Networks of Autonomous Real-Time Agents in Environments with Counteraction: Features and Components of the Model

V. Gorodetsky and M. Panteleev
Saint Petersburg Electrotechnical University «LETI», Saint Petersburg, Russia
E-mail: {vladim.gorodetsky, mpanteleyev}@gmail.com

Abstract. A group behavior and control model in a system consisting of autonomous real-world and Internet objects with opposition is considered. The urgency of the problem is due to the growing role of Internet technologies in competition models in economics, politics, computer security, etc. The problem model is analyzed in the context of the Boyd Cycle. The tasks of the cycle phases are formulated, a multi-agent model of the behavior of autonomous objects and the digital twin of the external environment in the control loop is substantiated. The necessity of using the model of a unified semantic knowledge and data space is argued.

1. Introduction
The field of information technology is very dynamic and the trend towards distributed systems of a network structure has been the most noticeable in recent years. The components of such structures are autonomous objects of various types (physical, virtual, social), each with built-in computing and communication capabilities. Typically, these objects are relatively simple in structure and functionality. However, due to the large number of objects in the system and intensive interaction, they are able to solve the most complex problems. Typically, such systems use such concepts as the Internet of Things, Cloud and Edge Computing that are currently the basic concepts for creating large-scale distributed systems. In such systems, distributed objects communicate with each other in the Internet environment, actively use its information, computing and communication resources, forming a complexly structured network object existing in a virtual environment to a significant degree. Applications of this class occupy more and more space in the modern digital world and play an increasing role in the economy, industry and the social sphere. They also ensure that the society is generally safe.

Those applications that operate in real time in a competitive external physical and/or virtual environment stand out in terms of complexity. This provides targeted opposition to their normal operation. Examples of such applications are competition models present in international markets, which often go beyond purely economic or market relations. This opposition of complex dynamic systems of the network structure in peacetime is most clearly observed in cyberspace. For more than 20 years, cyberspace operations aimed to violate the security of critical computer systems, the integrity and confidentiality of the information contained in them [1, 2] have been observed. Similar large-scale operations can be observed in the field of information wars and in the form of psychological influence on one or another population group of one or several countries simultaneously. It can be argued that technical, military and other types of intelligence actively use the resources of the virtual space and work in an environment with information security opposition.

The external environment with conscious opposition affects all aspects of a network object - its architecture and interaction models, behavior models and behavior control principles of autonomous
network objects and general system behavior. Such an external environment usually sets strict requirements for the structure of the information space of data and knowledge, for the structures of their presentation, storage, and for the mechanisms of use. In essence, a network object in an environment with resistance is a new class of objects and control problems, which is still poorly studied even at the conceptual level.

The relevance of tasks of such nature is constantly growing. It increases with the growing role of Internet technologies in various aspects of social and personal life, as well as due to the fact that various forms of competition and confrontation in the economy, politics, in the military sphere, ecology, computer and information security, and in other critical areas of public life are increasingly transferred to the virtual space and are practically implemented at the digital level.

The purpose of this work is a comprehensive analysis of distinctive behavioral models featured by network structure objects in an environment with resistance and the models of their control. The model of behavior and control under consideration is based on the control strategy provided by the Boyd cycle. Therefore, the features of objects of the network structure with purposeful behavior in an environment with resistance are analyzed first. Then the importance of the multi-agent architecture is substantiated for the specification of the behavior model of the interacting autonomous system objects and the model for managing them, as well as for the specification of the digital twin model of the opposing external environment in the control loop. Furthermore, the features of the models that support the implementation of the functions performed by the system at different phases of the Boyd cycle are analyzed, in particular, when assessing and predicting situations and when making decisions in the interests of implementing goal-directed behavior taking into account the opposition. The need to use the model of a unified semantic space of knowledge and data is substantiated. A separate section discusses the main features of the described problem and the proposed models. In the conclusion, the main results are summarized.

2. Control Features and Digital Twins in Counteraction Environments

In systems operating in a countermeasure environment, all processes are tied to real time and the timing of the system is critical. In accordance with the model of competitive interaction of two systems proposed by J. Boyd [3], competing systems have a standard closed cycle of functioning, consisting of four phases, namely observation, orientation, decision, action. In Figure 1, these phases are called Perception, Situation Assessment and Forecasting, Decision Making, Implementation of the Decision (Actions), respectively. This model, originally proposed for modeling and managing military operations, turned out to be workable in the abstract class of competitive systems interaction. This model is also adopted in this work.

Fig. 1. Boyd's cycle in a real-time control system

In the Boyd's model, each of the competing parties must structure their behavior in accordance with such a cycle, and each party must track the change of phases of the competitor's cycle in order to work successfully. The interaction of the competing parties takes place in real time, therefore, from the
From the standpoint of control processes, each party has to be a real-time system. Numerous studies and practical applications of the Boyd's model have shown that in this model of competition, the advantage is gained by the party that, along with qualitative models for assessing the situation and making decisions, spends less time on the implementation of the cycle and implements actions in accordance with the decision made earlier.

Naturally, in order to implement the Boyd's strategy, each side must have its own model and means of tracking the opponent's actions, which allows it to predict the phases of the opponent's Boyd cycle, make a forecast of his behavior, assess the situation from the standpoint of the attainability of its own goals in the context of the constructed forecast (within own Boyd's Cycle) and further develop (correct) your own decision and execute it. The described operation cycle of the network object control system sets the control scheme (meta model) in the real time, and also determines the requirements for the content of information on the opponent and for the rate of its updating.

Currently, the adversary's model is commonly referred to as its digital twin (DT). In general terms, a DT is understood as a physical object, system or process and as a certain set of its digital computer models and processes and/or services that implement them. The DT of some object, system or process is a computer program designed to numerically simulate the behavior of this object, system or process in order to assess its current state and/or behavior and predict these characteristics for a certain future horizon. Typically, information from the DT is used in conjunction with information that can be obtained at the same or close times about a real object, system or process. The same information is usually used to refine the model of the external environment DT by the discrepancy of its forecast for a certain point in time and the real state of the object, system or process, the model of which is represented by the DT. In short, the DT is used in a control loop with a customizable behavior model of the external environment and the adversary.

In the considered class of applications, the DT predicts the behavior of the adversary and the external environment in the Boyd cycle meta model, i.e. predicts the start and end times of the opponent's Boyd cycle phases and his behavior within the cycle phases. This information is then used in real time by the network entity to assess and predict the situation and make a decision in its own Boyd cycle. Thus, the DT of the behavior of the adversary and the external environment is an integral part of the advanced planning and control system for real-time systems in an environment with opposition [4, 5]. Figure 2 illustrates the place of the external environment DT and the behavior of the adversary in the process of control according to the Boyd cycle in a competitive environment.

In general, the mathematical model of a physical object that was built for the abovementioned purposes can be extended with other models, for example, machine learning models of the DT itself. It is customary to call such a DT a smart digital twin.

![Boyd cycle in a real-time control system with a DT of environment and opponent](image-url)

The term DT, generally speaking, is a metaphor, since no digital program can describe a physical object or the processes occurring in it in an exhaustive way and absolutely accurately copy the behavior of the object. For this reason, the DT always only approximately describes a real object, and...
it never describes all its aspects, and therefore it is not a double of a physical object in the usual sense. In the considered class of applications, such DT accuracy is impossible in principle, since the external environment is indefinite, and the information about the opponent available to an object, system or process is always incomplete, especially for solving forecasting problems.

The DT is always built so that it can be used to solve a specific set of applied problems, which ideally can be expanded by the user if necessary. For this reason, the DT is usually built on a modular basis. Each virtual (digital) module solves one or more applied problems from a given set, and the set of these modules fully covers a given set of use cases for the DT. In such a digital model, each of its individual components models one or another aspect of a physical object, and at the same time has a clear meaningful interpretation. Each such application module of the DT, displaying one or another aspect of a physical object or process, is usually called a digital shadow of a physical object or a process generated by it. It is important to note that currently there are other somewhat alternative definitions of the digital shadow concept.

3. Multi-Agent Architecture

The real-time systems of the described class solve the problems of the goal-directed group behavior control for an interacting set of various autonomous objects. Behavior control in such systems is called group control.

Currently, among the behavioral models of distributed objects, the leading place is held by multi-agent models and architectures. The first behavioral model of an agent and a multi-agent system was the model of R.A. Brooks, proposed in [6]. It describes the hierarchical architecture of a reactive agent (subsumption architecture), which specifies many options for its behavior (behavior patterns) depending on the state of the external environment. However, this architecture turned out to be unsuitable for formalizing group behavior and group control due to its weak expressive capabilities. Modern behavioral systems consider more complex models of behavior patterns and use the model of a state machine with an internal state for considering the history and current context.

Let us recall that it is customary to call an agent an autonomous computer program (system) capable of goal-directed behavior in a dynamic, unpredictably changing external environment. The key properties of an agent are its autonomy and the ability for goal-directed and proactive behavior. The agent autonomy is understood as its ability to work to achieve a set goal without the intervention of a person or other systems and at the same time to exercise self-control over its actions and internal state. The ability for proactive behavior means that when choosing a behavior, an agent takes into account not only the information it received from the outside world, but also the prehistory of its own actions and states of the external environment, which are reflected in his current internal state. For example, an agent can generate messages and send them to other agents even if there are no input events. For example, it can generate timeout violations. Let us note that the ability of an agent to behave proactively is the main difference between a computer program called an agent and a program called an object in object-oriented programming [7].

A multi-agent system (MAS) is defined as a network of loosely connected solvers of particular problems (agents) that exist in a common environment and interact with each other to achieve certain general goals of the system and/or particular goals of individual agents. Interaction can be performed by agents either directly – by messaging, or indirectly, when some agents perceive the presence of other agents through changes in the external environment with which they interact. The exchange of messages presented in a certain language is carried out either via dialogue or via protocol. The ability to exchange messages is another important distinguishing property of agents, which is called interactivity. Interactivity is defined as the ability of agents to exert some influence on each other, and it is in this sense that the network of agents in the MAS is called weakly connected. “From interactivity and autonomy follows the ability of a group of agents to generate solutions to complex problems with the help of side effects from joint coordinated actions” [8]. As is evident from the agent's property description, the agent’s and the MAS model are ideal for formalizing a system consisting of a set of autonomous objects coordinating their group behavior. In the MAS concept, each
autonomous network object is associated with an autonomous software agent. This agent, on the one hand, controls the internal behavior of the software and/or hardware components of the network node in various use cases (when the network node performs one or another role assigned to it). On the other hand, this agent is a representative of its node in the outside world, i.e. in a network of software agents, where its main function is to interact with software agents of other network nodes by exchanging messages in order to coordinate the individual behavior of its node in various tasks and subtasks that are solved jointly by a certain group of network nodes. This behavior is commonly called the agent's external behavior. Typically, external behavior is, in a certain way, an ordered sequence of messaging in the form of dialogues between a pair of agents or protocols.

It is important to note that at present, in the considered class of applications for the MAS model and technology, there are simply no alternatives, and this applies to both the models of group behavior and group control of an autonomous agent network, and the model of the digital twin of the rival and, possibly, the rest of the external environment.

In a distributed system, a mandatory component is a software and communication environment, usually called a digital platform which plays the infrastructure role that transforms many nodes of the agent network into a system with the ability for agents to interact (exchange messages) with each other. The infrastructure can support other functions (they are usually called services) available to all nodes (agents) of the network. In particular it may include an interface for accessing network agents to cloud resources and services, to information from external sensor networks, for example, the networks of the hydrometeorological service and its services for obtaining current and forecast weather data. Platform services can also include the functions of creating, storing and providing access to distributed data and knowledge presented in the network. One of the options for building such an infrastructure is considered in [9].

4. Concept of Situation and its Assessment in the Boyd Cycle

In accordance with the Boyd's cycle, after collecting available information about the outside world (such information will always be inaccurate, incomplete and possibly contradictory), the task of the system is to assess the current situation in the external environment with opposition, as well as to assess the situation in its own network of agents and propose its development. In complex real-time behavioral systems operating in an environment with opposition, the concept of a situation is considered as the most important concept that is the subject of special research. These studies have been carried out approximately since the beginning of the 1990s in relation to military action [10 - 13].

There are various meaningful interpretations of this concept, and they are usually given in comparison with the system state concept. In [10 - 13], the situation is defined as some dynamic characteristic of a complex controlled system in the external environment, which is described using a model of previous, current and future occurrences. A situation is a dynamic entity, which is described not only by a set of current states of distributed system and external environment objects but, most notably, by the set of relations assigned to them. This is the fundamental difference between a situation model and a state model of a complex system.

The concept of a situation is proposed for an integral description of processes in the system and in the external world from various points of view, and first of all, from positions that, in turn, characterize these processes from the standpoint of the attainability of its goals. The set of possible situations in the system and in the external world as assessed from a certain specific point of view is usually factored, so that an expert operates with qualitative situation assessments from each point of view, i.e. with classes of situations. For example, for a purposeful system, it is of paramount importance to assess the situation in terms of the attainability of the set goal, taking into account its current state, the available resources and actions of the competing party and its intentions (its predictable actions, for example). From this point of view, the entire set of situations can be divided into four classes, namely (1) the goal is achievable, (2) the goal is unattainable, (3) the goal is no longer relevant and (4) the goal is achieved. Situation evaluations can be based on revealing the rival side intentions, assessing one's own performance based on the technical diagnostics and available
resources, etc. However, all these factors are ultimately further used to assess the attainability of the group goal of the system and the corresponding optimization of its future strategy behavior.

The task of building a formal model of the situation, as well as algorithms for assessing the current situation from different points of view, which together will allow the system to assess the goal attainability, as well as to build a specific strategy for achieving it, is one of the modern complex tasks of group management for the behavior of a set of autonomous network objects. On the other hand, it corresponds with a Boyd Cycle phase which structures the system behavior in the context of the competing party’s actions. A model of a situation and algorithms for its assessment is one of the urgent tasks of complex systems, which is currently the subject of active research and development.

5. Boyd Cycle Decision Making

The decision-making process belongs to the third phase of the Boyd cycle (Fig. 1), and its essence is the search for control of the group behavior of the autonomous objects of the system. Usually, group behavior is understood as a structured set of individual behaviors of autonomous system objects, coordinated in the interest of achieving a group goal.

Formally, the model of goal-directed group behavior is described using two components. The first is the group action scenario. In many cases, it is specified in the form of a partially ordered set of actions that a group of autonomous objects must perform in order to reach their target state. This component does not indicate specific performers of actions, since the decision on the performer appointment can change during the system operation. The second component of the formal model of group behavior is the specific assignment of the action performers (group objects) for the scenario. This component is part of the control action and can be changed during the execution of the action script. A scenario of actions with assigned performers is called a group behavior scenario.

An action scenario usually has a hierarchical structure in which a scenario or pattern of behavior can be substituted into the scene of action, or this action can be elementary (indivisible). In other words, every script action at any level of abstraction can, in turn, be represented by a script with a hierarchical structure. It is believed that an elementary action should be performed by one network object. Formally, the scenario model can be described as follows:

\[ S = \{X, >\} \]

where \( S \) is the script identifier; \( X = \{X_i\}_{i=1}^N \) is the set of actions \( X_i \) of the script \( S \); > is the order relation on the set of script actions which is specified by the condition

\[ X_i > X_j, \text{ if action } X_j \text{ can be performed only after completion of action } X_i. \]

Taking into account the introduced definitions and assumptions, decision making in the Boyd cycle for a network object of the class under consideration at the initial moment of time consists of (1) forming a scenario for achieving the goal and (2) distributing the scenario tasks on a set of autonomous system objects. Formally, these problems are well known from literature as components of the action planning problem, and a lot of approaches, models and algorithms have been proposed for its solution. An algorithm for advanced scheduling in real time in the context of the Boyd cycle is proposed, for example, in [4, 5].

In a real-time system operating in an environment with opposition, the constructed plan (a scenario of group behavior with specific performers of its actions) may need to be periodically adjusted depending on the behavior of the adversary and the dynamics of the situation in the system itself and in the external world. In practice, in the considered class of applications, the action planning problem should be solved as a feedback control problem which must be solved periodically, depending on the obtained situation assessment [4, 5]. In some formulations of this problem, it is assumed that the scenario of actions is set rigidly, and then, depending on the constructed assessment of the situation, it is required to recalculate only the distribution of autonomous network objects on the set of actions in the scenario.

Thus, the formal statement of the decision-making problem in the third phase of the Boyd cycle can be formulated as follows:
Given:
1. A planning horizon \([T_0, T_K]\), where \(T_0\) and \(T_K\) are the (calendar) start and end times of the execution of the group behavior scenario, and the \(T_K\) value can set the latest admissible end time for the execution of the group work scenario or, in setting the task not set.
2. A group action scenario (1), i.e. a partially ordered set of actions \(X\), which must be performed on a given time horizon to achieve the set goal.
3. A set of autonomous objects, each of which is assigned a list of elementary script actions that it is capable of performing and the object's availability time intervals.
4. Optionally, a set of specific subject-oriented restrictions for the use of autonomous objects on a set of script actions (for example, by technical condition, by weather conditions, etc.).
5. The quality indicator of the planning problem solution \(F(P(X, D([T_0, T_K])))\), where \(P(X, D([T_0, T_K])))\) is the required distribution of the set of autonomous objects networks \(D([T_0, T_K]) = \{D_j([T_0, T_K])\}_{j=1}^M\) on the planning horizon \([T_0, T_K]\) over a set of actions \(X = \{X_i\}_{i=1}^N\) scenario.

It is required to find:
The distribution \(P(X, D([T_0, T_K])))\) of autonomous objects on the set of scenario actions, taking into account all imposed constraints, which optimizes the control quality index \(F\).

In the last phase of the Boyd's cycle, the decision is executed. This problem component of autonomous objects group behavior is largely subject-dependent and therefore is not considered here. The main requirement for it is that the execution of actions in each phase should, as far as possible, outpace the actions of the opposing side in a similar cycle.

6. Data, Knowledge and Unified Semantic Space
The core of the system for situation assessment and decision-making in the described class of behavioral systems is formed by knowledge, and in the considered class of applications, two types of knowledge are to be considered:
1. **Knowledge that is used in the DT** (Fig. 2). This is knowledge about the opponent and about the external environment, which allows us to assess the current situation in the external world (in the external environment with counteraction) based on the data obtained, assess the goals and intentions of the opponent and predict his behavior on a certain horizon of the future time.
2. **Knowledge that is used by a network of autonomous agents** to assess the situation in the context of the attainability of their own goal, as well as to make decisions on managing their own behavior (Fig. 2). It bears reminding that group management is the point at issue, which includes planning the scenario of group behavior and controlling its execution.

The main new problems related to the acquisition, presentation and use of knowledge in the considered class of applications are due to the distributed nature of the system. The coordinated work of all its distributed autonomous components (network agents) is possible only if all these entities operate in a single information space, and for the coordinated work at the knowledge level, this information space has to be organized in accordance with an ontology common to all agents.

The creation of such a unified semantic space of knowledge and data for an object consisting of a set of autonomous distributed entities is an urgent task in the field of knowledge representation and integration. This challenge is currently under active research and development. In accordance with the modern trends, the most promising is an integral knowledge model built using the concept of LPG (Labelled Property Graph) structures, which are implemented in graph database management systems (graph DBMS). At present, most of these DBMSs are developing in the concept of open source software, and therefore this development is proceeding quite quickly. In a graph DBMS, an ontology schema is represented by a graph with rich expressive capabilities. The search for answers to queries is performed using graph search procedures, and these procedures can be implemented efficiently, which is very important for real-time systems with complexly structured knowledge bases.
In Fig. 3 knowledge and data bases that form a single information space of an object of a network structure and a DT of the external world in an environment with resistance are presented in the context of tasks solved at different stages of the Boyd cycle.

![Diagram](image-url)

**Fig. 3.** Knowledge and data bases that form a unified information space of a network structure object and a DT of an environment with opposition

7. **The Results**

It is evident from the abovementioned that the model of group behavior and the model of group control of a distributed object of a network structure operating in an environment with opposition has a number of features that greatly distinguish them from similar models of a network object operating in an environment without opposition. Briefly, these differences and features are as follows:

1. The *real time* factor and the control phase implementation rate in the Boyd cycle are the key features of the model for such an object. The time frames for the execution of the scenario of group behavior as a whole, as well as the planned time intervals for the execution of work by specific performers, can be tied to real time. This binding makes the modeling and control system very sensitive to violations of the specified times during the script execution. It is obvious that the delay of any work will cause an avalanche-like time shift in the admissible start times of all subsequent operations of the script, associated with the delayed work by the relation of succession in time and their transitive closure in the partially ordered set of script actions. In these conditions, the accumulated time lag can make the current plan inoperable, and the distribution of the script actions executors will need to be recalculated.

An even more significant influence on the behavior of an object can be exerted by the counteraction factor. This factor rigidly sets the pace of the execution of the phases of perception and information processing for assessing the situation, as well as making decisions and executing it in accordance with the Boyd cycle. Indeed, an object can hope to achieve its goal in the face of opposition only if it spends less time on the cycle than its rival. This means that in the Boyd's cycle, modeling and control must be performed using algorithms that are ready to issue a final solution at any time (*anytime algorithm*), dictated, in fact, by the rival side.

2. The presence of a digital twin of the rival party in the control loop is mandatory. The composition of the digital twin's attributes, the speed and accuracy of estimating and predicting their values play a decisive role for the successful achievement of the goal for the object under control.

3. The problem of vector situation assessment (i.e. its assessment from different points of view) in real time is relatively new. In the control processes, with a given rate of updating this assessment in the context of the reachability of the target state by the distributed control object.
4. The scenario of agent group behavior is a dynamic concept, and its update may be required at each new Boyd cycle. For this reason, the control system must have a multitude of possible scenarios for actions, for which the assignment of performers can be carried out only in real time, depending on the current situation. In this regard, the progress in the tasks of group behavior control for an autonomous entity set will be associated with success in creating models of scenario knowledge bases and mechanisms for reasoning about behavior in them. The existing models are still based only on the mechanisms for selecting scenarios from a given set of them for a finite set of situations [14].

5. The model of a network object operating in an environment with opposition, as well as the model of a digital twin of an adversary, require the use of specific formalisms. Some features of such formalisms are formulated in [1, 2]. The main one is due to the fact that it is impossible to plan the scenario of actions and the executors of its work from beginning to end in such applications in advance. When working in the environment, only the intentions of the controlled object can be constant. But the same cannot be said about the intentions of the rival, whose model in the control loop is represented by his DT. In fact, the scenario of the group behavior of the network control object should be generated dynamically, with defining the scenario of object group actions one or several steps ahead. As a possible formal model of the dynamic scenario generation of purposeful behavior, the works [1, 2] consider the model of an attribute formal grammar. However, the model proposed in these works solves the problem of dynamic planning for a single object and a multistep scenario performed by it. In the case under consideration, this problem needs to be solved in a class of distributed objects, and therefore such a model needs significant revision. In [15], to formalize such models, an algorithm was applied based on a search on AND-OR graphs.

6. A special task in the problem under consideration is the task of creating a single semantic space of knowledge and data in conditions when all agents can work with their data sources. To maintain their mutual understanding in the execution of the scenario of group behavior, all agents of the team must use a common, unambiguously interpreted conceptual basis, as well as an unambiguous interpretation of relations defined on the set of these concepts. This places special demands on the (distributed) ontology. The technologies for creating such a unified semantic space are still far from perfect. There are other features of the models and the required algorithmic support in systems of the considered class. For example, the dynamics of autonomous objects in the context of wireless communications can lead to dynamics and loss of connectivity in the communication network that supports the exchange of messages between agents, and such situations can be purposefully created by the opposing party. These features are to be studied.

8. Conclusion
The paper examines the features and conceptual foundations of group behavior modeling and control in complex systems consisting of a large number of various autonomous objects, when a significant part of these objects and the system control components exist in the virtual environment of the Internet which in one form or another counteracts its normal functioning. Systems and tasks of this type were studied at the initial stage of the Internet and the Internet technology development. However, they have become really relevant at the present time, when the Internet technology has matured and become the technology of mass use.

The main focus of the work is on the description of the key components and features of such systems. In fact, a constantly growing multitude of applications of this class forms a new class of applications, which is still poorly studied. A goal of this work is to attract the attention of researchers to the abovementioned problems and emphasize the need to address those of them that are urgent.

References
[1]. Gorodetsky V.I. and Kotenko I.V. Teamwork of agents-hackers: the use of multi-agent technology for modeling distributed attacks on computer networks// Proc. of the VIII National Conf. on AI (KII-2002). M.: Physmatlit, 2002. (in Russian)
[2]. Gorodetsky V.I. and Kotenko I.V. Conceptual foundations of stochastic modeling in the
Internet/ Proc. of the Institute for System Analysis RAS. Vol. 9: Fundamentals of information technologies and systems. Ed. by S.V. Emelyanov. M.: URSS, 2005. (in Russian)

[3]. OODA loop/ https://en.wikipedia.org/wiki/OODA_loop (access 15.10.2020)

[4]. Panteleev M.G. Formal model of advanced iterative actions planning for real-time intelligent agents// Proc. of the XIV National Conf. on AI (KII-2014), Kazan, 2014, Vol.1, P.323-333. (in Russian)

[5]. Panteleev M.G. Advanced Iterative Action Planning for Intelligent Real-Time Agents. Procedia Computer Science, Vol. 150, 2019, P. 244-252.

[6]. Brooks R.A. Intelligence without Reasoning// Proc. of International Joint Conference on Artificial Intelligence (IJCAI91), 1991, Vol. 1, P. 569–595.

[7]. Odell J. Objects and Agents Compared. Journal of Object Technology, vol. 1, no. 1, May-June 2002, pp. 41-53. http://www.jot.fm/issues/issue_2002_05/column4/ (access 15.10.2020)

[8]. Ferber J. Foreword. In Cossentino, M., Hilaire, V., Molesini, A., Seidita, V. (Eds.). Handbook on Agent-Oriented Design Processes. Springer. 2014. P. V – VI.

[9]. Gorodetsky V., Larukchin V., Skobelev P. Conceptual Model of Digital Platform for Enterprises of Industry 5.0. In I. Kotenko et al. (Eds.): Proc. of the International Conf. on Intelligent Distributed Computing (IDC 2019), Studies in Computational Intelligence, vol. 868, P. 35–40, 2019.

[10]. Lambert G.A. Grand Challenges of Information Fusion // Proc. of International Conference on Information Fusion, Cairns, Australia, 2003. P. 213–220.

[11]. Steinberg A.N., Bowman C.L., White F.E. Revisions to the JDL Data Fusion Model/ AIAA Missle Sciences Conference of Naval Postgraduate School, CA, 1998.

[12]. Fischer Y. On Situation Modeling and Recognition// Proc. of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory. Ed.: J. Beyerer, 2009. P. 203-215.

[13]. Gorodetsky V.I., Samoylov V.V., Trotskiy D.V. The basic ontology of collective behavior and its extensions// Izvestiya RAS. Theory Systems for Control. 2015, No. 5, P. 102-121.

[14]. Fedunov B.E. Onboard intelligent systems of tactical level for anthropocentric objects M.: De Libri, 2018. 246 P.

[15]. Gorodetsky V.I., Samoylov V.V., Trotskiy D.V. Facilities for specification and instrumental support of command behavior of autonomous agents// Bulletin of the Southern Federal University, № 3, 2011, P. 116–133.