Block diagram of the automatic model of technical documentation

D Baratov¹, N Aripov¹, O Muxiddinov¹, X Jumanov¹

¹ Tashkent railway engineering institute, Tashkent, st. Odilxojayev 1, Uzbekistan

E-mail: baratovdx@yandex.ru

Abstract. The article explores the features of electronic document management of technical documentation of railway automation and telemechanics. The article deals with the synthesis of mathematical models of electronic document management of technical documentation of railway automation and telemechanics. For this purpose, the real processes of creation, verification and use of technical documentation were examined using the example of the process of accounting and control of railway automation and telemechanics devices, which made it possible to identify a structural diagram of a formalized model of technical documentation. A model of electronic document management of technical documentation is created, created using the mathematical apparatus of finite state machines. A block diagram of an automaton model of technical documentation is developed. The proposed block diagram of a formalized model of technical documentation consists of matrices of external microoperations, internal microoperations, code creation for the following microcomponents. The number of internal states of the firmware is completely determined by the number of logic elements of the algorithms. The size of the matrices depends on the number of operators and logical conditions in the logic circuits of the algorithms.

1. Introduction

The traditional way to automate electronic document management of technical documentation (EDMTD) is that the development of software gradually changes to the background mode, since even minor changes in the transport scheme and technological process mean the need to reprogram the application and consume more amount of time and money. As a result, there will be no time to update appeals, based on changing conditions and requirements of railway transport [1-3]. Actively developing systems associated with enterprise automation require a large number of trained personnel due to the high complexity. In addition, the number of qualified specialists in the field of automation is growing insufficiently fast [4]. Thus, the task is to create an easy-to-use, reliable automation tool, in the arsenal of which there are not only tools for describing a business process, but also its implementation. The possibility of implementation is important, since a pure descriptive tool is interesting only from the point of view of the analysis of technological processes and can be used only as part of a technical task when performing a specific model of the work process [5, 6]. EDMTD modeling is presented in the works of both Russian [7-12] and foreign authors [13-18].

Based on the logic schemes of algorithms (LSA) developed in the papers [19-21] for the process of accounting and control of railway automation and telemechanics devices (CARCD), this article presents a block diagram of an automatic model of technical documentation (TD), microcommands are...
formed according to the logic diagrams of algorithms TD, the problems of simplifying the structure of the automaton of the CARCD process are considered.

The purpose of this work is to describe and define an automated workflow model. EDMTD is considered as an object of modeling. To implement the model, an approved machine of the theory of automata will be used, adapted to modern programming technologies. The task of creating and implementing EDMTD is very urgent today. Significant financial, material and time resources are spent at enterprises and organizations of the railway to solve this problem.

2. Synthesis of an electronic document management model for technical documentation based on an abstract finite state machine

The most effective solution to the problems of automation of the CARCD process can be achieved by formalizing and applying mathematical methods to optimize coordination of interaction.

A number of methods are known for identifying the functioning algorithms of complex systems, namely: the method of simplifying the work [22]; drawing up structural informational-temporal schemes [23], flowcharts and organigrams [24]. The essence of these methods lies in the operational recording and analysis of the investigated process. The common drawbacks of these methods from the point of view of the CARCD study are: a limited set of conditional designations of operations; the complexity, and for a number of methods [25] and the impossibility of displaying the parallelism of the processes of CARCD, the complexity of filling out the survey forms.

There are languages for the direct description of discrete processes, which include Petri nets [26], logic circuits of LSA algorithms [27], logic circuits of requirements [28], parallel logic circuits of algorithms (PLC) [29].

CARCD taking into account the parallelism of processes can be considered as a stationary dynamic system with discrete time. According to [24], such a system is an automaton. Thus, it is advisable to use the methods of the theory of automata to simulate the functioning of the CARCD process.

The automaton approach [19] is to display the process as a mutual automation system (one automaton is placed in one or several states of another automaton) with the possibility of a call (one automaton is called by a certain event from the output state generated during the passage of another automaton) by exchange messages (one machine receives a message from another) and status (one machine checks the status of another machine). The internal organization of the process can be described by a sequence of events [20]. The number of machines installed in internal states is not limited by the depth of positioning. This representation allows a more compact description of the life cycle of a program, module, and in our case, the AP or the CARCD process. In turn, a compact presentation improves the look.

As a simulated object EDTD railway automation. The formulation introduced in [20, 21] is used as the basis for the automaton model.

Formally, the EDTD process is presented in the form of three limited sets and interconnections of the elements of this set [6]. The mathematical representation of this process is given in the following form

\[ D_T = \{ U, P, F \} \]

here \( D_T \) is a formal representation of EDMTD; \( U \) is the set of participants; \( P \) is the set of processes; \( F \) - state of the TD with a range of actual values.

A limited set of real participants in the working process is established, \( P \) - in the revised system of the working process is defined as a limited set of processes performed by participants. \( F \) are the limited states that can be accepted by the TD after performing the procedures from \( P \) by the participants of the indicated set \( U \).

Using the theory of automata [30-33], we determine the automaton that executes the EDMTD model.
S (many states) is the set of all states that can be accepted by a document as part of a simulated workflow. Using the formulated notation from [6], this definition will be written as follows: \( \{S\} \equiv \{F\} \).

Initial states refer to many states as a whole. \( S_0 \) is the initial set of states \( S \). Therefore, in the framework of the proposed model \( \{S\} \equiv \{F\} \), \( S_0 \) can be considered a subset of \( \{F\} \).

We define the relationship of the set of processes \( P \) from the definitions of EDMTD and the set of transition functions \( F_p \). When constructing a model of an automaton model, the corresponding elements of the set: \( \{F_p\} \equiv \{P\} \) to determine the set \( F \).

In the described model, identification is established between the alphabet of the EDMTD automaton and the set of participants: \( \{\{A\} \equiv \{U\}\} \).

After synthesizing the model \( \{U, P, F\} \), we obtain the automaton model of EDMTD, which is defined:

\[
M = (A, S, Z, s_0, F_p, E),
\]

here \( A \) is the input alphabet; \( S \) is the internal alphabet; \( Z \) is the output alphabet; \( s_0 \) is the initial status; \( H \) is the transition function defined by the transition table and denoting two sets \( A \times S \to S \); \( E \) is the exit function defined by the exit table and denoting two sets \( A \times S \to Z \).

For this example: \( A = \{a, b\}; S = \{1,2,3\}; Z = \{0,1\}; s_0 = 1; F_p = \{1,1,1\}; E = \{2,2,2\} \).

Application of the presented model allows us to combine the approach in the development and use of EDMTD systems. The introduction of the EDMTD system will make the process of storing TD more transparent and predictable, and reduce the personal influence of executive staff on the final result.

3. Development of a block scheme of an automaton model of technical documentation

On the basis of a microprogram machine (MA), a block diagram of a formalized TD model has been developed.

Based on the MA, the Wilks-Stinger scheme was implemented, which is used when there are no strict requirements for the speed of machine control in the synthesis of microprograms with the least control [34].

The proposed structural diagram of the formalized TD model consists of matrices of external micro-operations \( M_1 \), internal micro-operations \( M_2 \), code creation for the following micro-components \( M_3 \). The presented model has become a new system paradigm for the representation of TDs [35, 36].

Each external micro-operation \( Z_v \) is a control operator, and the internal one \( Z_{\alpha_{gg}} \) is a control logical \( \alpha_{gg} \) condition that checks a condition whose values are fulfilled \( (\alpha_{gg} = 1; \text{marked with a + sign}) \) or non-fulfilled condition \( (\alpha_{gg} = 0; \text{marked with a - sign}) \). On the structural diagram, a circle denotes a circuit that serves to query the value \( Z_{\alpha_{gg}} \) of the condition it is checking (Fig. 1).

Firmware, i.e. the sequence of microoperations is conveniently described in the language of LSA, and external microoperation \( V_v \) is associated with the operator \( Z_{v_v} \), and internal microoperation \( \alpha_{gg} \) is associated with the logical condition \( Z_{\alpha_{gg}} \).

When performing this operation, each micro-command includes only one micro-operation (external or internal) and includes only one operator or logical condition during each microtact. The number of internal states of MA is completely determined by the number of LSA elements. The size of the matrices \( M_1 \) and \( M_2 \) depends on the number of operators and logical conditions in the LSA. In the specific case, when only operators enter the LSA, the matrix \( M_2 \) is absent. In this case, successive microcommands are generated each time in the \( M_3 \) matrix.
If there are logical conditions in the LSA, the matrix M2 is needed, in which they are formed $Z_{\alpha_{gr}}$. If the value checked in $Z_{\alpha_{gr}}$ the logical condition (if it enters the LSA without inversion), the order of execution of the LSA elements is violated. Then, in M3, the necessary number of microcommands must be formed in order to correctly perform LSA.

$$\begin{align*}
&\text{Figure 1. Block scheme of a formalized TD model.}
\end{align*}$$

If you combine a separate micro-command with each element of the LSA, even a very simple algorithm requires that MA fulfill a large number of internal states. At the same time, there is no need to distinguish the internal position for each LSA element. Some external microoperations may not be performed sequentially, but simultaneously in a single microact. This occurs when the operator performing operations on the corresponding microacts can work in parallel. LSA statements executed simultaneously are not related to the internal state of a single MA statement. This minimizes the number of internal states of MA. The combination of individual microoperations over time leads to an increase in speed, since the number of microtacts required to execute the algorithm decreases.

Simplification of the MA scheme can be obtained by simultaneously performing external and internal microoperations. Then the number of microcommands of MA will be determined not by the number of LSA elements, but by the number of groups of simultaneously performed microoperations. Consider the LSA technological process of accounting and control of railway automation and telemechanics devices $A_T$, which includes planning and repair of equipment, control of reception and storage, as well as the movement of devices in the repair and maintenance department:

$$A_T = V_0 V_{711} V_{712} V_{713} V_{714} \downarrow V_{715} V_{716} V_{717} \downarrow V_{718} \alpha_{711} \uparrow V_{719} \omega \uparrow V_{715}$$

$$\times \downarrow V_{712} \downarrow V_{715} \alpha_{713} \uparrow V_{716} \alpha_{714} \uparrow V_{717} \omega \uparrow V_{716} \downarrow V_{718} \omega \uparrow V_{715}$$

The CARCD process is presented in the LSA language [37, 38]. The main elements are the operators $V_{qg}$, $q = 1, Q, g = 1, G$ that correspond to the elementary actions of the process, logical conditions $\alpha_{q}, k = 1, K$ - the probability of their fulfillment depends on the current state of the process CARCD, represented by arrows $\alpha_{q} \uparrow^r, p = 1, P$, where $p$ is the index of the arrow.
Here we can distinguish 6 groups of simultaneously performed microoperations:

\[ A_T = V_0 V_{711} V_{712} V_{713} V_{714} V_{715} V_{716} V_{717} V_{718} \alpha_{711} V_{719} \alpha_{712} \]

\[ \times V_{715} \alpha_{713} V_{716} \alpha_{714} V_{717} \alpha_{715} V_{718} \alpha_{716} V_{719} \alpha_{717} \]

It follows from (1) that MA will have six internal states. The M3 matrix in Fig. 2 is constructed when compared with microcommands 1, 2, 3, 4, 5, 6 of the code combinations 001, 010, 011, 100, 101, 110, respectively.

Thus, we examined the method of phasing the microcommands in stages, and as a result we switched from the micro-command, which includes only one external or internal micro-operation, to the micro-command, which includes the entire group of external and internal micro-operations. To form such microcommands, a firmware should be implemented in addition to the LSA, providing information on the possibility of simultaneous operation of various operators and the distribution of offsets for each operator. Obviously, the operator \( V_{qg} \) and the logical condition \( \alpha_{qg} \) cannot be included in one micro-command if the value can be changed by the operator \( V_{qg} \). The task of forming the smallest possible number of LSA microcommands is complex.

When considering various possible ways of constructing an MA circuit, it is considered that only one algorithm is implemented. However, a software control method is used precisely when several different algorithms need to be implemented in the machine.

The structure of the MA transition graph will largely depend on the choice of microcommands when comparing internal states with microcommands.
A special requirement is the development of methods that allow you to minimize and encode the machine, taking into account various requirements.

4. Conclusion
The article proposes a method for constructing a structural diagram of a formalized model of TD microcircuit formation, effective for solving electronic document management tasks [39, 40].

The presented model can be used not only for the automation of CARCD processes, but also for electronic document management processes in general. The application of the proposed methodology for constructing the TD model allows the development of reliable application software for solving operational document management tasks.

Acknowledgments
We express the department of science of the Institute for the created conditions.

References
[1] Baratov D Kh, Boltaev A Kh, Astanaliev E T 2019 *International Journal of Advanced Research in Science, Engineering and Technology* 6(3) 8572
[2] Shamanov V 2016 *Automation on Transport* 2(2) 163
[3] Aripov N M, Baratov D X 2016 *International scientific journal* 9 53
[4] Bulavsky P Y, Markov D S 2013 *Actual issues of the development of railway automation systems of telemechanics: Sat. scientific proceedings SPb* 52
[5] Aripov N, Aliyev R, Baratov D, Ametova E 2016 *Procedia Engineering* 134 175
[6] Bulavsky P Y 2011 *Transport of the Russian Federation* 1(32) 60
[7] Mamikonov A G 1973 *Development methods for automated control systems* (M.: Energiya) p 336
[8] Aripov N M, Baratov D Kh, Ametova E K 2017 *Bulletin Of The Tashkent Railway Transport Engineering Institute* 3-3 90
[9] Baratov D 2017 *International scientific journal Internauka* 4(1) 22
[10] Markov D S, Likov A A, Sokolov V B, Konstantinova T Yu 2017 *Collection: intelligent Transport systems* 49
[11] Vasilenko M N, Bulavsky P E, Baratov D 2009 *World of transport* 7(4) 110
[12] Stepanov M F, Stepanov A M 2018 *Information technologies and nanotechnologies* 1681
[13] Guo F, Jahren C T, Turkan Y 2019 *International Journal of Construction Education and Research* 1
[14] Sun M T, Hou J L 2003 *Journal of the Chinese Institute of Industrial Engineers* 20(4) 305
[15] Sokolov S S, Karpina, A S, Gaskarov V D 2016 *Vestnik of Astrakhan State Technical University Series: Management, Computer Sciences and Informatics* 3 40
[16] Afonso C M, Roldán Salgueiro J L, Sánchez Franco M J, González M D L O 2012 *The moderator role of Gender in the Unified Theory of Acceptance and Use of Technology (UTAUT): A study on users of Electronic Document Management Systems*
[17] Yatin S F M, Ramli A A M, Shuaimi H, Hashim H, Dollah W A K W, Zaini M K, Kadir M R A 2015 *Australian Journal of Basic and Applied Sciences* 9(3) 82
[18] Aurelia P, Ana T 2008 *A document management system modeling* (Analele Universităţii Din Oradea)
[19] Aripov N M, Baratov D Kh 2017 *Automation on Transport* 3(1) 98
[20] Baratov D Kh, Aripov N M 2016 *Europaische Fachhochschule* 8 33
[21] Baratov D, Aripov N, Ruziev D 2019 *IEEE East-West Design & Test Symposium (EWDTS)* 1
[22] Verneburg R 1969 *Implementation of mathematical models on a computer* (Moscow, Progress) p 189
[23] Mamikonov A G 1973 *Methods for developing automated control systems* (Moscow, Energy) p 336
[24] Mamikonov A G, Aven O I, Kulba V V 1970 Formalized presentation of analysis results and design of automated control systems (Moscow: Institute of management problems) p 57
[25] Shannon R Yu 1978 Imitating modeling of systems – art and science (Moscow, Mir) p 418
[26] Tal D A 1982 Automation and remote control 7 113
[27] Lazarev V G, Piyl Y I 1978 Synthesis of control automata (Moscow, Energy) p 408
[28] Roginsky V N 1975 Basics of discrete automation (Moscow: Svyaz) p 432
[29] Yanov Yu 1958 Problems of Cybernetics 2 75
[30] Kalman R E, Falb P L, Arbib M A 2004 Essays on mathematical systems theory (Moscow: Editorial URSS) p 400
[31] Dolinsky M, Korshunov I, Tolkachev A, Yermolayev I, Litvinov V 2003 Components and technologies 8 124
[32] Kleban V, Kleban F 2008 Scientific and technical Bulletin of SPbSU ITMO 53 286
[33] Sapozhnikov V, Sapozhnikov V 1992 Self-checking discrete devices (SPb: Energoatomizdat) p 224
[34] Wilkes M V, Stringer J B 2019 Mathematical Proceedings of the Cambridge Philosophical Society. Cambridge University Press 49(2) 230
[35] Furth S, Baumeister J 2016 Enterprise Big Data Engineering, Analytics, and Management (IGI Global) p 200
[36] Aguilar A, Lozoya C, Orona, L M 2019 Peer-to-Peer Networking and Applications 12(5) 1323
[37] Baratov D X 2017 The issues of creating a formalized model of the technical documentation 4(1) 22
[38] Sapozhnikov V V, Sapozhnikov V V, Efianov D V 2019 Fundamentals of reliability theory and technical diagnostics (Moscow: Lan) p 588
[39] Baratov D, Aripov N 2018 Basic and applied research 64
[40] Sapozhnikov V, Sapozhnikov V, Efianov D, Dmitriev V 2017 Automation and remote control 78(2) 300