration of a medical device integration and coordination framework. In 2009 31st International Conference on Software Engineering-Companion Volume, Vancouver, BC, Canada, pp. 433-434. <http://doi.org/10.1109/ICSE-COMPANION.2009.5071048>
- [13] Drager-medical, Protocol definition, Drager RS 232 MEDIBUS. https://hit.healthsystem.virginia.edu/index.cfm/_api/render/file/?fileID=DC02EE3A-17A4-77A0-3E1CF4BE2E1B9EA8&fileEXT=.pdf.
- [14] Philips, Intellivue Patient Monitor user guide MP20/30, MP40/50, MP60/70/80/90. http://its.uvm.edu/FAHC_Alarms/McClure5/Philips%20Monitor/User%20Manual%20Philips%20MP50.pdf.
- [15] West health organization. (2015). Missed connections: A nurses survey on interoperability and improved patient care. <https://www.westhealth.org/wp-content/uploads/2015/03/Nurses-Survey-Issue-Brief.pdf>.
- [16] Goldman, J.M. (2008). Medical devices and medical systems-essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ice)-part 1: General requirements and conceptual model. ASTM International.
- [17] ASTM - American Society for Testing and Materials. <https://www.astm.org/>.
- [18] Rubio-Conde, P., Villarán-Molina, D., García-Valls, M. (2017). Measuring performance of middleware technologies for medical systems: Ice vs AMQP. *ACM SIGBED Review*, 14(2): 8-14. <http://doi.org/10.1145/3076125.3076126>
- [19] Soroush, H., Arney, D., Goldman, J. (2016). Toward a safe and secure medical Internet of Things. *IIC Journal of Innovation*, 2(1): 4-18.
- [20] MDPNP-Medical Device Plug-and-Play interoperability Program. <http://www.mdnp.org>.
- [21] Arney, D., Bhatia, K., Bhatia, S., Sutton, M., Rausch, T., Karlinsky, J., Goldman, J.M. (2011). Design of an x-ray/ventilator synchronization system in an integrated clinical environment. In 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, MA, USA, pp. 8203-8206. <http://doi.org/10.1109/IEMBS.2011.6092023>
- [22] Arney, D., Goldman, J.M., Bhargav-Spantzel, A., Basu, A., Taborn, M., Pappas, G., Robkin, M. (2012). Simulation of medical device network performance and requirements for an integrated clinical environment. *Biomedical Instrumentation & Technology*, 46(4): 308-315. <http://doi.org/10.2345/0899-8205-46.4.308>
- [23] Plourde, J., Arney, D., Goldman, J.M. (2014). Openice: An open, interoperable platform for medical cyber-physical systems. In 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPs), Berlin, Germany, pp. 221-221. <http://doi.org/10.1109/ICCPs.2014.6843734>

- [24] HITSP-Healthcare Information Technology Standards Panel. <http://www.hitsp.org/>.
- [25] HL7-Health Level Seven. <http://www.hl7.org/>.
- [26] Beyer, M., Kuhn, K.A., Meiler, C., Jablonski, S., Lenz, R. (2004). Towards a flexible, process-oriented IT architecture for an integrated healthcare network. In Proceedings of the 2004 ACM Symposium on Applied Computing, 264-271. <http://doi.org/10.1145/967900.967958>
- [27] Gaynor, M., Yu, F., Andrus, C.H., Bradner, S., Rawn, J. (2014). A general framework for interoperability with applications to healthcare. Health Policy and Technology, 3(1): 3-12. <http://doi.org/10.1016/j.hlpt.2013.09.004>
- [28] Jung, M.Y., Kazanzides, P. (2016). An architectural approach to safety of component-based robotic systems. In 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 3360-3366. <http://doi.org/10.1109/ICRA.2016.7487511>
- [29] Wu, J., Chui, C.K., Teo, C.L. (2015). A software component approach for GPU accelerated physics-based blood flow simulation. In 2015 IEEE International Conference on Systems, Man, and Cybernetics, Hong Kong, China, pp. 2465-2470. <http://doi.org/10.1109/SMC.2015.431>
- [30] Archip, A., Botezatu, N., Șerban, E., Herghelegiu, P.C., Zală, A. (2016). An IoT based system for remote patient monitoring. In 2016 17th International Carpathian Control Conference (ICCC), High Tatras, Slovakia, pp. 1-6. <http://doi.org/10.1109/CarpathianCC.2016.7501056>
- [31] Corno, F., De Russis, L., Roffarello, A.M. (2016). A healthcare support system for assisted living facilities: An iot solution. In 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Atlanta, GA, USA, pp. 344-352. <http://doi.org/10.1109/COMPSAC.2016.29>
- [32] Sawand, A., Djahel, S., Zhang, Z., Nait-Abdesselam, F. (2015). Toward energy-efficient and trustworthy eHealth monitoring system. China Communications, 12(1): 46-65. <http://doi.org/10.1109/CC.2015.7084383>
- [33] King, A.L., Feng, L., Sokolsky, O., Lee, I. (2013). Assuring the safety of on-demand medical cyber-physical systems. In 2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA), Taipei, Taiwan, pp. 1-6. <http://doi.org/10.1109/CPSNA.2013.6614238>