TECHNOLOGY FOR IMPLEMENTATION OF OPERATING STATION WITH TCL-TK BASED STRUCTURES. APPLICATIONS

ТЕХНОЛОГИЯ ЗА ИМПЛЕМЕНТИРАНЕ НА ОПЕРАТОРСКА СТАНЦИЯ
С TCL-TK БАЗИРАНИ СТРУКТУРИ. ПРИЛОЖЕНИЯ.

Mag. Math. Bachvarov D.[1], Assist. Prof. PhD Student Eng. Ivanova V.[2], Assoc. Prof. PhD Eng. Ilcheva Z.[1],
Assist. Eng. Boneva A.[1], Eng. Baruh N.[3]
Institute of Information and Communication Technologies – Bulgarian Academy of Sciences,Sofia, Bulgaria[1]
dichko1952@abv.bg, zlat@isdip.bas.bg, a_boneva1964@abv.bg,
Institute of Robotics - Bulgarian Academy of Sciences, Sofia, Bulgaria[2]
iwanowa.w@abv.bg
SD "ELL – Danev, Bozhilov &", Sliven, Bulgaria[3]
n_baruh@abv.bg

Abstract: *In this publication is presented a method for design of an operating station based on a TCL / TK - portable open source programmed platform, implemented for different operating systems (Windows, Unix, Linux, Solaris, Android).*
In the article three possible approaches to building an operator station are considered:
(1) Classical approach which is realized by using of "event processing", "call back" and "graphical tools" features of TCL/TK;
(2) The "Multi- interpreters" realisation using a number of interpreters of TCL;
(3) "Multi- threads" mode, including cooperative working of one or more program threads.
The described approaches are illustrated by presented implementation of an operating station to wireless control system for a group of intelligent instruments which are designed for laparoscopic surgery.

Keywords: OPERATING STATION, PROCESS CONTROL, TCL/TK, INTERPRETERS, MULTITHREADS, WIRELESS NETWORKS, LAPAROSCOPY

1. Introduction

The operating station is a basic tool for the building of a user interface to controlling systems. According to the requirements of the controlled processes the operating station software must possess capabilities for real-time processing, to work in multitasking mode and to have tools for graphical visualization. There exist a number of software platforms useful for built of the operating station structures and functions. Some of them, as UNIX, Linux and Windows are designed as operating systems with embedded features needed for different applications.

Other approach is using of the open source program products, realized by the script languages, such as TCL/TK, Pearl and Piton. They allow executable code and resource requests for the operating system to be mixed in the procedure bodies. The procedures, written in concrete scripting language give to the developer free access to the system resources and shorten development time.

In this article is shown the developed process of Operating station, based on the use of TCL/TK under Windows.

2. Basic features of TCL/TK[1]

TCL/TK is a scripting language allowing the developers with abilities for accessing to the resources of the operating system. It is designed with "open source" GNU license and consists of two components: (1) TCL is C-like procedure oriented language, used for standard algorithms programming; (2) TK consists of language operators for making of requests to the operating system resources setting the resource parameters. Both TCL and TK operators included into TCL procedures are processing following common language rules. TCL/TK is a string oriented language. It has only a few fundamental constructs and relatively little syntax, which make it easy to use.

In the TCL/TK application could be used a number of program variables, global or local, and constants, each of them contenting string data. The basic language types of data are related to the string interpretations of the strings from the TCL/TK operators: text strings, numerical values, objects, lists of objects, indexed array of objects and etc. TCL/TK supports multilevel hierarchy space about data into the TCL procedures, assigning an access- number of each used variable. Global type variables having access- number 0 are accessed into all application procedures. The local variables having given them access- number are isolated from the variables with other access- numbers.

The operators in the TCL/TK applications could be internal (embedded into the language kernel libraries) or external names of procedures, defined and located anywhere into the application. TCL procedures have a number of formal parameters, presented as names of string variables and interpreted into the procedure body. Calling of the procedures is doing by their names, followed by a list of actual parameters. TK operators are of type "internal" only and have fixed number of named parameters with "default" pre-defined values. As the parameters are named, to each of the parameters could be assigned "actual" parameter value. If it hasn't done from developer the operator uses its "default" value.

TCL/TK[2] has embedded mechanism for processing of real-time applications. It is realized on the base of registering into the procedures of program events and binding to them TCL executing procedures. After registration, these events, at the moments of their occurrences, could be to start the corresponded processing procedures. The processing procedures can be programmed to be working in blocked or unblocked modes – in the first case the application is blocked until current event has occurred and corresponded procedure was executed, in the second the application and the processing procedures (to registered events) are allowed for executing at every moment . Registered events may be ones of the Windows program events, time events, TK events, communication events, global TCL/TK variables settings and etc.

Using the basic TCL/TK features is possible to create a number of program tools, useful for Operating station built, as Automata scheduler (State machine), Graphical user interface, Serial interface drivers, Embedded data base drivers, Server and Client TCP interfaces, Embedded HTTP servers and etc. Other important application of this approach is the mixing of TCL/TK and other external programs to be used the TCL/TK infrastructure and interfaces. For the purpose is necessary the external programs to be presented with their C code and linked with TCL/TK libraries together. It is possible to embed into TCL/TK application external C-coded functions by a modifying of their headers and converting them to TCL commands. This approach allows the developer to mount new interfaces to the applications.

3. Advanced features of TCL/TK.

3.1. Interpreters [2]

Typically, TCL/TK application uses one interpreter of all TCL and TK operators. The language model supposes the use of hierarchical structures including more than one interpreter, each of which having own set of operators and addressing space. On the top of the hierarchy is one full functionally interpreter of TCL/TK code – Master interpreter. It could create a number of Slave interpreters, restricting any of their standard commands or creating new commands. This process could be continued and the Slaves can create Sub- slaves ones, forming hierarchical structures. Each interpreter of the hierarchy structure has own name space, isolated from others. It can communicate with his creator only. The Master and each of the creators have abilities for accessing to its Slaves resources using alliances mechanism.

3.2. Threads [2]

Tcl/Tk is supporting multi- threads mechanism, allowing to the developer to create applications including a number of parallel processed tasks. Threads are multiple flows of execution within the same process. All threads within a process share the same memory and other resources. As a result, creating a thread requires far fewer resources than creating a separate process. Furthermore, sharing information between threads is much faster and easier than sharing information between processes.

The operating system handles the details of thread creation and coordination. On a single-processor system, the operating system allocates processor time to each of an application's threads, so a single thread doesn't block the rest of the application. On multi-processor systems, the operating system can even run threads on separate processors, so that threads truly can run simultaneously.

After starting, there is only one thread executing, often referred to as the main thread, which contains a single Tcl interpreter. Any thread can create another thread at will; it isn't limited to starting threads from only the main thread. There exist messages exchanging mechanism between different threads allowing task synchronizations. Other useful tools used for between tasks synchronizations are the shared variables and the muttexes. Each tread in TCL/TK has own Main interpreter, unassessed from other threads.

4. Operating Station Architecture [3],[4]

The typical architecture of an operating station includes a number of functional blocks as:

- Interface block, providing the connection of the operating station to external devices by wire or wireless means.
- Communication block, providing communication with a remote client or server over the Internet;
- Graphical User Interface (GUI), providing operator interface to controlled technological processes using program or graphical tools;
- Video control, including image processing program tools used in real time visualization;
- Data Base block, supporting the work with a number of embedded Data Base packages;
- Expert System block, a piece of software programmed using artificial intelligence techniques. Such systems use databases of expert knowledge to offer advice or make decisions in such areas as medical diagnosis;
- Simulation (training) block.[5] It contains program model of the controlled processes. The operator could train on this model before to send, by GUI and Interface block, his commands to the real external process devices.
- Custom processing block. It is used as internal control mechanism of the program processes and is based on the use of embedded software state machines (finite automata structures).

All functional blocks can be designed based on TCL/TK program structures and tools. As their functionalities are relative autonomous could be use multi-threads for their software realizations. When are hierarchical or other types of dependencies between the different block functions is existing the last can be realized using corresponded interpreter hierarchies structures.

Such solutions are useful as they allow one host machine program process to be decomposed to set of concurrent logical program processes, each of them designed as TCL/TK procedure. Other benefits of this approach are the abilities of data protection and restricted access to different block resources. On Fig.1 is shown typical architecture of Operating station.

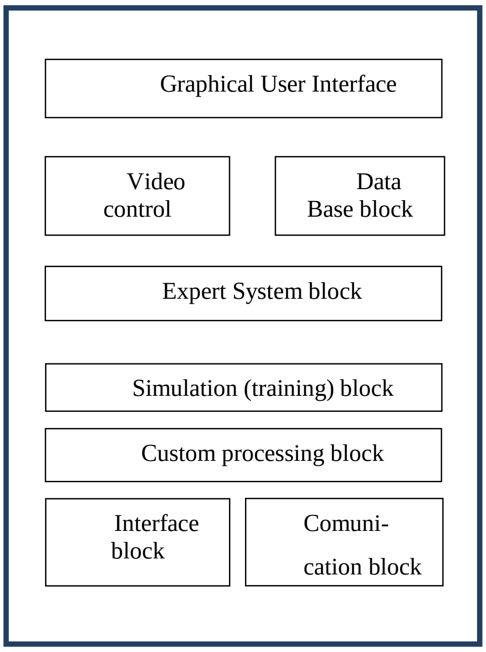


***Fig.1**. Operating Station typical architecture.*

5. Example of design process of working station application using TCL/TK under Windows [6].

On Fig. 2 is shown Operating station to wireless control system for a group of intelligent instruments which are designed for laparoscopic surgery designed using basic features of TCL/TK. It is hosted on Laptop PC under Windows 7. The each of intelligent laparoscopy instruments is sophisticated electro-mechanical device, executing specific laparoscopic surgery functions and having embedded wireless controller. The controllers of intelligent instruments and the operating station are forming LAN, used for between nodes communication of messages and commands and using custom designed protocol and network stack (uMac).

The example demonstrates the operation of the tool by searching for contact point, detecting the presence or absence of contact at the top of the tool with a surface, measuring the interaction force of the tool with a given surface. The obtained results are visualized in a graphical form and save in a database. The results are compared with other such results of the program results.

The range of the commands allows the user or doctor to control the device and motors, actuators, sensor –force and position, which are connected to the microcontroller.

Some of basic program functions are Commands for Motion - Start and Stop machine, command for insertion and retraction linear of the tools, Mode-Automatic and manual, current step positions of the motor, save in samples or save in results, visualization and comparison of the measuring and etc.

The program is designed for four instruments, but is only realized for one. The first point is to be selected which instrument has to work. The Fast Positioning button introduces a special mode to quickly search the working area.

Motion is a control program button with two alternative states: Start and Stop Motion of the instrument. It allows and prohibits the movements (insertion and retraction linear) of the laparoscopic instrument. Also the movements are forward and backward. According to the dimension of the step, the stepper motor respectively the instrument can work in four modes:

- a complete step;
- ½ step;
- ¼ step;
- 1/8 step.

The choice of the mode of the motion is via micro-switches

Automatic Control is associated with a program for fast searching of the force range where are starting the experiment measurements.

The search is making by doing of micro- step movements to forward direction, coupled with force measurements by the embedded sensors, while the force value is less then set by the operator. When the measurement force value is bigger or equal to set, a sequence of 300 consecutive micro step movements is staring. After each of the step movement is measuring, averaging and sending the result force value on the wireless network to the operating station.

History includes all commands and rapports during the communication sessions. They are also duplicated in a file (archive.txt from Folder Laparoscopy) by selecting the Save button, located in the top row of the initial screen.

DTBS Samples и DTBS Results are Graphical tools that provide the operator access to the files stored in the two databases for eventual visualization and benchmarking. They have the same organization and ways of working. Each one includes a list of filenames supported by the appropriate base at the current time, a sheet for locating a visible part of the list, and methods for selecting and positioning them in the lists, using several embedded program buttons.

Operation station of the example includes next functional blocks:

- On the left half of Fig.2 is shown GUI, consists different TK tools-Canvas, Buttons, Entries, Radio buttons, Text Boxes and etc. All elements are set by TCL procedures. They are using "binding" mechanism for detecting and processing pre-registered program events, connected to the operator controlling. GUI sets common variables for activating of other procedures, waiting on "VWAIT" loops. It is in dependence of other functional blocks and some of them are in dependence of execution of GUI actions;
- Interface block includes driver procedures for data stream supporting on the wireless network. It is based on the serial data processing of messages from wireless gateway device, connected to Operating station by wires USB. The receive byte event is "binded" to the driver. The send of byte is doing by "non- binded" driver. Interface block is in dependence to GUI;
- Communication block consists of server and client procedures built using the TCL/TK infrastructure for TCP processing. It is in dependence to GUI;
- Data Base block consists of a number of different data bases and is in dependence to GUI, Interface and Communication blocks;
- Video- control consists of the special image processing library BLT 2.0 and is in dependence to GUI;
- Expert system block consist of Rule Base and interpreter of making decision software. It is in dependence of GUI;
- Simulation block consists of the controlled processes model and State machine (finite automata controlling) and is in dependence of GUI. Some functions of GUI are binded to the program events generated from Simulation block software;

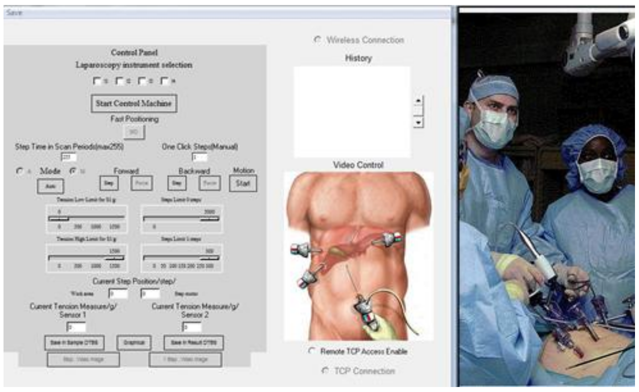
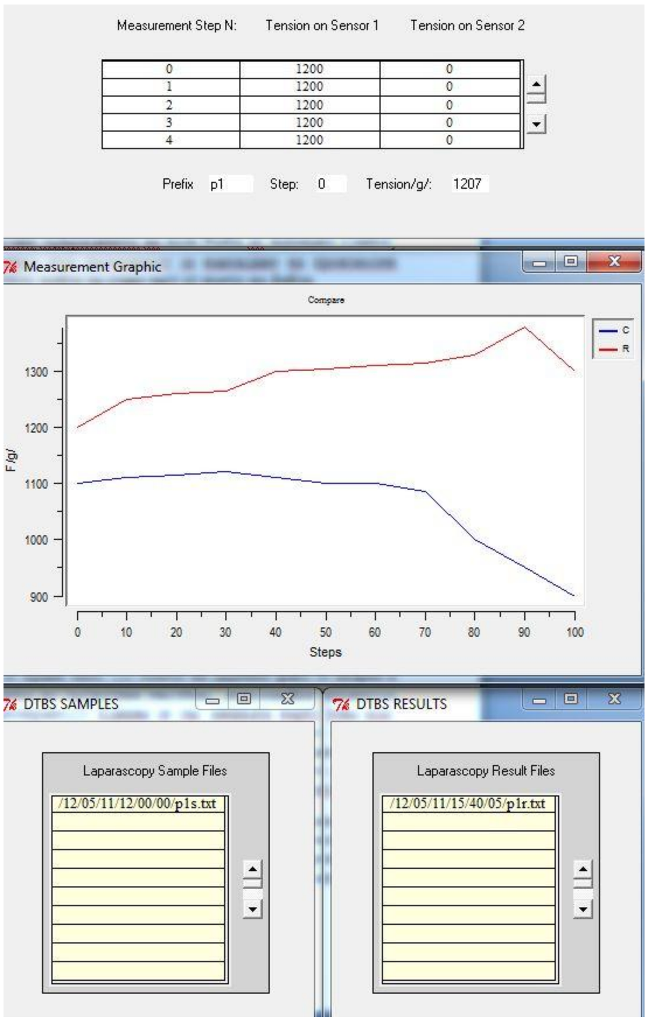


Fig.2. Operating station to wireless control system for a group of intelligent instruments (GUI).



Measurement Step N:	Tension on Sensor 1	Tension on Sensor 2
0	1200	0
1	1200	0
2	1200	0
3	1200	0
4	1200	0

Prefix p1 Step: 0 Tension/g/: 1207

/12/05/11/12/00/00/p1s.txt

/12/05/11/15/40/05/p1r.txt

Fig.3. TCL/TK tools used by Working Station.

- Some TCL/TK tools are using to give of the operator abilities for evaluation of the inteligent instruments functioning. They are built on Canvas mechanism infrastructure embedded in TK. Other tools are working with graphical data and are included in the library Graph. On Fig. 3. are shown applications realized by these tools. They allow draw in real time of graphs of incomming data from the network, saving data curves into special Data Bases, matching experimental data with prototypes data, controlling of the measurement process.

6. Conclusion.

The developping of Operating Station is big challenge in the designing of Control systems. It, being the main operator tool for monitoring and controlling of the technological processes, is sofisticated system integrating a lot of subsistems. The use of generic technology, common language (TCL/TK) with its features (virtual paralel processes and embedded librarries) significantly simplifies the developer's work. The language abilities for mixing C with TCL codes, adding and working with precompiled TCL extension packages, the protection of the application data (Safe TCL), multi interpreters and multithreading mechanisms, built-in network connectivity of the applications and multi-platform supporting makes the TCL/TK a powerful tool suitable for designing complex dynamic objects of the type of Operating Station.

References:

[1]. Practical Programming in Tcl and Tk. Brent B. Welch, Ken Jones, Jeffrey Hobbs. Prentice Hall Professional, 2003.

[2].www.tcl.tk (last visited 7.02.2018).

[3]. www.m2mic.free.bg (last visited 7.02.2018).

[4]. www.bpmio.free.bg (last visited 7.02.2018).

[5]. Batchvarov D., Geortchev V., Boneva A., Krasteva R., Tcl/Tk Based API for OMRON'S RS485 Interface Network, Second Meeting of Bulgarian – Austrian Automation's Day (08 June, 2001), j. Problems of Engineering Cybernetics and Robotics, Vol. 53, ISSN 0204-9848, pp. 80-86, BAS, Sofia, (2002).

[6]. D. Batchvarov, A. Boneva, Z. Ilcheva, S. Angelov , V. Ivanova, Tools for control of mechatronic objects using the wireless network stack uMAC, Proceedingsfor International Conference AUTOMATICS AND INFORMATICS' 2017 4-6 October 19, 2017, ISSN 1313-1850, CD: ISSN 1313-1869, John Atanasoff Society of Automatics and Informatics,Sofia, Bulgaria, pp. 77 – 80