DOI 10.34185/1562-9945-2-133-2021-08
UDC 519.71

M.O. Poliakov, C.O. Subbotin, O.M. Poliakov

INTEROPERABILITY OF INTEGRATED HIERARCHICAL SYSTEMS

Abstract: The content, properties and peculiarities of interoperability are considered in relation to integrated hierarchical systems. The structure of a unified element for the functional interaction of the levels of the hierarchical system is proposed. This element is common for interacting systems, performs different functions in these systems, and is built in the form of a finite state machine with a controlled structure. A computer program for modeling the interaction of the levels of a hierarchical system is proposed. An example is given of modeling the interaction of systems using the proposed unified element, which confirmed the possibility of interaction between systems by controlling the behavior of one automaton with the help of another.
Keywords: interoperability, integrated hierarchical system, control automata, system interaction simulator.

Introduction. The issues of modeling and integration of systems are considered in energy, mechanical engineering and other industries, which is accompanied by a variety of approaches and terminologies. In this work, a system is understood as a finite set of functional elements and relations between them, isolated from the environment in accordance with a specific goal [1]. According to [2], an integrated system is a case when there is a set of two or more interconnected systems, in which the functioning of one of them depends on the results of the functioning of the second (others).

According to the authors, the interconnectedness of systems is a sign of a supersystem consisting of these systems, and not a sign of an integrated system. A sign of the system's integration is the presence of common elements in its subsystems that perform different functions in these subsystems [3].

Another definition [4] of an integrated system: "a system in which all subsystems included in it work according to a single algorithm, i.e. has a single control point"as a defining feature it introduces a difference from an interoperable system, that is, a system" in which its subsystems operate according to independent algorithms, do not have a single control point". That is, integration and interoperability are incompatible. Although the term interoperability, as "the ability

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

of two or more information systems or components to exchange information and to use information obtained as a result of the exchange” [4] does not deny the interoperability of integrated systems, which is the subject of this work.

There is also a widespread class of integrated systems in which there are unidirectional control and information links between the systems that are part of it. The essence of these connections lies in the fact that one of the systems acts in relation to the other as a control system, and the other - as a control object that informs the first. One direction of the control link creates a control hierarchy in the system. The known literature does not consider the issues of constructing functional structures that simultaneously possess the properties of integration and hierarchy and ensure the interoperability of their systems, which complicates their modeling and selection.

2. Research objectives. The aim of the work is to increase the interoperability of the systems of an integrated hierarchical system by developing a typical element of interaction based on models with controlling and simultaneously controlled automata..

3. Analysis of publications. The finite state machine [5] is a common model of control devices for discrete systems. It is described by the tuple:

$$A = \langle X, Y, S, s_0, \mu, \lambda \rangle, \quad (1)$$

where X is a set of inputs; Y is a set of outputs; S is a set of states; s_0 is an initial state; μ is the function of outputs; λ is the transition function.

And the requirement of combining the functions of the object and the subject of control in one element is satisfied by a controlled automaton, which is described by a tuple [6]:

$$A = \langle X, Y, S, s_0, C, c_0, F \rangle, \quad (2)$$

where the names of the sets X, Y, S coincide with the similar names behind the tuple (1), but the elements of these sets can be binary or nonbinary; C is a set of controls; c_0 is an initial control; F is the set of functions of the machine in its states.

4. Framework for the functional interaction of systems. The block diagram of the interaction of the control automaton (CA) with the system object, the external environment and the next level in the hierarchical system is shown in Fig. 1.

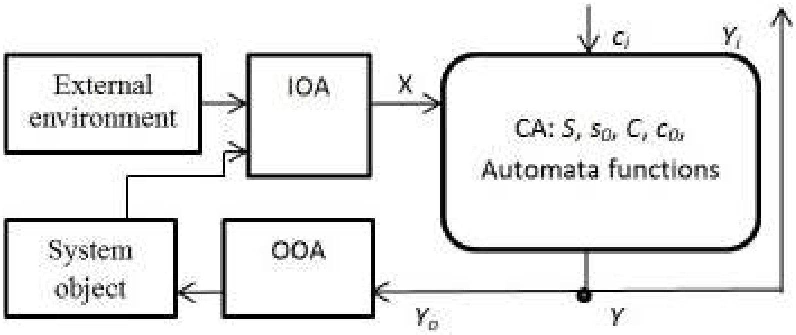

Figure 1 - The block diagram of the interaction control automaton

The scheme is based on the set-theoretic definition (2) of an automaton. The external interface of the machine consists of:

• A set of inputs *X*, inputs operating automata (IOA) that process signals from sensors of parameters of the system object and the environment;

• Signal input of the current control of the structure c_i;

• The set of outputs *Y*, which, through the output operating automata (OOA), control the impact on the object of the Y_o system and inform the higher level Y_i.

The behavior of the automaton CA is determined by the set of states' *S*, the initial state s_0, the initial control c_0, the functions of outputs and transitions of the automaton as a whole [5] or functions of its states [6]. In the loop "CA – Y_0 – OOA – system object – IOA – *X* – CA" the automaton implements the object control behavior, which consists in responding to the signals of the sensors of the parameters of the object and the external environment of a given level by affecting the object in such a way as to change its state as desired. In this case, the nature of the response is set by the control signal at the input c_i, which is formed by a higher-level CA.

In fact, CA is a set of automata, whose sets of inputs and outputs are subsets of the corresponding sets of CA. These machines have various sets of states and transition and output functions. Prior to the arrival of the control signal, the CA acts as if the signal c_0 is present at its control input.

The control signal activates only one of the automata included in it. After activation, the active automaton is in its initial state s_0. Only the active automaton generates the CA outputs that affect the system object and inform the superior automaton about its current state.

One of the options for the interaction of automata of two levels in an integrated hierarchical system is shown in Fig. 2.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

In fig. 2 shows an integrated system consisting of two subsystems with a common element – an automatic machine CA 1, which performs the function of a control machine in the system "System object – IOA 1 – CA 1 – OOA 1", and in the system "CA 1 – CA 2 – IOA 2" – the function of the control object. Events X_2 depending on Y_{12} can be the fact that CA 1 is in a certain state or the fact of recognizing a certain sequence of changes in the states of FA 1. Interrelation of the integrated system Fig. 2 with the higher level is carried out using signals C_2, Y_{23}.

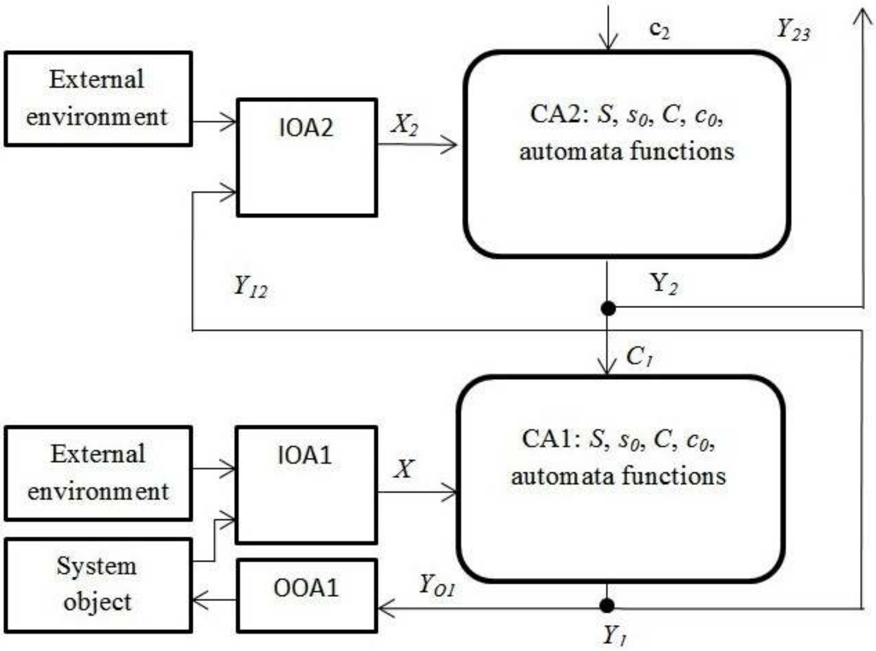

Figure 2 - Integrated system structure

In order to confirm the possibility of the integrated behavior of two automata, consider an example of an integrated system of two automata with the structure shown in Fig. 2. The automaton is selected as a controlled automaton, the graph of which is shown in Fig. 3.

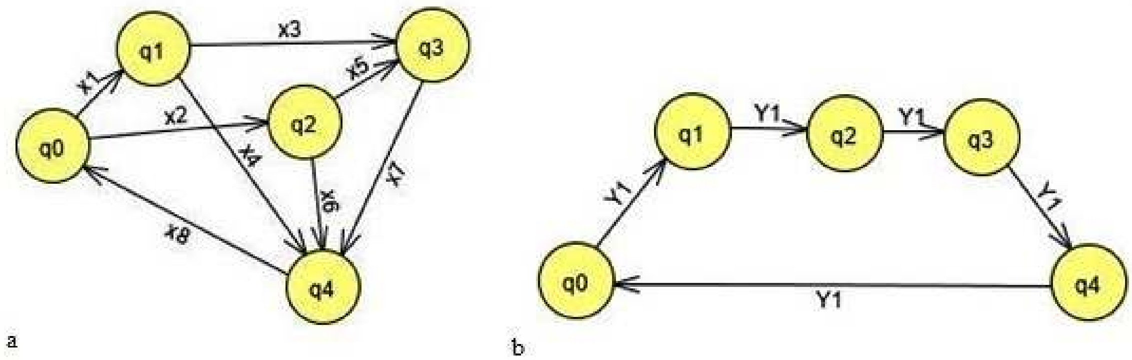

Figure 3 - Graph of controlled (a) and control (b) automaton

Automatic machines Fig. 3 are initial, with an initial state q0. In the automaton Figure 3a, four cycles are possible, which are listed in Table 1.

Table 1

Cycles of a controlled automaton

Cycle designation	Sequence of states and inputs along the cycle
P1	q0 – x1 – q1 – x3 – q3 –x7 – q4 –x8 – q0
P2	q0 – x1 – q1 – x4 – q4 –x8 – q0
P3	q0 – x2 – q2 – x5 – q3 –x7 – q4 –x8 – q0
P4	q0 – x2 – q2 – x6 – q4 –x8 – q0

The availability of cycles and their combinations depends on the set of inputs used and is shown in table. 2

Table 2

Cycles of a controlled automaton

№	Permitted cycles	Permitted inputs
1	P1	x1, x3, x7, x8
2	P2	x1, x4, x8
3	P3	x2, x5, x7, x8
4	P4	x2,x6, x8
5	P1, P2	x1, x3, x4, x7, x8
6	P1, P3	x1, x2, x3, x5, x7, x8
7	P1, P4	x1, x2, x3, x6, x7, x8
8	P2, P3	x1, x2, x4, x5, x7, x8
9	P2, P4	x1, x2, x4, x6, x8
10	P3, P4	x2, x5, x6, x7, x8
11	P1, P2, P3	x1, x2, x3, x4, x5, x7, x8
12	P1, P2, P4	x1, x2, x3, x4, x6, x7, x8
13	P1, P3, P4	x1, x2, x3, x5, x6, x7, x8
14	P2, P3, P4	x1, x2, x4, x5, x6, x7, x8
15	P1, P2, P3, P4	x1, x2, x3, x4, x5, x6, x7, x8

Naturally, the choice of the optimal control (allowed cycles) depends on the specifics of the control object. At the same time, if we take into account some additional features, then it is possible to single out typical control algorithms and control automata that implement them: a control algorithm with a sequential increase / decrease in functionality, increase / decrease in the path length in a cycle, decrease / increase the number of inputs used. The combination of the listed options

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

determines the typical control behavior at this level of the hierarchy. For example, suppose that the options for allowed cycles in Table. 2 are ranked according to the increase in functionality, and then to implement the behavior of a sequential increase in functionality, we use the control automaton shown in Fig. 3b. The outputs of this automaton determine the list of permissions for the inputs of the automatic machine CA 1 in the given state of the automatic machine CA 2. The control option is changed at the moment of transition through the initial state of the controlled automatic machine CA 1, which is perceived by the automatic machine CA 2 as input *Y*1.

The FSM - Simulator program is designed to simulate a two-level hierarchy "controlled - controlling automaton", which takes place in an integrated system. The program was developed in the Visual Studio Code [7] environment in the Type Script language [8]. The program code is available at the link [9]. The program allows you to describe the specification of simulated automata by entering text files.

The program screen contains the *data* folder, the files of which describe the structure of simulated machines first (control object – CA 1) and second (subject of control – CA 2). In fig. 4 shows the folder structure of these machines and the contents of text files that describe the machine CA 1.

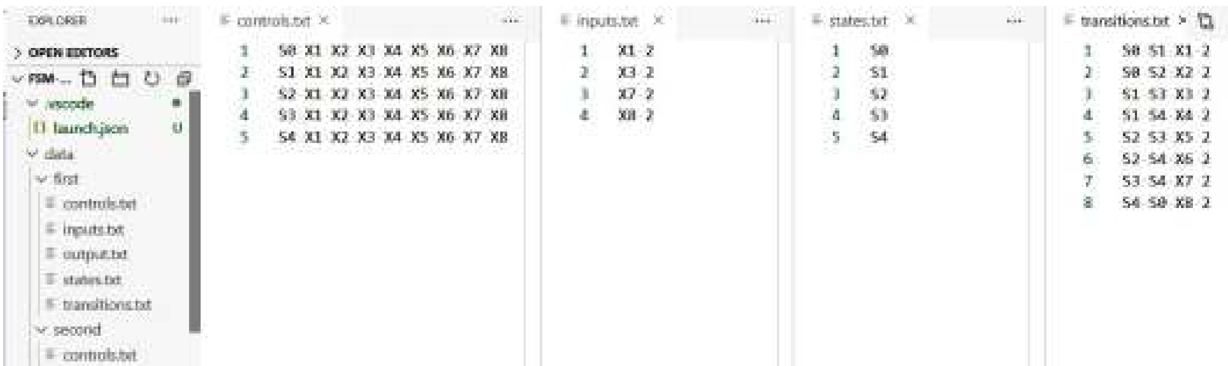

Figure 4 - Screen of the program "FSM – Simulator”

Each machine is described by the following files with the .txt extension:

- *Controls* – describes the initial enable of the machine inputs to exit each state when it is active. In the example in Fig. 4 in each state, all inputs are enabled. During the simulation, some of the permissions are removed;
- *Inputs* – describes the input signal (designation and value) in each cycle in the mode of reading inputs from a file. For example, the record "3 X7 2" means that on the third clock cycle, the X7 signal acts at the input, which has the value 2. In the generation mode, any input signal arrives at the input of the machine with equal probability. In this case, if the signal at the input is not allowed for this state in the

controls and *transitions* files, then it is ignored and the previous state is preserved at this clock cycle;

- *Outputs* – shows the cycle number, designation and value of the input signal, the value of the states of the machine, as well as the list of inputs allowed for transition to the next cycle. It is assumed that the CA is a Moore ternary automaton [6] and the value of its states uniquely determines the value of the outputs. The state in such a machine, apart from the name / identifier, has the following meanings: "0" – passive; "1" – transitional; "2" is active. A fragment of the outputs file is shown in Fig. 5. This fragment of the simulation result, which shows that the active states of the automaton change cyclically (S0 – S1 – S3 – S4 – S0) when the sequence of inputs X1 – X3 – X7 – X8 arrives. The rest of the inputs are ignored. After the end of the cycle, the machine switches to another cycle with the enabled inputs X1, X4, X8. The presence of simultaneously active and transient states in the operating cycle of the automaton is a feature of the behavior of the ternary automaton [6];
- *States* – the designations of states included in the machine are listed. For an example of an automaton fig. 4 these are five states (S0 - S4);
- *Transition* – each line of this file describes one arc of the automaton graph and the vertices incident to it. So the record S0 S1 X1 2 means that there is an arc X1 between the vertices S0 and S1 and the transition is performed if X1 = 2.


```
|  №  ||  X  | Value ||  S0 |  S1 |  S2 |  S3 |  S4 || Allowed x(before transition
|  1  ||  X2 |   2   ||  2  |  0  |  0  |  0  |  0  || X1 X3 X7 X8
|  2  ||  X1 |   2   ||  1  |  2  |  0  |  0  |  0  || X1 X3 X7 X8
|  3  ||  X5 |   2   ||  1  |  2  |  0  |  0  |  0  || X1 X3 X7 X8
|  4  ||  X2 |   2   ||  1  |  2  |  0  |  0  |  0  || X1 X3 X7 X8
|  5  ||  X6 |   2   ||  1  |  2  |  0  |  0  |  0  || X1 X3 X7 X8
|  6  ||  X3 |   2   ||  0  |  1  |  0  |  2  |  0  || X1 X3 X7 X8
|  7  ||  X5 |   2   ||  0  |  1  |  0  |  2  |  0  || X1 X3 X7 X8
|  8  ||  X1 |   2   ||  0  |  1  |  0  |  2  |  0  || X1 X3 X7 X8
|  9  ||  X8 |   2   ||  0  |  1  |  0  |  2  |  0  || X1 X3 X7 X8
|  10 ||  X2 |   2   ||  0  |  1  |  0  |  2  |  0  || X1 X3 X7 X8
|  11 ||  X7 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  12 ||  X3 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  13 ||  X7 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  14 ||  X5 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  15 ||  X3 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  16 ||  X7 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  17 ||  X6 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  18 ||  X2 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  19 ||  X7 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  20 ||  X2 |   2   ||  0  |  0  |  0  |  1  |  2  || X1 X3 X7 X8
|  21 ||  X8 |   2   ||  2  |  0  |  0  |  0  |  1  || X1 X3 X7 X8
|  22 ||  X4 |   2   ||  2  |  0  |  0  |  0  |  1  || X1 X4 X8
|  23 ||  X3 |   2   ||  2  |  0  |  0  |  0  |  1  || X1 X4 X8
```

Figure 5 - Fragment of the results of modeling the behavior of the system

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

In general, the simulation results demonstrate the ability to control the behavior of one automaton with the help of another, as a result of which the interoperability of the systems that form an integrated hierarchical system is ensured.

Conclusion. The terminology in the field of interoperability of integrated systems is in its infancy and needs to be clarified. Therefore, the term “interoperable integrated system” is used to mean “two or more systems that carry out information and control interaction through a common element, the functions of which are different in these systems.

It is proposed to use a controlled finite automaton as a model of a typical element for the implementation of interoperability of systems. The essence of the interaction lies in the choice by the control machine of the behavior of the controlled machine using the information provided by the controlled machine through its outputs.

The content of interaction of systems in an integrated system is determined by its purpose, goals of functioning and other factors. At the same time, typical targets are identified in the interaction of the control and controlled automata. For example, it is expanding the functionality of object control. These goals determine the typical behavior in the system, which we will call the behavior of the automaton control. The number of the automaton behaviors, the number of active inputs, the length (number of stages) of the path in the control cycle is indicators of the level of interoperability.

Flexibility and variety of types of interaction are achieved by describing the structure of the machine in the form of data files in text format and executing the machine with such a structure in a universal software environment.

Using the developed program "FSM – Simulator", the simulation of the interaction of such automata was carried out, which confirmed the possibility and flexibility of the functional interaction of the systems of a hierarchical integrated system using a standard node based on a controlled automaton.

REFERENCES

1. Sagatovsky, V.N. Fundamentals of systematization of general categories (Osnovy sistematizatsii vseobshchikh kategoriy). / V.N.Sagatovsky. - Tomsk: Publishing house of TMI, 1973. - 431 p.

2. GOST 34.003-90 Information technology (IT). Set of standards for automated systems. Automated systems. Terms and Definitions. M.: Standartinform, 2009. 15 p.

3. Poliakov, M.A. Set theoretical model of an integrated control system (Teoretyko-mnozhestvennaya model′ yntehryrovannoy kontrollernoy systemy upravlenyya). System technologies. Regional interuniversity collection of scientific works, №4 (63). Dnipropetrovsk. 2009. P. 130 - 137.
http://st.nmetau.edu.ua/journals/63/17_a_ru.pdf
4. GOST R 55062-2012. Information technologies. Industrial automation systems and integration. Interoperability. Basic principles. Moscow, Standartinform. 2018 –8p.
5. Xavier, E. Theory of automata, formal language and computation.– New age international publishers – 360 p.
6. Poliakov, M.A., Andrias, I.A. Finite state machines with nonbinary elements of sets. (Konechnye avtomaty s nebynarnymy élementamy mnozhestv). System technologies. Regional interuniversity collection of scientific works. Dnipro. 2019. №2 (121). P. 85 – 94.
http://st.nmetau.edu.ua/journals/121/10_a_ru.85-94.pdf
7. Visual Studio Code Site. Available: https://code.visualstudio.com/docs/ [Accessed: Feb. 24, 2021].
8. Type Script language Organization Site. Available: https://www.typescriptlang.org [Accessed: Feb. 24, 2021].
9. FSM Simulator Code. Available: https://gitlab.com/Eonus/fsm-simulator. [Accessed: Feb. 24, 2021].

Received 11.02.2021.
Accepted 15.02.2021.

Интероперабильность интегрированных иерархических систем

Рассмотрены содержание, свойства и особенности интероперабильности применительно к интегрированным иерархическим системам. Отмечено, что терминология в области интероперабильности интегрированных систем находится в стадии становления и нуждается в уточнении. Поэтому термин «интероперабильная интегрированная система» применен в значении «две или более системы, осуществляющие информационно-управляющее взаимодействие через общий элемент, функции которого в этих системах различны».

Предложена структура типового элемента функционального взаимодействия уровней иерархической системы. Этот элемент построен в виде конечного автомата с управляемой структурой. Изменение этой структуры происходит под управлением вышестоящего автомата, что обеспечивает интероперабильность уровней системы. Предложена компьютерная программа для моделирования взаимодействия уровней иерархической системы. Гибкость и многообразие видов взаимодействия достигнуты за счет описания структуры автомата в виде файлов данных в текстовом формате и ис-

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

полнения автомата с такой структурой в универсальной программной среде. Приведен пример моделирования взаимодействия систем с использованием предложенного типового элемента, который подтвердил возможность взаимодействия между системами путем управления поведением одного автомата с помощью другого.

Выделены следующие типовые цели взаимодействия: расширение/сужение функциональных возможностей управления объектом; расширение/сужение множества активных входов автомата, которые используются в процессе управления; минимизация/максимизация длины пути в цикле управляемого автомата. Эти цели определяют типовое поведение в системе, которое названо поведением управления автоматом. Характеристики автоматов по этим целям предложено использовать в качестве показателей уровня интероперабильности.

Інтероперабільність інтегрованих ієрархічних систем

Розглянуто зміст, властивості й особливості інтероперабільності стосовно інтегрованих ієрархічних систем. Відзначено, що термінологія в області інтероперабільності інтегрованих систем знаходиться в стадії становлення та потребує уточнення. Тому, термін «інтероперабільна інтегрована система» застосований у значенні «дві або більше системи, що здійснюють інформаційно-керуючу взаємодію через спільний елемент, функції якого в цих системах різні».

Запропоновано структуру типового елемента функціональної взаємодії рівнів ієрархічної системи. Цей елемент побудований у вигляді скінченого автомата з керованою структурою. Зміна цієї структури відбувається під керуванням вищого в ієрархії автомата, що забезпечує інтероперабільность рівнів системи. Запропоновано комп'ютерна програма для моделювання взаємодії рівнів ієрархічної системи. Гнучкість і різноманіття видів взаємодії досягнуті за рахунок опису структури автомата у вигляді файлів даних в текстовому форматі та виконання автомата з такою структурою в універсальному програмному середовищі. Наведено приклад моделювання взаємодії систем з використанням запропонованого типового елемента, який підтвердив можливість взаємодії між системами шляхом керування поведінкою одного автомата за допомогою іншого.

Виділено такі типові цілі взаємодії: розширення / звуження функціональних можливостей управління об'єктом; розширення / звуження множини активних входів автомата, які використовуються в процесі керування; мінімізація / максимізація довжини шляху в циклі керованого автомата. Ці цілі визначають типову поведінку в системі, яку названо поведінкою керування автоматом. Характеристики автоматів по цим цілям запропоновано використовувати в якості показників рівня інтероперабільності.

Поляков Михайло Олексійович - професор кафедри електричних та електронних апаратів національного університету «Запорізька політехніка».

Субботін Сергій Олександрович - завідувач кафедрою програмних засобів Національного університету «Запорізька політехніка».

Поляков Олексій Михайлович - студент кафедри технічної кібернетики Національного технічного університету «Кієвський політехнічний институт ім. Ігоря Сікорського».

Поляков Михаил Алексеевич - профессор кафедры электрических и электронных аппаратов Национального университета «Запорожская политехника».

Субботин Сергей Александрович - заведующий кафедрой Национального университета «Запорожская политехника».

Поляков Алексей Михайлович - студент кафедры технической кибернетики Национального технического университета «Киевский политехнический институт им. Игоря Сикорского».

Poliakov Mykhailo Oleksiyovich - Professor of the Electrical and Electronic Devices Department of National University «Zaporizhzhia Politechnic».

Subbotin Sergey Oleksandrovich - Chair of Software Tools Department National University «Zaporizhzhia Polytechnic».

Poliakov Oleksii Mykhaylovich - Student of Technical Cybernetics Department National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)