Keywords: railway station; real-time information system; asynchronous automaton; floor equipment

Igor ZHUKOVYTS'KYY
Dnipropetrovsk National University of Railway Transport
Lazaryan 2, 49010, Dnipro, Ukraine
Corresponding author. E-mail: ivzhukl@ua.fm

USE OF AN AUTOMATON MODEL FOR THE DESIGNING OF REAL-TIME INFORMATION SYSTEMS IN THE RAILWAY STATIONS

Summary. The author proposes to develop special models that are built based on finite-state machine, Mealy, in order to display information about technological processes at railway stations. The input alphabet of such machines is represented by the real signals (from the floor equipment, the related information systems and the dispatch office personnel). This representation allows one to formalize the software design process of real-time information management systems for these technological processes. The article demonstrates the possibility of formal transition from the automaton model to the software algorithms. The proposed approach was tested when designing the information system for Nizhnedneprovsk junction railway yard.

1. INTRODUCTION

Railway transport plays a key role in the overall picture of cargo and passenger transportations in many European countries and particularly in Ukraine. It is a mechanism with huge infrastructure, a large number of vehicles and a complex unmanageable control system. Currently, the traffic control in the railway transport cannot be represented without automated information management systems. The information from these systems is used by supervisors, operators and other management workers of railway transport. It is used for analysis and forecasting the situations on the railway operating domain, for making decisions on train control and management of railway stations operation, etc.

The problems of automation of information input in such systems and rational use of the information, which is stored in there, continue to be relevant. For railway stations, information systems have been developed that display current state of the technological process [1]. One can use signals from the line equipment (track circuits, points, track sensors, etc.) in order to input the information about current state of technological process. Such signals may be received by the information system at different stages of technological process. Correct interpretation of such signals and their correlation with certain stage of technological process is an important and relevant task when projecting the algorithms of operation of the information system.

2. LITERATURE REWIEW, PURPOSE AND STRUCTURE OF PAPER

Traditionally, information support of the railway stations operation has been realized by powerful automated systems. For example, in Ukraine, for this purpose, the systems of freight transportation control are used. These are the automated control system of Ukraliznytsia`s freight transportations (ACS UFT) and automated control system of railway yards (ACS RY) [2]. Such systems are common in the other European countries and the United States [3, 4]. The main task of such systems is collection of data, formation of statistical information, forecasting, planning, etc. The weak point of these systems is the situation display at the station in real time. To solve this problem, it is necessary to provide automatic input of information on the state of technological process at the station. The scientists of Dnipropetrovsk National University of Railway Transport, named after Academician V.

Lazaryan, have been solving such problems [1]. Automatic input of information was implemented using the floor equipment signals (track circuits, switches, color light signals, etc.). For correct interpretation of input signals, it was necessary to build an adequate model of the technological process at the station [5].

For a long time, scientists of many countries with developed rail service have been dealt with the problems of simulation model construction for the railway stations operation [6 - 9]. The different simulation mechanisms are also used. Thus, a number of studies show the use of Petri nets for simulation of railway stations operation [10 - 12]. In other studies, the authors use their own software [13] for modeling the technological process at the stations. The work [7] observes the whole complex of the author's software models describing a significant part of the station operation aspects. The work [14] describes the possibility of using the automaton model for simulation of the station operation.

The simulation models are used for non-systemic analysis of station operation, i.e., construction of simulators for the dispatching personnel. They can determine the rational progress of technological process and give suggestions for improvement of gridiron of the stations.

To construct the information systems working in real time, it takes a special approach to modeling. In the simulation model, the input signals are generated by the software components or the system operator. In real-time systems, the input model signals are the real signals from the floor equipment, from the systems of higher level, and in some cases, from the operational staff of the station.

The real-time information system in the station monitors the number of technological processes. Each process should have its own model. The paper sets the problem to develop a standardized methodology for constructing of the automaton model, which makes it possible to monitor each technological process of the station with the desired properties in real time. Moreover, this model should be implemented by the standard software components. For this purpose, it is necessary to create a mechanism for the design of such components.

3. AUTOMATON MODEL DESIGN

The paper considers technological processes that have clearly defined discrete states. The transition between these states occurs under the influence of certain signals or in case of any events. At the railway stations, most of technological processes can be represented in such a way. For example, the state "train inspection" is initiated by the signal "switch on of the safe guard". Transition to the state "completion of the train processing" is influenced by the signal "switch off the safe guard". Controller of information-management system, which is connected with such technological process (Fig. 1), can be presented as a classical representation of abstract machine [15], i.e. in the form of a six-component vector $S=(A,Z,W,\delta,\lambda,a1)$, where:

$A = \{a_1,a_2,\ldots,a_M\}$ – is a set of states (state alphabet);

$Z = \{z_1,z_2,\ldots,z_N\}$ – is a set of input signals (input alphabet);

$W = \{w_1,w_2,\ldots,w_K\}$ – is a set of output signals (output alphabet);

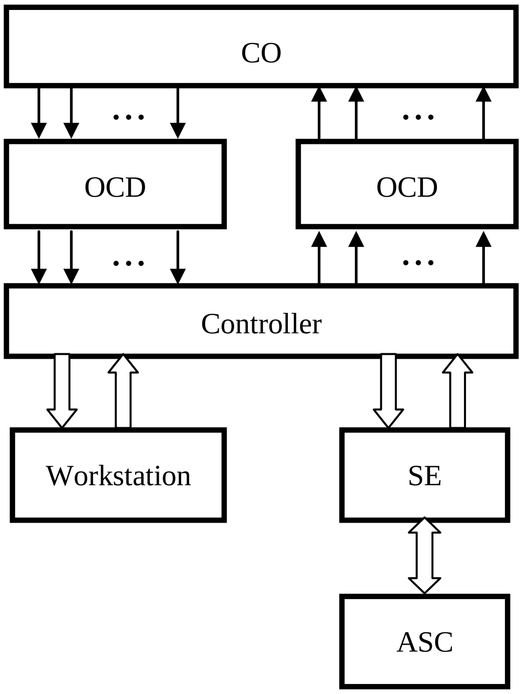
δ - is a transition function, which associates the pairs "state-input signal" (a_m,z_n) with the machine state: $a_s = \delta(a_m,z_n)$, $a_s \in A$;

λ = is an output function, which associates the pairs "state-input signal" (a_m,z_n) with the output signals of the machine $w_k = \lambda(a_m,z_n)$, $w_k \in W$;

$a_1 \in A$ – is an initial state of the machine.

The state alphabet can be represented by the set of (finite) states of the technological process.

A set of input signals (input alphabet) can be associated with a set of input signals and messages received by the controller (the signals from the track circuits, control of the switches position, messages from the ACS of the highest level, etc.), as well as event set, resulting from the combinations of these signals and messages (for example, "the input color light signal opened + occupation of input track circuit + message on the train arrival from the ACS of the highest level").



CO – is a controlled object; OCD – is the object communication device; SE – is switching equipment;
ACS – is an automated control system of the highest level.

Fig. 1. The controller managing technological process

The output alphabet can be represented by a set of actions performed by the controller in response to the input signals (for the information system, it is the messages that are sent to the automated working station and ACS of the highest level, and for the control system, it is also a set of control signals sent by the controller to the operating actuators).

Unlike classical representation of the abstract machine [15], which operates in discrete time receiving non-negative integer values t = 0,1,2,… the machine representing technological process is working in continuous time. The signals received at its input come at random times. To bring this machine to the canonical form, these moments can be represented as a finite discrete numbered set (array), i.e., $t = t_0, t_1, t_2, \ldots$

The developer should clearly define the set of states of the technological process, the input and output alphabet. This problem is complex and its success is largely determined by the proficiency of the developer, i.e., his/her knowledge of all the details of technological process.

As an example let us take the part of automaton representation of information subsystem of the receiving yard (Fig. 2), which is the part of the SCAT RY (system of complex automatization of railway yard [16]).

In this example, the processing workflow of the train that arrived at the receiving yard is represented by the asynchronous machine Mealy. This means the output signal is determined by the state of machine and the input signal, which transferred the machine to this state.

The graph nodes represent the state of machine, or rather the state of the train processing workflow at the receiving yard. At the beginning of arrows, which show transition from the one state to another, there are the signals initiating this transition. The operations performed by information and control system (output signals of the machine) during this transition are given in italics at the end of the arrows.

Initial state of the machine is "The way is clear". The sign "Train reception" is being formed as a set of events. Switch of entrance signal from the permissive to the forbidding one, signals of point positions determine route of the train from the arrival track to this accumulation track. The signal of section track circuit clearing is activated before the given accumulation track. Combination of these signals determines transition of machine to the state "Train has arrived". This transition initiates formation of the special message to computer workstation of duty officer and to the information system of freight traffic about train arrival to the station.

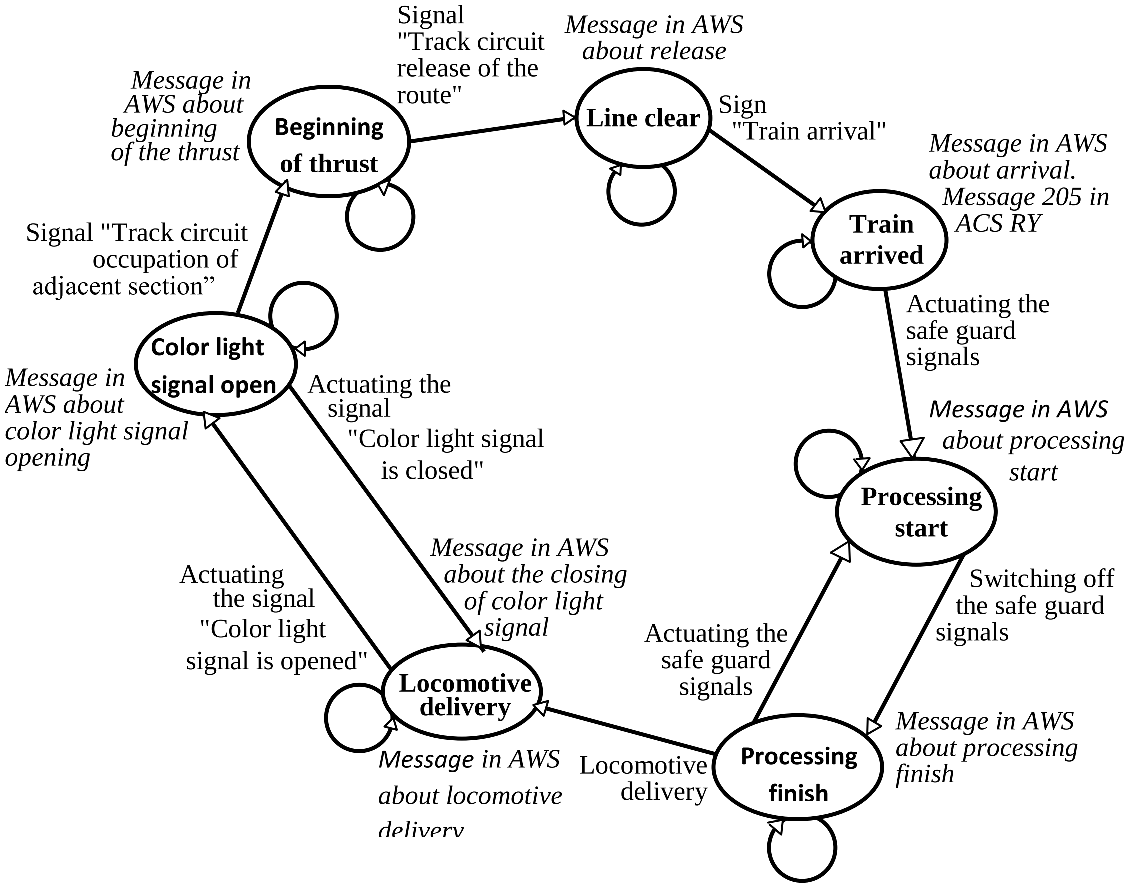


Fig. 2. Part of the automaton representation of information subsystem of the receiving yard

After the train arrival at receiving yard (and locomotive uncoupling), one should perform train processing (technical inspection and preparation for breaking-up). Inspection team starts operation after activation of special protecting signals. The presence of these signals determines transition of machine to the state "Start of processing". Deactivation of these signals means transition to the state "End of processing". The transitions are accompanied by formation of appropriate messages in the computer workstation of duty officer. In some cases, after deactivation of the protecting signals, an additional inspection is required (transition from the state "End of processing" to the state "Start of processing").

Using some signal combinations of track circuit and shunting traffic lights, one can determine the fact of locomotive delivery to the processed train (the sign "Locomotive delivery", transition to the state "Locomotive delivery", formation of message in the computer workstation).

According to the corresponding signals shown in the Fig. 2, further transitions take place: "Opening the traffic light" (signal from the traffic light) and "Start of humping" (the hump engine began to push the train to gravity hump, and the track circuit signal from the section adjacent to the track has been formed). After clearing the section adjacent to the track (signal from track circuit), the machine transits to the state "The way is clear". All transitions are accompanied by forming and transferring of messages to the workstation of duty officer in the park.

It is assumed that for each of the several tracks of receiving yard there is its own automatic model. These models should work in parallel.

It can be assumed that there are technological processes that are described by the asynchronous machine Moore. That is, the output signal of these processes is determined by their state only, independent of the input signal, which transferred them to the present state.

4. SYNTHESIS OF SOFTWARE PROGRAM

In the classical theory of finite-state automaton, the abstract representation is necessary for the subsequent structural synthesis. Since we consider the known in advance technical structure (Fig. 1), the next stage after the automaton representation of the technological process will be the stage of synthesis of software program that implements this representation in the controller.

There are a number of mechanisms for synthesizing microcontroller programs [17]. In December 1993, the International Electrotechnical Commission (IEC) recognized five standard programming languages that could be used for implementing either process or discrete programmable controllers. The IEC is an organization that prepares and publishes international standards for all electrical, electronic, and related technologies, including controllers. The organization identified five programming languages and their common abbreviations as follows: ladder diagram (LD), instruction list (IL), function block diagram (FBD), structured text (ST), and sequential function chart (SFC). The third edition was published in February 2013.

The sequential function chart (SFC) is the closest to the automatic description. These charts divide the sequential task into steps, transitions and actions. They are figured to describe the sequence of interactions [17]. In [18], it is briefly shown how to implement charts using the Ladder Diagram (LD) programming language.

In order to control technological processes at railway stations, complex algorithms and real-time operating systems are required. One of the most advanced real-time systems is the QNX. This system contains its own means of implementing parallelism. This makes it possible to use the description of manufacturing process of the station in the form of set of separate finite state machines, which operate in parallel and independently.

Graphical environment Photon Application Builder (PhAB) is often used to develop real-time systems in the QNX environment. C language is offered for software development in this environment. Therefore, further on, it will be considered the possibility of synthesizing a software product in the PhAB environment using C language. The software product should implement the technological process presented in the form of Mealy machine.

Let us consider the fragment of asynchronous automaton Mealy (Fig. 3). We assume that the set of input signals that do not transit the machine to another state do not generate output signals.

For each state a_i we define a set of input signals $\widetilde{Z}_i \subseteq \boldsymbol{Z}$, $\widetilde{Z}_i = \{z_{i1}, z_{i2}, \ldots, z_{ik_i}\}$, which transfer machine from the state a_i to any other state $a_j \in A$, according to the transition table, wherein $a_j \neq a_i$, a k_i is a number of such signals (it is evident that $k_i \leq N$).

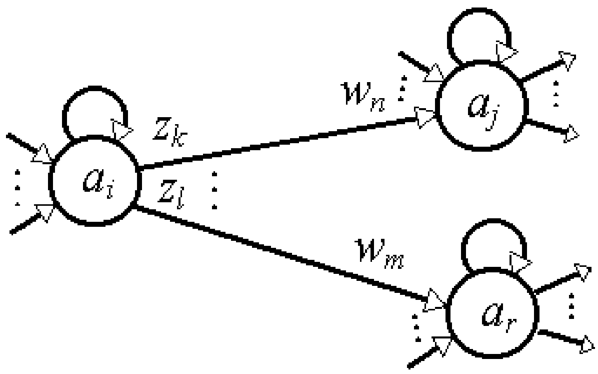


Fig. 3. Fragment of asynchronous automate Mealy

One can propose a formal transition algorithm from the machine graph (for example, Fig. 3) to the algorithmic diagram of the control program (Fig. 4). It is triggered at the receipt of some signal $z \in Z$ (it should be noted that the term "input signal" means the real signals from the floor equipment, and the messages from the ACS of the highest level, as well as the events resulting from the combinations of these signals and messages).

First of all, the current state of the machine is analyzed (the machine state is associated with the state of technological process, for example, as it is shown in the Fig. 2). The input signal is analyzed for each state. When for the *i* state the input signal $z_k \in \tilde{Z}_i$ is received, then the algorithm branch is executed, the operations of which correspond to the output signal w_n. Then the current state is assigned with the value a_j (w_n and a_j are determined by the tables of outputs and the machine transition respectively).

Thus, with the state alphabet, the input and output alphabet, the tables of outputs and transitions of the machine, as well as a number of sets $\tilde{Z}_i \subseteq \boldsymbol{Z}$ (i=1,2,…,M) as the initial data, it is possible to create an algorithm of system response to the receipt of some input signal using the above mentioned rules.

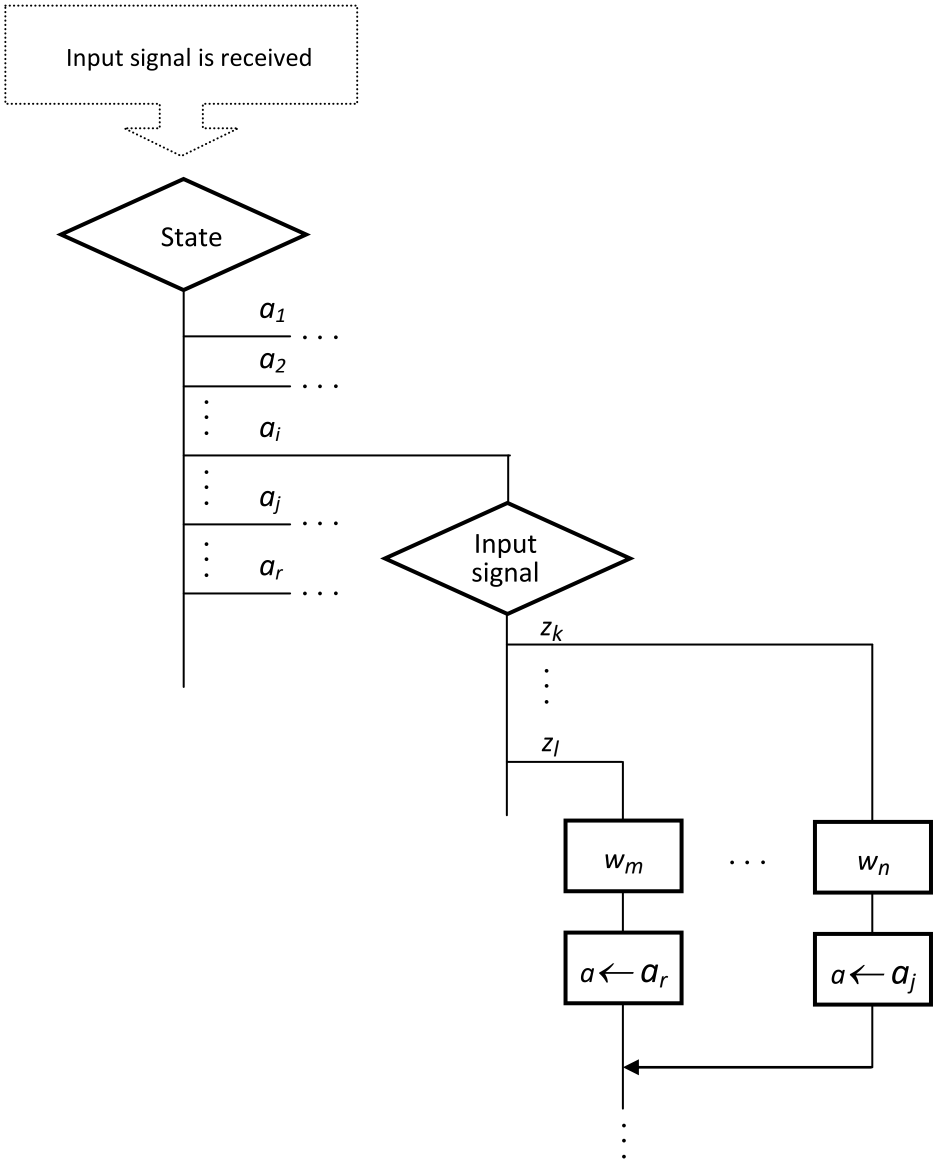


Fig. 4. The fragment of algorithmic diagram of the system response to the receipt of input signal (corresponding to the machine Fig. 3)

The block, in which are implemented the operations corresponding to the output signal w (Fig. 4), can be associated with activation of the corresponding procedure. Thus, it is possible to implement a universal software implementation of the above mentioned algorithm. A setting for a specific machine is performed in the module, which describes the initial machine data (tables), as well as procedures that implement the operations corresponding to the output signals.

This technique was used during design of software for information systems of Nizhnedneprovsk junction railway yard. The system was developed in a short time and was successfully tested in the university laboratory.

5. CONCLUSIONS

During design of real-time information and information management systems for railway stations it is convenient to represent the technological processes at the stations in the form of classical finite-state machine (Mealy or Moore). At this, a set of (finite) states of the technological process corresponds to the state alphabet, and a set of input signals and messages received by the system controller corresponds to the input alphabet. A set of operations performed by the system controller in response to the input signals corresponds to the output alphabet. Such representation of technological process makes it possible to formalize the software design for the controller of information or information management system.

References

1. Шафит, Е.М & Жуковицкий, И.В. & Косорига, Ю.А. & Елисеев, В.В. Совершенствование автоматизированных систем управления технологическими процессами на сортировочных станциях. *Информационно-управляющие системы на железнодорожном транспорте.* 2003. No. 5. P. 53-55. [In Russian: Shaphit, E, & Zhukovyts'kyy, I. & Kosoryga,Y. & Eliseyev, V. Improvement of the automated control systems of technological processes in the marshalling yards. *Information-controlling Systems on the Railway Transport*].
2. Косолапов, А.А. & Жуковицкий, И.В. Резервы архитектуры автоматизированной системы управления грузовыми перевозками Украинских железных дорог. *Залізничний транспорт України.* 2013. No. 1. P. 10-13. [In Russian: Kosolapov, A & Zhukovyts'kyy, I. Provisions architecture automated control system of freight Ukrainian railways. *Rail Ukraine*].
3. Cherneva, G. & Andonov, A. & Hubenová, Z. The problem of synthesis of functionally steady information systems in railway transport. In: *5th International Scientific Conference "Theoretical and Practical Issues in Transport"* (11-12 February 2010, Pardubice, Czech Republic). Available at: http://hdl.handle.net/10195/37758.
4. Константінов, Д. & Мисько, М. Дослідження перспективних систем диспетчерського управління на основі сучасних інформаційних технологій. *Збірник наукових праць УкрДАЗТ.* 2013. No. 140. P. 91-96. [In Ukrainian: Konstantinov, D. & Mysko, M. Research Policy supervisory control system based on modern information technology. *Proceedings of UkrDAZT*].
5. Жуковицкий, И. Синтез программного обеспечения информационно-управляющих систем на основе теории абстрактных автоматов. *Інформаційно-керуючі системи на залізничному транспорті.* 2001. No. 5. P. 39-41. [In Russian: Zhukovyts'kyy, I. Synthesis software information management systems based on abstract theory of automata. *Information-controlling Systems on the Railway Transport*].
6. Лещинский, Е. *Имитационное моделирование на железнодорожном транспорте*. Москва: Транспорт. 1977. 176 p. [In Russian: Leschinkiy, Ye. *Simulation modeling on the railway transport.* Moscow: Transport].

7. Вернигора, Р. & Малашкин, В. Моделирование работы систем станционной автоматики в эргатических имитационных моделях железнодорожных станций. *Сб. науч. трудов Днепропетр. нац. ун-та ж.д. тр-та «Транспортные системы и технологии перевозок».* – 2011. No. 2. P. 31-37. [In Russian: Vernigora, R. & Malashkin, V. Modeling of the railway automation systems in the ergatic models of railway stations. *Collection of Scientific Papers of the Dnipropetrovsk National University of Railway Transport «Transport systems and transportation te hnologies»*].
8. Szűcs, G. Railway Simulation with the CASS NDRA Simulation System. *Journal of Computing and Information Technology*. 2001. Vol. 9. No. 2. P. 133–142.
9. Сукач, Е. Стенд имитационного моделирования сортировочной станции железнодорожной сети. *Проблемы программирования*. 2009. No. 3. P. 81-89. [In Russian: Sukach, Y. The stand for simulation modeling of marshalling yard on railway network. *Programming problems*].
10. Нагорный, Е. & Алешинский, Е. Моделирование функционирования комплекса «Сортировочная станция – прилегающие участки» с помощью сетей Петри. *Информационно- управляющие системы на железнодорожном транспорте.* 2000. No. 2. P. 98–103. [In Russian: Nagorniy, Y & Aleshinskiy, Ye. Modeling of complex functioning "Marshalling Yard – adjacent areas" with Petri nets. *Information-controlling Systems on the Railway Transport*].
11. Giua, A. & Seatzu, C. Modeling and supervisory control of railway networks using Petri nets. *Automation Science and Engineering, IEEE Transactions on.* 2008. Vol. 5. No. 3. P. 431-445.
12. Milinkovic, S. & Markovic, S. & Veskovic, M. & et al. A fuzzy Petri net model to estimate train delays. *Simulation Modelling Practice and Theory*. 2013. No. 33. P. 144-157.
13. Бобровский, В. & Вернигора, Р. Функциональное моделирование железнодорожных станций в тренажерах оперативно-диспетчерского персонала. *Математическое моделирование*. 2000. No. 2(5). P. 68-71. [In Russian: Bobrovskiy, V. & Vernigora, R. Functional simulation of railway stations in the simulators for dispatching personnel. *Mathematic modelling*].
14. Bobrovskiy, V.I. & Kozachenko, D.N. & Vernygora, R.V. Functional simulation of railway stations on the basis of finite-state automata. *Transport Problems*. 2014. Vol. 9. No. 3. P. 57-66.
15. Баранов С.И. *Синтез микропрограммных автоматов*. Ленинград: Энергия, 1979. 232 p. [In Russian: Baranov, S. *Synthesis of firmware automats*].
16. Шафит, Е. & Жуковицкий, И. & Косолапов, А. & Туник, Т. Выбор технических средств и разработка вариантов технической структуры СКАТ-СС. *Інформаційно-керуючі системи на залізничному транспорті.* 2000. No. 4. P. 127. [In Russian: Shaphit, E. & Zhukovyts'kyy, I. & Kosolapov, A. & Tunik, T. The choice of instruments and options for the technical structure of СКАТ-СС. *Information-controlling Systems on the Railway Transport*].
17. Fialkowski, Ch.M. Sequential function chart programming. Processing sequential and parallel operations based on time or events. Available at: https://www.isa.org/intech/20140804/
18. McWhinnie, J. Sequential Function Charts for All. Implementing Sequential Function Chart Designs using Ladder Diagram Programming Language for Programmable Logic Controllers. Napier University, Edinburgh, Scotland. Available at: http://www.plcdev.com/sequential_function_charts_all

Received 03.10.2015; accepted in revised form 02.12.2017