

Industrial Transport of Kazakhstan
ISSN 1814-5787 (print)
ISSN 3006-0273 (online)
Vol. 22. Is. 4. Number 88 (2025). Pp. 81–102
Journal homepage: <https://prom.mtgu.edu.kz>
<https://doi.org/10.58420/ptk/2025.88.04.007>
UDC 681.5

MODELS AND INFORMATION TECHNOLOGIES FOR FORMALIZATION OF DESIGN PROBLEMS AND RAILWAY STATIONS DEVELOPMENT AUTOMATION

V. Lakhno¹, D. Zhamangarin^{2}, G. Muratbekova³, A. Tabylov⁴*

¹National University of Life and Environmental Sciences of Ukraine, Kiev, Ukraine;

²Kazakh University of Technology and Business, Astana, Kazakhstan;

³Academy of Civil Aviation, Almaty, Kazakhstan;

⁴Caspian University of technologies and engineering named after Sh. Yessenov, Aktau, Kazakhstan.

E-mail: dus_man89@mail.ru

Valerii Lakhno — Doctor of Technical Sciences, Professor, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine

E-mail: lva964@gmail.com, <https://orcid.org/0000-0001-9695-4543>;

Dusmat Zhamangarin — PhD, Kazakh University of Business and Technology, Astana, Kazakhstan

E-mail: dus_man89@mail.ru, <https://orcid.org/0000-0002-3141-5984>;

Gulzhan Muratbekova — Candidate of Technical Sciences, Civil Aviation Academy, Almaty, Kazakhstan

E-mail: gv170471@mail.ru, <https://orcid.org/0009-0003-0005-0470>;

Abzal Tabylov — Candidate of Technical Sciences, Sh. Yesenov Caspian University of Technology and Engineering, Aktau, Kazakhstan

E-mail: abzal.tabylov@bk.ru, <https://orcid.org/0000-0002-7532-5681>.

© V. Lakhno, D. Zhamangarin, G. Muratbekova, A. Tabylov

Abstract. The research topic is dedicated to planning technical support, developing automated control systems (ACS), and improving modeling methods for railway stations (RS) technological processes. Modern design engineers require a holistic and systematic understanding of models representing the research object. These models should reflect all operational aspects of future technical systems, including the structure and main processes of the object. The aim of the study is to improve the methodology for formalizing technological processes (TP) of railway stations based on the visual programming paradigm. Objectives include creating use case diagrams (UCD), state diagrams (DIS), and activity diagrams (DIA) in UML to visually represent TPs; describing each object, its executors, and works; formalizing the sequence of technological operations. The results show that using UML allows visualizing the interaction of RS objects and executors, parallel and synchronized processes, and the execution stages of each operation. DIS and DIA diagrams help determine the states of individual objects and the phases of operations. Automated modeling enhances TP efficiency and ensures mutual understanding between designers and operators. Furthermore, graphical-analytical models developed in Rational Rose and AutoCAD facilitate automation of TP planning and calculation of key performance indicators. In conclusion, the proposed methodology enables efficient design, documentation, formalization, and visualization of RS TPs, simplifying technical decision-making and improving the quality of ACS and IS development for railway stations.



Keywords: railway stations, technological process, UML, use case diagram, state diagram, activity diagram, automated control

For citation: V. Lakhno, D. Zhamangarin, G. Muratbekova, A. Tabylov Models and information technologies for formalization of design problems and railway stations development automation // Industrial Transport of Kazakhstan. 2025. Vol. 22. No. 88. Pp. 81–102. (In Eng.). <https://doi.org/10.58420/ptk/2025.88.04.007>.

Conflict of interest: The authors declare that there is no conflict of interest.

ЖОБАЛАУ МӘСЕЛЕЛЕРІН РЕСІМДЕУГЕ ЖӘНЕ ТЕМІРЖОЛ ВОКЗАЛДАРЫН ДАМУ ТУДЫ АВТОМАТТАНДЫРУҒА АРНАЛҒАН МОДЕЛЬДЕР МЕН АҚПАРАТТЫҚ ТЕХНОЛОГИЯЛАР

В. Лакно¹, Д. Жамангарин^{2}, Г. Муратбекова³, А. Табылов⁴,*

¹Ұлттық биоресурстар және табиғатты пайдалану университеті, Киев, Украина;

²Қазақ бизнес және технология университеті, Астана, Қазақстан;

³Азаматтық авиация академиясы, Алматы, Қазақстан;

⁴Ш. Есенов атындағы Ксапий технологиялар және инжиниринг университеті, Ақтау,
Қазақстан.

E-mail: dus_man89@mail.ru

Валерий Лакно — т.ғ.д., профессор, Ұлттық биоресурстар және табиғатты пайдалану университеті, Киев, Украина.

E-mail: lva964@gmail.com, <https://orcid.org/0000-0001-9695-4543>;

Дусмат Жамангарин — PhD, Қазақ бизнес және технология университеті, Астана, Қазақстан.

E-mail: dus_man89@mail.ru, <https://orcid.org/0000-0002-3141-5984>;

Гульжан Муратбекова — т.ғ.к., Азаматтық авиация академиясы, Алматы, Қазақстан

E-mail: g.muratbekova@alt.edu.kz, <https://orcid.org/0009-0003-0005-0470>

Абзал Табылов — т.ғ.к., Ш. Есенов атындағы Ксапий технологиялар және инжиниринг университеті, Ақтау, Қазақстан.

E-mail: abzal.tabylov@bk.ru, <https://orcid.org/0000-0002-7532-5681>

©В. Лакно, Д. Жамангарин, Г. Муратбекова, А. Табылов

Аннотация. Зерттеу тақырыбы техникалық қолдауды жоспарлау, автоматтандырылған басқару жүйелерін (АБЖ) құру және теміржол станцияларының (ТС) технологиялық процестерін модельдеу әдістерін жетілдіруге арналған. Қазіргі заманда жобалау инженерлері зерттелетін объектінің моделін толық және жүйелі түсінуі қажет. Модельдер болашақ техникалық жүйелердің жұмысын толық сипаттап, объектінің құрылымы мен негізгі процестерін көрсетуі тиіс. Зерттеудің мақсаты – теміржол станцияларының технологиялық процестерін (ТП) визуалды бағдарламалау парадигмасы негізінде формальдендіру әдісін жетілдіру. Міндеттері: UML тілінде қолдану жағдайлары диаграммаларын (UCD), күй диаграммаларын (DIS) және әрекет диаграммаларын (DIA) жасау арқылы ТП-ны графикалық көрсету; әр объектіні қызмет көрсететін орындаушылар мен жұмыстар бойынша сипаттау; ТП орындау тәртібін формальдендіру. Нәтижелер көрсеткендей, UML тілін қолдану арқылы ТС объектілері мен олардың қызмет көрсетушілерінің өзара әрекеттесуін, ТП-ның параллельді және синхронды процестерін визуалдауға мүмкіндік туады. DIS және DIA диаграммалары әр объектінің күйін және әрбір операцияның орындалу кезеңдерін анықтауға мүмкіндік береді. Автоматтандырылған модельдеу ТП-ның тиімділігін арттырып, жобалаушы мен оператор арасындағы өзара түсіністікті қамтамасыз етеді. Сонымен қатар, Rational Rose және AutoCAD орталарында

әзірленген графикалық-аналитикалық модельдер ТП жоспарларын автоматтандыруға және негізгі көрсеткіштерді есептеуге мүмкіндік береді. Қорытындылай келе, ұсынылған әдістеме ТС ТП-ның моделін тиімді жобалауға, оны құжаттауға, формальдендіруге және визуализациялауға мүмкіндік береді, бұл техникалық шешімдерді қабылдау процесін жеңілдетеді және АБЖ мен ИС жобалау сапасын арттырады.

Түйін сөздер: теміржол станциялары, технологиялық процесс, UML, қолдану жағдайлары, күй диаграммасы, әрекет диаграммасы, автоматтандырылған басқару

Дәйексөздер үшін: В. Лахно, Д. Жамангарин, Г. Муратбекова, А. Табылов Жобалау мәселелерін ресімдеуге және теміржол вокзалдарын дамытуды автоматтандыруға арналған модельдер мен ақпараттық технологиялар // Қазақстан өндіріс көлігі. 2025. Том. 22. № 88. 81–102 бет. (Ағыл. тіл.). <https://doi.org/10.58420/ptk/2025.88.04.007>.

Мүдделер қақтығысы: Авторлар осы мақалада мүдделер қақтығысы жоқ деп мәлімдейді.

МОДЕЛИ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ ФОРМАЛИЗАЦИИ ЗАДАЧ ПРОЕКТИРОВАНИЯ И АВТОМАТИЗАЦИИ РАЗРАБОТКИ ЖЕЛЕЗНОДОРОЖНЫХ СТАНЦИЙ

В. Лахно¹, Д. Жамангарин^{2}, Г. Муратбекова³, А. Табылов⁴*

¹Национальный университет биоресурсов и природопользования, Киев, Украина;

²Казахский университет бизнеса и технологий, Астана, Казахстан;

³Академия гражданской авиации, Алматы, Казахстан;

⁴Каспийский университет технологий и инжиниринга имени Ш. Есенова, Актау, Казахстан.

E-mail: dus_man89@mail.ru

Валерий Лахно — д.т.н., профессор, Национальный университет биоресурсов природопользования, Киев, Украина

E-mail: lva964@gmail.com, <https://orcid.org/0000-0001-9695-4543>;

Дусмат Жамангарин — PhD, Казахский университет бизнеса и технологий, Астана, Казахстан

E-mail: dus_man89@mail.ru, <https://orcid.org/0000-0002-3141-5984>;

Гульжан Муратбекова — к.т.н., Академия гражданской авиации, Алматы, Казахстан

E-mail: g.muratbekova@alt.edu.kz, <https://orcid.org/0009-0003-0005-0470>;

Абзал Табылов — к.т.н., Каспийский университет технологий и инжиниринга имени Ш. Есенова, Актау, Казахстан E-mail: abzal.tabylov@bk.ru, <https://orcid.org/0000-0002-7532-5681>.

© В. Лахно, Д. Жамангарин, Г. Муратбекова, А. Табылов

Аннотация. Исследование посвящено планированию технической поддержки, разработке автоматизированных систем управления (АСУ) и совершенствованию методов моделирования технологических процессов (ТП) железнодорожных станций (ЖС). Современным инженерам-проектировщикам необходим системный и комплексный подход к моделям, описывающим объект исследования. Модели должны отражать все аспекты функционирования будущих технических систем, включая структуру объекта и основные процессы. Цель исследования – совершенствование метода формализации ТП ЖС на основе парадигмы визуального программирования. Задачи включают создание диаграмм прецедентов (UCD), диаграмм состояний (DIS) и диаграмм активности (DIA) с использованием UML для графического представления ТП; описание объектов, исполнителей и выполняемых работ; формализацию порядка выполнения технологических операций. Результаты показывают, что применение UML позволяет визуализировать

взаимодействие объектов ЖС и исполнителей, параллельные и синхронные процессы, а также этапы выполнения каждой операции. Диаграммы DIS и DIA помогают определить состояния отдельных объектов и фазы выполнения работ. Автоматизированное моделирование повышает эффективность ТП и обеспечивает взаимопонимание между проектировщиками и операторами. Графико-аналитические модели, разработанные в средах Rational Rose и AutoCAD, позволяют автоматизировать составление планов ТП и расчет ключевых показателей функционирования ЖС. В заключение, предложенная методика обеспечивает эффективное проектирование, документирование, формализацию и визуализацию ТП ЖС, облегчая процесс принятия технических решений и повышая качество разработки АСУ и ИС ЖС.

Ключевые слова: железнодорожные станции, технологический процесс, UML, диаграмма прецедентов, диаграмма состояний, диаграмма активности, автоматизированное управление

Для цитирования: В. Лахно, Д. Жамангарин, Г. Муратбекова, А. Табылов Модели и информационные технологии для формализации задач проектирования и автоматизации разработки железнодорожных станций // Помышленный транспорт Казахстана. 2025. Т. 22. No. 88. Стр. 81–102. (На англ.). <https://doi.org/10.58420/ptk/2025.88.04.007>.

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Introduction.

The planning of technical support, an automated control system (ACS), the development of technologies for the operation of objects of the corresponding subject area (SbAr) are usually based on modeling methods. At the same time, design engineers need to have a holistic, systematic understanding of the models that describe the object of research. In turn, these models should reflect all aspects of the operation of future technical systems. By the models of the corresponding SbAr (and, in particular, railway stations, hereinafter referred to as RWS), we mean a system that is capable of simulating the structure or main aspects of the operation of the research object. Assessment aspects of SbAr modeling are associated with determining the effectiveness of the implementation of automated processes at the object (Bobrovsky, 2000: 68–71; Milinkovic, 2013: 144–157).

It is advisable to create modern graphic models using specialized software on a computer.

Reproduction of existing production processes in the form of simple diagrams and brief descriptions helps to achieve a common understanding of the current norms and operational procedures between the executor and the customer of the RWS development projects.

All of the above mentioned has determined the relevance of our research on this topic.

The Unified Modeling Language (UML) is a standard tool that allows to create diagrams of software and business processes. UML can be used to implement such operations as visualization, specification, construction and documentation of software system artifacts (Milinkovic, 2013: 144–157; Kozachenko, 2013: 47–55).

The constructive use of the UML is based on general principles used in modeling of complex objects and systems. And besides, using the UML, you can take into account many of the features of the processes of object-oriented analysis and planning of such systems and objects. The use of UML will allow solving the problems associated with documenting the system architecture, taking into account important details of technological processes (TechP) at the RWS. The UML toolkit offers its own language for formulating instructions for the RWS systems and provides tools for modeling of work during the planning and versioning phase of the RWS project.

The organizational structure of the RWS is a set of links (employees, structural division) and connections between them. A common method for representing the structure of the RWS is an organizational chart.

The organizational chart shows the place of each position and each division in the overall structure of the RWS and illustrates the distribution of powers and responsibilities.

The initial stage of planning is the synthesis of a use case diagram (hereinafter UCD). The basic elements of UCD include:

- use case (technological process - TechP);
- the executor and service object at the RWS, see Fig. 1.

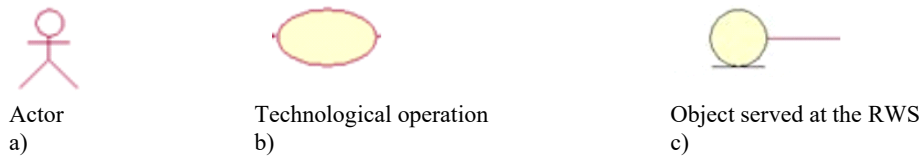


Fig. 1. Examples of conditional images used in the UML of UCD during the design of the RWS system

UCD in UML is used to graphically describe the general features of the actions of objects and systems. However, at this stage, the modeling does not consider its internal structure. For example, UCD can be useful for describing the reception of a train at the RWS, obtaining information about the clients' solvency for a freight RWS, displaying data on the arrival of a train on information boards of the station, etc. On the UCD the TechP is depicted as an oval. In the field of the oval or under it, there is a phrase that explains the precedent (technological operation) (Kozachenko, 2013: 22–28; Voevoda, 2009: 169–174)

As previously shown in works (Szűcs, 2001: 133–142; Bobrovskiy, 2014: 57–65), the creation of UML of UCD for the RWS systems is necessary for:

- (at the initial stages of the RWS planning) defining the general boundaries and a context of a specific project (taking into account its features);
- (at the initial stages of the RWS planning) the formulation of general instructions for the algorithms of actions at specific RWS, depending on its features - freight, passenger, etc.;
- creation of the initial conceptual model of the RWS. At the subsequent stages, detalization is required in the form of logical, physical and software models;
- preparation of initial design documentation for the interaction of the RWS designers with its customers and executors of specific types of work.

Therefore, the literature review showed that many aspects of the use of UML diagrams to describe the functioning of complex systems on the railway transport are not fully disclosed. This led to the main goal of our research, which is to improve the method of formalizing the description of the TechP at the RWS based on the visual programming paradigm for simulating the RWS operation.

Materials and Methods.

Modeling the graphical representation of the TechP at the RWS by means of the UML visual language is achieved by creating diagrams of state (hereinafter DIS) and activities of various degrees of detalization (Kirichenko, 2017: 142–154).

For example, for a freight RWS, there was created a UCD, shown on Fig. 2. This diagram shows the technical processing of the train upon arrival at the RWS.

In order to achieve a clear understanding of the TechP at the RWS, it is necessary to highlight its main components: objects requiring the actions of the executor; set of works; persons performing work.

A certain number of operations (works) provided for by the TechP are performed by each object at the RWS, and the execution of technological operations (hereinafter referred to as TechO) at the station is provided by executors (shunting locomotives, marshalling yard, etc.). Each TechO should be carried out by executors of a strictly defined specialization (for example, a technical inspection team (TIT) performs an inspection of wagons, the formation of a train - a shunting locomotive and a marshalling yard, etc.). At the same time, the executor of a particular specialization can perform several different operations (for example, the signalman performs the fastening of the train and the cleaning of brake shoes) (Bobrovskiy, 2014: 57–65.).

Modeling of the TechP at the RWS is achieved by synthesizing UML diagrams of state and activity, respectively, DIS and DIA.

DIS in UML describes the process of changing the states (completed works of TechP) of only one object. In this case, a change in the state of an object (train, group of wagons) can be caused both by internal processes and due to the action of external pathogens. The main purpose of this UML diagram in the formalization of the TechP at the RWS is to describe all possible sequences of work, which together will form the options for the actions of objects during their stay at the RWS.

DIS in UML notation is essentially a graph of a special kind. This graph can be represented as a certain automate. The vertices of the graph will be the work performed at the RWS. Besides, such a graph contains some other types of automaton elements. These elements will be rendered as adequate graphical notations (conventions) in a specific modeling environment. The arcs of the graph are designed to represent transitions from state to state. The corresponding states describe the completion of the work and the transmission of the necessary signal to the executors of the next work in the diagram. DIS can be nested within each other (Kozachenko, 2013: 47–55.).

The diagram of activity in the UML language (hereinafter referred to as DIA) reflects the TechP typical for the RWS. The diagram of activity is also a graph that represents a certain automate, but it has the following differences:

- on DIA, both states can be distinguished, and actions can be shown, and actions, in turn, can be represented as a new DIS or DIA, receiving nested diagrams;
- DIA has the operator of "choice" in the set of tools for presentation;
- on DIA, you can show the parallelism of the processes performed at the RWS;
- on DIA there is a possibility of presenting the processes of synchronization at the RWS.

When formalizing the processes of the RWS, the DIS describes the change in the phases of servicing objects during the execution of TechP and work with individual objects (for example, disbandment of a train, etc.). In this case, changes in the states of objects can be caused both by internal processes and due to the action of external factors. An example of a diagram of state is shown on Fig. 3.

When describing the RWS, there are used DIA to describe technological operations performed with objects within individual states. On DIA, the executors of a certain specialization correspond to separate paths, operations - states of activity, cause-and-effect relationships between operations - transitions between the states.

Additional elements of the DIA are decision nodes and nodes of merging (merging and splitting), as well as points of input and output of signals.

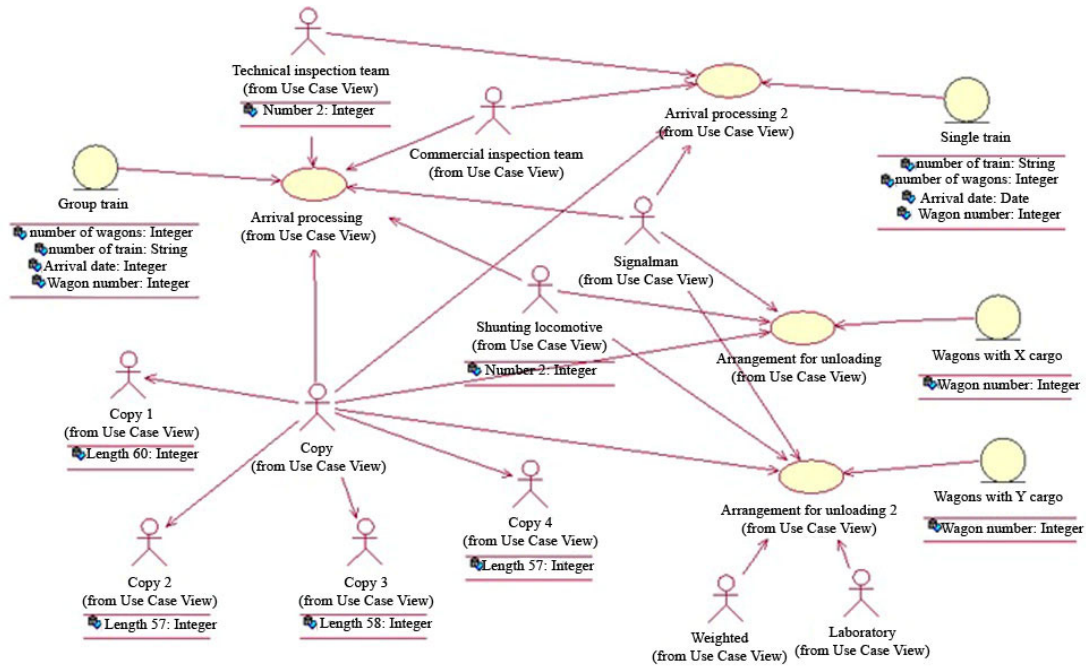


Fig. 2. UCD for freight RWS

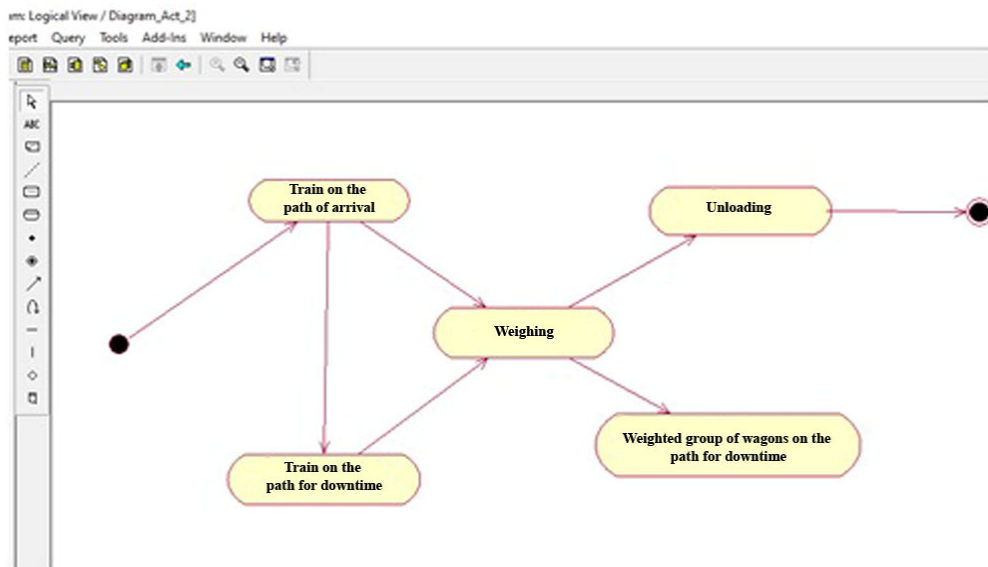


Fig. 3. DIS for servicing train wagons with grain

An example of DIA for servicing a train upon arrival is shown on Fig. 4.

For the TechP, modeled using the diagram of activity, it is typical to reflect the parallelism of the activities of the objects involved in the TechP data. For TechP, in parallel, there are involved the path on which the train arrived, a shunting locomotive, a signalman, a technical inspection team, a commercial inspection team (CI). The states of the graph show the work that is performed while processing the train. The execution time of operations is indicated by the label "do".

Work status labels of DIA indicate: *entry* – work, or an incoming document required to perform work at the time of entering the state;

exit – work (output objects) performed at the time of exit;

do – work done throughout the allotted time;

event – separate action performed while the system is in a particular state.

Such "events" must be ordered in time.

Only one transition can be made from specific states at given moments in time. In this way, it is ensured that a dual result is avoided for any event. There are two special types of states: 1) entry; 2) exit.

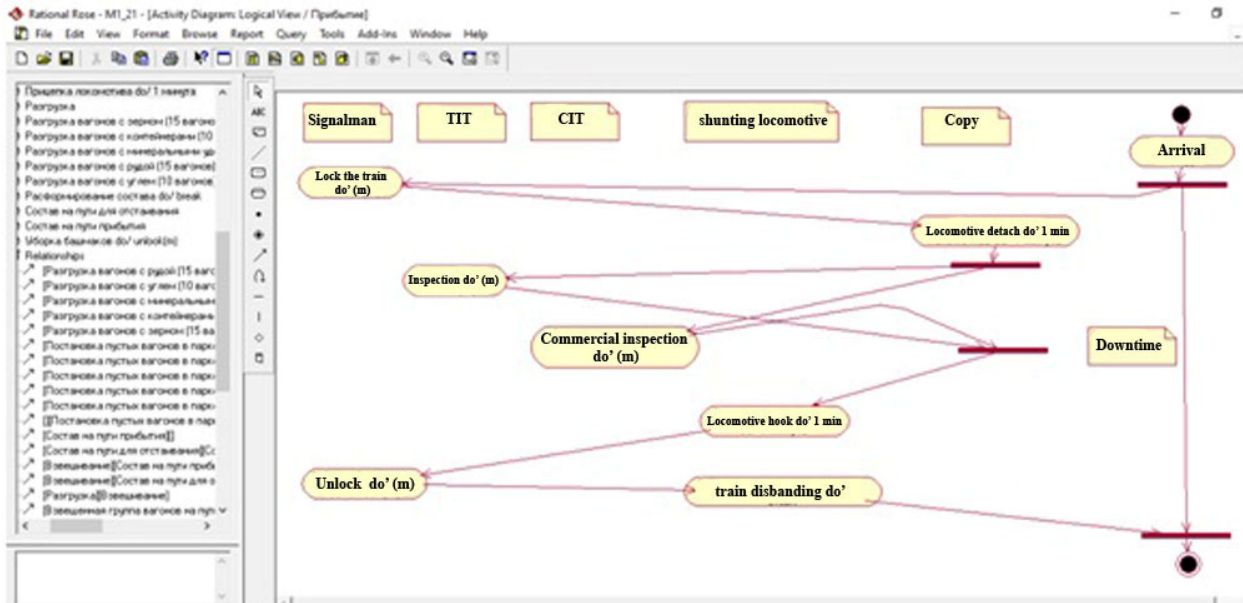


Fig. 4. DIA for processing a train at the RWS

Any action that is associated with an entry event is performed when objects enter the appropriate states. Exit events are executed when objects leave these states.

In the behavior of the train in the system we can distinguish actions that are reflected by transitions and actions of the displayed states. Although both are processes that are implemented, as a rule, by some "executors" of the TechP, they are interpreted in different ways. Transition activities are considered as instantaneous or continuous. State works can take quite a long time. Work may be interrupted by some external event.

In addition, there are two special transitions on the DIA: connection position and distribution position. This representation in the diagram shows the execution of parallel works. The Fig. 4 shows that in the technological process of processing a train upon arrival, two parallel works are performed: maintenance of the RS and its inspection. The Fig. 4 shows both operations performed in parallel and operations separated by connection and distribution positions. Each such action is characterized by a function of execution or time of completion of work. For parallel work, we can create diagrams of various degrees of detalization. For the diagram of the completion of the train stay in a state with parallel works, it is necessary that all parallel works finish their execution simultaneously. This is the main condition for moving to the next work in the diagram.

The graphical-analytical representation of TechP is visual for the development, understanding and subsequent creation of new functional blocks of TechP at the RWS and reduces the time spent for its study. It is also possible to present TechP of various degrees of detalization. After receiving a schematic representation of the TechP in the IBM Rational Rose environment, the user receives a text file describing the process. The file can be used in the analysis of TechP and to perform calculations of the formalization indicators of the TechP at the RWS.

Let us note that the scientific novelty and practical significance of these studies lies in the methodology proposed for creating mathematical models of the TechP at the RWS using the unified UML language.

This approach makes it possible to reduce the time spent on creating a model of the RWS operation and to present a specific technological process for each station, as well as to specify, design, document and formalize the technical process, to develop work sequence diagrams of various degrees of detalization.

The design of the TechP at the RWS and the creation of their input model is characterized by a high level of interaction between the design engineer and the computer. This stage is characterized by the creation of an effective graphical representation focused on the visualization and formalization of the TechP at the RWS.

At this stage, a set of formalization diagrams of the TechP at the RWS is represented as a set of graphic objects Q_{ex} . The following types of objects are highlighted:

D_p – use case diagram; D_{Sch} – DIS; D_{act} – DIA.

Each of the given diagrams is associated with a set of tools for their graphical display.

Use-Case Diagram (UCD) is presented as a directed graph and is described by the following set:

$$D_p = \{I_d^p, V, E, f_b, f_e, vt\}, \quad (1)$$

where I_d^p – use case diagram identifier; V – list of vertices of the graph; E – list of transitions; f_b – initial vertex of transition; f_e – final vertex of transition; vt – transition type function.

The initial and final vertices of the transition are defined as:

$$f_b : E \rightarrow V, f_e : E \rightarrow V, \quad (2)$$

$$vt : V \rightarrow VT, \quad (3)$$

$$VT = \{actor, entity, function\}. \quad (4)$$

Information about the diagram is contained in the file that describes the resulting model and contains:

list of vertices of actors - V_a . Here V_a describes the vertex that its identifier defines, the field *quid* and *stereotype* are defined by the function VT . Each actor can have a list of parameters. The list of parameters is defined as *class_attribute_list* and each attribute is defined as *ClassAttribute*;

list of vertices – functions V_f that are defined by identifier, name, by the field *quid* and additional urgent information

list of edges E . An edge is determined by the link type *Association*, identifier, name, by the field *quid*, field *roles*, a list of two objects *Role*. For each edge, there is a list of two vertices, final and initial.

The State Chart Diagram is described by the following set:

$$D_{Sch} = \{I_d^{Sch}, V, E, V_{start}, V_{stop}, f, f_b, f_e\}, \quad (5)$$

where I_d^{Sch} – state chart diagram identifier; V – set of vertices (states); E – list of transitions; V_{start} – vertices of the initial state of the diagram; V_{stop} – vertices of the final state of the diagram; f_b – initial vertex of transition; f_e – final vertex of transition; f – runtime function.

The initial and final vertices of the transition are defined as:

$$f : V \rightarrow R. \quad (6)$$

Information about a diagram, which is displayed in an identical file, consists of structures:
list of vertices - names of works (V). The parameter (V) describes the vertex, which is defined by its identifier - fields *quid* and *type* containing a line "StartState". The attributes are presented in the field *actions* in the form of a list, each element of this list is an attribute of the vertex, the name of which is indicated in the field *ActionTime*. The field *ActionTime* contains the attribute of the operation execution at the top of the graph of values of the set W ;

list of edges (E). The list of edges (E) is defined by a field *transitions* and a service word *list transition _list*, each edge is defined by a structure *objet State _Transition*. Each edge is identified by an edge identifier, by the field *label*. The final vertex of the edge is defined by the service word *supplier* and *supplier_quidi* - the identifier of the final vertex. The initial vertex of the edge is determined by the service word *client* and *client _quidu* - the identifier of the initial vertex. The edge is characterized by the execution of the transition action and is determined by the field *Event*. The field *Event* is described by the name of the line type, by the identifier and the service message defined by the field *sendEvent*. The message has its own identifier;

vertices of the initial state of the diagram are presented in the file by the field *objet State* and are indicated by the line "\$UNNAMED\$ 0". The vertex is defined by the field *type "StartState"* as the initial state for the graph;

the vertices of the final state of the diagram are presented in the file by the field *objet State* and are indicated by the line "\$UNNAMED\$ 1". The vertex is defined by the field *type "EndState"* as the initial state for the graph.

DIA is described by the following structure:

$$D_{act} = \{I_d^{act}, V, E, V_{start}, V_{stop}, f, f_b, f_e, S\}, \quad (7)$$

where I_d^{act} - DIA identifier; V - set of vertices (states); S - list of parallel t existing actors; E - list of transitions; V_{start} - vertices of the initial state of the diagram; V_{stop} - vertices of the final state of the diagram; f_b - initial vertex of transition; f_e - final vertex of transition; f - runtime function,

$$V = VA \cup VC \cup VD \cup VG, \quad (8)$$

where VA - set of vertices of the diagram transition; VC - set of vertices of the conditional transition; VD - set of vertices of separation point transition; VG - set of vertices of connection point transition.

The sets VA, VC, VD, VG are pairwise disjoint. The set VD can have only one entry to the edge, and it takes the following form:

$$(\forall v \in VD) Card \{e \in E, f_b(e) = v\} = 1, \quad (9)$$

$$(\forall v \in VG) Card \{e \in E, f_e(e) = v\} = 1, \quad (10)$$

$$s : V \rightarrow S. \quad (11)$$

Information about the diagram that is displayed in the model file consists of such structures:
 a list of vertices - the names of works or transitions of a special type (transition type *State* ; conditional transition; the transition of a separation point of a type *SynchronizationState* is determined by an identifier - the field *quid* . A separation point is characterized by a list of separation edges. The list is defined by the field *transitions* (*list transitions_list* and a list of edges with a field *object State_Transitions*). Each vertex is defined by the field *quid* , by the edge identifier and a list of edges of the initial state and final states. The initial edges are described by the field *supplier* and contain links to the initial state of the graph and the final states described by the field *client* . All transitions have their identifiers and data transfer events during transitions;

the transition of the connection point of the type *SynchronizationState* is determined by the identifier - the field *quid* . A connection point is characterized by a list of connection edges. The list is defined by the field *transitions* (*list transitions_list* and a list of edges with a field *object State_Transitions*). Each vertex is defined by the field *quid* , by the edge identifier and a list of edges of the initial state and final edges. The initial edges are described by the field *supplier* and contain links to the initial state of the graph and the final states described by the field *client* . All transitions have their own identifiers and a data transfer event occurs when transitions are made;

identifier *V* - describes the vertex that the list of edges *E* defines. The list of edges is determined by the field *label* , the final vertex of the edge is determined by the service word *supplier* and *supplier_quidu* - the identifier of the final vertex, the initial vertex of the edge is determined by the service word *client* and *client_quidu* - the identifier of the initial vertex. The edge is characterized by the execution of an action during the transition and is determined by the field *Event* . The field *Event* is characterized by the name of the line type, by the identifier and a service message, which is defined by the field *sendEvent* , and the message has its own identifier;

vertices of the initial state of the diagram are presented in the file by the field *object State* and are indicated by the line "\$UNNAMED\$ 0" . The vertex is defined by the field *type "StartState"* as the initial state for the graph;

the vertices of the final state of the diagram are presented in the file by the field *object State* and are indicated by the line "\$UNNAMED\$ 1" by the field *type "EndState"* . The vertex is defined as the initial state for the graph;

the list of executors *S* is defined in the diagram as a section *partitions* (*list Partitions* , the list of executors is initiated by the field *object Partitions* , the identifier *S* is determined by the field *quid* , the field *class* contains data on the link of belonging to the UCD entity, the field *persistence* defines the type of entity.

In general, the presented model Q_{BX} is a list of graphical objects that are written in a specific order.

Therefore, in the input model, there is proposed a description of the graphical-analytical representation of the RWS technological process, while the description is made in the IBM Rational Rose environment, which allows at the next stage to proceed to the stage of designing interfaces for the ACS or IS of RWS.

The considered diagrams of states and activities prescribe the rules for the functioning of the RWS model. A station is a collection of automate interacting with each other in discrete time.

Some automate exist all the time the model is running (permanent), and some are created and destroyed in the process of operation (temporary).

Permanent automates correspond to the RWS resources (shunting locomotives, paths, a technical inspection team, etc.). Trains correspond to temporary automate.

In a relationship with a temporary automate, a permanent automate accepts their requests for resource allocation or satisfies the request if it has a sufficient number of resource units at its disposal. At the end of the work of the temporary automate, the resource that was used to perform the work is returned to the permanent automate.

A temporary automate is created upon activation of a node or vertex of the type UCD function. In this case, the request to perform the following activities, which are described by the corresponding diagram of state, can be initiated multiple times. The state of the automate is set by the active vertex of DIS. Upon completion of the work corresponding to this vertex, the automate moves along the edge from this vertex to a new active one.

If several edges come from one vertex, then the transition occurs along one of them. The rule for selecting an edge is a vertex attribute. The work of the automate ends and the temporary automate is destroyed upon reaching the final state.

An ambiguous task is to simulate the operation of the RWS when the temporary automate is at one of the vertices of the diagram of state. DIA is assigned to this vertex, the vertices of which describe groups of works, and some works can be performed in parallel.

The internal model should be the basis for the development of effective methods of functional modeling of the RWS operation. An internal model is created automatically based on the input model using the models developed in the article.

Based on the UCD, there are formed lists of executors E and object templates D , each element of which is described by the following data structure:

executor $e_r \in E$

$$e_r = \{i_e, P_e\}, \quad (12)$$

where i_e – executor identifier; P_e – list of executor parameters; object template $d_y \in D$

$$d_y = \{i_d, P_c, E_d, A\}, \quad (13)$$

where i_d – object template identifier; P_c – set of object properties with default values; E_d – list of executors required for object maintenance; A – state machine (SM), describing the procedure for performing technological operations (TechO) with an object.

The formation of the SM describing the procedure for performing TechO with an object is based on the DIS. To describe the automate, there is used an approximate parametric graph whose vertices correspond to the states of the SM, and the arcs correspond to transitions. Vertex incidence lists are used to describe the structure of the graph. Moreover, each state of the automate is described by the structure

$$s_a = \{i_a, R, X\}, \quad (14)$$

where i_a – automate state identifier; R – TechO of the object in this phase of TechP; X – list of transitions.

Each element of the list of transitions is described by the structure

$$x_q = \{z, i_a, f_p\}, \quad (15)$$

where z – input signal; f_p – transition function.

The technology for object maintenance in certain phases of TechP is formed on the basis of the DIA. The technology model is a directed graph $T = \{O, G\}$. TechO corresponds to the vertices of the graph $o_j \in O$, as well as the points at the beginning and at the end of the technological process, branching, merging, and decision making.

In particular, TechOs are represented by structures

$$q_i = \{i_q, i_{qn}, f_s, f_d, f_e, E_q\}, \quad (16)$$

where i_q – vertex identifier; i_{qn} – next vertex identifier; f_s, f_e – functions performed respectively at the beginning and at the end of the operation; f_d – function that determines the duration of the operation; E_q – list of executors.

Each element of the list of executors is determined by the specialization of the executor required to start the work and by a parameter that indicates the order in which it is released from the operation.

As noted earlier, at the upper levels, the RWS models are collections of interrelated states. Each of these states can be represented as a sub-automate. Accordingly, the work of such sub-automate as part of the RWS will be conducted in parallel. The states that characterize the RWS operation will correspond to the various technological resources at the RWS. And in addition, the RWS scheme is supplemented by such objects as a dispatcher (the main function is the appointment of TechO executors) and an incoming flow generator (IFG) of service requests.

The work of the dispatcher and the IFG can be described using external algorithms.

From the point of view of representing the model of the RWS functioning as a finite-automaton model, both the dispatcher and the IFG can be represented as an automate with a single state. That is, the dispatcher and IFG will generate signals that will control the RWS operation. The interaction of the SM with each other is realized by means of signals automates generated during the operation.

For a typical structure, a SM of the RWS will respond to the following signals:

- 1) "Perform (object, type of work)". Such a signal is received by the dispatcher, processed by him, and then the executors are appointed;
- 2) "Execution (object, type of work, executors)". Executors (or executor) are SMs. The SM data will correspond to some units of technological resources of the RWS;
- 3) "Completion (object, type of work)". This signal serves to synchronize messages from executors about the completion of work;
- 4) "Time". A signal τ that is needed to measure the time intervals that are allocated for the execution of work.

As an example, let us consider the diagram of changes of the SM states for a shunting locomotive (technological resource, see Fig. 2), which is assigned to arriving trains, see Fig. 5.

The operations shown on fig. 5 can be represented as a graph describing the operation of a state machine. The Figure 5 gives a representation in the form of a graph for an automate (shunting locomotive), which transports wagons during their unloading.

In the course of modeling the operation of such a state machine, the task of obtaining information on the duration of the execution of specific TechOs is of paramount importance.

From the point of view of the description of the SM, it is necessary to assess the possibility of transition from the state A for which the objects can perform TechO with

duration t into a certain state B . If the ultimate task is to automate the object control processes at the RWS, then obviously it is necessary to receive a signal that actually initiates the transition.

Next, you need to decide on the place where you should store information about the transition time. Within the framework of modeling with UML diagrams of the TechP at the RWS, we assume that all incoming signals will be external in relation to the initial state of the object. Tracking the time of work performed by a shunting locomotive can be applied using two approaches:

The first approach involves processing the SM states for each moment of event processing τ ;

The second approach assumes the need to process SM events, which is characterized by a complex structure.

When using each of the above approaches, we mean that it is necessary to create SM based on its rules for transitions between the states.

Therefore, if we take the rules for creating SM as a basis, applying the first approach, then it is necessary to process its states at every moment τ . In this case, it is necessary to store in the ACS or IS time arrays describing the change in the states of the SM and the signals generated at the moment τ . It is also necessary to check the transitions to the following states of the SM. Consequently, the SM operation will continue until at a certain moment in time τ the SM state will correspond to the position ("Train sent").

The second approach is also generated on the basis of SM states. But in this case, it is more important to create SM based on the general principles of deduction - from general provisions to detailed ones. In addition, the SM is controlled through external signals. It is also typical for the second approach that transitions from one state to another will be accompanied by quantitative data.

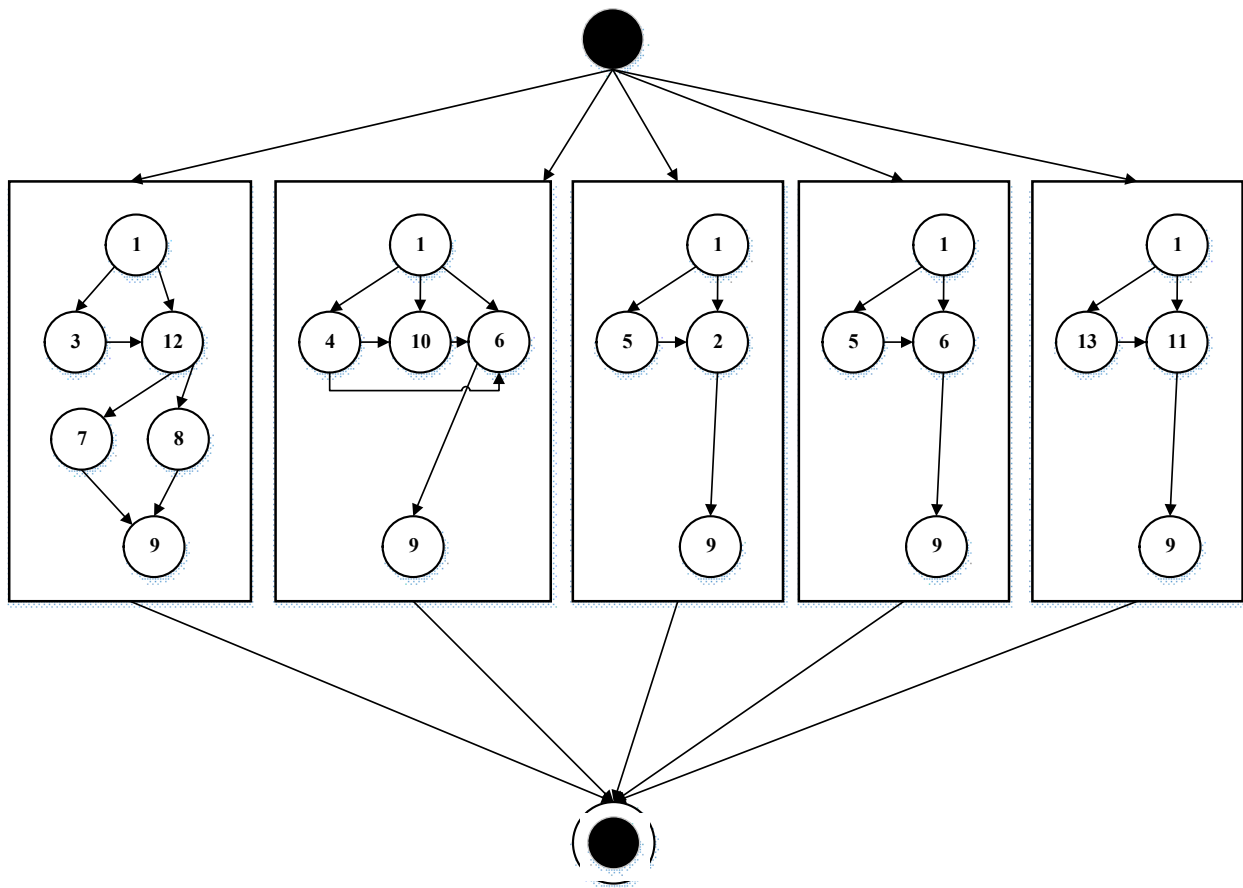
Using these considerations, there was created a UML diagram for the SM, which shows the TechP for unloading wagons with different cargos, see Fig. 6.

Thus, this SM describes the model of the TechP at the RWS in UML notations and, in fact, is a set of hierarchical automates. In this case, each SM will correspond to a unit of the RWS technological resource (for example, for the resources shown on Fig. 2: path, shunting locomotive, etc.). Separate elements of the SM will be required to simulate the activities of the dispatcher and to develop an ACS or IS for the RWS.

We also note that the strategies of action of these elements will be described by a set of parameters, which at the next stages of designing an ACS or IS for the RWS can be considered as variables for the multi-criteria optimization task of the RWS operation. At first approximation, it is better to describe the activities of the dispatcher using the simplest strategy of the SM operation. With this supply, the wagons undergoing processing will correspond to the parameters of the SM state.

The models of the technological process described using UML diagrams, and the methodology for creating RWS models as hierarchical SMs are well suited for the development of object-oriented applications for ACS and IS of the RWS.

However, in some cases, it is advisable to perform the automation of the development process and analysis of the technology of its work using simpler modeling methods. It is also advisable to do this if we are not talking about the design and implementation of object-oriented software products, but about the need to complete the stage of documenting work based on traditional design automation tools, for example, using the AutoCAD package. Such a supply of models focused on automated design systems of the RWS will increase the efficiency of human-machine interaction and provide automation of the processes analysis at the RWS, based on the minimum information entered by the operator. In such situations, it will be quite effective to use the method of representing the processes of the RWS functioning as graphical-analytical models.



SM states:

1. Arrival of the locomotive.
2. Arrangement of a group of wagons (GrWag) for unloading at VO-1.
3. Arrangement of GrWag on the path of the grain park.
4. Arrangement of GrWag on the path of the coal park.
5. Arrangement of GrWag on the path of the ore park.
6. Arrangement of GrWag for unloading at VO-2.
7. Arrangement of GrWag for unloading at the wagon unloading station No. 1.
8. Arrangement of GrWag for unloading at the wagon unloading station No. 2.
9. Arrangement of empty wagons on the path of the receiving-departure yard
10. Arrangement of GrWag for defrosting
11. Arrangement of GrWag for containers unloading.
12. Performing laboratory analysis of cargo.
13. Arrangement of GrWag on the path.

Fig. 5. SM state change diagram for a shunting locomotive assigned to arriving trains

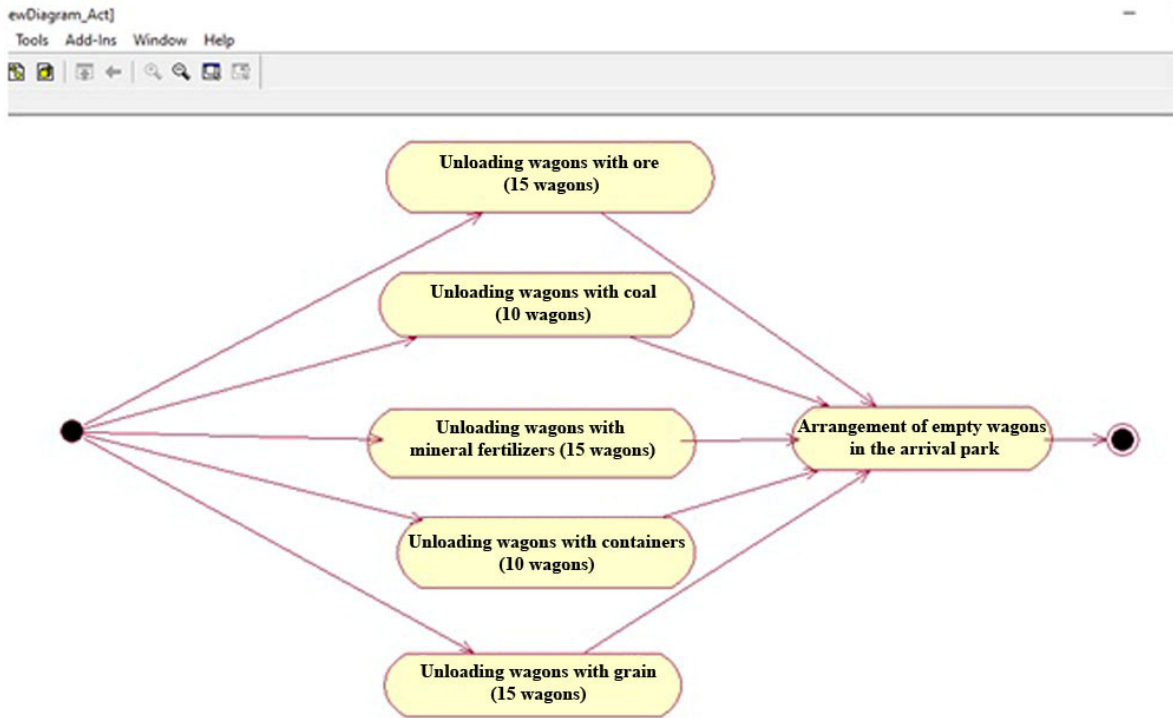


Fig. 6. UML diagram for a state machine that shows the TechP for unloading wagons with different cargos

In addition to SM using as a tool for modeling the technical equipment of the RWS, we can also use the theory of graphs, since for a number of cases, for example, when we are talking about modeling a special technical equipment of the RWS, this approach will be less laborious than developing a state machine. In this case, individual technical means (TechM) used at the RWS to automate TechP and the corresponding executors will be positioned as the vertexes of a tree (graph). The arcs of the graph will correspond to connections. Then the set of tree vertices (V) can be divided into such subsets (V_r, V_g, V_s). The vertexes V_r will represent the leaves of the tree to which the TechM, as well as the executors, will correspond to. As in the case of using the notations of UML diagrams or SM, the executors mean separate paths, shunting locomotives, TechM for automation of loading and unloading operations, technical inspection teams, commercial inspection team, etc.). The parameter V_s will correspond to the root of the tree, which forms the RWS. The nodes V_g will correspond to the TechM group used at the RWS. Moreover, the nodes V_g will be associated with the TechM groups, which are grouped according to certain principles (cargo handling points, a park of shunting locomotives, a laboratory, etc.). With this formulation of the problem, each vertex must be associated with its own list of parameters. For example, the vertex type (line, line group, RWS) will determine the parameter t_b . In order to determine the structure of the tree as a whole, it is necessary to assign a vertex v for each of the vertices u_b , i.e. $u_b \rightarrow v$. The rest of the parameters will be determined by the type of the vertex.

Then, the vertices $v_r \in V_r$ in the computer memory can be represented as follows:

$$v_r = \{t_b, u_b, s_r, n_r, y_r, h_r, z_r\}, \tag{17}$$

where s_r – vector of specialization of the executor (for example, TechM), which correspond to the types of operations performed; n_r – executor name (TechM); y_r, h_r, z_r –

respectively ordinate, height and visibility of lines in daily plans-schedules of the RWS operation.

The vertices $v_g \in V_g$ in the computer memory can be represented as follows:

$$v_g = \{t_b, u_b, n_r, w_g\}, \quad (18)$$

where n_r – the name of the groups of executors (TechM); w_g – width of a specific group (TechM) on daily plans-schedules.

The vertex v_s is represented by the structure:

$$v_s = \{t_b, u_b, p_s, s_s, w_s\}, \quad (19)$$

where p_s – period for which the simulation is performed; s_s – horizontal graph scale; w_s – column width for line title.

The RWS functioning can be represented as a process of station objects maintenance by individual executors. Such objects can include: a train, a wagon, a group of wagons, a locomotive, etc. Then, the object maintenance model will be a directed graph of the form $G(O, L)$. The vertices of the graph will correspond to individual TechOs, which the executors perform during the maintenance. The arcs of the graph will correspond to the cause-and-effect relationship between TechOs. The structure of such a graph can be represented as incidence lists. Then, each vertex (o) is associated with a list (or lists) of previous (p_o), as well as subsequent (n_o) vertices.

The implementation of individual TechOs in some cases requires the involvement of several executors (several TechMs). For example, when using a marshalling yard, it is necessary to use the following TechMs: arrivals paths, thrust paths, shunting locomotive, signalman. Then, the vertices (a) and (c) have the form in which we will consider ($a \rightarrow c$) and ($c \rightarrow a$) as the simultaneous involvement of several executors (TechMs) to perform one TechO.

Then, each memory operation in a computer can be represented as follows:

$$o = \{p_o, n_o, t_o, p_o, b_o, v_o, w_o, l_o, d_o\}, \quad (20)$$

where t_o – TechO type; b_o – identifier of the object for which this TechO is implemented; v_o – identifier of a specific TechO executor; x_o, w_o – respectively, the beginning and duration of TechO; l_o – pointer to a modeled point on the graph; d_o – vector of additional parameters (this vector depends on the specifics of TechO).

Additional parameters that determine the specifics of TechO may be as follows: train number; the number of wagons, etc. Constants can be used as parameters of operations. You can also use a parameter b_o . The duration of a TechO can be represented as a constant, or you can apply special functional dependencies that were obtained for different TechOs.

The Figure 7 shows an example of the representation of the technical equipment of the RWS in the form of a graph for the case when it is necessary to simulate the transit trains maintenance at the RWS.

Individual operations on the graph are displayed as corresponding icons. Combining operations imported, for example, from a *.mdl Rational Rose file into appropriate groups, is implemented using the lists p_o, n_o . Then the TechO group can be assigned to the object b . This is done by specifying its identifier b_o . Each of the objects can be associated with a list of parameters d_b .

By changing the parameters of objects, it is possible to provide a synchronous change of the corresponding parameters of all TechOs related to the object. Then the TechO of objects at the RWS can be formalized using a directed graph of the form $H(T, L)$.

A graph $H(T, L)$ is like a graph $G(O, L)$. In the graph $H(T, L)$, it is assumed that T – the graph of vertices or templates of TechO, and L – arcs (connections) between the vertices. Individual operations can then be represented as follows:

$$t = \{p_t, n_t, t_o, s_t, x_t, w_t, l_o, d_t\}, \quad (21)$$

where s_t – specialization of the executor who performs TechO; x_t – conditional start point of TechO; w_t – duration of TechO; d_t – vector of default values of additional parameters that depend on the TechO type.

Creation of a technology description used at the RWS can be performed in separate editors that are suitable for the RWS technologist. This can, for example, be AutoCAD, KOMPAS, etc. Moreover, the list of works can be easily imported from UML diagrams files, for example, below there is a fragment of the listing of the list of works at the RWS, imported from a *.mdl file describing the UML diagram for the TechP of the state machine for unloading wagons with different cargos, see fig. 6:

```
(list States
(object ActivityState "Unloading wagons with coal (10 wagons)"
(object ActivityState "Unloading wagons with ore (15 wagons)"
(object ActivityState "Unloading wagons with mineral fertilizers (15 wagons)"
(object ActivityState "Unloading wagons with containers (10 wagons)"
(object ActivityState "Unloading wagons with grain (15 wagons)"
object ActivityState "Arrangement of empty wagons in the arrival park")
(object ActivityState "Train on the path of arrival")
(object ActivityState "Weighing")
(object ActivityState "Weighted group of wagons on the path for settling")
```

Executors' identifiers (v_o) are replaced by their specializations from the list (s_r). If new objects are added to the UML diagrams, then the computer will select the executor, guided by their specialization. This will ultimately allow minimizing the duration of work on specific TechO.

Taking into account the fact that maintenance at the RWS is performed using standard technologies, the formalization of the TechP description can have a positive effect, which consists in reducing the time for developing and modifying maintenance schedules of TechOs. The developed graphic-analytical model, implemented, for example, in AutoCAD, see fig. 8, makes it possible to automatically determine part of the indicators of the RWS functioning.

For example, it is possible to automatically carry out calculations for the load factors of TechO executors, as well as for TechM (6)

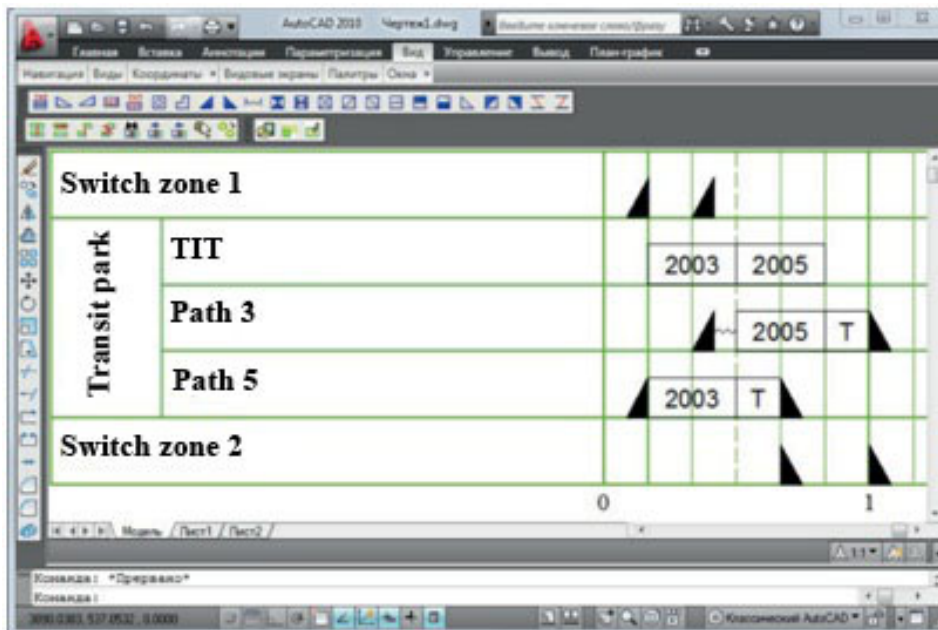


Fig. 8. An example of integrating work lists from a state machine graph or UML diagram into the AutoCAD design system

The formal representation of the processes of the RWS functioning will also allow automatically generating files with the initial data for the simulation of processes at the RWS.

Taking into account the fact that the development of schedule plans is usually performed by a RWS technologist, the use of a schedule plan, as well as tools for design automation packages such as AutoCAD, should not cause any particular difficulties. In fact, in this case, the development of a plan-schedule resembles the usual process associated mainly with adding, removing and modifying icons on the plan. At the same time, the developed interface in any computer-aided design environment, for example, AutoCAD, will be intuitive for RWS technologists and will not require additional training, unlike the skills of designing UML diagrams. Let us note that formalization of the process of drawing up schedules, based on the models described in the chapter, will significantly reduce the time for creating such schedules. Also, this approach, in conjunction with the possibilities of combining the Rational Rose and AutoCAD package tools, increases the capabilities of both technologists and programmers of ACS and IS of RWS to make adjustments to already existing diagrams and plans-schedules of TechP, depending on the specific circumstances at the RWS, as well as coordination of modifications of operations with objects. The presence of models makes it possible to automate the processes of calculating the main indicators of the RWS operation, to visualize the results of calculations on the loading of TechM, downtime, etc.

Formal representation of the processes of the RWS functioning will also allow to automatically generate files with the initial data for the simulation of processes at the RWS

Conclusion.

The method of formalizing the TechP description of the RWS based on the visual programming methods for the simulation of the RWS operation has been improved.

The UML diagrams of state and activity have been adapted to represent the RWS operation technology. When formalizing the RWS description, the diagram of state is submitted taking into account the specifics of the description of the change in the phases of objects maintenance in the process of performing the maintenance of individual objects.

It is shown that the diagram of state for RWS is a state machine (SM) that models the sequence of changing the states of an object. The detalization of the behavior of objects serviced at the RWS has been completed. Detalization is done using the diagrams of activity. The diagrams of activity are used to formally describe the TechO with objects and executors of work at the RWS.

There is proposed a methodology for creating RWS models as hierarchical SM. It is proposed to visualize state machines taking into account the features of the RWS in the form of Harel diagrams (UML state diagrams).

On the basis of the use of hierarchical SM, there have been improved the methods of functional modeling of the RWS.

Based on the use of hierarchical spacecraft, the methods of functional modeling of the LDS have been improved.

It is shown that functional UML models of the RWS can be used to analyze changes in structures, technical equipment, and technologies for work technology at the RWS. Moreover, these models can be used to assess the compliance of the technical and technological equipment of the RWS with both the existing and future scope of work at the station.

It is shown that signals about the beginning and the end of various works at the RWS allow synchronizing the operation of the model. It is shown that the use of actions to describe the states and signals will make it possible to use external algorithms for the design of ACS of the RWS. This is especially important when the application of the finite-automaton formalism is difficult or inconvenient. The chapter discusses specific examples of the SM description that simulate the operation of the RWS for the TechO of the wagon acceptance.

The described methodology was implemented using the UML in the Rational Rose environment. This significantly reduced the complexity of the work on the synthesis of the corresponding models of the RWS.

It is shown that in the course of the development of effective automated systems for analyzing the technologies of the RWS operation, it is required to increase the efficiency of man-machine interaction when creating models of RWS processes. It has been established that the structure of the model of the RWS functioning should ensure the implementation of possibilities for the automatic analysis of RWS processes based on the minimum human participation in the process of entering information. The achievement of the solution to this problem is facilitated by graphical-analytical models of the RWS.

It has been established that the graphical-analytical models of the RWS operation are capable of providing a high speed of human-machine interaction and reducing the barriers between the cognitive perception by the engineer of the graphic models of TechP at the RWS and the UML notations, which are designed for programmers. There has been developed a model of graphic-analytical description of the RWS, in which, in contrast to the existing ones, there was added the ability to automatically correct the list of objects and technologies, as well as additional links between TechOs, which will reduce the load on designers during the development and analysis of technologies of the RWS functioning. Moreover, the list of objects and technologies can be automatically imported from UML diagram files.

REFERENCES

- Bobrovsky, 2000 — Bobrovsky V.I. Functional modeling of railway stations in the simulators of operational dispatching personnel. *Mathematical Modeling*. — 2000. — 2(5). — Pp. 68–71. [Eng.]
- Bobrovskiy, 2014 — Bobrovskiy V. Functional simulation of railway stations on the basis of finite-state automate. *Transport Problems*. — 2014. — 9(3). — Pp. 57–65. [Eng.]
- Giua, 2008 — Giua A. Modeling and supervisory control of railway networks using Petri nets. *IEEE Transactions on Automation Science and Engineering*. — 2008.— 5(3). — Pp. 431–445. — <https://doi.org/10.1109/tase.2008.916925> [Eng.]
- Kozachenko, 2013a — Kozachenko D.N. Object-oriented model of railway stations functioning. *Science and Progress in Transport. Bulletin of the Dnipropetrovsk National University of Railway Transport*. — 2013. — 46. — Pp. 47–55. [Eng.]
- Kozachenko, 2013b — Kozachenko D.N. Mathematical model for assessing the technical and technological indicators of the RWS operation. *Science and Progress in Transport. Bulletin of the Dnipropetrovsk National University of Railway Transport*. — 2013.— 45. — Pp. 22–28. [Eng.]
- Kirichenko, 2017 — Kirichenko A.I. Methods of creating intelligent automated control systems for the delivery of goods on the railway. *Science and Progress in Transport. Bulletin of the Dnipropetrovsk National University of Railway Transport*. — 2017. — 2(68). — Pp. 142–154. [Eng.]

- Leonenkov, 2007 — Leonenkov A.V. UML 2. BHV-Petersburg. — 2007. — 560 p. [Eng.]
- Milinkovic, 2013 — Milinkovic S. A fuzzy Petri net model to estimate train. *Simulation Modelling Practice and Theory*. — 2013. — 33. — Pp. 144–157. — <https://doi.org/10.1016/j.simpat.2012.12.005> [Eng.]
- Szűcs, 2001 — Szűcs G. Railway simulation with the CASSANDRA simulation. *Journal of Computing and Information Technology*. — 2001. — 9(2). — Pp. 133–142. — <https://doi.org/10.2498/cit.2001.02.04> [Eng.]
- Voevoda, 2009 — Voevoda A.A., Romannikov D.O., Zimayev I.V. Application of UML diagrams and Petri nets in the development of embedded software. *Scientific Bulletin of Novosibirsk State Technical University*. — 2009. — 4. — 169–174. [Eng.]

