

# SCADA-System Multiservice for Automated Configuring Telemechanic Means

Oleksandr Kulchytskyi, Tamara Loktikova, Andrii Morozov, Nataliia Krushynska

**Abstract**—The development of SCADA-system multiservice for automated configuring telemechanic means was introduced in the article. It was described the structure, the working algorithm of the multiservice developed, it was considered its most important and responsible constituent parts. The latest architectural tendencies and the modern programming language were used in the process of design. It was proved that the developed multiservice enables to increase and improve the quality of setting telemechanical system.

**Keywords**—Automated configuring, C#, Information and management telemechanical complex, SCADA-system, Telemechanics .

## I. SETTING THE GENERAL PROBLEM

SCADA-system multiservice for automated configuring telemechanical means (multiservice – further in the context) is the programme (or the application) fulfilling the function of the interface for connecting the user and enables to create and develop the telemechanic system quickly and conveniently without necessary involving the highly-qualified staff. Multiservice means implementing the principle of using patterns: if there is the class of typical tasks, thus they should be solved automatically with the minimal involvement of human resources, how-ever, at the same time multiservice should not be only a set of the ready patterns.

In particular, there exists the whole class of the tasks with the setting that even the high-level specialist would prefer to solve in batch, formulaic that is. These are the tasks of the primary setting that occur while installing the system.

For the first, there occurs the necessity to set all the system parameters at once, or, at least, set them into the certain state by default. The number of such settings is estimated as thousands, and there always exists a probability of a human error. Using the patterns while setting decreases the probability of an error, however even if it occurs, it will have a minimum influence and it will be possible to be corrected later on.

For the second, such setting will be too time-consuming for the complete writing all the system parameters manually nowadays, even using the proper manuals.

For the third, the majority (practically about 80%) of settings by default, if they are assigned automatically or on the basis of a short dialogue, will not need to be changed at all. These are the settings of language, hardware, network etc.

While developing multiservice there should not be any doubts left as for what is responsible for this or that part of the functionality of the programme. Practice shows that a poorly organized property or an instrument gets not only unused within the time, but also impossible for usage as a countless number of expansions occurs and they are not taken into account by these properties or instruments. Any new multiservice expansion should be clearly explained to the user to avoid potential mistakes. All stated above is a regular practice of describing or arrangement, however, exactly in case of creating multiservice, when there is a number of properties and opportunities of working with them, everything that may be not clear or taken

O. Kulchytskyi, PE «Research and Production Enterprise «PROMEX», Zhytomyr, Ukraine (e-mail: kuliastik@i.ua).

T. Loktikova, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine (e-mail: tamlokt@ukr.net).

A. Morozov, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine (e-mail: morozov@ztu.edu.ua).

N. Krushynska, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine (e-mail: natkrushyn@gmail.com).

wrong will be, oddly, not clear or taken wrong. Therefore there occurs a burning issue of correct regulation, description, and comprehensibility of objects properties or instruments needed for the user.

One of the important questions lies in the interconnection between functionality or the objects of multiservice. It is worthy to start from the fact that avoiding the interconnection among separate elements close according to their functionality is impossible. For example, a few modules perform serialization, thus which one will be responsible for forming common elements and creating the file? If one always tries to combine them into one module, then instead of separate expansions to functionality one will eventually get a non-comprehensive monolith for an intellect. Therefore it will be harder and harder (or impossible) to expand and maintain it. The solution of this problem lies in selecting common units out of such modules into a separate module. This separate module is possible to extend in the shape of another module that enables to avoid conflicts among the modules. It allows avoiding the situation when designing each new expansion becomes a more efforts-taking and unreliable matter [1].

The main advantage of multiservice implies solving the user's tasks. The wider the range of these tasks is, the more interesting multiservice itself – on condition that the interface does not press the limits of user's object model [2].

As multiservice tunes the system of telemechanics itself, thus it has a wide range of application areas: objects of energy, mines, the Tram-Trolleybus Office, outdoor city lighting, rail transport, subway, large industrial objects, oil fields, airports, water utilities [3].

## II. PRINCIPLES OF CONSTRUCTING INFORMATION AND MANAGEMENT TELEMECHANIC COMPLEXES

The modern information and management telemechanic complex (hereinafter referred as IMTC) is the system of telemechanics of the 4th generation characterized by using microprocessors and microcomputers in its content. There are the two types of IMTC: one-level IMTC (it has one central receiving and transmitting station (hereinafter referred as CRTS) and a multi-level IMTC (it has two and more CRTS).

IMTCs are constructed on the principles of SCADA-systems.

SCADA-system (Supervisory Control And Data Acquisition) is a human-machine interface catering for the interaction of a person (an operator, a dispatcher) with an automated technological process.

Modern SCADA-systems fulfil numerous functions, in particular: tuning on the certain task; dispatcher management; automotive management; saving the history of the processes; fulfilling safety functions; fulfilling general system functions.

However the main characteristic of SCADA-system is the availability of the interface with the user. Without this interface management gets automatic, unlike dispatcher management.

As the system of dispatcher management SCADA should provide fulfillment of the following tasks:

interaction with an operator (providing visual and sound information, transmitting operator's commands to the system); assistance to the operator in decision making; automated alarm in case of failures and critical situations; transmitting informative messages to the operator panel; event logging; extracting the information from the archives and presenting it to the operator in the comfortable view for perception; preparing reports; accounting development of technological equipment.

For fulfilling these tasks of SCADA-system it is necessary to create a system within the following multiservices: «Configurator», «Core», «Reflecting frames», «Graphs of altering meanings of sensors», «Retrospective», «Frames database». We would like to stop on «Multi-service-Configurator» in particular.

In order to design SCADA-system multiservice for automated configuring telemechanic means it is necessary to clarify the structure of the system to be modelled.

The structure of the modern system of telemechanics or one-level IMTC contains the following constituents: central receiving and transmitting station or control center (CRTS); remote terminal unit (RTU); control board (CB) or interactive video-wall; sensors of alerting, metrological, code signals (SAMCS); power units (PU); computer (C). Devices may be connected via radial lines of communication (RLC); mainstream lines of communication (MLC); transit or chain lines of communication (TLC).

The structure of the multiple-level IMTC is characterized by having two and more CRTS in its constitution. It is organized a direct information communication among RTU of different levels. Considering this, IMTC gets a rank of a network system.

Devices of CC (CRTS) and RTU may be constructed by different principles: mainstream-module, distributed, mixed [4].

Nowadays there are also three generally known variants of telemechanics objects automation: centralized system, distributed system and mixed system [2].

### III. THE MAIN TELEMECHANIC PROTOCOLS

For the connection between telemechanic means among the most frequently used are the following: Modbus RTU (Modbus TCP), IEC 101 (IEC 104) and «Granite micro» («Granite TCP»). The consolidated characteristics of these protocols and their TCP expansion are introduced in **Table I**.

TABLE I. CONSOLIDATED CHARACTERISTICS OF THE MAIN TELEMECHANIC PROTOCOLS

	Modbus RTU	IEC-101	«Granite micro»
Protocol that is TCP expansion	«Modbus TCP»	«IEC-104»	«Granite TCP»
Support of RS-485 / RS-232	+	+	+
Length of frame	256 bytes	120 bytes	256 bytes
Length of frame for TCP expansion	260 bytes	278 bytes	1024 bytes
Work mode	«Master» – «Slave»		
Work mode for TCP expansion	«Server» – «Client»		
Internal addressing	Slave ID (from 0 till 247);	RTU address (from 0 till 255)	RTU address (from 0 till 8192); channel number (from 0 till 100); place of functional module (from 0 till 15)
Additional addressing for TCP expansions	Transaction number (2 bytes), IP(4 bytes)	IP(4 bytes)	IP(4 bytes)
Supporting transmission modes	«Reading» «Reading-Recording»	– Sending without response; – sending with confirmation; – sending on request	«Service» «Facilitator» «Enquiry»
Control sum	CRC16 (polynomial $2^{16}+2^{15}+2^2+1$ )*	Arithmetic sum in module 256 of all bytes	CRC (polynomial $2^{15}+2^{12}+2^5+1$ )
Time markers	Absent	Astronomic (up to 1 ms)	Relative (up to 1 ms)
General reliability	+	++	+++
General efficiency	+	++	+++
Maximum transmission rate	115200 Bod		
Probability of undetected mistake	$10^{-8}$	$10^{-4}/10^{-8}$	For analogue sensors – $10^{-8}$ , signal sensors – $10^{-12}$ , commands of teleoperation – $10^{-16}$
*TCP is responsible for Modbus TCP protection			

#### IV. OBJECTIVE OF THE WORK

It is offered the development of multiservice of SCADA-system, corresponding to all the requirements stated above and works with telemechanical devices according to protocols «IEC 870-5-101», «IEC 870-5-104», «Modbus TCP», «Granite», «Granite-Micro», «Granite Ethernet TCP/IP», «Info Proto».

#### V. THE BASIC MATERIAL SUPPLY

The structural scheme of the multiservice is presented in Fig. 1.

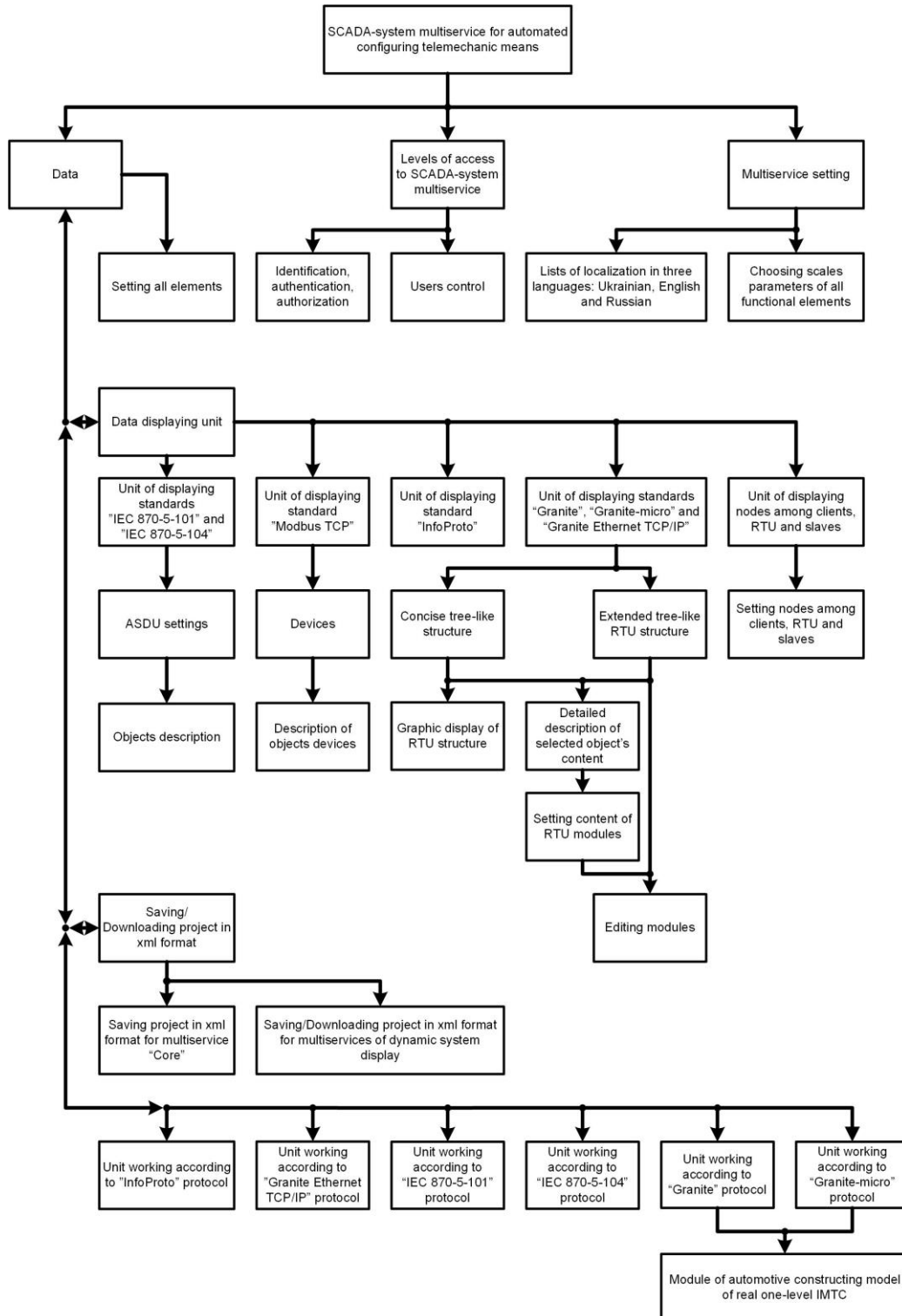


Fig. 1 Structural scheme of multiservice

The names of the node for a given circuit reflect their purpose and principles of their functioning.

Introducing multiservice of SCADA-system to the user starts from the main form of programme, shown in **Fig. 2**.

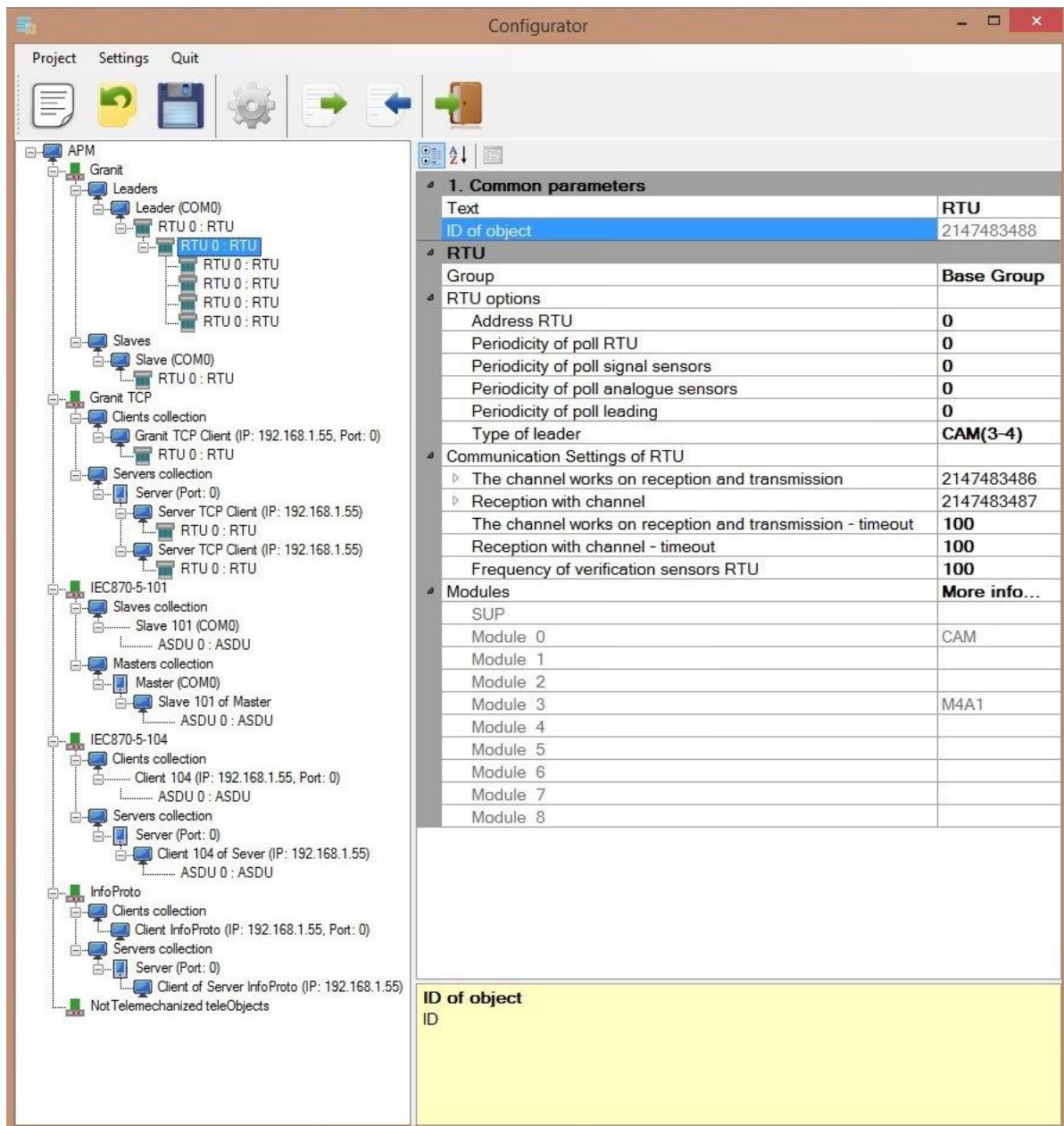


Fig. 2 The main multiservice form

The menu «MenuStrip» is located above. The insets of the given element enable to create the new project, open the already existing project, open the project within the ones opened before, save in xml format, edit dynamic lists, add new users of multiservices of SCADA-system and set up the programme in the way convenient to the user (language, fonts). [5]

The element «ToolStrip», located under the element «MenuStrip», functionally fulfills the same actions as «MenuStrip», however unlike it, each element has got a picture button with a graphic image that intuitively helps the user to find the needed element. Such duplicating functionality is created in order to allow the user to choose those instruments for his work that would be more convenient for operation.

On the left (**Fig. 2**) there is system elements structure tree. Data in the programme are presented via nodes united in the structure element «TreeView». The programme comprises two tree-like structures: the first, containing the information about data, and the second, which is a smaller copy of the previous one and it is reflected in the main form tree. This decision increases system comprehensibility for a person.

Having chosen any node of the tree by the mouse button, the expanded information concerning the object on the right in the element of reflecting the content of the object «PropertyGrid» will appear [6].

It is possible to switch from the main form to the form «Expanded editing modules». In this form (**Fig. 3**) the element «TreeView» reflects a part of all the nodes with the information about data. In this part of nodes the first mode may be only RTU, that is the highest in the system structure. The principle of work for «TreeView» and «PropertyGrid» in this form is the same as in the main form. On the left from «TreeView» there is a dynamic graphic image of RTU selected in «TreeView» (the node is additionally highlighted in a grey color), and RTU, located higher and lower according to the hierarchy. As it is shown in **Fig. 3**, modules, containing RTU, names of RTU and names of the channels are reflected. If «arrow control» is included into the settings, then modules may be replaced not only by arrow keys in the dynamic graphic display of RTU, but also from the keyboard. This graphic panel is scaled and moved if necessary [6].

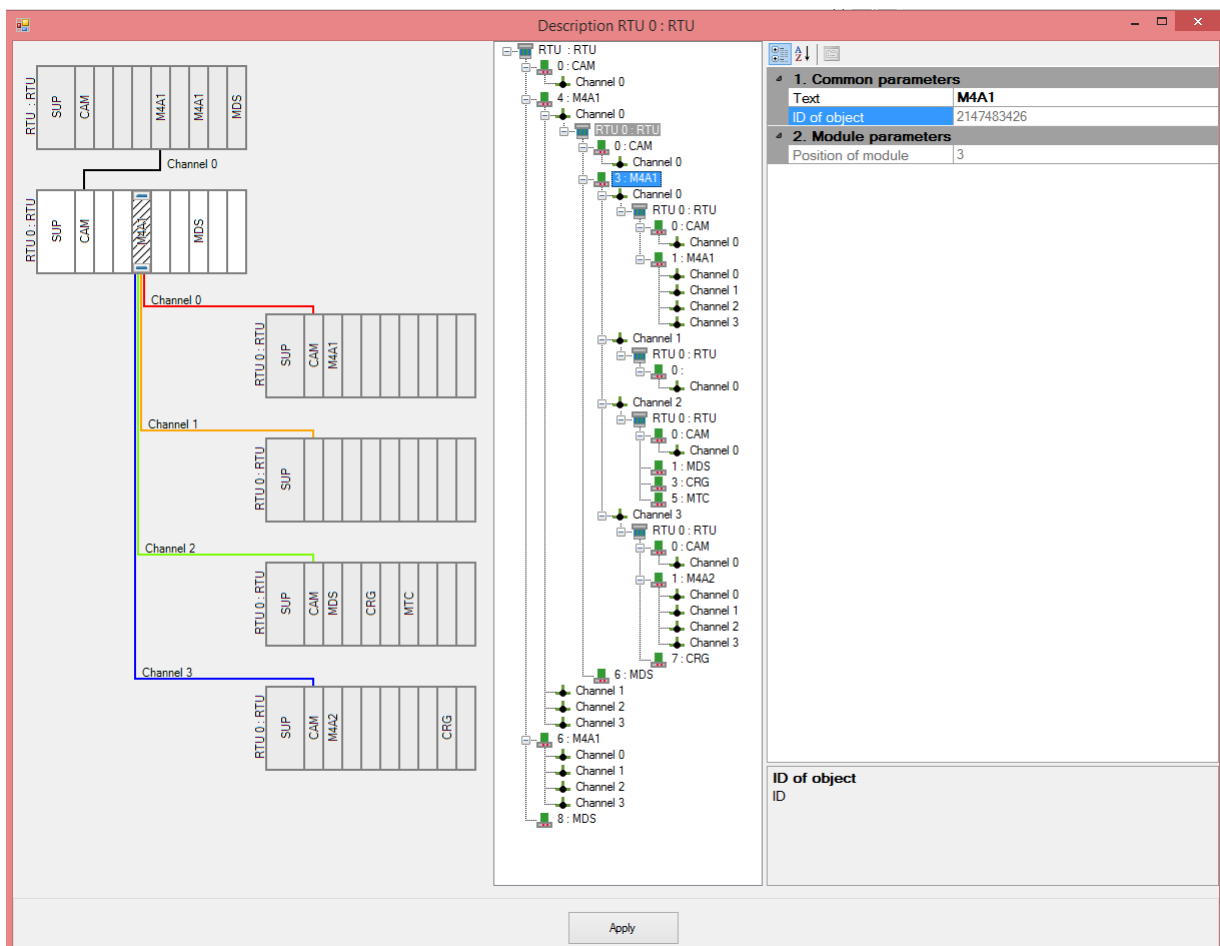


Fig. 3 Form for expanded editing modules

In **Fig. 3** it was introduced a universal form «Detailed editing objects» for modules RTU, Device or Application Service Data Unit (later in the text – ASDU).

On the left of the form in the element «Property Grid» it was shown the characteristics of

the main object, in the center in the element «Tree View» sub-objects of the main object are introduced, and on the right hand-side in the element «Property Grid» it was shown the characteristics of the selected sub-object in the tree. At first it was made a decision to create a unique form for each object, however practice showed that such unique forms have a very similar set of elements. Creating exactly the universal form is easier for user's perception as he approaches to work in a formulaic way, moreover, in terms of programming we get a more convenient and a more qualitative decision [6].

Having chosen the node of any tree by the right mouse button, it is possible to see the context menu. According to object's type, it may be deleted, cut, inserted, copied and, if it is RTU – open the form, introduced in Fig. 3. For the elements «Master» and «Slave» it is presupposed the inset «Scan», after that multiservice constructs a model of a real one-level IMTC according to protocols «Granite» and «Granite-micro».

The performance algorithm (according to protocols «Granite» and «Granite Micro») is the following. On the basis of data implemented into the element «Master» and «Slave», the connection is created on the consequent port with the stated parameters. After that a request is created, the programme waits for it in the form of data package with ID of module type according to the number of RTU and the place of module in it. If there is no answer or the unnecessary information appears (system packages may appear), then the algorithm will recognize it and RTU (and a module with it on the parallel) is not added into a programme. If the byte of ID type of a module is in the list of ID-bytes of modules, then this module is added to RTU (if the module is the first in it, then RTU is created itself).

In case if the added module is the module of data relaying, then a special request (or requests) is additionally created for checking the range of channels (the number of channels and ranges is known at once according to the module type) that gives the opportunity to find out how many RTU are connected via this module. The system of ranges is implemented to enable flexible expansion of the system in order not to reset it completely through the addresses of new RTU. When it was acquired the information about the existence of not yet created RTU in the programme, then the algorithm is transformed into recursion – the request is created again, the programme waits for it in the form of data package, where it will be ID of module type according to the number of RTU and the place of module in it.

The group of elements «Tree View» and «Property Grid» are used on all the voluminous forms. «Tree View» allows to comprehend easily the hierarchy of the system and find a necessary node of the «tree». «Property Grid» provides expanded information about the element selected in the tree. Data (as it is seen from Fig. 4) are presented as a pair: «name of data field» – «value». Having highlighted such a pair in «Property Grid» it will be given its brief description at the bottom. Multitude of such pairs and their description is formed automatically on the basis of attributes (attribute «Display Name» is responsible for «the name of data field», and «Description» – metadata) of properties of the class C# object, which was assigned the element «Property Grid» from «Select Object». The attributes may be reasserted and, for example, «the name of the data field» may be localized. It is also possible add the work with recalculation (enum в C#) via reasserting «Enum Converter». The drawback of «Property Grid» is the fact that it is impossible to set up displaying hierarchy of subclasses, for example, to skip the name of the subclass in the hierarchy of displaying. It is also necessary to use displaying in order to change signatures, descriptions, feature «only reading», concealing or displaying properties [6].

Each object in the system has got an individual identify number (ID) that enables to work effectively with the object that are not structurally interconnected. «ID» also allows to connect the objects between multiservices of SCADA-system. «ID» is created by the programme and the user cannot replace it, he can only look through its value.

Multiservice has got the lists of the localized names in the three languages – Ukrainian, English and Russian. Such a list may be easily expanded in case of creating a new content.

Fig. 4 Form of detailed editing RTU

For saving data of the project it could be possible to use database (DB) that enables to work faster in comparison with xml files. However database should be always maintained, telemechanic systems are rather complicated and thus it is needed to use the tree-like database that makes it confusing and inconvenient, transferring database requires extra efforts, there occurs the difficulty with the compatibility of versions, and, what is the most important – multiservice is used for creating the model of a telemechanic system and then converts data for other multiservices and it is not used in the dynamic tracking of telemechanic system. Due to these drawbacks of using database in the particular case it was chosen to save the project in xml format. The structure and small by the size elements are kept in the main file of serialization, and each element of RTU, Device and ASDU has got the unique file. That allows not only to navigate the structure of the whole system while analyzing the files of serialization, and also to create the method of a delayed downloading that primarily deserializes the main file and afterwards the file highlighted by the user. It will greatly decrease downloading time (SCADA-systems may have hundreds of elements and thousands of sensors).

Along with the files for saving the suggested development, specialized simplified files for multiservice «Core» and the complex of programmes for dynamic displaying of SCADA performance. These specialized files are created on the basis of multiservice data.

## VI. TESTING FUNCTIONALITY OF SCADA-SYSTEM MULTISERVICE

Performance, the main functions and opportunities of forms in SCADA-system multiservice for automated configuring telemechanic means were described above.

The experimental check-up is reasonable to be carried out directly via considering the performance of the automotive construction module of one-level IMTC module according to the protocols «Granite» and «Granite-micro».

It is necessary to give the definitions to the modules that will be used in testing: CAM –



combine module of controller, adapter and modem; CDG – controller-drive-gateway; M4A1 – module of four-channel line adapter for information exchange with external devices according to interfaces «current loop» and «RS-485»: M4A2 – module of four-channel line adapter for information exchange with non-modulated code-impulse signals; MDS – module of discrete signal management.

Experimental research was carried out with two RTUs. «The first» (on the right in **Fig. 5**) may contain three modules and it is connected to the computer via RS-232, and also via its own module of CAM – to «the second». «The second» (on the left in **Fig. 5**) may contain four modules and it is communicated with «the first» also via its own CAM.

As it may be seen in **Fig. 5**, in the first test «the first» RTU comprises the following modules: CAM (0 place), M4A2 (the 1st place) and MDS (the 2nd place), and «the second» RTU – CAM (0 place), CDG (the 1st place), CAM (the 3rd place).

In order to construct the model of one-level IMTC in multiservice it is necessary to apply the context menu for the node «Master» or «Slave» (**Fig. 2**) and opt for «Scan». Then the form for setting up the range of addresses. It is necessary to choose the range; let it be within 0 till 12 (it is allowed that RTU from testing will have the addresses from this range). At first only zero places within the given range of RTU addresses are questioned and, if this place is not empty, then 8 more RTU places are questioned (that is the largest possible number). That takes some time, depending the selected range and real-life RTU of the scanned structure.

After finishing the process in order to review the unfolded structure it is necessary to move on to the form for detailed modules editing (it is possible to do via double-clicking on RTU in the tree «Tree View» of the main form or in «Property Grid» (selecting RTU in the tree «Tree View» before that) via the button «Read more...»). The results are introduced in Figures 7–8 for both tests where the graphic display of the researched structure is shown under arrow «1» and it is possible to assess on which places of RTU there are its modules; under arrow «2» – a researched structure in the form of the tree, and under arrow «3» it was given the display of the parameters for «the second» RTU in the element «Property Grid», where it is also possible to estimate which places of RTU have its modules [6].



Fig. 5 Real equipment in the first test

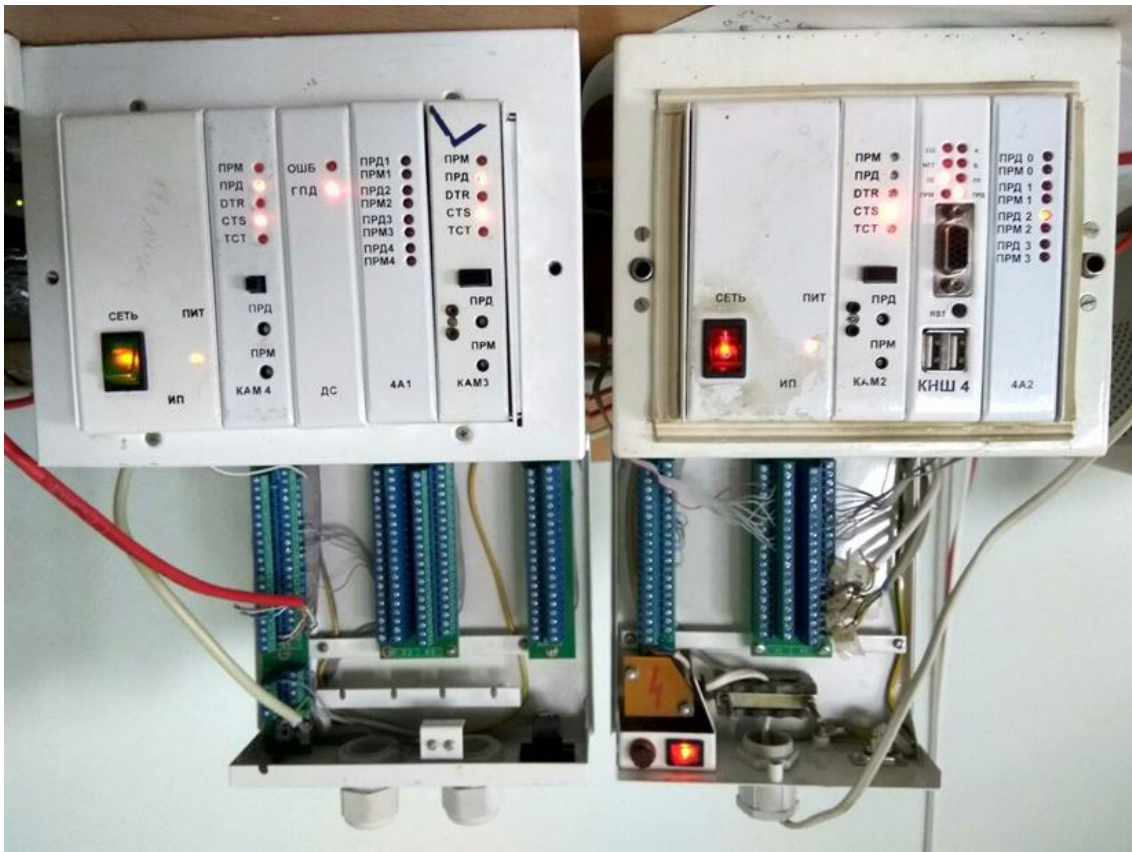


Fig. 6 Real equipment in the second test

**Description RTU 0 : RTU**

**1. Common parameters**

Text	RTU
ID of object	2147482616
RTU	
Group	Base Group
RTU options	
Address RTU	0
Periodicity of poll RTU	0
Periodicity of poll signal sensors	0
Periodicity of poll analogue sensors	0
Periodicity of poll leading	0
Type of leader	KAM 1
Communication Settings of RTU	
The channel works on reception and transmission	2147482615
Reception with channel	2147482614
The channel works on reception and transmission - timeout	100
Reception with channel - timeout	100
Frequency of verification sensors RTU	100
Modules	More info...
SUP	
Module 0	CAM
Module 1	CDG
Module 2	
Module 3	CAM
Module 4	
Module 5	
Module 6	
Module 7	
Module 8	

**ID of object**

Apply

Fig. 7 Results of the first test

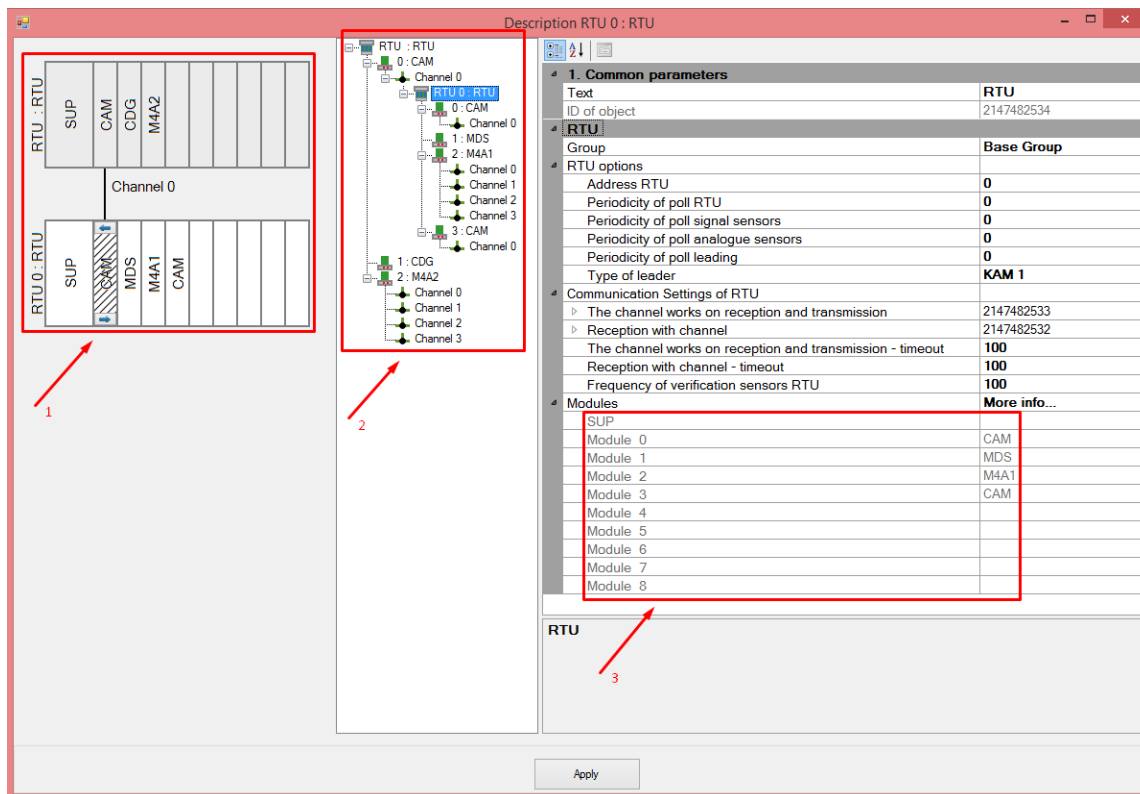


Fig. 8 Results of the second test

On the basis of the given results and introduced researched structures (**Figures 5–8**) it is possible to conclude that the module of the automated construction of the model of one-level IMTC according to the protocols «Granite» and «Granite-micro» corresponds to the set requirements, it fulfills the set tasks in a correct and a distinct way.

It is clear that real telemechanic systems may have hundreds of RTUs which, in their turn, comprise up to 9 modules, however this simplified model enables to show the essence of purpose for this multiservice module: instead of creating the structure of one-level IMTC and writing objects' parameters manually by the human, all of this is performed automatically (it is necessary only to fill in the names of the objects for the users) and without common mistakes (of course, if there are not any mistakes in physical settings, however it may occur the modernization of the automotive construction module which may detect even such mistakes that will cause decreasing their number which already stays minimal).

## VII. CONCLUSION

The designed multiservice has got the following advantages:

- works with telemechanic means according to the protocols «IEC 870–5–101», «IEC 870–5–104», «Modbus TCP», «Granite», «Granite-micro», «Granite Ethernet TCP/IP», «Info Proto»;
- developed using language C#, which was recognized as a flagship language for Microsoft corporation as it uses new opportunities of .NET, supports a lot of modern technologies of information displaying, interacts with different databases etc. [7];
- using designing templates is one of the developer's tools that enables him to save the time and make the decision more qualitative and comprehensive.
- the given visibility of the interface of the user makes multiservice ergonomic and it allows easy navigation in the structure of a telemechanic system and it enables a quick set up of the necessary elements, add sensors, expand or, vice versa, remove the part of

the devices;

- keeping the project in in xml format gives the opportunity for easy complementing or changing the model, without sufficient loss that enables the specialist to set up the telematic system remotely or analyze it without using additional specialized programmes. While downloading a big project from xml file there is a possible few seconds delay (or even a bit longer). Saving the project not in one large, but in a quite big number of files allows to avoid such a delay while using the method of delayed downloading. This method presupposes downloading not the whole project at once, but only the structure without internal objects. While clicking the element of the tree-like structure according to the user's interest reloading the internal objects from the file takes place;
- the module of the automotive construction of the model of one-level IMTC according to the protocols «Granite», «Granite-micro».
- once written complement keeps its working ability in any future versions of the programme on condition that the new multiservice declares a reverse compatibility with an old version. Along with it, appearance of the new applications does not interfere with the work of those written before (except the case, when the new complement is created to replace the old one).

Thus a «proper» multiservice provides convenient tools for modeling user and system areas, for creating the interface and for writing logics of transformation downwards and upwards, and also the instruments for interaction between the modules and their control.

#### REFERENCES

- [1] Martin R. Clean Code A Handbook of Agile Software Craftsmanship [Text]: Handbook/ R. Martin. – Massachusetts: Prentice hall, 2008. – 431 p.
- [2] Kulchytskyi O.R., Loktikova T.M., Ishchenko O.S. Systema avtomatichnogo konfigurovannya telemekhanichnykh zasobiv: tezy Vseukrainskoyi naukovo-praktychnoyi on-line konferenciyi aspirantiv, molodykh vchenykh ta studentiv, prysviachenoyi Dniu nauky .Zhytomyr, 16–18 travnya, 2018. – s.83–84
- [3] Loktikova T.M., Kulchytskyi O.R., Bolshoi V.A. Doslidzhennya suchasnogo stanu ta tendency rozvytku informaciyno-upravliayuchykh telemekhanichnykh kompleksiv: Tezy dopovidey IX Mizhnarodnoyi naukovo-tekhnochnoyi konferencii «Informaciyno-kompyuterni tekhnologii 2018» (Zhytomyr, 20–21 kvitnya 2018). – Zhytomyr, 2018. – S.176–177.
- [4] Loktikova T.M. Metody ta zasoby obrobky i peredachi informaciiv systemakh I merezhakh peredachi danykh [Tekst]: navchalnyi posibnyk/ T.M. Loktikova, A.V. Morozov, V.A. Bolshoi, N.O. Kushnir. – Zhytomyr: Zhymyrskiy derzhavnyi tekhnologichnyi universytet, 2015. – 162 s.
- [5] Martin D., Professional XML [Text]: tutorial / D. Martin, M. Birbek, M. Key etc. – Birmingham: Wrox Press, 2000 – 1200 p.
- [6] Microsoft [Electronic resource]: The official website Microsoft – System. Windows. Forms – Microsoft – Access mode: <https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms?view=netframework-4.8>.
- [7] Rikhter, J. CLR via C# [Text] tutorial / J. Richter. – Washington: Microsoft Press, 2013. – 813 p.
- [8] Soroko V.I. Avtomatika, telemehaniika, svyaz I vychislitilnaya tekhnika na zheleznykh dorogakh Rossii [Tekst]: Enciklopediya. T1./ V.I. Soroko, V.M. Kainov, G.D. Kaziev. – M.: NFP «Planeta», 2006. – 736 s.
- [9] Portnov E.M., Ishchenko A.S. Sistemotekhnika sovremennykh informaciyonno-upravlyayushchikh kompleksov// Visnyk inzhenernoyi akademii Ukrainy. – 2006. – №1. – s.39–46
- [10] Boltenkov V.I. Konfigurirovaniye I nastroyka avtomatizirovannykh system [Tekst]: uchebnoye posobiye/ V.I. Boltenkov, A.L. Litvinov, N.V. Lycheva. – Belgorod: Izdatelstvo BelGU, 2004–81 s.
- [11] Portnov E.M. Analiz sostoyaniia proizvodstva, principov postroyeniya I tendency razvitiya informaciyonno-upravlyayushchykh kompleksov dlya ASU raspredelennykh enegroobyektov i proizvodstv [Tekst]/ E.M. Portnov. – M.: MIET, 2002. – 78 s.