Graphical interface software for the surgical table control system to monitor and adjust the parameters of surgical equipment

V A Filippenko¹*, A N Epikhin¹², D A Korochentsev¹, O A Safaryan¹

¹Don State Technical University, Rostov-on-Don, Russia
²Rostov State Medical University, Rostov-on-Don, Russia

*e-mail: philippenko@inbox.ru

Abstract. The work is devoted to the development of software with a graphical interface for the surgical table control system, which allows you to monitor and adjust the parameters of surgical equipment. The object of the work is the processes between the computer and the microcontroller that control the surgical equipment. The subject is the software tool, which is used to ensure the operation of the surgical equipment control system. In the work there is an analysis of requirements and methods necessary for creation of safe medical software. After that, the algorithmic design takes place, which includes: selection of the design pattern, the structure of the codograms and the algorithm of the operation start.

1. Introduction

Human eye and retina surgery are highly specialized surgery, the equipment and instruments of which must be strictly according to their purpose. These operations use sophisticated high-tech equipment, and the success of the operation largely depends on the proper operation of the entire set of devices, instruments and complex equipment, which only adds difficulties in managing all of them. At the same time, each of the surgical instruments contains individual parameters with which a surgeon can set or edit the behavior of a particular surgical instrument.

The purpose of this work was to develop software to manage the parameters of surgical instruments. The software was developed for a computer that communicates with the microcontroller via a standard bidirectional serial port of the computer, which transmits data via RS-232 protocol.

The patent [1] describes a device with a GUI system that controls various parameters of surgical instruments. According to one of the interface variants presented in this work, the screen of display and parameter control includes a graphical interface, which, in addition to the presence of parameters of the surgical device, also has a representation of the different stages of the procedure. As you move from one stage to the next, the parameters of the surgical instrument are automatically changed to the parameters required for that stage, which are represented by phacoemulsification.

An example of a device capable of combining a vitreoretinal and cataract microsurgical console is presented in a patent [2]. It states that the surgical equipment used for ophthalmic surgery can be adapted to perform operations on the anterior or posterior part of the eye, or support both options. Such a surgical console can provide many functions depending on the type of surgery and type of surgical equipment.

The patent [3] provides a system for collecting and transmitting video image data with superimposed parameter data. The design includes an input device configured to receive data as separate data streams.
The information presented in the patents considered above will be useful when working on this project.

2. Requirements for medical software development
As the software functionality becomes more intricate, concerns arise regarding efficacy, safety and reliability [4]. To ensure the secure operation of medical devices containing the software, an appropriate understanding of the tasks that the software must perform is required. Also, programs such as these that meet the previous requirement should not cause unexpected or unacceptable risks when the application is running that could be harmful to the patient [5]. In the past, hardware was more important than software, but in the recent times software one of the elements for overall implementation has surpassed the prominence than hardware [6]. It is important to understand that any software, in itself, does not present a risk to patients. However, the complex architecture of a software tool may allow for a sequence of actions that may result in various, unforeseen errors that may cause dangerous situations during surgery.

In order to avoid such problems in software design, a certain standard for medical devices was adopted, following which during the development, it is possible to improve the security of the developed software. This standard is known as GOST ISO 14971-2011 [7]. There was also adopted a national standard GOST R 55544-2013/IEC/TR 80002-1:2009 [8], which is aimed at more detailed explanation of certain requirements necessary to ensure the planned and uninterrupted operation of medical software.

According to it, potential actions that may lead to unexpected errors can be divided into two categories:

1. A sequence of actions that can lead to unexpected software responses to input data that is not handled by the software.
2. A sequence that may lead to errors due to incorrect approach to software design.

The above categories are characteristic of software whose architecture was incorrectly implemented.

When performing risk analysis for a developed application, most attention should be paid to potential anomalies that lead to dangerous situations. The risks of unexpected situations due to software anomalies should also be classified by the severity of the damage.

Risk assessment for medical software design should start at the earliest design stages. The proactive application design principle implies that the test development process will be initialized as early as possible in order to detect and eliminate anomalies as early as possible. Based on this, it can be determined that a proactive approach will be an effective and quick way to obtain secure medical software.

Different methods can be used to identify different anomalies during the execution of a program, which use different approaches and can be useful at certain stages of development [9].

Examples of these methods can be: Fault Tree Analysis (FTA) and Fault Type and Sequence Analysis (FMEA).

The Fault Tree Analysis method is a graphical tool to study the causes of system level failures. It uses a logical and probabilistic model to combine a series of lower-level events. This approach is primarily a top-down approach to identify component level failures that may have caused a system level failure.

The Failure Type and Effect Analysis method is a structured approach to identify potential failures that may exist during product or process development. Using this approach is able to prevent certain consequences of a failure of any part of the software, if an error occurs in the operation of the higher hierarchy of the element.

Since it is rather difficult to determine the element in which, when executing a program, an error can occur, it is best to identify the categories of defects, for each of which there is a well-known risk management measure.
As a result, software risk and anomaly identification activities should focus primarily on the severity and relative probability of harm in the event of a failure, rather than trying to determine the probability of each potential failure.

Design and implementation of a medical software need to consider the software engineering principles and methodologies [10]. An agile methodology of software development was used while working on this project.

3. Requirements for medical software interface

The availability of a graphical user interface for medical software raises an additional problem, which is that some parts of this interface may affect unexpected behavior of operators, which may result from improper interaction with the product. Most often, such situations can be caused by a misunderstanding of the overly complex GUI by the user. Therefore, it is important to anticipate the wrong usage and to design the interface in such a way as to avoid such situations as much as possible.

To build more efficient interfaces, it is recommended to use the principles:

1. Accuracy depending on the context - interface elements can be represented both in digital and analogue formats. At the same time, when accuracy is more important than context - it is recommended to give preference to the digital format.
2. Availability - it is recommended to use a unique identifier for each specific case of different signals. These identifiers should differ not only in color but in form as the operator may have some vision defect.
3. Visual hierarchy is the essence of separation of more important for perception objects from secondary ones.
4. Comparison of static and dynamic identification - use of animation is a powerful factor to attract operator's attention. At the same time, it is not necessary to overdo it when designing such software, because additional resources of the system may be spent on their drawing, and the operator's attention may be detached from more important elements.
5. Using a visual perspective in relation to the accuracy of perception - 2D graphics will be visually perceived better than 3D. Also, for drawing two-dimensional elements will be allocated fewer resources.

4. General architecture of the application

In order to abstract the operator who controls the parameters of the surgical equipment and the program as a whole from the logic of the application, it is necessary to develop a single graphical user interface to manage the application. There are many design patterns for developing graphical user interfaces, each of which is suitable for specific types of tasks. The main ones from the bottom are templates such as: Model-View-Controller (MVC), Model-View-Presenter (MVP), Model-View-ViewModel (MVVM).

After analysing the above patterns, it was decided to use MVC. Model-View-Controller allows you to divide into modules the main parts of the software, which will be responsible for storage and access to data, for the operator's interaction with controls and for displaying data that have been changed. This division of data can simplify the software architecture and also allows you to change certain parts of the software, without affecting others. Thus, based on the selected design pattern, it turns out that the main parts of the developed product will be: a graphical user interface, a module of communication with the COM-port and the module-controller.

The graphical interface will contain the controls, such as:

- sliders with the ability to change the range of indicators to control the parameters of surgical equipment, as well as display these parameters;
- pop-up windows that will display the slider and the numerical value of the selected and actual parameter;
- buttons, by means of which the parameters containing only two values will be regulated;
- drop-down lists, by means of which it will be possible to select the type of operation, the doctor;
- additionally, created elements to control the pedal parameters.

It should be noted, that for some graphic interface elements, the touch control logic should have been additionally implemented. Also, custom-created elements can significantly reduce the amount of code and build a more advanced architecture of the interface and its interaction with the backend.

To communicate with the microcontroller via COM port, an additional module will be implemented, which will receive data from the port and transmit it above for disassembly. Also, a method for writing data to the port will be implemented in the module. In this case, the implementation of the module will be built so that when reading or writing data, these operations will not block the execution of each other.

Module-controller will record incoming data from the COM port, disassemble them, and after parsing update the information on the graphical interface, as well as receive modified data from the GUI and transfer them for writing to the COM port. In addition, it will work with the connection and communication module with the SQLite database, which will store the default parameters for doctors.

5. GUI development
The main view of the graphical user interface will contain:
- scales showing the parameters of each tool;
- a pedal control element with a visual representation of each of its parameters;
- the control element of the operation starts;
- the upper bar, which displays the operation data;
- the window of the video stream broadcasting;
- the lower bar for control of different parameters.

For the development of the graphical user interface, the declarative programming language QML will be used. The elements that make up the language's building blocks are inherited from the QtDeclarative (C++) module on which the QML execution environment is based. In this way, developers can implement logic in their applications without using C++. This set of technologies is supported by the Qt Creator tool. Thus, an overview of the interface is shown in Figure 1.

![Figure 1. General view of the graphical interface](image-url)
Scales that visually display the parameter of each surgical instrument, as well as its actual value in the form of a number that is located just below the scale. When you touch on each of the scales, you will see a pop-up window where you can edit the instrument parameter. In addition, this window contains the actual parameter and the new one. The new value will be applied to this tool after the pop-up closes. An example of such a window may be seen in Figure 2.

![Figure 2. Example of the pop-up window](image)

Each of the pedal parameters is represented as a pie chart for better visual perception. Clicking on this element will open a pop-up window, where you can adjust the pedal parameters. An example of this window may be seen in the figure 3.

![Figure 3. Pop-up window for adjustment of pedal parameters](image)

To start the operation, touch the "Start operation" button, which is located on the operation control element, Figure 4.
Figure 4. Operation control element

In addition, this element contains a stopwatch displaying the duration of the operation as well as the current time and date.

6. Computer communication with microcontroller

RS-232 (Recommended Standard 232) - physical layer standard for UART (Universal Asynchronous Receiver-Transmitter) asynchronous interface. The serial port is a convenient tool for communication between different peripherals, which can be assembled independently on the basis of a microcontroller.

According to RS232 standard, there are two types of devices involved in data exchange:

1. Data Terminal Equipment (DTE) (control device).
2. Data Circuit-Terminating Equipment (DCE) (controlled device).

When turned on, the microcontroller performs self-test and cyclically displays information from sensors (pressure, pedal) in the joint RS-232.

When receiving the command "Operation Start" from the computer, the microcontroller begins to respond to the pedal and control the pumps and milling machine. When receiving the "Stop" command from the computer, the microcontroller stops the pumps and milling cutter. Before issuing the "Startup" command, the computer must load the procedure parameters into the microcontroller.

To transfer data to the RS-232 microcontroller and read data from it was implemented class SerialPort. The main object, which operates this class is QSerialPort [11].

After a successful opening QSerialPort tries to define the current port configuration and initializes itself. Once the port is available for reading or writing it is possible to use the read () or write () methods. In this case, if not all data has been read at once, the rest of the data will be available for further use as new incoming data is added to the internal QSerialPort read buffer.

7. Implementation of the main part of the program

C++ general-purpose programming language was selected for application development. The main framework to be used for software development will be a cross-platform framework for software development in the C++ programming language - Qt Open Source [12].

The main class in the application is MainHandler. It takes a pointer to a parent item and also captures by reference an object of type QQmlApplicationEngine [13], which is used to load the graphical part of the program. In the class constructor, the following pointers to objects are initialized using the initialization list:
- serialPort - used to communicate the program tool with the MC;
- sixCodogramm - used to store parameters and build the sixth codogram for recording;
- sevenCodogram - used to store parameters and build the seventh codogram for recording;
- eightCodogram - used to store parameters and build the eighth codogram for recording;
- startButton - used to start and end the operation;
- stopwatch - used to fix the time of operation duration.

By default, the class expects the startButton to signal the start of the operation. Immediately when the signal was received - connects to listen to signals about the willingness to record a codogram in RS-232 from sixCodogramm, sevenCodogram, eightCodogram. Also, once the signal to start the operation was received, the class artificially emulates signals about the readiness to record a codogram, as they must go before the codogram "start the operation. After the basic codogram was written to the port - sent a codogram of the start of the operation and initialized counter of its duration. Figure 5 shows a more detailed architecture that contains the basic modules of the software tool.

![Architecture of the software tool](image)

Figure 5. Architecture of the software tool

After the start of the operation is pressed again, the class stops listening to signals from the codogram handler objects, sends a message about the end of the operation, resets the operation duration timer.

When receiving a codogram from the serialPort object from the microcontroller - onPortHasRead method begins to check the header and checksum of the message by adding the first 6 bytes of the codogram. If the checksum does not match the data recorded in the sixth byte, then the MC may have sent a wrong value. In this case, the codogram is not processed and is skipped. If the checksums coincide, the parsing process of the codogram starts. The parsing starts with the second byte, because the message type is specified in it. After that, the data is extracted, converted to the required types and passed to the objects for display - codogram handlers, and then to controls.

8. Storage of surgical equipment parameters
In order to implement the database of doctors for whom there is an opportunity to download any default parameters, add a new doctor, with the ability to configure the personal parameters and the list of operations for which the default parameters of surgical devices - it is necessary to implement communication software tool with the database. In this case, the data shall be stored in a local database, as it makes no sense to set up a separate local server to store such data.
As a database was chosen SQLite, which is an embedded, cross-platform open-source database that supports a fairly large set of SQL commands. In addition, it is embedded, which greatly simplifies the work with data, when deploying the software.

To work with it, the DataHandler class was created. The main object operated in this class is QSqlDatabase. This class provides an interface to access the database through a connection, which provides access to the database through one of the supported database drivers.

After creating a connection, DataHandler requests a list with the names of doctors and a list of default operations. After that, two models with data are formed, which are transferred to the ComboBox and then displayed on the graphical user interface.

After the operator selects one of the parameters, DataHandler asks for a list of surgical equipment parameters, creating a structure with parameters. This structure is then transferred to the setAllData method, which is declared in MainHandler. This method already passes the parameters to the microcontroller as a codogram.

When you scroll through the ComboBox entries to the end, you can see an empty field. By tapping it, you can enter a doctor's name or the name of an operation to save it to the database. The selected parameters for this record will be collected and saved automatically.

9. Conclusion
As part of this work, the requirements for the design, testing, and development of medical software have been studied. The RS-232 standard - the physical layer standard for asynchronous UART interface - was considered in detail. The analysis and selection of the basic development tools necessary for this project were made, the design pattern Model-View-Controller was chosen, which allows to divide the main parts of the software into modules. The general architecture of the software tool was developed. Also, the software with graphical user interface for the surgical table management system was designed, developed and tested, allowing to monitor and adjust the parameters of the surgical equipment, meeting the following requirements:

1. Graphic interface, with the possibility of touch control.
2. Displaying the video process of the operation. Possibility to record the operation.
3. Graphic panel displaying the current parameters of the surgical device (irrigation, aspiration, vacuum, etc.) with the possibility of their correction.
4. Provide a stable connection between the computer and the microcontroller to communicate via COM port.
5. To ensure the collection of parameters of surgical equipment to form a codogram, which will be transmitted to the microcontroller.
6. To provide parsing of the codograms coming from the microcontroller and display them on the user interface.
7. Database of doctors for whom it is possible to download any default parameters. Adding a new doctor, with the ability to configure personal parameters.
8. List of operations for which the default parameters of surgical devices are set.

Currently, the software tool is implemented at the Department of Ophthalmology of Rostov State Medical University.

Acknowledgments
A.N. Epikhin was supported by the Government of Russia grant no. 14.Z50.31.0046.

References
[1] Patent US 9119700B2. Graphical user interface system and method for representing and controlling surgical parameters. Boukhny M. Publ. 11/30/2004
[2] Patent RU 2682484C2. Surgical console graphical user interface. Toe D., Buhnj M., Satliff, T., Vuldrizh, K., Yang, M., King, K. Publ. 07/17/2014
[3] Patent US 8982195B2. Digital video capture system and method with customizable graphical overlay. Claus, M., Urich, A., Rockle P. Publ. 10/12/2007
[4] Jetley R, Sudarsan S, Sampath R and Ramaswamy S 2013, Medical Software – Issues and Best Practices, Distributed Computing and Internet Technology, 69-91
[5] Caffery F, Casey V, Sivakumar M, Coleman G, Donnelly P and Burton J 2012, Medical Device Software Traceability, Software and Systems Traceability, 321-339
[6] Alekya T P, Venkatesh P and Pramod Kumar T M 2018, Software as medical device, International Journal of Drug Regulatory Affairs, 4(2) 10-18.
[7] State Standard 14971-2011. Medical devices. Application of risk management to medical devices. Moscow, Standartinform Publ., 2011.
[8] State Standard R 55544-2013. Medical devices software. Part 1. Guidance on the application of ISO 14971 medical devices software. Moscow, Standartinform Publ., 2011.
[9] Wallace D and Kuhn D 2002, Failure modes in medical device software: An analysis of 15 years of recall data, International Journal of Reliability, Quality and Safety Engineering, vol.08, no. 04, 351-371
[10] El-Masri, S 2013, A New proposed Software Engineering Methodology for Healthcare Applications Developmen, International Journal of Modern Engineering Research, 1566-1570.
[11] Summerfield M 2010 Advanced Qt Programming: Creating Great Software with C++ and Qt 4 (Upper Saddle River: Pearson Publishers) p 806
[12] Monteiro F R, Garcia Mario A P, Cordeiro L C and Eddie B. de Lima Filho 2017, Bounded model checking of C++ programs based on the Qt cross-platform framework, Software: Testing, Verification and Reliability, vol. 27, no. 3, 1-17
[13] Schlee M Qt 5.10. Professional'noe programmirovanie na C++ [Qt 5.10. Professional C ++ Programming] / Sankt-Peterburg: «BHV-Peterburg» 2018 - 1073 pages.