Application of growing nested Petri nets for modeling robotic systems operating under risk

E V Sorokin, A V Senkov

National Research University "Moscow Power Engineering Institute" in Smolensk, Energeticheskiy proezd, 1, Smolensk, 214013

E-mail: scorpwork@mail.ru; a.v.senkov@mail.ru

Abstract. The paper studies the peculiarities of modeling robotic systems engaged in mining. Existing modeling mechanisms are considered, which are based on nested Petri nets, and a new formalism of growing Petri nets is presented that allows modeling robotic systems operating under risk. Modeling is provided both for the regular operation mode and for non-standard modes in which individual elements of the system can perform uncharacteristic functions. The example shows growing Petri nets that are used for modeling extraction of flat coal seams by a robotic system consisting of several different-type autonomous robots.

1. Introduction

The use of robotics for underground mining is determined by many factors and directly affects the performance and safety of the works. Use of automated and automatic machines is primarily connected with human safety. As the depth grows, the injury rate increases due to occurrence of rock pressure. The injury rate among underground workers is 4,4 times higher (7,2 times for great depth works) than on the surface [6]. With the widespread mechanization, the number of accidents in the course of operating mining machines began to increase. High dustiness, humidity, temperature and pressure differences, noise, vibration, insufficient illumination in the workplace lead to occupational diseases and workplace accidents, including fatal [1].

Creation of effective robotic systems is not seemingly possible without designing adequately accurate models of such systems. Discrete-event modeling is the most common method of simulation and consists in representing an object behavior in the form of a discrete events sequence [5, 7].

The main difficulty in modeling robotic systems is the consideration of risk events, to which such systems are exposed. One of the main risks associated with the operation of autonomous systems is failure of individual elements of systems as a result of internal or external issues. In most papers dealing with discrete-event modeling, for example [3, 4], an approach is proposed for modeling risk events with different levels of implications, however, none of them consider adaptation of the system to the changed conditions. Available discrete-event modeling tools do not provide continuation of the system simulation after an occurrence of a risk event.

2. Extraction of flat coal seams using a robotic system

Let us consider the problem of extracting flat coal seams by a robotic system. The technology consists in interaction of four types of mobile robots. The excavating robot is designed on the basis of a cutterloader, the transport robot is based on a self-propelled car. The information-control robot and the fastener robot are developed specifically. Figure 1 schematically shows the main process of extracting flat seams using four interacting robots. Going up to the ventilation drift, excavating robot 1 removes the rock and fills the cart of transport robot 2.

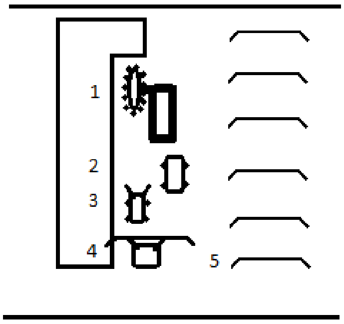

Figure 1. Scheme of flat seams extraction

The transport robot makes shuttle runs between the loader and the transport drift. The excavating robot is followed by control robot 3, which transmits information about the state of the worked-out area, that has not been bolted, as well as other data important for the system, e.g. temperature, ground state, etc. Robot 4 consistently moves the section supports, ensuring safe operation in the shaft section. Let us take a look at the model of the above system which is based on nested Petri nets [2, 8].

3. Nested Petri nets

A nested Petri net is a net that may use other Petri nets as tokens. A nested net consists of a system net which describes the operation of the system in general and elementary nets representing operation of individual system elements (individual robots in our case). While at the system net level each robot is described by a separate token, at the element net level the tokens allow one to determine the state of the robot [11, 13].

In contrast to classical Petri nets, nested Petri nets modeling is implemented through four types of steps:

1) a transfer step – firing the system net transition according to standard Petri nets rules;
2) an object-autonomous step only changes the marking the elementary net;
3) a horizontal synchronization step – simultaneous firing two transitions in two elementary nets;
4) a vertical synchronization step – simultaneous firing the system net transition together with some elementary nets transitions.

A nested Petri net may be formally described [9,10,11] as the set:

NPN = (Atom, Lab, SN, (EN1, . . . ,ENk), Λ), where

1) Atom = Var∪Con is a set of atoms (intersection of variables and constants sets);
2) Lab = Labv∪Labh is a set of net labels;
3) SN is a system Petri net;
4) (EN1, . . . ,ENk) is a finite set of ordinary elementary Petri nets;
5) Λ is a partial transition marking function that labels some transitions of the system net with labels of Labv and some transitions of the elementary nets – with labels of Labv∪Labh.

The simulation model of the robotic system given above can be represented as a two-level nested Petri net. The system net is represented in figure 2. In the initial position P1, transport, control and fastener robots are located, the loader performs the main work in position P4. The transport and control robots are advanced to the loader through transitions T1 and T6 respectively; the fastener robot maintains its position and installs specialized supports over the machines in transition T9, the main movement of this robot will take place after a step of the other robots operation in position T10.

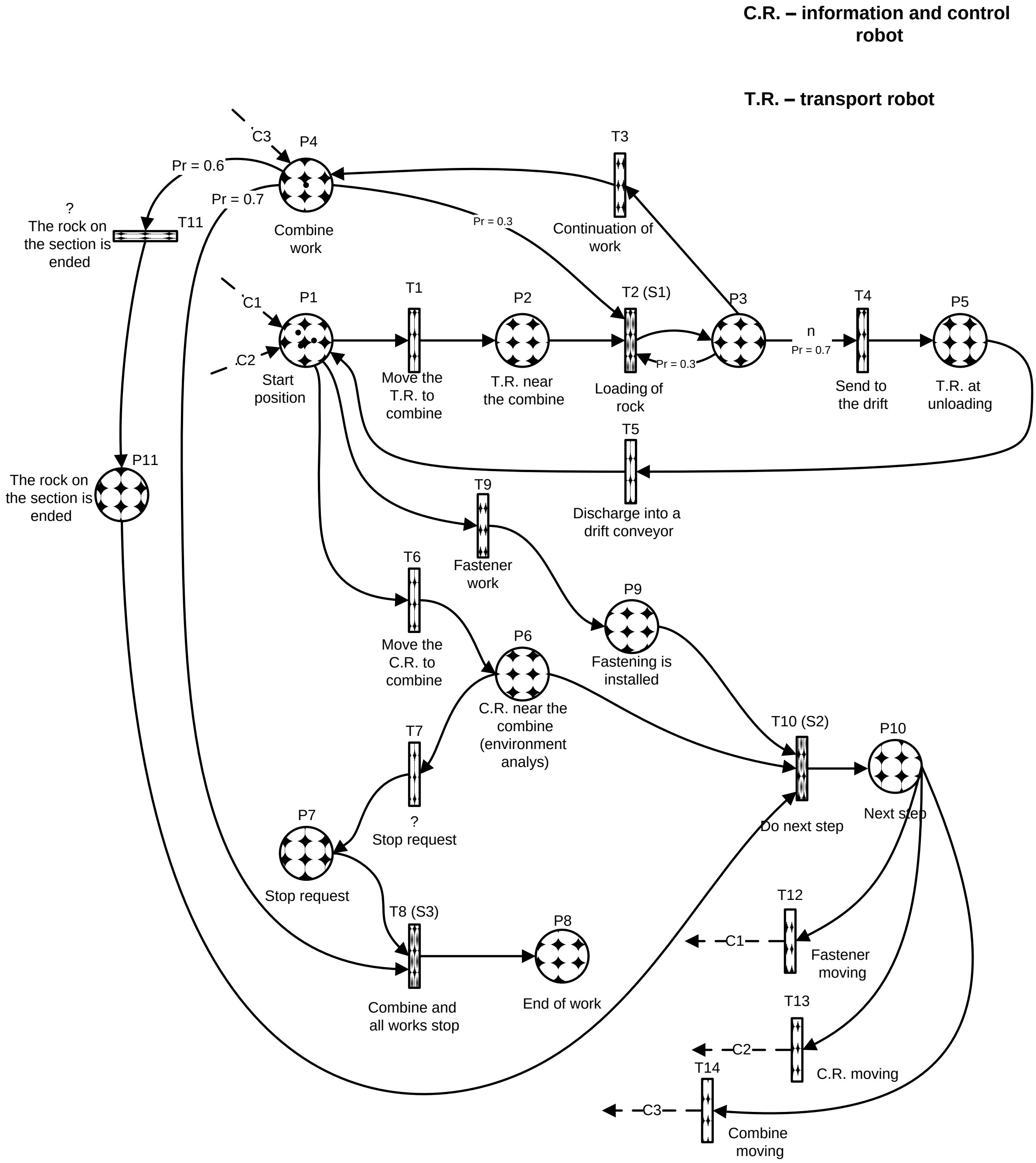

Figure 2. System Petri net for extracting flat coal seams

The rock is loaded from the loader to the transport robot via the vertically and horizontally synchronizing transition T2 (S1). Loading is possible if the loader is working at the face, and the transport robot is next to a free container.

Transition T2 provides vertical and horizontal net synchronization of the transport robot and the loader. The elementary net of the transport robot is shown in figure 3.

The transport robot is sent to the drift upon loading n units of the rock into the vessel, after which the run to the transport drift T4 and unloading of coal T5 are carried out.

The work of the information-control robot is contained in position P6. The control robot constantly monitors the main parameters of the environment, provides the machine with regular breaks, determines the degree of the rock looseness and corrects the loads and speeds of the controlled machines. The mining out command or the end of the shift are represented by transition T7. If the synchronizing transition T8 (S3) fires, the control robot and the loader will stop, which will inertially result in the subsequent stop of the other machines. The elementary net of the information-control robot is shown in figure 4.

Transition T10 (S2) of the system net reflects transfer to the next iteration of the face mining, for example, in case of the loader, reaching the ventilation drift.

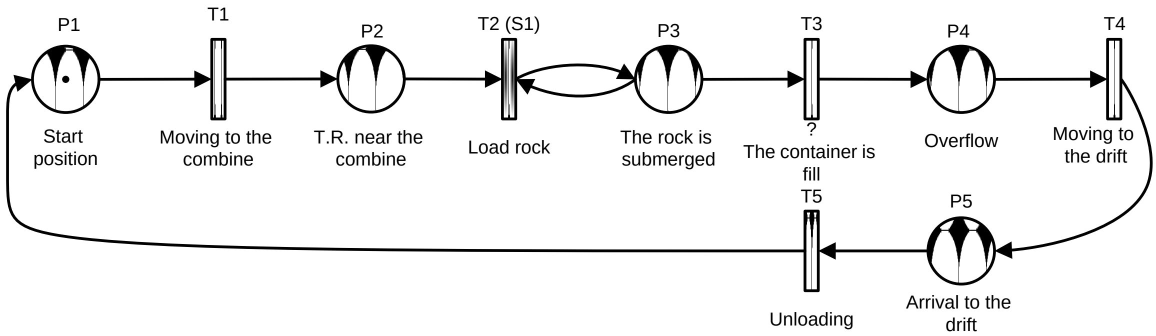

Figure 3. Elementary net of transport robot

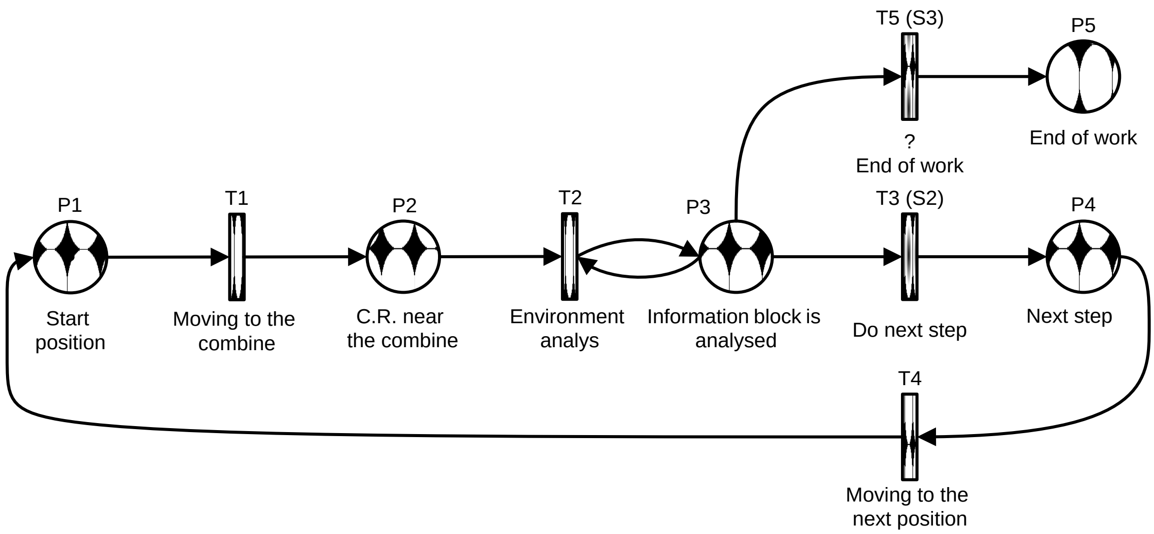

Figure 4. Information-control robot

4. Growing Petri nets

In the course of operation of the described facility risk, events of different nature may occur from time to time. When such events occur, the shortcomings of the classical nested Petri nets become apparent. Environmental influence, sudden changes in operating conditions and possible failures make the presented model inappropriate, and its further use becomes impossible since nested Petri nets have insufficient flexibility and are unable to automatically adapt to the changing conditions of the robotic system operation. Thus inability of a transition to fire may block the operation of the entire facility.

Depending on the level of the occurring problem, the model should try to adapt to the changed conditions. Implementing such system is suggested using "growing" Petri nets. Growth in this case will be understood as changing the net using behavior patterns created by an expert in advance for the key situations that were worked out in good time.

The formal description of a growing nested Petri net has a form of the vector GNPN = (Atom, Lab, SN, (gSN1, . . . ,gSNk), (EN1, . . . , ENk),
(gEN1, . . . ,gENk), Λ, fS, fE), where

1) gSN is a finite set of supplementary Petri nets of the system model;
2) gEN1, . . . ,gENk is a set of supplementary Petri nets for elementary models 1-k;
3) fS is a function of matching the supplementary model and the system net;
4) fE is a function of matching the supplementary model and the elementary net.

Let us single out two mechanisms for the growth of a nested Petri net.

1) Unified growth – involves developing a net without changing the current structure. In such situations, there is some pattern or a fixed-type net that can be mixed into the working system to perform certain actions, for example, to fix the problem. Such supplementary net can generally be applied for any transition of the elementary or system net.

2) Specialized growth – based on the specific features of the objects being modeled. It is used to develop the elementary net and involves interference in the structure – bypassing inefficient transitions, breaking links, etc.

Let us assume that any risk event can be reduced to the impossibility of firing of a transition of the elementary net robot or the system model shown in Figure 2. Let us consider the transport robot engine malfunction that occurred during the loading of the rock. A stone fell from the loader belt and broke through the front cooling radiator of the transport robot. To eliminate any malfunction, one can allocate some time for repair by starting the net growth with a module for eliminating failures, for example, as it is shown in [1]. Often, a time-out or a restart of the working element can solve the problem. An example of a Petri net growth with pending the repairing or eliminating the problem by restart is shown in Figure 5 and represents a unified type of the net growth.

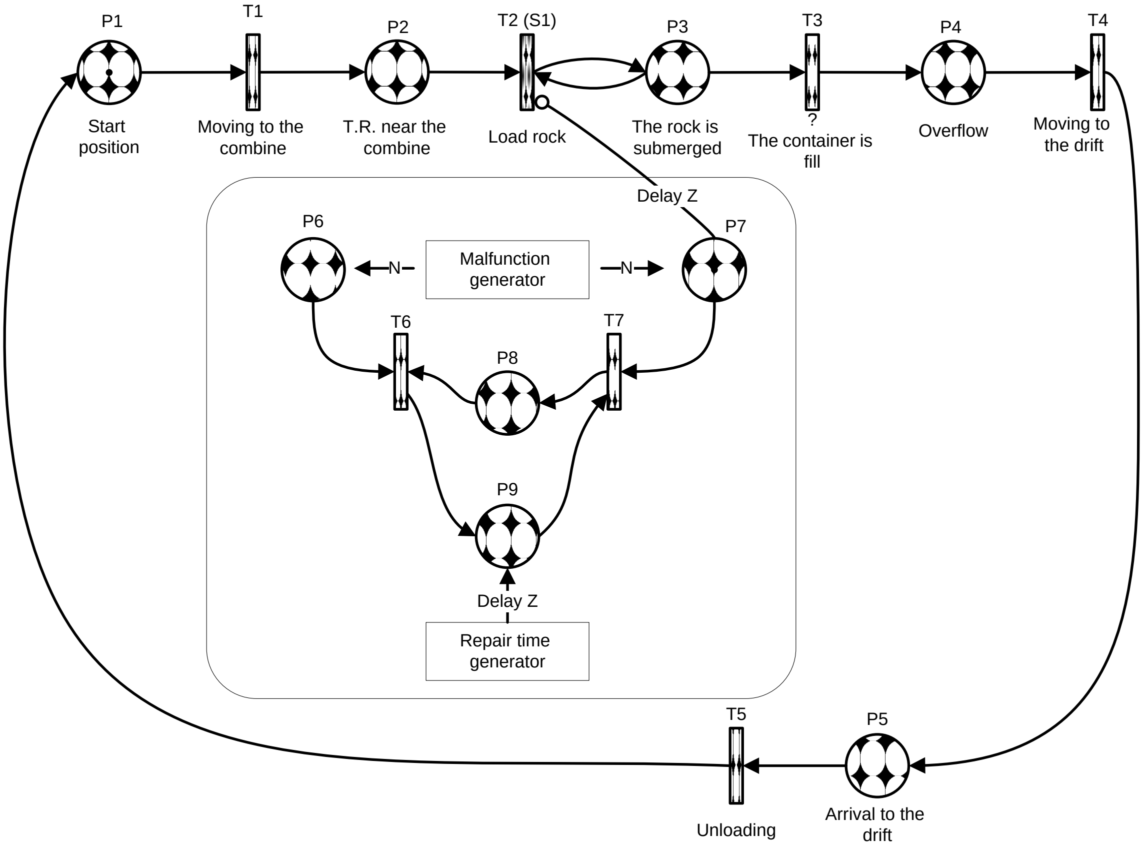

Figure 5. Transport robot malfunction module

If at least one token appears in position P7, transition T2 will be closed by an inhibitor arc. The failure rate is simulated by appearance of N tokens in transitions P7 and P6 from the failure generator. The complexity of the failure is indicated by the time-out Z. After elimination of the failure the token goes to position P8, releasing transition T2 after the time-out Z.

Long-term repairs are generally unacceptable in such systems and the failure module can be replaced with another troubleshooting option developed by an expert engineer. For example, if capable, the control robot may commit itself to making shuttle runs from the loader to the transport drift. In this case, the elementary net of the control robot will grow in a specialized way, as shown in figure 6.

5. Conclusion

The formalism of growing nested Petri nets, in contrast to existing ones, provides modeling of robotic systems operating under risk. At the same time modeling is provided both for the regular operation mode and for non-standard modes in which individual elements of the system can perform uncharacteristic functions. Such mechanism is convenient for assessing options for solving contingencies with consideration for time and resource costs. Moreover, representing models in the form of Petri nets makes it possible to test them for deadlocks, reachability of operations, which is of no small importance in designing complex robotic systems that are intended to solve heterogeneous operations in a changing environment.

Such representation is primarily suitable for modeling the studied processes, planning the behavior of the future system. In the future, it is planned to consider the management of robotic systems based on the formalism of growing Petri nets. To this end, the proposed growing Petri nets will be complemented by resource accounting mechanisms. Another direction of developing the proposed kind of Petri nets is supplementing them with tools of modeling the system under uncertainty of arising risk events.

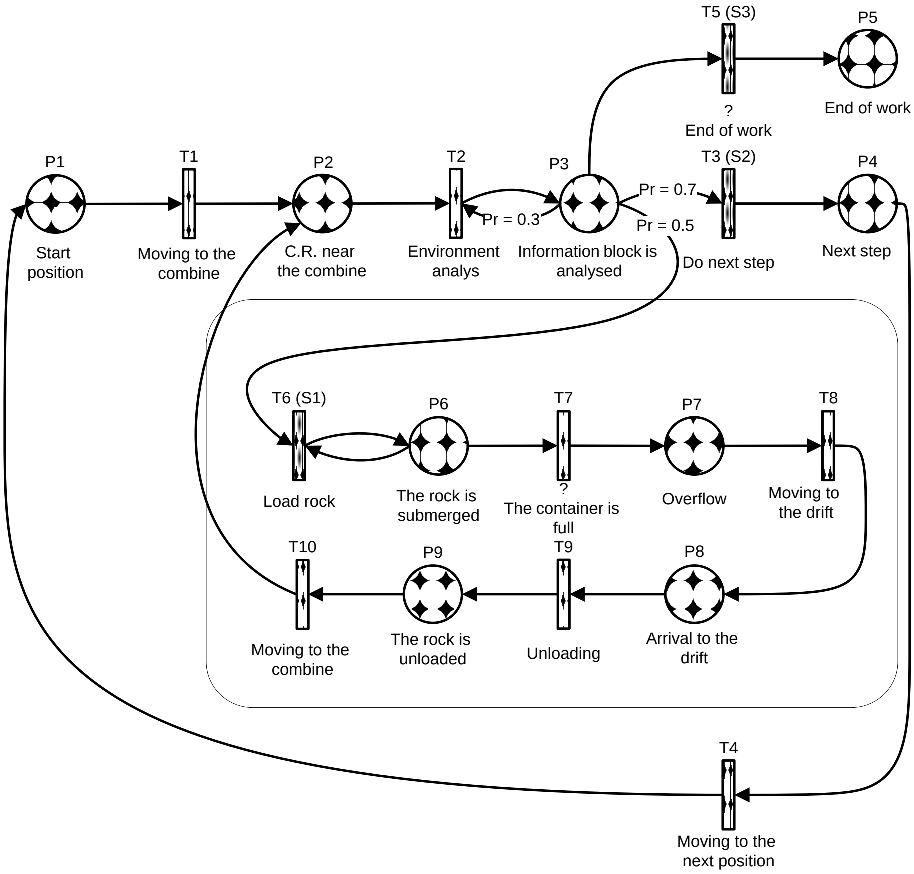

Figure 6. Growth of information robot elementary Petri net

Acknowledgments

Report study was founded by the Council for Grants of the President of the Russian Federation, project № MK-6184.2016.8

References

[1] Konuh V L, Zinoviev V V 2011 *Discrete-event modeling of underground mining*. (Novosibirsk: SO RAN) p 243

[2] Lomazova I A 2004 *Nested Petri nets: Modeling and analysis of distributed systems with object structure*. (M.: Nauchniy mir) p 208

[3] Konuh V L 1991 *Preproject analysis of mine robotic systems.* (Novosibirsk: Nauka. Sib. Br.) p 182

[4] Sturgul D 1998 Computer simulation of mining operations *Gorniy vestnik.* **1** 77 – 81

[5] Livshic A L, Malc E A 1978 *Statistical modeling of queuing systems* (V.: Sov. Radio) p 248

[6] Arsham H, System Simulation: The shortest Path from Learning to Applications [electronic resource]: http://ubmail.ublat.edu/~harsham/simulation/sim.htm

[7] Banks J, Carson J, Nelson B, Nicol D 2001 *Discrete-Event System Simulation* (NY: Prentice Hall) p 594

[8] Zernov M I, Sak-Sakovskiy V I, Senkov A V, Sorokin E V 2016 *Variant of management of resource risks of elements of complex organizational-technical systems on the basis of embedded Petri nets* Fifteenth National Conference on Artificial Intelligence with International Participation KII-2016. Proceedings of the conference. (Smolensk: Universum) **2** 332-336

[9] Lomazova I A 1997 *On Proving Large Distributed Systems: Petri Net Modules Verification*. Proc. 4th Int. Conference on Parallel Computing Technologies. Lecture Notes in Computer Science. **1277** 70–75

[10] Lomazova I A 2000 Nested Petri nets — a Formalism for Specification and Verification of Multi-Agent Distributed Systems *Fundamenta Informaticae* **43(1–4)** 195–214

[11] Lomazova I A 2001 Nested Petri nets: multi level and recursive systems *Fundamenta Informaticae* **47(3-4)** 283–293