

Computational and Technological Models of Cognitive Monitoring Systems

Alexander Vodyaho¹, Evgeniy Postnikov², Osipov Vasiliy³, Nataly Zhukova³, Michael Chervontsev², Nikolai Klimov^{*4}

¹Saint Petersburg Electrotechnical University, Saint-Petersburg, 197022, Russia

²Research and Engineering Center of Saint Petersburg Electro-Technical University, Saint-Petersburg, 197022, Russia

³Saint Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, Saint-Petersburg, 197022, Russia

⁴Saint Petersburg National Research University of Information Technologies, Mechanics & Optics, Saint-Petersburg, 197022, Russia

ARTICLE INFO

Article history:

Received: 21 January, 2019

Accepted: 13 March, 2019

Online: 26 March, 2019

Keywords:

Cognitive monitoring,

Computational models synthesis

ABSTRACT

The approach to the construction of promising cognitive monitoring systems is considered. Based on the analysis of known solutions, the problem of automatic synthesis of computational models of cognitive monitoring is posed. In the synthesis of models, it is proposed to reduce them to relatively finite operational automata. An example of a synthesized model of cognitive monitoring is given.

1. Introduction

Monitoring systems (MS) is one of the important subclasses of information systems (IS). Nowadays one can observe an increasing of interest to this class of IS because of a high relevance of these IS in practice. Due to increasing amount of observable objects, their complexity and variability of its properties, existing MS very often are not able to solve the challenges they face. Many creative tasks of monitoring are solved only with the participation of a person. Significant time and material resources are spent. Among such difficult creative tasks are the following: constructing models of objects and situations, assessing their state and the impact of external and internal factors on possible scenarios for the development of events, synthesis and implementation of monitoring programs and others. Solution of these problems can be provided prospective cognitive monitoring systems (CMS).

The CMSs should allow machine control of the monitoring processes of the target objects. Such management should be carried out proceeding from solved applied problems. CMSs form a separate subclass of cognitive systems (CS). For CMSs the input data are received from observed objects. Data can include target information and measurement results of object state parameters. The tasks to be solved include obtaining data, processing them, managing the collection of data about objects.

CMSs are intelligent IS that are able to perceive information about the state of the external world and their own state. These systems must bind the observed events and form their space-time models. On the basis of these models, management of monitoring

itself should also be carried out. Ability to work at these models allows CMSs, in contrast to traditional management systems, realize content adaptive processing of information streams received from observable objects. Content adaptive is the processing of the semantic content of information flows taking into account the observed context, which is determined by monitoring conditions, set goals. Such processing allows identify information elements in data streams, to establish relations between them. The resulting coupled structures can be considered as models of real world objects, reflecting the states of these objects and their variation in space and time. In the process of IS operation, the constant reconstruction of these models is carried out. It should be noted that CMS can realize cognitive behavior, and can solve specific application tasks, which can relate to a variety of subject domains.

The article suggests an approach to the construction of promising cognitive monitoring systems. This approach develops the existing theoretical and methodological apparatus of cognitive monitoring from the standpoint of implementing such systems. In the second section of the article, an analysis of modern CS, the models, methods and architectural solutions used in them is carried out. In the third section, the problem of synthesis of computational models of CS is formulated. The fourth section proposes a method for their synthesis. In the final section, an example of synthesis is considered.

2. Analysis of modern CS and their capabilities

The principles of CS organization are investigated already throughout not less than thirty years and the considerable experience of creation of such systems is accumulated.

*Corresponding Author: Nikolai Klimov, Email: hocico16@gmail.com

Some models, methods and architectures of CS have been proposed in research [1, 2, 3, 4]. In the early CS, such formalisms as cognitive maps were widely used, in which factors (elements of the system) and relations between them were determined [5, 6]. In later systems, mechanisms for working with situation patterns and fusion mechanisms [7 - 10] were applied. In accordance with the existing approaches to creation of CS models of information representation and working methods with them are classified according to the types of cognitive architectures. There are three main types of cognitive architectures — symbolic, emergent and hybrid [7]. The first type is characterized by the use of production and graph models of knowledge, for the second type - the use of associative neural network methods, as well as other competitive models and methods. Hybrid architectures involve the sharing of knowledge models and neural network methods. Examples of systems are given in 15.

The well-known are such CS as SOAR [11], ICARUS [12], ACT-R [13], which position themselves as IS with cognitive architecture. A detailed overview of IS with cognitive architecture is given in [14].

The first approach to constructing the CS has been developing for a long time. Early work done in this direction was of an experimental nature. They can only partially simulate the intellectual activity of man through the reproduction of individual heuristic algorithms. Now, the attention of research teams working in this direction is mainly focused on the study of static cognitive structures. The questions of constructing and using dynamic cognitive structures are practically not being researched [15 - 21].

In a significant part of modern CSs try to use the neural networks. However, so far the possibilities of usage the big neural networks are significantly limited due to high requirements for computing resources and imperfect architectures these networks.

In practice, engineers prefer to use symbolic and hybrid CSs. In all cases the architectural issues of the creation and maintenance of the CS remain, notably the issues of the accumulation and use of architectural knowledge, the development of flexible architectures, as well as mechanisms for self-monitoring and self-adjustment of such systems.

The main difficulty in using readymade cognitive models, methods and architectures for solving monitoring and management tasks is that they do not provide for controlled reorganization under changing monitoring conditions. In particular, content adaptation is not provided for data processing, adaptation to changing monitoring and control requirements, and monitoring environment capabilities.

Some works are known, in which some issues of creating the CS in the interests of solving problems in the field of monitoring and management are considered, but their number is very limited. In addition, almost all of them are focused on solving particular problems for specific subject domains. For example, in [22], we consider possible ways of realizing cognitive possibilities in the field of transport.

In general, the analysis of the state of modern CMSs has shown that they are rather scattered, a general approach to the

construction of CS has not been formed to date, and existing solutions are poorly applicable for the construction of CMSs.

From the point of view of creating CMS, an approach focused on working with models of objects and monitoring processes is of considerable interest. Nowadays this approach finds practical implementation. The idea of the model-oriented approach is based on the methods of automatic synthesis [23]. The created apparatus determines the theoretical basis of CM and provides a sufficiently powerful methodical apparatus. It allows build agile systems.

Agility is achieved due to an informed choice of the values of dozens of tuning parameters and their timely correction. There are separate implementations of systems, which were tuned by IT specialists. For practical usage of this the approach, it is necessary to have a tool to develop models that determine how to implement and configure CMS in practice. First of all it is computational models of these systems.

The CMS can be considered as the domain specific CS.

The system of models which describes, both structure, and behavior of a target system is the basis for functioning of CSM. CSM has to solve rather wide class of tasks, in particular: i) automatic creation of models, ii) check of relevance of models, iii) presentations to stakeholders information about target system status in the required form; iv) to solve problems of optimization of target system structure and target system business of processes and many other problems.

The following main requirements are imposed to CSM: i) CSM must have all properties of CS, ii) models can be used for describing both target system dynamic structure and dynamic business processes; iii) models can describe multilevel systems and parallel processes; iv) procedures of automatic creation of models (model synthesis) must be available; v) models must be suitable for describing target system in time.

Different approaches to creation of models can be used/ First of all following approaches are used: i) ontology, ii) Petri nets and graph models, iii) neural networks iv) automata models.

Table 1: Advantages and disadvantages of different approaches

N	Technology	Advantages	Disadvantages
1	Ontologies	Can describe dynamic structures Logical inference is available	Problems with description of target system behavior and problems with synthesis
2	Petri nets and graph models	Can describe target system behavior. Algorithm of process mining are available	Problems with description of multilevel structures
3	Neural networks	High speed of operation	Problems with description of multilevel structures and problems with synthesis
4	Automata models	Enough good speed of operation. Algorithm of synthesis are available	Problems with description of parallel processes.

One can effectively use ontologies for the description a structural component of a target system, there is a good query language but for describing of target system behavior they are not very good.

Petri nets and graph models (work flow graphs) can be effectively for business processes modeling. They can describe parallelism and can be automatically generated from log files. These models are widely used within Process Mining [24]. This approach is rather good for describing target system behavior but not for working with multilevel structures. Neural networks show very good performance but there are problems with synthesis and working with multilevel structures. Automata models are used relatively seldom for CS realization but have a number of advantages, in particular, they can work with multilevel structures and can be generated automatically.

Advantages and disadvantages described above approaches are presented in the table 1.

Thus, it is apparent that one model can't solve all problems and a multimodel approach is to be used. Nowadays primary interest for CMS practical implementation is the solution of 2 problems: i) development of multimodel for target system description and ii) solving the problem of describing multilevel structures i.e. with the help of automata models.

3. The problem of synthesis of a computational model

In a broad sense, a computation model determines the behavior of the system as a whole, based on the behavior of its individual parts. The construction of the computation model involves the definition of the set of permissible operations used for computations, as well as the relative costs of their application. The computation model allows to characterize the necessary resources - the execution time, the amount of memory, as well as the limitations of algorithms or hardware. The main purpose of the computational model is to ensure the possibility of efficient use of available resources in solving problems of CM. When evaluating computational models, parameters of the generated models of cognitive monitoring can also be taken into account.

Then a computational model can be defined as $B(R,E,T)=F(u(E), v(T),g(R))$, where $u(E)$ - is the function of monitoring subtasks distribution among the elements of target systems E , $v(T)$ is a function of technologies definition T , which are used for monitoring problems solving, $g(R)$ - is the function of requirements for technologies of monitoring problems forming which are defined by monitoring formal models R . So, for building a computational models of CMS 3 models are to be determined: the formal model of monitoring, monitoring environment model and the technological model. These models determine the conditions for constructing computational models, their structure and parameters. The position of computational model in a general CM model system is shown in the fig. 1.

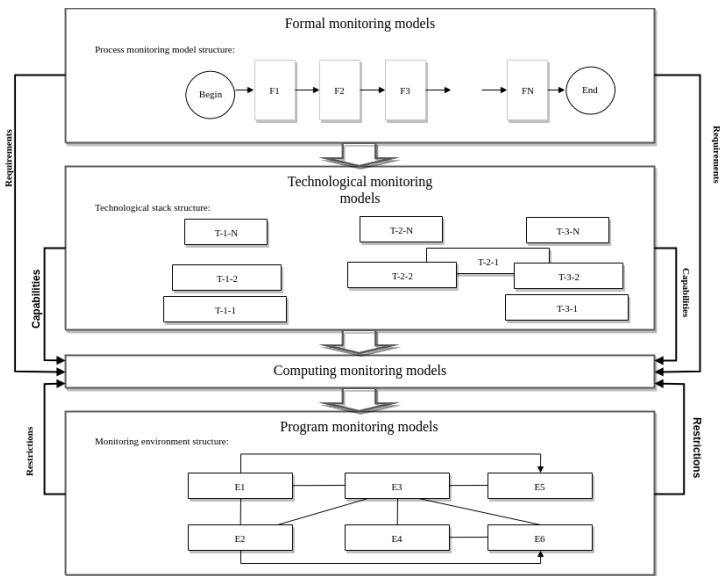
The CM formal models can be defined in the frames of general CM theory [25]. The CM process model can be presented as a system of correlated cognitive functions $\{F_i\}$, $F_i F$, which allow build models of real world objects.

The model of the monitoring environment can be presented as a set of elements $\{E\}$, $E_i E$. Monitoring environment as a rule is

multilevel distributed media with permanently changing parameters. Monitoring systems in this case can have several levels. The levels are defined by types of used hardware. Monitoring hardware can be located inside the objects of a target system such as local servers, clusters etc.

The CM technological model is defined as a set of technological stacks $\{T_i\}$, $T_i T$. Each stack T_i includes the set of technologies $\{T_j\}$, $T_j T_i$ which are similar mentioned above but they have different parameters $\{T_j\}$ $T_j T_i$. The example of technological model, which supports 3 stacks (T_1, T_2, T_3), is presented in the fig.1. In each stack N technologies are defined. For CM process realization one can use different technological solutions depending on the monitoring environment and goal. So we have the following dependency $T(R,E)$.

Generally, the problem of CM model synthesis can be formulated in the following way. It is necessary to find an effective computational model B_0 for which W_{opt} - extremum of the key indicator of efficiency (minimum of amount of resources which are necessary for CM procedure realization) $W_{opt}(B_0)=extr_v W(M(B_v))$, in case of restrictions for auxiliary indices of efficiency $M(B_v)$ which are defined by the parameters of CM models to be formed: $M(B_v)=F(R(B_v),E(B_v),T(B_v))$, $M(B_v)DM, R(B_v)DR E(B_v)ER, T(B_v)TR$, where A is a set of variants of CM models organization, DM is the area of admissible parameter values of CM models, DR is the area of admissible variations of requirements to parameter values of CM models, DE are restrictions defined by the CM environment, DT is the set of available technologies. The main CS model parameters are accuracy, reliability, completeness, etc.



4. Synthesis of the CM computational model

According to the given above definition of the CM Computational model it can be presented in the form of three dimensional structure: $A = \{a_{i,j,k}\}$, $a_{i,j,k} \in \{0,1\}$ In this case i -th dimension reflects realized CM functions, j -th dimension describes components of monitoring environment and k -th dimension defines components of technological stack. In this matrix an element $a_{i,j,k}$ is equal to 1 if i -th cognitive function is

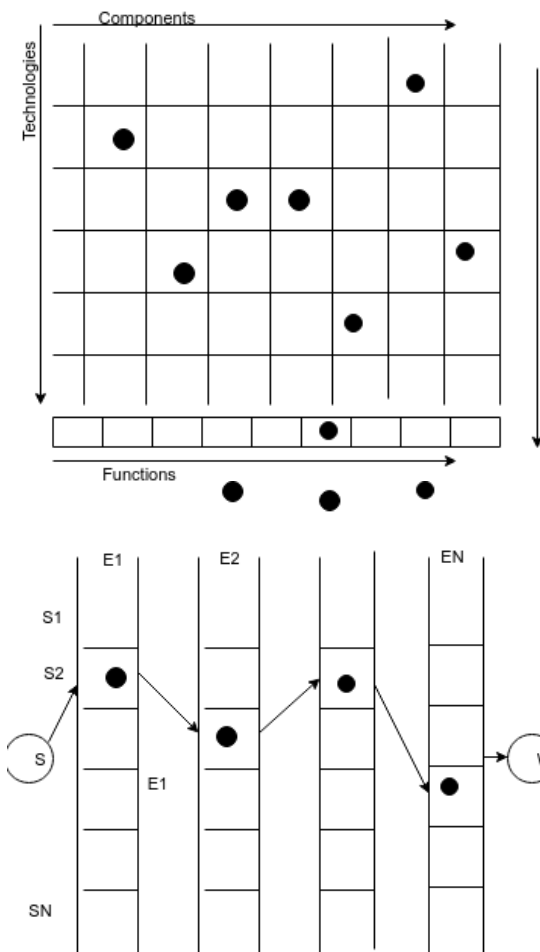
realized on j -th component using k -th technology. Matrix projections on two dimension space $j \times k$ i.e. on the monitoring environment space and technological components stack for defined element i is shown in the fig 2a.

In order to synthesize such structures, it is proposed to use relatively finite operational automata (FOA)²⁶. The synthesis scheme of the computational model is shown in Fig. 2b). In accordance with²⁷, each automate FOA_r in r -th moment of the time is described by a set of ten parameters:

$$FOA_r = \{da_r, db_r, dc_r, F_r^b, F_r^c, DA(db_{r-1}), \dots\}$$

$$\{\dots, DB(db_{r-1}), DC(db_{r-1}), FB(db_{r-1}), FC(db_{r-1})\}$$

, where da_r - vector of input data; db_r - vector of internal state parameters; dc_r - vector of exit state parameters. Functions F_r^b of transitions of an automate from one internal state to another and functions F_r^c are written in the form: $db_{r+1} = F_r^b(da_r, db_r)$, $dc_r = F_r^c(da_r, db_r)$. States da_r , db_r , dc_r , and functions F_r^b , F_r^c , characterizing the automaton at the r th moment of time, must satisfy certain conditions: $da_r \in DA(db_{r-1})$, $db_r \in DB(db_{r-1})$, $dc_r \in DC(db_{r-1})$, $F_r^b \in FB(db_{r-1})$, $F_r^c \in FC(db_{r-1})$, where $DA(db_{r-1})$, $DB(db_{r-1})$, $DC(db_{r-1})$, $FB(db_{r-1})$, $FC(db_{r-1})$ - allowable sets of states and functions of the automate defined with respect to $r-1$ time. The transition from the FOA_r to the FOA_{r+1} automate by $r+1$ moment can be written as: $F_r^b : FOA_r, da_r \rightarrow FOA_{r+1}$



It is envisaged to build your machine for each of the functions. As a result, the computational model at time r will be described by a family of automata: $FEOA = \{FOA(F1), FOA(F2), \dots, FOA(FN)\}$. FOA for different functions can be constructed in any order.

During synthesis of FOA for i -th functions can take into account other previously synthesized automata, then there is a dependence $FEOA_r(FEOA_{r-1})$. This approach is justified when certain functions can be expressed through other functions, i.e. $F_k = G(F_i), i \in \{1..N\}$. In this case, one can speak of a system of interconnected automata.

5. An example of a CM model

CM and control systems can be used in many subject domains including the "smart house" systems. These systems can be divided into three subclasses: centralized, bus and mixed.

Each smart house system consists of various modules, for example: supply module, security module, security module, management module. The supply module may include the following submodules: a power supply submodule, a water supply submodule, a submodule of equipment and household appliances. The security module can include an alarm submodule, a submodule of video surveillance, a submodule of fire alarm. The control module, as a rule, consists of a submodule of access control and access control, a fire suppression module, a climate control submodule and lighting control.

The domain description of the smart house system includes the following main components: automated functions, data processing tasks and procedures, users, information elements and relationships between them, characteristics of information elements and procedures for data processing, relationships between information elements and procedures.

Thus, the domain model is represented in the form: $M = \langle F, H, P, O, R \rangle$, where F is the set of automatized functions, H is the set of tasks (procedures) for data processing, P is the set of users, O is the set of objects and automation processes, R is the set of relations (relationships) between the components $\{F, H, P, O, V\}$, V is the complete set of informational elements of the domain.

Systems of the smart house class can be implemented in the form of CM and management systems. Various modules of such a system can act as automated monitoring units (fire alarm, burglar alarm and power consumption) capable of adjusting control actions as a result of analyzing and processing incoming data and constructing an environment model and a monitoring model based on this data. Information from monitoring units can also be processed at the central control point of the system from which the subsystems also receive control actions.

As one of the monitoring subsystems, consider the fire monitoring module. This subsystem consists of modules for monitoring fire alarm and control module (fire suppression module). The behavior of the fire alarm module at each point in time is set by the monitoring environment model, at each time formed on the basis of data coming from groups of fire sensors (smoke and heat detectors), as well as historical indications. The control action can consist in forming a notification of a sensor failure, generating a pre-fire notification and sending control commands to the fire suppression subsystem in accordance with a predetermined algorithm.

Definition of a set of automation objects: a group of smoke detectors, a group of thermal detectors, a group of address points, a peripheral microcontroller of a substation, a group of fire extinguishing elements (relays, etc.), a central controller of a substation.

Definition of a set of automated functions: monitoring of detector malfunction, monitoring of room temperature, monitoring of optical density of the environment, start-up of the fire-extinguishing system, stopping of the fire-extinguishing system, generation of an alarm notification on the fact of sensor failure detection.

Definition of a set of data processing procedures: collection of information about the failure from the central controller, rotation of obsolete (irrelevant) data, archiving of fault notifications, aggregation of historical data for the subsequent analysis of the current situation on the gradient.

Definition of a set of users of the system: the administrator (owner), the module of the central database.

Definition of a set of information elements of the fire alarm system: device identifier, status identifier, event identifier, event date, event time, event data storage time, event response event, system response date, system response time.

Consider the possible scenario of cognitive monitoring and management of the fire alarm and firefighting system:

The monitoring system periodically poll groups of sensors and stores the readings. In real time, the derivatives are calculated from the changes in the readings. Data processing occurs on several levels:

1. At the level of the peripheral controller, the data undergoes preliminary filtration and purification, the malfunction of specific sensors is analyzed;
2. At the central controller level, data from the peripheral controller is stored and the fire is analyzed for the gradient;
3. At the level of the fire control and fire control subsystem, decisions are made on the control effect;
4. At the level of the general monitoring system, high-level notifications from the fire alarm subsystem are aggregated, an administrator is notified.

The monitoring procedure is adjusted in real time depending on the internal and external input data:

1. If a defective detector is detected, it can be excluded from the list of the system elements being polled; If a possible fire on a gradient is detected, the frequency of polling of sensors in the area of the alleged incident is increased;
2. According to indications from adjacent subsystems (volume and presence sensors, climate control systems), thresholds for fire alarm elements can be adjusted.

Consider the synthesis of a computational model of a fire system at the level of the central and peripheral controllers. The state machine for the peripheral controller is described by the following sets:

1. Set of Input Data $DA(\underline{db_{r-1}})$ - integer values of optical density of sensors;
2. Set of internal state $DB(\underline{db_{r-1}})$ - the indicator of serviceability of fire loop elements, the frequency of polling of the elements of the fire loop;
3. Set of output state $DC(\underline{db_{r-1}})$ - aggregated indicators of sensors, fault messages;

4. Set of transition functions between internal states F_r^b - making a decision about the condition of the sensor.

The state machine for the central controller is described by the following sets:

1. Set of Input Data $DA(\underline{db_{r-1}})$ - aggregated readings from peripheral controllers, data from adjacent climate control subsystems and firefighters;
1. Set of internal state $DB(\underline{db_{r-1}})$ - a fire indicator by a gradient in a particular area;
2. Set of output state $DC(\underline{db_{r-1}})$ - notification of malfunctions in the system, warnings and fires;
3. Set of transition functions between internal states F_r^b - making a decision on a fire by a gradient.

With the help of control actions, the load can be redistributed at different levels of the system: in the case of a fault diagnosis, the central controller decides on reducing the frequency of polling the problem sensor by the peripheral controller; In the diagnosis of the premozharnoy situation, the frequency of the poll, on the contrary, a decision is made to increase the frequency of polling the target and adjacent sensors.

Consider the synthesis of the computational model of the subsystem of fire alarm and fire extinguishing in the "smart house". Set of input parameters $DA(\underline{db_{r-1}})$ in the system - numerical values of the optical density and temperature obtained from the sensors. Set of internal states of the monitoring system $DB(\underline{db_{r-1}})$ is set by the polling rate of each sensor, as well as by one of the operating modes of the system: Normal, Pre-fire, Sensor Fault, Fire. As a computational model, the derivative of the change in the readings of the sensors P.

In the "Normal" state, the load on the central and peripheral controllers polling the sensors is distributed evenly, the polling period of each sensor is t_1 . The transition to the "Pre-fire" state is determined by the excess of the derivative P of the admissible value P1. In this case, a redistribution of the computational load occurs: the frequency of polling of the target and adjacent sensors to t_2 increases. Also, the technological models and functions used are changed: instead of the derivative P, the received sensor observations begin to be compared with a certain threshold value R1. The transition to the "Fire" state passes when the fire hypothesis is confirmed, at which the value of the observation begins to exceed the threshold value R1. Polling of sensors in the prospective area of the fire is stopped, and the computing resources are transferred to the control subsystem, activating the fire suppression algorithm in this area.

The CMS of the fire situation in the smart house has a number of advantages over the classical systems of fire alarm and fire extinguishing:

1. On the basis of a real time model of the monitoring environment, it is possible to quickly recognize the dysfunctional parts of the system (fire detectors) for the purpose of their subsequent replacement, as well as the formation of premonitory notifications based on the calculation of the derived values of the sensor readings;

It allows in real time to change the composition of the monitored monitoring environment and to generate control actions more localized.

Conclusion

Usage of CMSs allows solve a problem of managing distributed IS with a very big number of elements and variable structure i.e. allows build IS with higher levels of complexity. CMSs assume operation on the model level. The key problem of CM is on the fly generation of target IS model. In suggested article the problem of model generation from data is discussed. This is only the first step which allows build rather simple CMS. The next step to be done is looking for methods of model synthesis methods on the base not only raw data but knowledge, including knowledge from the external ontologies.

References

- [1] M. Glodek, F.Honold, T. Geier, G. Krell, F. Nothdurft, S. Reuter, F. Schüssel, T. Hörnle, K. Dietmayer, W. Minker, S. Biundo, M. Weber, G. Palm, F. Schwenker Fusion paradigms in cognitive technical systems for human-computer interaction *Journal Neurocomputing archive Volume 161 Issue C, August 2015 Pages 17-37*
- [2] Osipov V., Vodyaho A, Zhukova N., Glebovsky P. Multilevel Automatic Synthesis of Behavioral Programs for Smart Devices, Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), 2017
- [3] Madl, T., Chen, K., Montaldi, D., and Trapp, R. (2015). Computational cognitive models of spatial memory in navigation space: a review. *Neural Netw.* 65, 18–43. doi:10.1016/j.neunet.2015.01.002
- [4] Zagoruiko, NG Cognitive analysis of data. - Academic publishing house "GEO", 2012. - 203 p.
- [5] Axelrod, R. The cognitive mapping approach to decision making.// In: *Structure of Decision. The Cognitive Maps of Political Elites*, R. Axelrod (ed.), (3-18), ISBN: 069107578-6, Princeton University Press, Princeton. 1976.
- [6] Avdeeva Z.K. Comparative analysis of selective cognitive maps by the degree of formalization // *Cognitive analysis and management of the development of situations (CASC'2009): Proceedings of the International Conference (November 17-19, 2009, Moscow)*. Moscow: IPP RAS, 2009. P.11-22.
- [7] Duch W., Oentaryo R., Pasquier M., Cognitive architectures: where do we go from here? *Proceedings of the Second Conference on AGI, 2008*.
- [8] Gardner K. M., Rush A., Crist M. K., Teegarden B. *Cognitive Patterns: Problem-Solving Frameworks for Object Technology*. Robert K Konitzer. Zurich, Switzerland. 2011. 166 p.
- [9] Glodek M., Honold F., Geier T., Krell G., Nothdurft F., Reuter S., Schüssel F., Hörnle T., Dietmayer K., Minker W., Biundo S., Weber M., Palma G., Schwenker F. Fusion paradigms in cognitive technical systems for human-computer interaction. *Neurocomputing* 161. 2015. p. 17–37
- [10] Perlovsky L. I. Cognitive high level information fusion *Information Sciences* 177 (2007) 2099–2118
- [11] Laird J., Rosenbloom P., Newell A., Soar: an architecture for general intelligence, *Artificial Intelligence* 33. (1987).
- [12] P.Langley, An adaptive architecture for physical agents, in: *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2005*.
- [13] Anderson J. R., Lebiere C. The newell test for a theory of cognition, *Behavioral and BrainSciences* 26 (2003).
- [14] I. N. Rozenberg Cognitive Transport Management. *TheStateCounsellor*, No. 2, 2015. From 47-52
- [15] Osipov V.Y. Development of Information Processing Processes in Associative Intelligent Machines // *Mechatronics, Automation, Control*, No. 6, 2014, pp. 8-13.
- [16] Osipov V.Y. A recurrent neural network with layer structure in the form of a double helix // *Information Technologies*, No. 7, 2014, pp. 56-60.
- [17] Osipov V.Y. Associative intelligent machine with three signal systems // *Information-control systems*, No. 5, 2014, P. 12-17.
- [18] Osipov V.Y. Limits of memory of recurrent neural networks with erasure of outdated information // *Scientific Bulletin of Novosibirsk State Technical University*, No. 3 (56), 2014, P. 115-122.
- [19] Neurodynamic Systems Of Multiagent Modeling Based On The Modern Theory Of Catastrophes Nechaev Y.I. *Neurocomputers: development, application*. 2017. No. 1. P. 16-29.
- [20] Neurodynamic modeling of nonlinear nonstationary systems based on the modern theory of catastrophes Nechaev Y.I., Lav N.S. *Neurocomputers: development, application*. 2017. № 3. P. 3-16.
- [21] Concept of neural networks under the functioning of self-organizing systemsS Nechaev Y.I. In the collection: *Proceedings of the XI International Conference on Nonequilibrium Processes in Nozzles and Jets (NPNJ'2016) Moscow Aviation Institute (National Research University)*. 2016. P. 445-447.
- [22] Osipov V. Yu. Automatic Synthesis of Action Programs for Intelligent Robot // *Programming and Computer Software*. 2016. Vol. 42. No. 3. pp. 155 -160.
- [23] *Encyclopedia of Management*. URL: <http://www.enotes.com/management-encyclopedia/management-thought>
- [24] Wil van der Aalst. *Process Mining Data Science in Action Second Edition* (DOI 10.1007/978-3-662-49851-4) Springer-Verlag Berlin Heidelberg 2016. 467 p.
- [25] Osipov V. Sintez rezultativnyih programm upravleniya informatsionno-nyichislitelnyimi resursami [Synthesis of resultive programs for information and computing resources management] *Priboiry i sistemyi upravleniya*. 1998. #12. S.24-27