UDC 681.5:656.257
DOI: 10.15587/1729-4061.2017.118833

DEVELOPMENT AND STUDY OF TECHNOLOGICAL VISUAL PROGRAMMING OF LOGIC CONTROL PROBLEMS

I. Furman
Doctor of Technical Sciences, Professor*
E-mail: ifurman@ukr.net
A. Allashev
Assistant*
E-mail: alex.freeman.ua@gmail.com
A. Piskarev
PhD, Associate Professor, Head of Department*
E-mail: post@3g.ua
S. Bovchaliuk
PhD, Associate Professor*
E-mail: bovchaliuk@ukr.net
*Department of Automation and
Computer-Integrated Technologies
Kharkiv Petro Vasylenko National Technical
University of Agriculture
Alchevskyh str., 44, Kharkiv, Ukraine, 61002

Розглядається підхід до розробки програм логічного керування за допомогою табличної проблемно-орієнтованої технологічної мови. Запропоновано концепцію технологічного візуального програмування, що дозволяє залучити безпосередньо інженерів-технологів до створення програм логічного керування. Розроблено метод автоматичної генерації програмного коду на стандартних мовах сучасних засобів автоматизації за таблицями технологічної циклограми

Ключові слова: проблемно-орієнтовна мова, таблична циклограма, автоматизація програмування, програмований логічний контролер

Рассматривается подход к разработке программ логического управления с помощью табличного проблемно-ориентированного технологического языка. Предложена концепция технологического визуального программирования, позволяющая привлечь непосредственно инженеров-технологов к созданию программ логического управления. Разработан метод автоматической генерации программного кода на стандартных языках современных средств автоматизации по таблицам технологической циклограммы

Ключевые слова: проблемно-ориентированный язык, табличная циклограмма, автоматизация программирования, программируемый логический контролер

1. Introduction

In recent years, there has been an increasing interest on the part of various specialists to languages and technologies of programming of systems of logic control over industrial sites. With increasing level of automation of industrial sites, requirements for quality of software of logical control systems are becoming increasingly high. Nowadays, in industrial logic control systems, the basic element is the programmable logic controller (PLC). PLC is a universal microprocessor device. PLC is designed to operate in real time mode under production conditions. At present, there are a great number of industrial controllers of different manufacturers in the market.

For standardization of a large number of variants, in 1993, the International Electrotechnical Commission issued the standard IEC 61131-3, which describes language means of programming of industrial controllers. The standard includes three graphical and two text languages [1]:

– SFC (Sequential Function Chart) is the graphical language that allows description of the control algorithm in the form of a set of pairs of steps and transitions;

– LD (Ladder Diagram) is the language of ladder diagrams, graphical language, constructed on the principles of relay-contact schemes;

– FBD (Function Block Diagram) is the graphical language of functional units;

– ST (Structured Text) is the high-level text language with Pascal-like syntax;

– IL (Instruction List) is a low-level text language, representing a simplified assembler for problems of logical tasks.

When creating IEC 61131-3 standard, developers described existing at that time popular PLC programming languages. As early as then, it was clear that PLC programming languages should be simple and visual for engineers-technologists to be able to create control programs themselves. Nowadays, however, professional programmers are involved in creation of PLC control programs [2, 3]. When implementing the logic control algorithm, a programmer creates a program code in the language of the standard in a specialized environment. The program code remained “closed” for understanding and verification for an engineer-technologist, who is the most competent expert in the subject area of the automated process, but does not have knowledge of languages programming.

Thus, today there is a need to involve engineers-technologists, who are the most competent experts in the field of automated technological process, in the process of development of logical control programs. An urgent task is to create and explore new technological linguistic means and unconventional approaches to programming logical control problems.

2. Literature review and problem statement

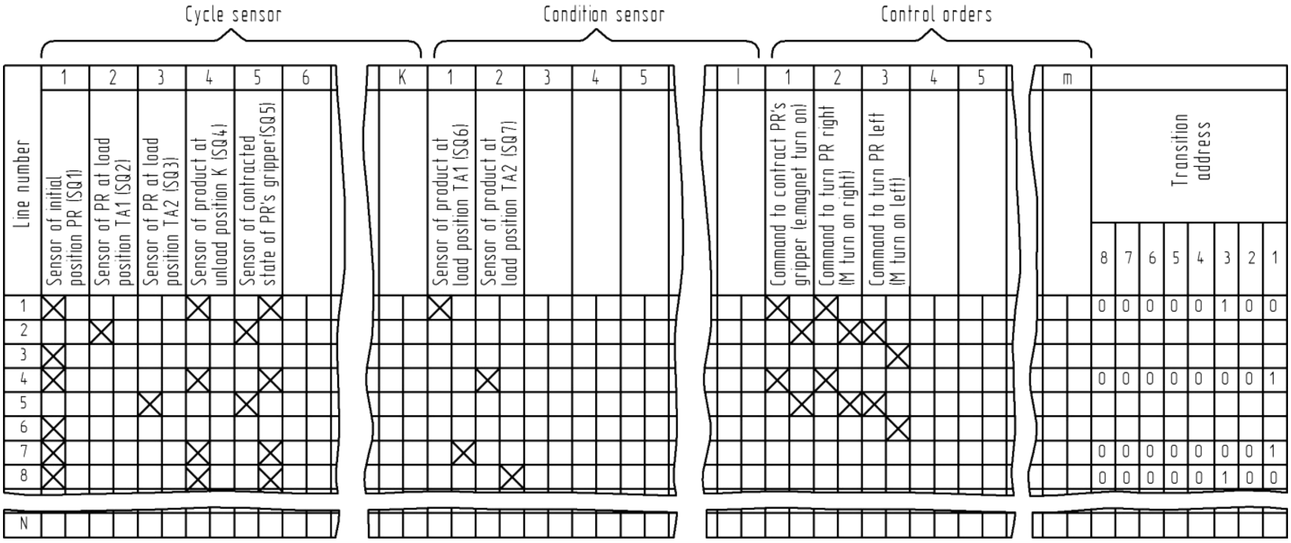
While developing programs, professional programmers in most cases use particular approaches to writing a code that depend on subjective experience and preferences of a particular developer. It should be noted that, given the specific subject area, development and program implementation of algorithms of logic control over technological sites is a separate trend of modern programming [4]. Development of logical control systems implies a problem-oriented approach with close involvement of engineers-technologists, who are experts in the field the automated process, at all stages of life cycle of the created software [5].

Today, the most common linguistic means of creation of control systems software is the set of languages of IEC 61131-3 [6]. Each of the five languages of the standard focuses on one specific area of problems of logical control. When describing the complex control algorithms, programmers are forced to use several languages of IEC 61131-3 standard in one project. In such a program, a part of the algorithm is described in the same programming language, and the rest of the programs are described in another programming language. Many experts point out obvious shortcomings of using the languages of the standard, “closeness” for understanding and using these languages for engineers-technologists [8]. That is why there appear different works that describe PLC programming languages that are alternative to the standard. The ideas of using a universal visual language of UML modeling to create logical control programs are described [9]. Another trend is adaptation of universal text programming languages to problems of logical control [10]. However, the use of universal programming languages and modeling is accessible only for professional programmers. It is offered to use the method of automatic generation of the program code in standard PLC programming languages for mathematical models, such as from the mathematical package MathCAD [11]. In addition, a variety of specialized modeling languages for solving a certain class of automation tasks are proposed [12].

However, a programming error in the logic of control over a complex technological site can lead not only to serious material losses, but also create an emergency situation threatening health and lives of people.

Since the advent of the first industrial controllers, various experts pointed out the need to create visual problem-oriented languages. These languages should be simple and should be able to describe the algorithm of logic control in terms of technological process alone [13]. Such an approach will allow engineers-technologists themselves to create and edit control programs. For example, at the end of the 1980-s, within the framework of the project “Buran”, development of graphic programming language DRAGON (Friendly Russian Algorithmic language that will Provide Visibility) was being carried out. The main task of developers was to create a single universal programming language that due to its visibility and availability was to replace specialized languages of the project. “DRAGON” language designers created it as a universally accessible visual language that was intended to describe the structure of human activity and programming in particular. This language was also designed to visualize and formalize knowledge for designing and modeling of systems and to be use in the learning process, etc. The main requirement for the visual language “DRAGON” was to improve comprehension of graphical representation of algorithms (dragon-schemes) for human visual perception [14]. All other requirements were considered less important, though they did not lose their relevance.

In the 90-s, in Kharkov Institute of VSRITElectromach, there were developed and put into production at many enterprises of the former USSR the logical control automatic machines of parallel action (LCAPA); while using them for formalization of control algorithms, it was proposed to apply tabular technology cyclograms (Fig. 1).



	Cycle sensor														Condition sensor												Control orders																			
Line number	1		2		3		4		5		6		K		1		2		3		4		5		l		1		2		3		4		5		m		Transition address							
	Sensor of initial position PR (SQ1)		Sensor of PR at load position TA1 (SQ2)		Sensor of PR at load position TA2 (SQ3)		Sensor of product at unload position K (SQ4)		Sensor of contracted state of PR's gripper (SQ5)						Sensor of product at load position TA1 (SQ6)		Sensor of product at load position TA2 (SQ7)										Command to contract PR's gripper (e.magnet turn on)		Command to turn PR right (M turn on right)		Command to turn PR left (M turn on left)								8	7	6	5	4	3	2	1
1	☒						☒			☒					☒												☒		☒										0	0	0	0	0	1	0	0
2			☒						☒																			☒		☒	☒															
3	☒																															☒														
4	☒						☒			☒							☒										☒		☒										0	0	0	0	0	0	0	1
5					☒				☒																			☒		☒	☒															
6	☒																															☒														
7	☒						☒			☒						☒																							0	0	0	0	0	0	0	1
8	☒						☒			☒								☒																					0	0	0	0	0	1	0	0
N																																														

Fig. 1. Example of control algorithm in the form of a table of technological cyclogram

This language was called "language of parallel logic controllers" (LPLC) [15]. Description of the control algorithm in the form of tables of a cyclogram was simple, visual and readable for engineers-technologists. Thus, engineers-technologists created a control program in this tabular language, checked the control algorithm and eliminated logic errors in the program. When creating a tabular language LPLC, the main requirement was also comprehensibility of tabular representation of the control algorithm in the form of a cyclogram of visual perception by an engineer-technologist [16].

At present, to attract specialists from different subject areas to solving applied tasks, the approach, based on creation and use of problem-oriented languages in the process of program designing, is being developed [17]. Lots of problem-oriented languages for creation of programs in the field of management of databases, finance, mathematical packages and other problems are currently used in practice. To attract engineers-technologists to development of control systems based of the standard PLC, it is also proposed to use a variety of subject-oriented languages for description of technological algorithms [18].

Thus, it would be appropriate in the context of the problem-oriented approach to use the tabular language of technological cyclograms for various microelectronic means of logic control as a language of formalization of the control algorithm. A technological cyclogram is visual, its concepts and terminology comply with automated technological process, due to which it can be used by different professionals at different stages of the life cycle of a logical control system. However, for implementation of possibility of programming different industrial controllers directly by a tabular cyclogram, it is essential to explore the requirements for specialized instrumental environment of visual programming and for the method of integration with existing PLC programming packages in standard programming languages.

3. The aim and objectives of the study

The main aim of preset research is to improve the quality of logical control programs for different industrial controllers by using the problem-based tabular language of technological cyclograms as a programming language.

To accomplish the set goal, it is necessary to solve the following problems:

– to develop the problem-oriented language of programming of means of logic control and technology and its applications;

– to develop requirements for environment of programming of logic controllers with the use of the proposed problem-based technological language;

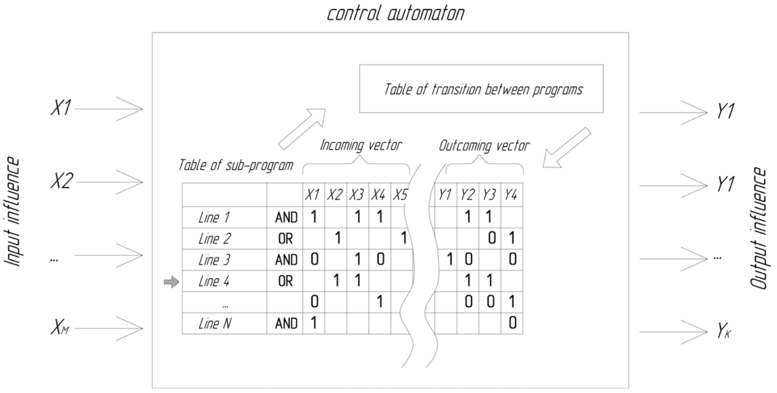
– to explore the proposed means of programming in tasks of creation of programs of logical control of technological equipment.

model in the problem-oriented language, which specializes in a specific subject area and does not depend on specific software and hardware. Then the debugged control model is automatically converted into a real code in a standard language.

With development of microprocessor technology and applied information technologies as a programmer of modern PLC, the use of a personal computer started.

Today, manufacturers of microprocessor PLC offer for their controllers either branded SoftLogic – programming systems or integration of their controllers with third-party SoftLogic systems. In these instrumental systems, a developer can create a control program in the languages of IEC 61131-3 standards, compile a project into a machine code and directly download the compiled code in an industrial controller. Thus, it is advisable to use standard PLC programming languages as implementation languages. Automatically generated by the problem-oriented model, a control program in the language of the standard can be compiled and uploaded in a specific PLC model with the help of the existing SoftLogic-system.

The proposed technological visual programming was implemented within the rules and concepts of problem-oriented programming, involving creation of software abstractions of the solved problem, which are called domain models [19, 20]. Even in the middle of the last century, various experts indicated visibility and convenience of describing complex situations in the tabular form while analyzing a solved problem, and the possible use of tables of solutions directly as a programming language [21–23]. Thus, it is appropriate to use the improved tabular language of cyclograms as a problem-oriented language. Technological cyclogram meets the necessary requirements of clarity, simplicity, accessibility for understanding of engineering personnel. An engineer-technologist will be able to formalize the whole algorithm of logical control directly in the proposed tabular language. The basic concept of programming with the help of the cyclogram language is the "state" of the controlling machine. The reaction of the controlling machine to the input influence will depend on which line of a cyclogram control is (Fig. 2).



		X1	X2	X3	X4	X5	Y1	Y2	Y3	Y4
Line 1	AND	1		1	1			1	1	
Line 2	OR		1			1			0	1
Line 3	AND	0		1	0		1	0		0
⇨ Line 4	OR		1	1				1	1	
...		0			1			0	0	1
Line N	AND	1								0

Fig. 2. Formalization of logic of controlling machine in the form of tables of cyclogram language

4. Development of technological visual programming

4. 1. Creation of a problem-oriented language based on tabular cyclograms

Software development within the problem-oriented approach at the first stage implies description of the control

When constructing a cyclogram, a developer operates input, output and internal variables. Input and output variables are Boolean vectors. Input variables are state of inputs of the controller, i. e. states of the controlled object and the environment. Output variables are states of outputs of the controller, that is the generated control commands. The program consists of the descriptive and the operator parts. The descriptive part includes the link of the program elements to the control object. The operator part includes a tabular record of a cyclogram of operation of technological aggregate. The operator part is a sequence of lines, each of which is a logic operator.

4. 2. Technology of creation of control programs in the language of tabular cyclograms

Technological visual programming technology in practice is implemented by:

1) creation of a visual object-oriented language of formalization of the logical control algorithm (for example, different variations of the tabular language of technological cyclograms);

2) development of the specialized programming environment for the problem-oriented language with powerful visual capabilities (tabular cyclogram editor, etc.);

3) development of the program analyzer in the created problem-oriented language;

4) development of the rules of translation (program code generation) of the tabular cyclogram into program code in the standard language of the selected hardware platform.

Within the framework of the proposed TVP-programming, the tabular language of technological cyclograms, which allows visual description of control logic in terms of technological process, was used as the language of formalization of the control algorithm. The program is created as a set of tables in a specialized table editor. A set of software elements and structures of the tabular cyclogram language can be different. But it is necessary to develop the rules for analysis of a cyclogram and translation into a program code in a standard programming language for each variant of the tabular cyclogram language.

For implementation of the proposed approach to programming of means of logic control in practice, it is necessary to create a specialized programming environment, which should include the following subsystems:

– a subsystem of input of project variables (input, output, internal variables, counters, timers, etc.), tasks from the list of subprograms of a cyclograms depending on the implemented algorithm;

– a cyclogram table editor that makes it possible, operating a list of project variables, to describe control logic in the form of lines of cyclogram tables;

– an error-control subsystem of control program in the form of cyclogram tables;

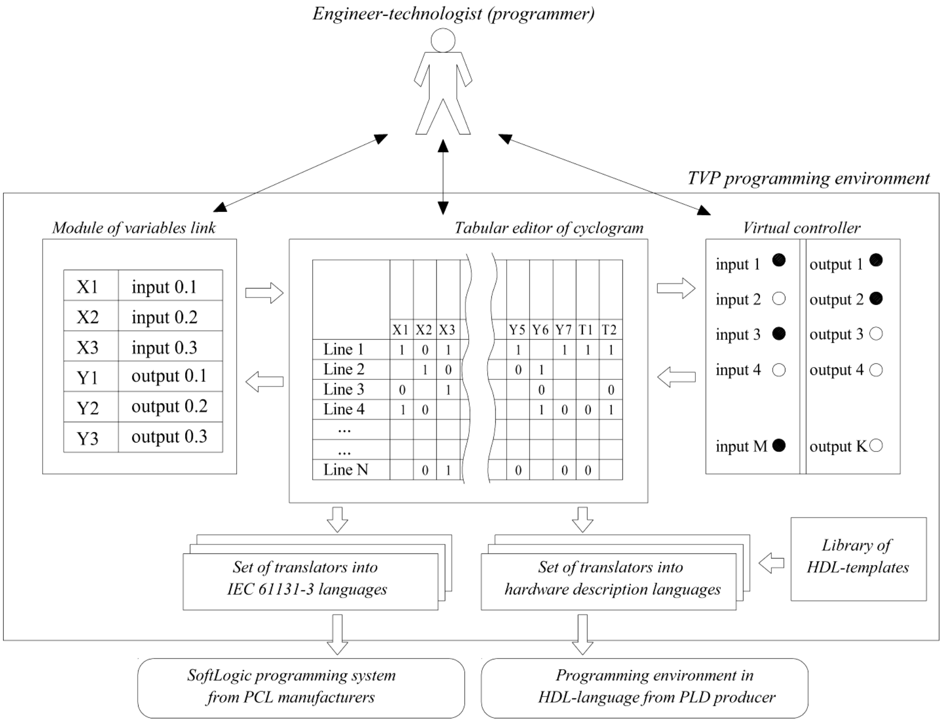
– a virtual logical controller that allows emulation of operation of the controlling machine depending on cyclogram tables (which allows checking in the emulation mode what output impacts will be formed depending on the test sequence of incoming vectors);

– a set of specialized translators into standard IEC 61131-3 languages, which make it possible to automatically transform control logic in the form of tables of the debugged cyclogram into a program code into standard PLC language;

– a specialized translator, which generates description of a controlling digital device in the hardware description language (for example, full HDL-description of the logical control automatic machine of parallel action) by cyclogram tables and the selected HDL-pattern.

4. 3. Creation of a specialized programming system in language of tabular cyclograms

With the proposed approach, a personal computer with the developed programming environment in the language of technological cyclograms immediately becomes a device of programming of microelectronic means of logic control. In technological visual programming, an engineer-technologist himself describes the logic control algorithm in a tabular form. Programming of discrete systems of logic control is formalization of a sequence of logical equations. When creating control programs, a developer operates logical variables that are connected with discrete inputs-outputs of the controller. Programming in the tabular cyclogram implies that for each of the expected set of input and internal variables, control command is formalized as a set of states of output discrete variables. Thus, a mandatory part of programming TVP-environment is the module of link of cyclogram variables to specific physical inputs-outputs of the industrial controller (Fig. 3). Operating given variables of the project, a developer describes control logic in the table editor of TVP-environment in the form of cyclogram lines. To have a possibility to verify control logic without uploading the program to the hardware controller, TVP-environment must contain module "Virtual controller". This module should make it possible to perform a cyclogram in the emulation mode in TVP-environment, that is, to offer the possibility to generate a sequence of input vectors and verify implemented control.

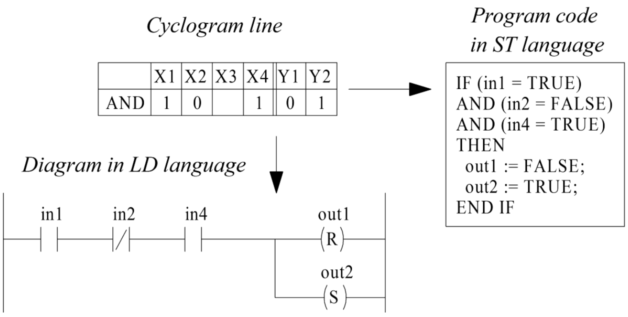

Engineer-technologist (programmer)
TVP programming environment
Module of variables link
Tabular editor of cyclogram
Virtual controller
input 1 ●
input 2 ○
input 3 ●
input 4 ○
input M ●
output 1 ●
output 2 ●
output 3 ○
output 4 ○
output K ○
Set of translators into IEC 61131-3 languages
Set of translators into hardware description languages
Library of HDL-templates
SoftLogic programming system from PCL manufacturers
Programming environment in HDL-language from PLD producer

X1	input 0.1
X2	input 0.2
X3	input 0.3
Y1	output 0.1
Y2	output 0.2
Y3	output 0.3

	X1	X2	X3	Y5	Y6	Y7	T1	T2
Line 1	1	0	1	1		1	1	1
Line 2		1	0	0	1			
Line 3	0		1		0			0
Line 4	1	0			1	0	0	1
...								
...								
Line N		0	1	0		0	0	

Fig. 3. Structure of TVP-environment of programming of industrial logic controllers

Created and debugged control program in the form of a cyclogram is automatically translated into the program code in the language of implementation for the selected hardware of logic control. Since PLC is a standard microprocessor device with the system software, built-in by a manufacturer, application control program in IEC 61131-3 language contains only control logic. That is why a completely applied control program in one of five languages can be obtained by a debugged cyclogram. For this, TVP-environment should be composed of a set of translators of a cyclogram into a program code for each language of IEC 61131-3 standard. Translation includes an analysis of the variables of a project and all cyclogram tables and subsequent synthesis – generation of a complete code of the equivalent program in the selected language of implementation. During cyclogram processing, a translator must find out whether the generated cyclogram is incorrect by any rule. If the generated cyclogram is incorrect, a detailed message is given to a developer. If a cyclogram is correct, it is necessary to generate by translation rules the original code in the implementation language on all the lines in the cyclogram tables. For example, ST language is a standardized text language that is similar to the Pascal language. A logical equation in this language is a conditional operator “IF”, which lists the logical variables involved in this equation. If the condition of the operator “IF” becomes equal to “TRUE”, a set of operators of section “THEN” is fulfilled. Thus, one logical operator of the program in ST language corresponds to one line of the cyclogram table. If a variable is used in the left side of a cyclogram’s line (expected event), the check of this variable gets into the condition of operator “IF”. If a variable is used in the right side of the cyclogram’s line (generated control order), a change in the value of this variable gets in section “THEN” of the same conditional operator in ST language (Fig. 4).



	X1	X2	X3	X4	Y1	Y2
AND	1	0		1	0	1

```
IF (in1 = TRUE)
AND (in2 = FALSE)
AND (in4 = TRUE)
THEN
  out1 := FALSE;
  out2 := TRUE;
END IF
```

Fig. 4. Example of translation of a cyclogram line into a program code in ST text language and LD graphic language

A program in a standardized graphical language of ladder LD diagrams visually resembles an electric circuit with closed and open contacts. Presence or absence of current corresponds to the result of a logical operation (true – there is current; false – there is no current). On the left and on the right, the circuit is limited by vertical lines — power buses. Between them, there are chains, formed by contacts and relay coils, similar to conventional electronic circuits. On the left, any chain begins with a set of contacts that send left-to-right “ON” or “OFF” states, corresponding to logical values “TRUE” or “FALSE”. A logical variable corresponds to each contact. Sequential connection of contacts forms a logical operation “AND”, parallel connection of contacts forms a logical operation “OR”. In the right part of the circuit, there can be any number of coils (relay), which can be assembled only in parallel. Thus, one circuit in the graphic diagram in LD language corresponds to one line of the cyclogram table. If a variable is used on the left side of a cyclogram’s line (expected event), a correspondent contact, connected to the variable, appears on the left side of LD chain. If a variable is used on the right side of the cyclogram’s line (generated control order), a correspondent coil appears on the right side.

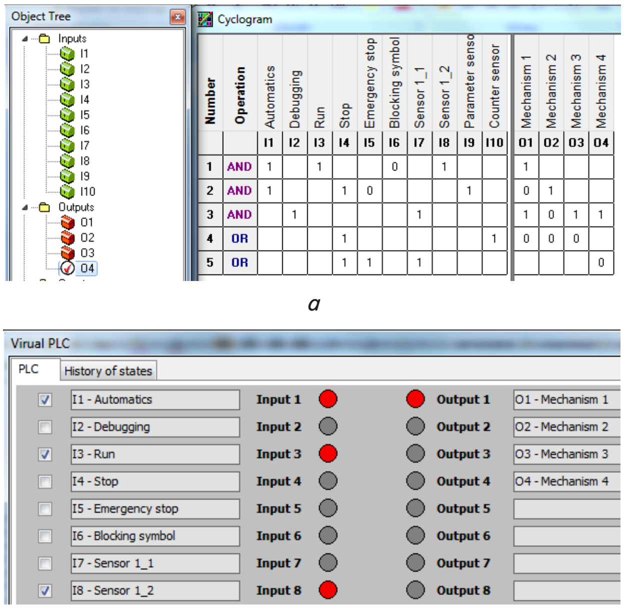
If the industrial PLD-controller was selected as hardware means of logic control, a developer before translation must select an accessible template of a digital device from a library of TVP-system. HDL-template contains complete description of typical architecture of a digital device (for example, description of a logical controlling machine of parallel action of increased reliability). Specialized translator of TVP-environment, relying on control logic, implemented in a cyclogram, by the selected template generates a complete text in the language of description of hardware of a digital control device with the selected architecture. Thus, when adding a new HDL-template to the library of TVP-environment, it is necessary to implement the rules of embedding into this template of logic control, based on the technological cyclogram.

Automatically generated code without modifications is loaded by a developer of the standard programming environment of the selected hardware, from where the connected industrial controller is programmed.

One of the first attempts of the authors to implement the system of programming PLD-controller using the language of technological cyclograms was a simple environment of programming in LPLC-M language. This system made it possible to generate a program code in AHDL language of hardware description, describing the machine of parallel action for controlling the railway crossing, using the cyclogram table for automatic system of controlling railway crossings. Then, the editor of simple one-table cyclograms and the translator of these cyclograms into the program code in corporative STL language of industrial controllers of Festo company was created. Within the framework of this work, we studied requirements for the universal TVP programming environment, which will make it possible in the tabular problem-oriented language Cyclograf, operating variables of a project, to describe different control algorithms, in the stepwise emulation mode to debug control logic in TVP-environment itself, and to work out sequentially a set of input vectors in emulation mode. After creating and debugging control logic in TVP programming environment, a developer selects a specific model of the industrial controller (microprocessor PLC, industrial PLD-controller) and a standard implementation language, available for this model. A set of specialized translators of TVP-environment automatically generates a complete program code by the created cyclogram (and by a selected template of a digital device). To implement the proposed approach in practice, the specialized table programming environment and the sub-system of modeling of control algorithm “Virtual controller” were created (Fig. 5).

The translator into the program code in ST language of the international IEC 61131-3 standard was created as a part of the developed programming environment.

A key challenge that was solved when creating TVP-technology was development of effective visual programming language, oriented to a user, not having software development skills. When creating this programming method, the authors set for themselves minimum and maximum tasks. To solve the minimum task, it was necessary to provide an accessible level of programming code visibility for an engineer-technologist who has no professional skills in the field of informatics and programming languages. To solve the maximum problem, it is necessary to exclude a programmer from the technological process of preparation of controlling programs altogether and pass over the software development directly to an engineer-technologist, providing him with respective tools.



Object Tree

Inputs
I1
I2
I3
I4
I5
I6
I7
I8
I9
I10
Outputs
O1
O2
O3
O4

Cyclogram

Number	Operation	Automatics	Debugging	Run	Stop	Emergency stop	Blocking symbol	Sensor 1_1	Sensor 1_2	Parameter senso	Counter sensor	Mechanism 1	Mechanism 2	Mechanism 3	Mechanism 4
		I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	O1	O2	O3	O4
1	AND	1		1			0		1			1			
2	AND	1			1	0				1		0	1		
3	AND		1					1				1	0	1	1
4	OR				1						1	0	0	0	
5	OR				1	1		1							0

a

Virual PLC

PLC | History of states

☑	I1 - Automatics	Input 1	●	●	Output 1	O1 - Mechanism 1
□	I2 - Debugging	Input 2	○	○	Output 2	O2 - Mechanism 2
☑	I3 - Run	Input 3	●	○	Output 3	O3 - Mechanism 3
□	I4 - Stop	Input 4	○	○	Output 4	O4 - Mechanism 4
□	I5 - Emergency stop	Input 5	○	○	Output 5	
□	I6 - Blocking symbol	Input 6	○	○	Output 6	
□	I7 - Sensor 1_1	Input 7	○	○	Output 7	
☑	I8 - Sensor 1_2	Input 8	●	○	Output 8	

b

Fig. 5. Environment for programming logic control in the language of cyclograms: *a* — table editor of subprograms of a cyclogram; *b* — module "virtual controller"

5. Practical example of application of technological visual programming technology

Let us consider an example of creating a program of logic control in the form of a technological cyclogram that implements the algorithm of control over an industrial robot, which serves two sections of the technological line (Fig. 6). The robot under consideration is portal and moves around the rectangular coordinate system: the first direction of its motion is along the axis of the portal (right/left), the second direction of its motion is perpendicular to the axis of the portal (lowering/lifting of the gripper).

On the first and the second sections of the production line, at the first level of the hierarchy, control of separate technological units (TU) is performed using local control systems (LCS). Each technological unit has load position "L" and unload position "U", at which existence of workpieces is controlled by position sensors SQ13-SQ20. An industrial robot performs transferring of workpieces from the unload position of the first or the second section to the load positions of the first or the second section. The state of the gripper is controlled by sensors SQ1 (gripper is released) and SQ2 (gripper is contracted). Vertical position of the gripper is controlled by SQ3 sensors (gripper is in the upper position) and SQ4 (gripper is in the lower position). Horizontal position of the gripper on the portal is controlled by sensors SQ5 (gripper is in load position TU1), SQ6 (gripper is in unload position TU1) and similarly by SQ7–SQ12.

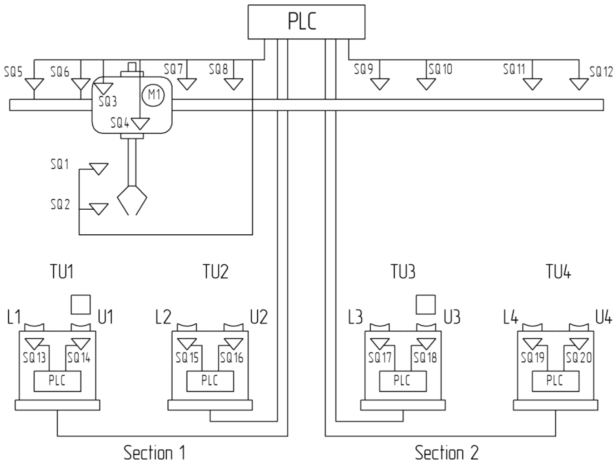


Fig. 6. Functional diagram of the system of production line control

Algorithm, describing control of the robot for transferring the workpiece from unload position TU1 to load position TU3, is presented in Fig. 7.

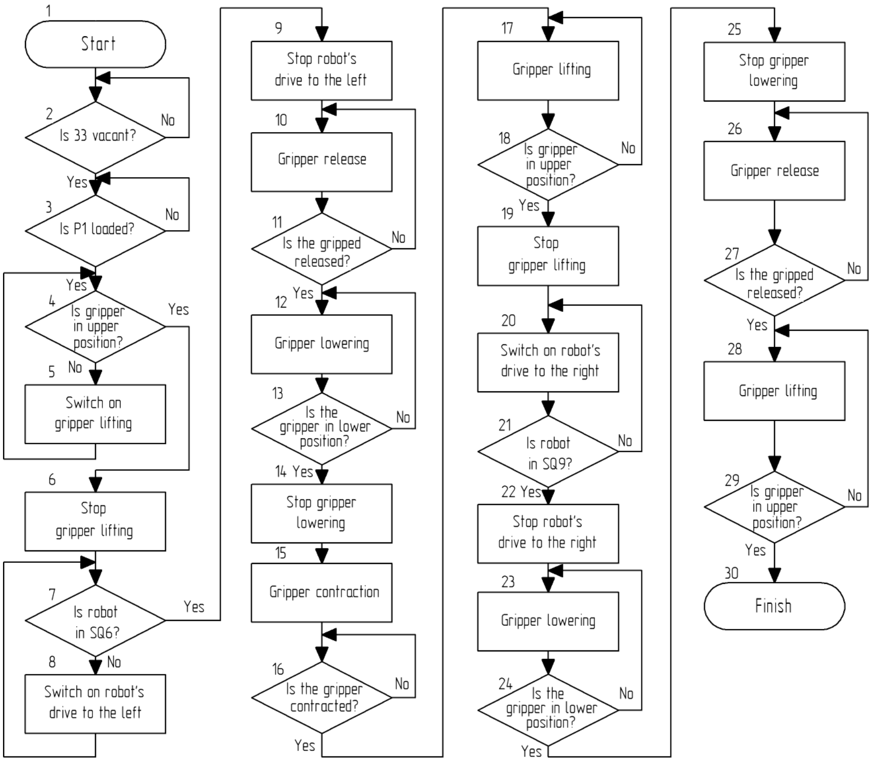


Fig. 7. Block diagram of the algorithm of program control of industrial robot's motion for workpiece transfer

The algorithm of workpiece transfer by an industrial robot is a strict sequence of actions. First, it is checked if the required load position is free and if there is a workpiece in the unload position. Then the gripper is lifted in the upper position and moved to the required unload position. After moving to the desired position, the gripper is released, lowered and contracted. Thus, the transferred workpiece is grasped by the gripper. Then the gripper is lifted and moved to the required load position. After being moved to the desired position, the gripper is lowered and released, that is the workpiece has been transferred to the load position. After that, the gripper is lifted.

Engineers-technologists as the most qualified experts in the field of automated industrial facilities are involved in the process of development of the control algorithm. With the traditional approach to creation of control software, the created technological algorithm is passed over to a programmer, who, based on his knowledge in the field of programming languages and his personal preferences, creates a program code in the selected standard language. Described by a professional programmer, the algorithm of control of the motion of an industrial robot in a graphic language of relay-contact LD symbols will visually represent the electric circuit with open and closed contacts. The same control algorithm, implemented as a program code in the text programming language ST, will represent a large amount of a text code, similar to the program code in the Pascal language. Thus, the quality of the manually generated program code and correctness of implementation of the technological algorithm in it depend completely on competencies of a programmer, his complete understanding of details of the process. Created control program in a standard language is not accessible for checking and verification to an engineer-technologist as the most competent specialist in the field of the automated technological process.

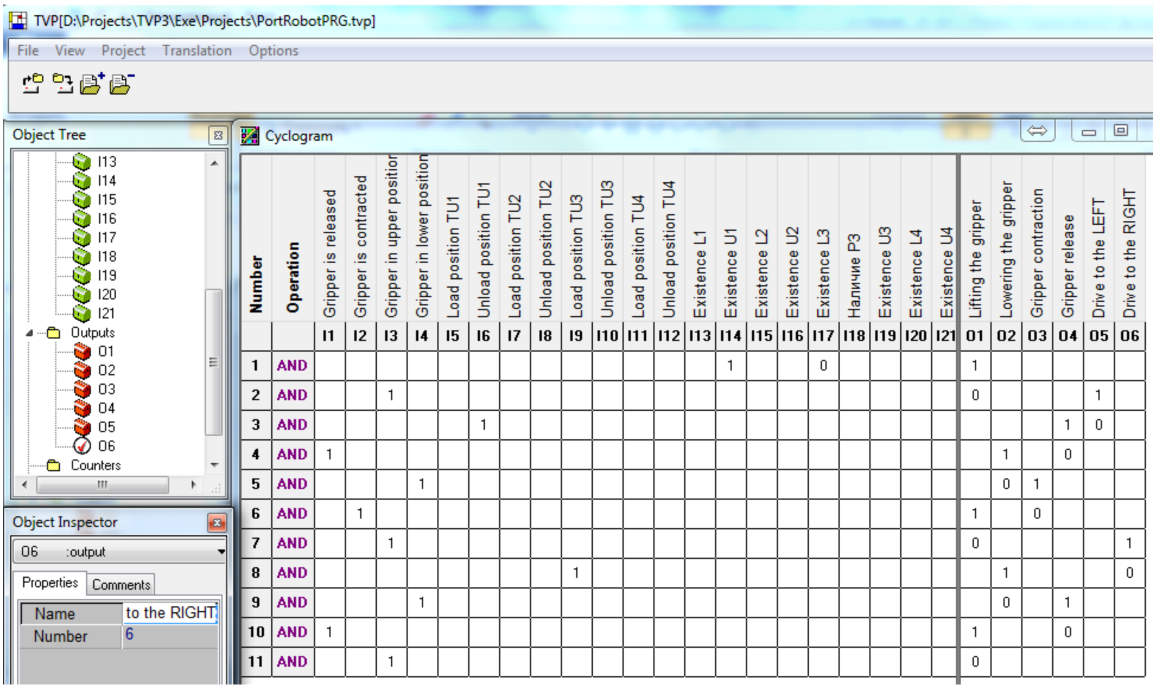
Within the proposed TVP-technology, the control algorithm is formalized in a specialized editor in the form of tables of a technological cyclogram. The above sequence of robot's actions algorithm for transferring a workpiece is represented in the cyclogram as a sequence of lines in a table, where each line is a logical equation. In the left part of the line, a user formalizes a logical condition, and in the right part – an action that is performed (Fig. 8).

The presented table cyclogram fully describes the technological control algorithm. When creating a control program in a tabular cyclogram language, a developer operates with exceptionally technological terms. A cyclogram is clear, visual and psychologically natural for engineers-technologists, which enables them to participate in development and verification of programs of logical control.

In accordance with the above technological visual programming technology, the created and debugged tabular cyclogram is automatically translated by the specialized translator into the program code in one of the languages of the standard. Without changes, the text, generated by the program is uploaded into the selected PLC model with the help of SoftLogic programs. Unlike the traditional approach, the control program in the form of tables of a cyclogram can be used by a variety of specialists for understanding the technological algorithm at the stages of launch-adjusting operations, modernization of a control system, as technical documentation, etc.

6. Discussion of results of development and research into technological visual programming of problems of logic control

Tabular language of cyclograms as a programming language for industrial controllers with parallel architecture was described in [17]. Today, however, for a wide usage of the tabular language of technological cyclograms for programming of industrial controllers in various automation tasks, it is necessary to develop a specialized programming environment and a set of specialized translators of the created cyclogram into the program code in the standard languages.



06 :output

Name: to the RIGHT
Number: 6

Number	Operation	Gripper is released	Gripper is contracted	Gripper in upper positio	Gripper in lower position	Load position TU1	Unload position TU1	Load position TU2	Unload position TU2	Load position TU3	Unload position TU3	Load position TU4	Unload position TU4	Existence L1	Existence U1	Existence L2	Existence U2	Existence L3	Наличие P3	Existence U3	Existence L4	Existence U4	Lifting the gripper	Lowering the gripper	Gripper contraction	Gripper release	Drive to the LEFT	Drive to the RIGHT
		I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14	I15	I16	I17	I18	I19	I20	I21	01	02	03	04	05	06
1	AND														1			0					1					
2	AND			1																			0				1	
3	AND						1																			1	0	
4	AND	1																						1		0		
5	AND				1																			0	1			
6	AND		1																				1		0			
7	AND			1																			0					1
8	AND									1														1				0
9	AND				1																			0		1		
10	AND	1																					1			0		
11	AND			1																			0					

Fig. 8. Technological cyclogram of motion of industrial robot for workpiece transfer

The vast majority of microprocessor PLC are programmed in languages of the IEC 61131-3 standard. That is why creation of a set of translators in five different languages of the standard within the proposed TVP-environment allows programming a large set of different PLC using the tabular language of technological cyclograms. In the context of the proposed problem-oriented approach, the tabular language of cyclograms is the language of description of the model of logic control, and standard programming languages of industrial controllers are languages of implementation of this model on the specific hardware platform of logic control. Existence of the model of emulation of implementation of the created cyclogram "Virtual controller" in the TVP-environment allows debugging control logic in TVP-environment at early stages of development when it is not yet known what hardware device will be selected as a controlling machine. The existence in TVP-environment of an extensible library of templates of digital devices of logic control and rules of template generation of the program code in languages for description of hardware on a technological cyclogram allows engineers-technologists to program industrial PLD-controller for a variety of control tasks. The proposed approach at all stages of the life cycle of the software of the automated control system allows us by the modified technological cyclogram to obtain a complete program code for specific hardware device of control, or without big labor to replace one model of the industrial controller with another (replace the PLC, which is programmable only in one language, with the PLC, which is programmed in another language of the standard, to replace microprocessor PLC with the industrial PLD-controller and vice versa). Thus, TVP programming environment allows involvement of engineers-technologists as the most knowledgeable experts in the field of automated technological process into the process of development and modernization of control programs.

However, the shortcoming of this approach is the need to develop a specialized powerful visual table editor. Each version of the language of technological cyclograms must have its table editor, unlike universal text languages, the program code of which can be edited in any text editor. It should also be noted that IEC 61131-3 standard describes only appearance and syntax of graphical languages (LD, FBD, and SFC), but does not describe the format of program files storage. Therefore, if the format of storage of the graphic programs of a specific SoftLogic-system is "closed" by its developer, the control program in a graphic language, automatically generated by TVP-environment, is not transferrable to this SoftLogic-system of PLC programming (it will be necessary to type it in).

The control program in the language of technological cyclograms is simple, visual and easy to understand and modify for engineering staff. In practice, cyclograms successfully describe some classes of automation tasks (different cyclic industrial robots, coiling machines, packaging automatic machines, railway automation subsystem, etc.). However, unlike text programming languages and multilingual programming languages (one software PLC project can include different languages of IEC 61131-3 standard), strictly formalized tabular form of a technological cyclogram is not so flexible and universal. Thus, it would be worthwhile studying what classes of industrial automation problems are possible to program using tables of technological cyclogram and what set of programming constructions of cyclograms is required for each class of problems. Related studies would offer an opportunity to develop a variety of tabular languages for different automation tasks and create a fully functional environment for programming industrial controllers with the help of technological cyclograms. An important area for further research is also identification of the zones for rational use of the language of technological cyclograms within TVP-technology in comparison with the existing traditional approach to logic control programming.

7. Conclusions

1. In the framework of the proposed technological visual programming of different industrial controllers, we developed a tabular language of technological cyclograms, which is the continuation of the programming language for logical controlling machines of parallel action. It is also proposed to use the tabular language of cyclograms as a problem-oriented language of the logical control model. The language of technological cyclograms is hardware-independent language of formalization of the control algorithm. When specifying control logic in the form of tables of a cyclogram, a user operates exclusively with the terms of the technological process and does not take into account what microelectronic control device will be used. Under the proposed TVP-programming, a complete program code in the selected standard language of implementation is automatically generated by the created and debugged implemented control model in the form of a cyclogram. By the same technological cyclogram at any stage of the lifecycle of control system software, it is possible to generate a control program both in the programming text language PLC, or in the graphic language PLC, as well as complete description of a digital device of logic control in the hardware description language by the selected HDL-template (for example, description of a logical controlling machine of parallel action of increased reliability).

2. The composition of specialized environment for programming various industrial automation tasks with the use of the language of technological cyclograms was proposed and substantiated. TVP programming environment should include a tabular editor of different variants of cyclogram tables, the module of links of inputs-outputs of the controller and editing variables of the project, a set of specialized translators into standard programming languages of IEC 61131-3, an extensible library of HDL-templates of digital logic control devices and rules of template generation of description of a digital device on a technological cyclogram, the "Virtual controller" module for emulation of the control model, formalized in a tabular cyclogram, in TVP implementation environment. The control program in the form of the tables of technological cyclogram can be widely used at stages of launch-adjusting works, maintenance and modernization of the control systems, and as technical documentation.

3. The rules of algorithmically equivalent transformation of a technological cyclogram into a program code in the standard implementation languages were developed. In the created information environment of TVP-programming with the use of a practical example of programming of a pick-and-place robot, the possibility to formalize the control logic in the form of a language of technological cyclograms was shown.

References

1. IEC 61131-3. Programmable controllers. Part 3: Programming languages [Text]. – 2nd ed. – Int. Electrotechnic Commission, 1998.
2. Zyubin, V. E. Programmirovanie PLK: yazyki MEK 61131-3 i vozmozhnye al'ternativy [Text] / V. E. Zyubin // Promyshlennye ASU i kontrollery. – 2005. – Issue 11. – P. 31–35.
3. Shalyto, A. A. Switch-tekhnologiya. Algoritmizatsiya i programmirovanie zadach logicheskogo upravleniya [Text] / A. A. Shalyto. – Sankt-Peterburg: Nauka, 1998. – 628 p.
4. John, K. H. IEC 61131-3: Programming Industrial Automation Systems [Text] / K. H. John, M. Tiegelkamp // Concepts and Programming Languages, Requirement for Programming Systems, Decision-Making Tools. – Springer-Verlag Berlin Heidelberg, 2001. doi: 10.1007/978-3-642-12015-2
5. Sinichkin, S. G. Programmiruemye kontrollery i ih primenenie dlya modernizatsii sistem upravleniya tekhnologicheskim oborudovaniem [Text]: ucheb. pos. / S. G. Sinichkin; NGTU im. R.E. Alekseeva. – Nizhniy Novgorod, 2008. – 211 p.
6. Demenkov, N. P. Yazyki programmirovaniya promyshlennyh kontrollerov [Text]: ucheb. pos. / N. P. Demenkov. – Moscow: Izd-vo MGTU im. N. E. Baumana, 2004. – 172 p.
7. Parr, E. Programmiruemye kontrolery: rukovodstvo dlya inzhenera [Text] / E. Parr. – Moscow: BINOM. Laboratoriya znaniy, 2007. – 516 p.
8. Tatarchevskiy, V. A. Problemy primeneniya yazykov standarta IEC 61131-3 i vozmozhnye puti resheniya [Text] / V. A. Tatarchevskiy // Informatsionno- matematicheskie tekhnologii v ekonomike, tekhnike i obrazovanii. – Ekaterinburg: UGTU-UPI, 2007. – P. 239–241.
9. Dubinin, V. Engineering of Validatable Automation Systems Based on an Extension of UML Combined With Function Blocks of IEC 61499 [Text] / V. Dubinin, V. Vyatkin, T. Pfeiffer // Proceedings of the 2005 IEEE International Conference on Robotics and Automation. – 2005. doi: 10.1109/robot.2005.1570732
10. Zyubin, V. E. «Si s protsessami»: yazyk programmirovaniya logicheskih kontrollerov [Text] / V. E. Zyubin // Mekhatronika. – 2006. – Issue 12. – P. 31–35.
11. Yang, C. (John) CAutomated Model Transformation between MATLAB Simulink/Stateflow and IEC 61499 Function Blocks [Text] / C. (John) Yang, V. Vyatkin // IFAC Proceedings Volumes. – 2009. – Vol. 42, Issue 4. – P. 205–210. doi: 10.3182/20090603-3-ru-2001.0302
12. Shah, A. A. Multi-view Modeling to Support Embedded Systems Engineering in SysML [Text] / A. A. Shah, A. A. Kerzhner, D. Schaefer, C. J. J. Paredis // Lecture Notes in Computer Science. – 2010. – P. 580–601. doi: 10.1007/978-3-642-17322-6_25
13. Allashev, O. Yu. Prohramuvannia lohichnykh kontroleriv z vykorystanniam tekhnolohichnykh mov korystuvachiv [Text] / O. Yu. Allashev, I. O. Furman // Enerhetyka ta kompiuterno-intehrovani tekhnolohiy v APK. – 2015. – Issue 1 (3). – P. 76–77.
14. Baltrushaytis, V. V. "GRAFIT-FLOKS" – tekhnologiya razrabotki programmnogo obespecheniya bortovyh vychislitel'nyh mashin [Text] / V. V. Baltrushaytis // Sistemy i kompleksy avtomaticheskogo upravleniya v kosmonavtike i narodnom hozyaystva. – 1998. – P. 79–81.
15. Furman, I. A. Kontrollery i protsessory s parallel'noy arhitekturoy [Text] / I. A. Furman, V. A. Krasnobaev, M. A. Malinovskiy, S. V. Panchenko. – Kharkiv: UkrGAZHT, 2006. – 416 p.
16. Malinovskiy, M. L. Kontseptsiya sozdaniya tablichnyh yazykov opisaniya apparatury [Text] / M. L. Malinovskiy, I. A. Furman, A. P. Konishcheva, A. Yu. Allashev, A. V. Svyatobat'ko // Radioelektronni i kompiuterni systemy. – 2010. – Issue 6. – P. 289–291.
17. Spinellis, D. Notable design patterns for domain-specific languages [Text] / D. Spinellis // Journal of Systems and Software. – 2001. – Vol. 56, Issue 1. – P. 91–99. doi: 10.1016/s0164-1212(00)00089-3
18. Wenger, M. Model-driven engineering of networked industrial automation systems [Text] / M. Wenger, A. Zoitl, R. Froschauer, M. Rooker, G. Ebenhofer, T. Strasser // 2010 8th IEEE International Conference on Industrial Informatics. – 2010. doi: 10.1109/indin.2010.5549621
19. Parr, T. Language Implementation Patterns. Create Your Own Domain-Specific and General Programming Languages [Text] / T. Parr. – Pragmatic Bookshelf, 2010. – 380 p.
20. Evans, E. Domain-Driven Design: Tackling the Complexity in the Heart of Software [Text] / E. Evans. – Boston, MA: Addison-Wesley, 2004. – 560 p.
21. Pollack, S. L. A decision table approach to system analysis [Text] / S. L. Pollack, H. Hicks, W. Harrison // ACM SIGMIS Database. – 1970. – Vol. 2, Issue 1. – P. 6–12. doi: 10.1145/1040700.1040701
22. Engelien, M. Rekursive Projektierung von Algorithmen und Entscheidungstabellentechnik [Text] / M. Engelien // Wissenschaftliche Zeitschrift der TU Dresden. – 1980. – Vol. 29, Issue 3/4. – P. 873–880.
23. Hambi, E. Programmirovanie tablits resheniy [Text] / E. Hambi. – Moscow: Mir, 1976. – 86 p.