Simple Graphic Language Developed For Systems Serving Housing and Communal Services

Arkady Kluchev[1][0000−0002−3892−8424], Alexey Platunov[1][0000−0003−3003−3949], Vladislav Zhovnitsky[1][0000−0001−7673−3532], Vladislav Kluchev[1][0000−0002−3052−2200], Yaroslav Gorbachev[1][0000−0001−5419−6422]
kluchev@gmail.com
aeplatunov@gmail.com
vzhovnitsky@gmail.com
vakluchev@gmail.com
yaroslav-go@yandex.ru

[1] ITMO University, 49 Kronverksky Pr., St. Petersburg, 197101, Russia

Abstract. As a means to solve a number of functional tasks of automation of housing systems, a special problem-oriented programming language based on functional block diagrams (FDB) is proposed. The proposed language will improve the quality of the system by offering: -Reduction of entry threshold; -Limiting user capabilities to increase security (protection against incompetence), providing an interface for working with the system, and not an API for programming the system; - Simplification of the perception of logic due to the simple logic of the low level of abstraction and the number of operations "if-¿ then"; - Extending the boundaries of functionality due to the flexibility of adding algorithms.

Keywords: Graphic programming languages · Cyberphysical systems · Housing and utilities · Functional illiteracy · [1]

Currently, there is the problem of introducing computer technology in various areas of our lives. On the one hand, there is a generational conflict, expressed in the fact that modern technologies are not always intuitive and accessible to people of middle and older ages, and on the other hand, there is a problem with programming among young specialists - the willingness to work with high-level APIs without diving into low-level issues. This problem is relevant, in particular, in the automation of housing and communal services in areas remote from large cities, where due to natural causes a large number of people are washed out who are able to master and maintain systems with a high entry threshold. We have practical experience in implementing management systems for the housing and communal complex (Housing and Communal Services) of various cities.

The main functions of such systems are: monitoring obsolete power substations, controlling outdoor lighting, collecting data on the consumption of various energy carriers. The beginning of work - 2003. The work was carried out by LMT LLC with the active participation of ITMO employees. At the moment, systems for housing and communal services are complex scalable cyber-physical systems that allow solving problems both in large cities and in small cottage villages.

From our experience, the following were revealed:

- Management systems for housing and communal services are complex cyber-physical systems that require highly qualified personnel during deployment and operation;
- AOLCS (Automated Outdoor Lighting Control System) "Luch" deployed in about 30 cities of Russia and only two of them found such specialists.

The most painful places are:

- System configuration (setting up controllers, setting up server components);
- Application programming;
- Low level of training of local specialists.

A big problem is functional illiteracy [1, 6], which manifests itself even among the senior staff of the units responsible for maintaining such systems. The habit of local technical specialists to act within a limited number of templates makes the introduction of modern technologies and automation systems problematic. This report discusses the creation of a special programming language that would allow, through high-level abstractions, to lower the entry threshold and help employees solve simple problems of algorithmization and automation without the need for serious retraining and advanced training courses.

What are the specifics of housing and communal services:

- There are no people with the necessary set of competencies (in fact, the competencies of a system administrator, an entry-level programmer, and a competent PC user are sufficient)
- There is no way to independently obtain the necessary knowledge;
- No higher education;
- Functional illiteracy is common;
- There is no desire to gain new knowledge (especially for local specialists aged);
- The local administration is very far from the problems of using automation systems.

Housing and communal services differs from such industries as the oil industry, finance, industry and retail in small financing.Regional small cities are experiencing a constant outflow of specialists to large financial and industrial centers. The local education system is limping. So it is necessary to abandon the

classical approaches to configuration and applied, user programming of automation systems in favor of systems with the most simple, intelligent controls that do not require the attention of specialists.

The aim of this work is to solve the problem of introducing computer technology into housing and communal services by using a tool - a language that allows you to hide the high complexity of the system from the user by providing an abstract interface.

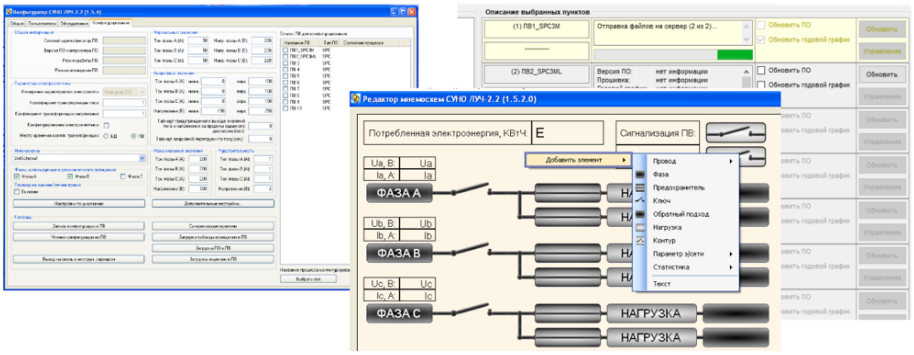

Fig. 1. Examples of Interfaces That Have Understood Understanding

At present, there are a large number of such visual programming languages that are designed to solve various problems:

- The Sequential Function Chart (SFC) is a graphical language for programming industrial logic controllers PLC and structuring programs;
- FBD is the language of Functional block diagrams like LD language of VP (visual programming), where each block represents a mathematical operation;
- And other languages that solve a variety of problems, from languages for teaching children programming [2] and about the grammar of visual web applications, to programming languages in rocket and space technology [3].

A number of articles describe the way of thinking that people use at the initial stages of their development. This way of thinking is conditioned reflex thinking, that is, to solve the problem, a person uses ready-made templates that are described by a simple formula: “if something else”, going beyond such simple recipes-algorithms, as a rule, drives a person into a dead end, absence the finished template and the need for enumeration makes it difficult or impossible to solve the problem. In order to solve problems using people with this way of thinking described above, it is necessary to exclude all complex constructions that do not fit into the behavior predictable for humans.

The language should be described by simple verbal constructions and should be accessible for explanation in terms of frequently encountered things, such as a water supply system or a motorway. Also, the language should not contain abstract objects and mechanisms that are difficult to understand that go beyond the everyday life of an ordinary person (for example: cycles, inheritance, processes, queues, interrupts, and so on).

There are examples of similar tools that are used to solve the problems of managing and controlling home automation systems: IFTTT (If This Then That), Blynk and the like. When developing such a system or programming language, it is worth remembering such an important problem as user errors that arise during the compilation of algorithms and system programming [4].

In this work, we develop a language that allows combining various elements of infrastructures and solve some problems of automation of objects, such as: turn on watering plants depending on humidity and lighting, turn on and turn off street lighting depending on precipitation and cloud cover, control thermostats of an automated object when reaching user specified geographic coordinates, etc.

The proposed language is a set of square-functional blocks representing any action and having the following parameters: two inputs, one of which is inverse, two outputs and some applied parameters. As a computation model for implementing such a programming language, a process network computation model is proposed [5], based on the interaction of processes through FIFO queues (first in, first out). The programming process consists in connecting the squares with each other, when the faces of the squares are in contact - two processes are connected, the output of one process is combined through queues (blocking queues, if there is no data, the process is blocked until the data appears) with the input of the other.

SGL Key Ideas:

- The SGL language is based on the discrete event model;
- The operators of the language are logical elements, and the connections between the elements are the faces of square elements;
- SGL does not have Turing completeness, that is, it is impossible to implement any computable function on it. Feedback for the implementation of automata cannot be done.

Language Features:

- Readability (ease of reading and understanding programs) and simplicity (language consists of several types of constructions) come first;
- Lack of cycles;
- Reliability;
- Low cost.

The programming process consists in connecting the squares with each other, when the faces of the squares are in contact - two processes are connected, the

output of one process is combined through queues (blocking queues, if there is no data, the process is blocked until the data appears) with the input of the other. SGL language constructs are function blocks. The function block is presented as a square with two inputs and two outputs.

Function Block Composition:

- Inputs (DIN, AIN, MODBUS cell, MQTT topic, etc.);
- Outputs (DOUT, AOUT, MODBUS cell, MQTT topic, etc.);
- Elementary logical elements (AND, NOT, OR-NOT, etc.);
- Filters (averager, median filter, level limiter, etc.);
- Complex functional blocks (annual schedule, control algorithm, etc.);

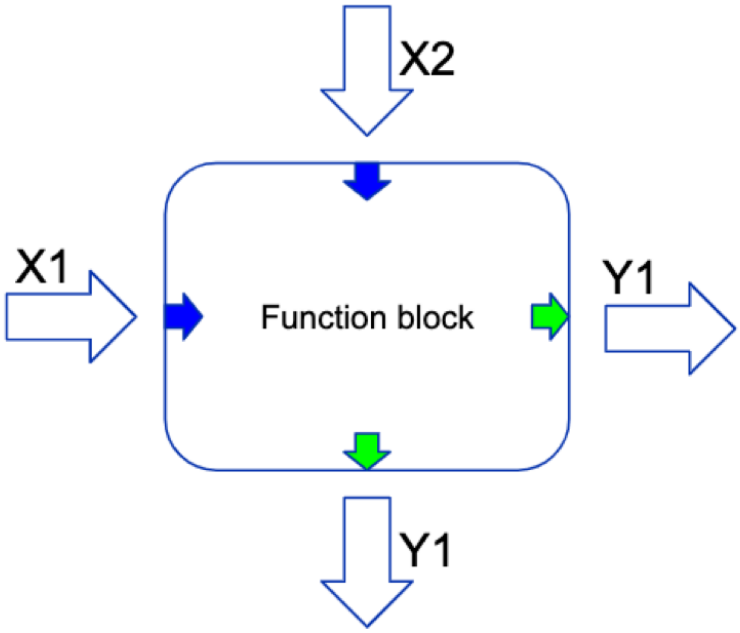

Fig. 2. Function block of SGL

As shown in the figure 2, X1, X2 - inputs, Y1, Y2 outputs Inputs X1 and X2 can have different meanings, for example, X1 - input for receiving data, X2 - settings.

Each unit is responsible for a specific process, for example, determining the user's location using geolocation or receiving data from a temperature sensor installed on the object. The list of such blocks includes: geolocation, temperature sensor, switch, triggers, inputs and outputs, message / notification generator, timers, counters, etc. The list of tasks that can be solved is mainly limited by the user's imagination, and the extension of this language by Adding a specific function block is not a high cost task.

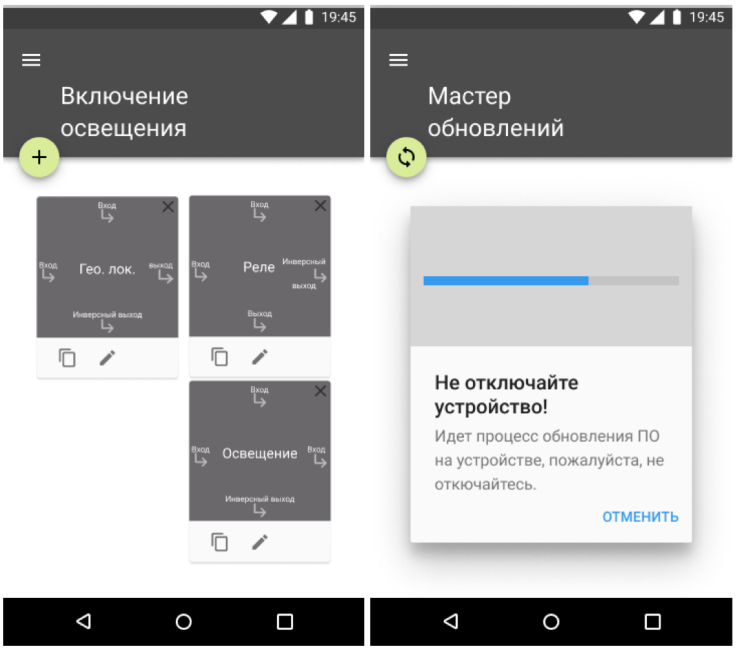

Fig. 3. Graphical configuration interface with configuration "wizard" for cyber-physical systems serving housing and communal services.

And also it is very important to notice that the most popular electronic device available to the widest segments of the population is an Android smartphone.It is logical to assume that the IDE for programming and tuning systems should also be placed in a familiar environment for the user.

An example of solving one of the problems of automation of a housing and communal services object can be heating the room before the arrival of the system user directly to the object. Such a problem is solved within the framework of the developed language by connecting three blocks. The coordinate determination unit (or constant geolocation specified in the unit settings) and the temperature

sensor transmit data to the thermostat inputs, as shown in Figure 2, which in turn transmits signals to the relay control unit at the output.

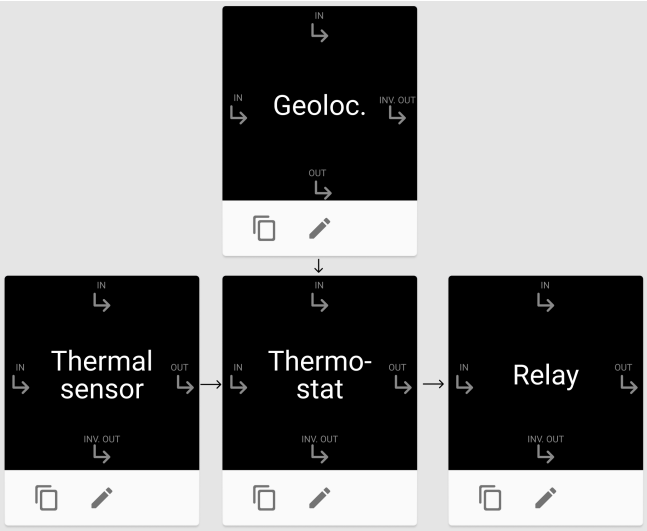

Fig. 4. Programming automation of heating the housing and communal services facility using a visual programming language

Implementation of local automation tools (if it is not an oil rig or a large plant) requires a sufficiently large investment related to the need for field technicians to travel expenses for business trips, maintenance, adjustment and maintenance, as well as separation from the work process. There are two possible solutions to this problem. Further training of field staff is one of such solutions that is not always effective and often costly. And the second option is to lower the threshold of entry and risk management, described above at the stage of development of these systems.

In conclusion on the findings:

- Outside of successful industries, it is very difficult to build a competent IT infrastructure for servicing modern systems;
- The programming tools used in conventional process control projects are too complex for existing staff;
- To automate such an area as housing and communal services, it is necessary to create as simple as possible, and preferably even completely automatic cyberphysical systems;
- SGL language allows you to reduce the entry threshold for creating simple configuration systems; it is well accepted by service personnel accustomed to working with electric networks;

The visual programming language proposed as part of the development of cyberphysical systems serving the housing and communal services allows expanding the system's functionality through user programs, simplifying user interaction with a complex engineering system and lowering the entry threshold for the system.

References

1. Kulyutkin Yu. Adult education and the problem of functional illiteracy // Man and Education. - 2008. - No. one. (in Russian)
2. Litvinov Yu. Realization of visual programming tools for robots for the study of computer science in schools // Computer tools in education. - 2013. - No. one. (in Russian)
3. Kalentiev A., Tyugashev A. Use of graphic languages :in the life cycle of on-board software for spacecraft // Bulletin of Samara State Aerospace University named after Academician S.P. Korolyov (National Research University). - 2010. - No. 2. (in Russian)
4. Palekar M., Fernandes E., Roesner F. Analysis of the Susceptibility of Smart Home Programming Interfaces to End User Error //IEEE Workshop on the Internet of Safe Things (SafeThings)(SafeThings' 19). ACM, New York, NY, USA. – 2019.
5. Stefanov T. et al. System design using Kahn process networks: the Compaan/Laura approach //Proceedings of the conference on Design, automation and test in Europe-Volume 1. – IEEE Computer Society, 2004. – C. 10340.
6. Baskakova M., Soboleva I. New facets of functional illiteracy in the digital economy // Educational issues. 2019. (in Russian)