Distributed stream data processing system in multi-agent safety system of infrastructure objects

S S Valeev[1], N V Kondratyeva[1], A S Kovtunenko[1], M A Timirov[2] and R R Karimov[1]

[1]Ufa State Aviation Technical University, K. Marx Str., 12, Ufa, Russia, 450008
[2]Bashkir Production Association "Progress", Kommunisticheskaya str., 23, Ufa, Russia, 450076

e-mail: vss2000@mail.ru

Abstract. The solution of the problem of resource management in distributed computing systems of processing stream data in safety systems of distributed objects is considered. The tasks of streaming data processing in a multi-level multi-agent evacuation system in an infrastructure object are considered. The features of the mathematical model of a distributed stream data processing system are discussed.

1. Introduction

An infrastructure of socio-technical systems includes various complexes of buildings that have a large area and the possible presence of a large crowd of people. The goal of risk management with the security system of an infrastructure is minimizing collective risk in the event of a critical situation in the cyber-physical system or in the context of infrastructure degradation. For the analysis of the state of these objects, various systems for collecting and processing information are used: sensors for the state of physical parameters, video surveillance systems and telecommunication systems. It should be noted that the part of the information is transmitted and processed as streaming data. In the event of a critical situation information about the state of the information field can be lost due to factors affecting the elements of the computer system. So we can lose the accuracy of measurements of the physical parameters of the internal space of the infrastructure [1-9].

So, let the observable information field be the state of the internal physical and information space of the considered infrastructure. In this case, the task of reallocating computing system resources in a technical safety system, in which information about the state of an infrastructure object is collected from various sensors in real time, becomes extremely important. This system should be considered in the class of self-learning control systems for distributed objects that have a variable structure [10-15].

The state of the information field may change depending on internal and external factors affecting the physical energy flow, which, in turn, affects the distribution of risk levels in the event that this energy goes out of control. In this case, the solution to the problem of risk management can be based on the application of self-learning elements of a multi-level management system at its planning and coordination levels. The formation of scenarios for the development of a critical situation and possible ways to minimize risks is possible with the use of a self-learning multi-agent data collection system.

The considered multi-agent system can be implemented in various ways: we can use a multi-agent software platform, such as JADE, or implement software agents based on different programming languages: Python, Java, etc. [16-18].

Although the databases can be implemented on the basis of various modern approaches, it is necessary to solve the tasks of reallocating computing power in real time in order to ensure the efficiency of solving problems of collecting and processing, including streaming data, in the safety systems of the distributed infrastructure facility [19-28].

As it is known, modern airports and other similar structures are complex socio-technical systems. For example, the system for ensuring the process of evacuating passengers in critical situations includes technical means of detecting and responding to the alarms, various sensors, computing, telecommunications and human resources of engineering services and building security systems. In addition, it is necessary to take into account the many uncertainties that affect the effectiveness of decision-making procedures in the evacuation process. Currently, the solution to the problem of reducing the risk of evacuation is still an important scientific and practical task.

Information support in critical situations based on a decision support system can serve as a risk management tool. In this regard, it is advisable to develop an intelligent decision support system (DSS) based on self-learning technologies and multi-agent modeling using streaming information processing.

2. Architecture of intelligent decision support system

It is quite difficult to ensure effective decision making in critical situations for large socio-technical systems under the conditions described above, using a centralized control scheme. One of the solutions is the use of a three-level decision support system, which allows managing the complexity of the decision-making problem and, as a result, reduce risk levels (figure 1). Upstream information from sensors I_1 from the control level of sensors and actuators is transmitted to the level of online solutions. In the opposite direction, control actions are transferred to actuators and technical equipment that support evacuation management processes I_2.

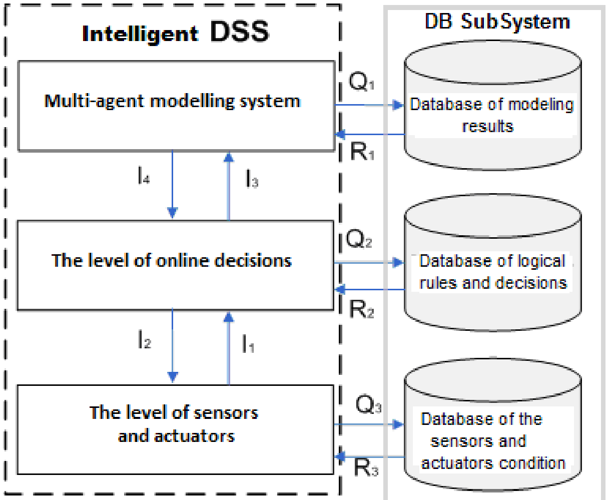

Intelligent DSS
DB SubSystem
Multi-agent modelling system
Q₁
Database of modeling results
R₁
I₄
I₃
The level of online decisions
Q₂
Database of logical rules and decisions
R₂
I₂
I₁
The level of sensors and actuators
Q₃
Database of the sensors and actuators condition
R₃

Figure 1. The information structure of intelligent decision support system.

A database storing information transmitted from sensors and actuators receives a stream of data Q_3 on the state of the information field and elements of the executive level. Information R_3 is transmitted in the opposite direction, which is necessary for the effective functioning of sensors and instruments.

Decision making at the level of online decisions in real time is based on a set of specified logical rules R_2 received in response to Q_2 requests. At the same time, the database stores information about the effectiveness of decisions made on the basis of Q_2 rules. The flow of data on the state of the critical situation I_3 is transmitted to the multiagent modeling level, which, in turn, is looking for effective solutions for transmitting information I_4 to the operational management level.

The data on the simulation results are transferred to the database of simulation results. If necessary, they can serve as input for the next stage of the simulation.

The risk of inefficient evacuation was adopted as a criterion for the effectiveness of an intelligent decision support system:

$$Q_{\mathrm{E}} = 1 - P_{\mathrm{E}}, \tag{1}$$

where P_{E} is the probability of a successful evacuation, for example, in the event of a fire.

Probability of escape P_{E} is defined in accordance with standard methods of fire risks evaluation in accordance with the requirements of GOST 12.1.004-91 “Fire safety. General Requirements” as follows:

$$P_E = \frac{0{,}8 \cdot t_{bl} - t_{est}}{0{,}8 \cdot t_{bl}},\ t_{est} < 0{,}8 \cdot t_{bl}, \tag{2}$$

where t_{bl} is the time of blocking the building exits, it depends on fire conditions, building configuration, etc. Also, as it is given in GOST 12.1.004-91 “Fire safety. General Requirements”, t_{bl} = 288 sec. in considered case; t_{est} is the estimated evacuation time determined for each of the best evacuation route. Estimated evacuation time t_{est} depends on information flows presented on figure 1 and, therefore, on the adopted solutions in an emergency evacuation. It also can be obtained on fire and evacuation integrated modeling results:

$$t_{est} = f(I_1, I_2, I_3, I_4, R_1, R_2, R_3, Q_1, Q_2, Q_3). \tag{3}$$

It should be noted that figure 1 shows only a generalized information structure of the hierarchical decision support system. In real techno-social systems, every single information flow is more powerful, all of them are transmitted in real time, and each of them by itself has a very complex information structure. With this in mind, the practical implementation of intelligent DSS requires high performance computing technologies and smart integration of hardware and software.

3. Multiagent system of infrastructure object safety

A set of agents of the coordination level use the information received by the system planner to solve the task of minimizing risks through interaction mechanisms in multi-agent systems. At the level of coordination, among other things, the problem of ensuring the integration of information on the state of the control object using multi-agent sensor systems must be solved, which makes it possible to minimize errors of the first kind and of the second kind due to the effect of self-organization of different sensors. At this level, the problem of developing control actions on the ventilation system, fire protection system, information support for evacuation must be also solved.

The architecture of multiagent system of cyber physical system safety is presented in figure 2. The structure of this system is based on hierarchical architecture.

In the sense of Big Data concept here we have different flows of information and different formats of data. We need to gather information from different types of sensors and store these data in Databases: DB_Video – Data Base of Video Sensors Data, DB_Sensor1 – Data Base of Infrastructure Sensors, DB_Sensor2 – Data Base of Dynamic Objects, DB_BD – Data Base of Mix Content, DB_M – Data Base of Simulation Results Data.

Multigent system of safety of cyber-physical system includes: Agent_Image_An – agent of Video Stream Flow Analyzer, Agent_Dy_Objects – agent of Dynamic Objects trajectory Identification, Agent_Inf_An – agent of Infrastructure State Identification, Agent_Integ_Date – agent of Integration of different flows from sensors, Agent_Mod – Agent of Simulation of the different Scenarios of Critical Situations, Agent_Integ_Dm – Agent of Analysis the state of Cyber-Physical System, Agent_R – agent of risk field identification, Agent_F – agent of integrated risk identification, Agent_U – agent of control and management of minimizing risk, Agent_U_I – agent of control signal generation for actuators, Agent_U_S – agent of management decisions realization, C&MC – implementation of control and management system.

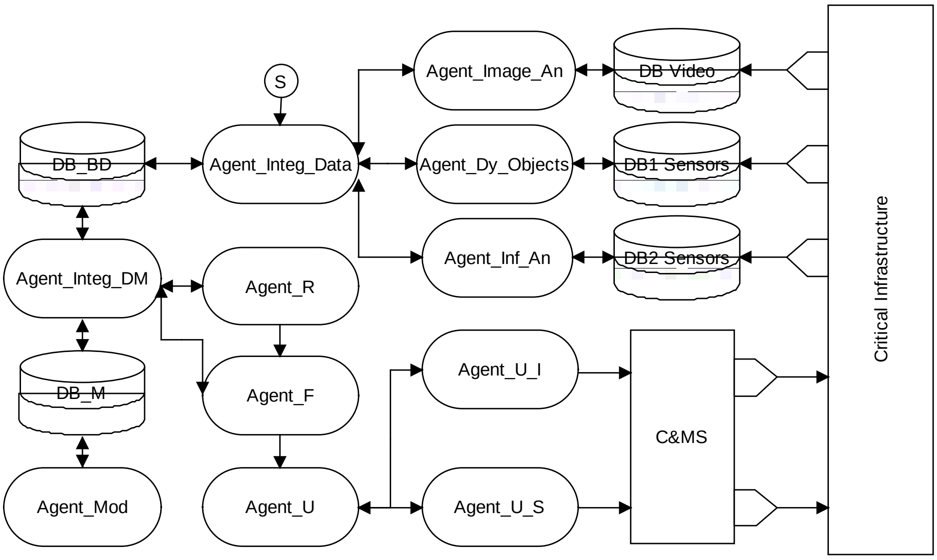


Figure 2. Multiagent system of infrastructure object safety.

So, here we have the massive application of different streaming data flows and we need to solve the resource allocation problem in case of degradation of the information system.

4. Formulation of the optimal resource allocation problem

The model of the processing and storing of streaming data (PSSD) based on the computer network includes resource model of the computer network *N*, model of data processing system (DPS) and the relation *D* describing system deployment in the network.

$$IS = \langle N, DPS, D \rangle . \tag{4}$$

Here:

- *N*– the computational network,
- *DPS* – the PSSD model,
- *D* – the deployment relation that shows $\forall p \in P$ on what node it physically deployed.

$$D = \{(w, p) : w \text{ executes } p;\ w \in W,\ p \in P\}. \tag{5}$$

Let $B : W \times R \to \mathbf{R}^+$ – function that shows $\forall w \in W$ amount of the involved resource. Values of the function can be calculated from resource capacities of data processing model and network model as follows.

$$B_{cpu}(w) = \sum_{\forall p \in D(w)} \frac{V(p)}{T(p)} . \tag{6}$$

$$B_{mem}(w) = \sum_{\forall a \in (D \circ C)(w)} M(a) . \tag{7}$$

$$B_{net}^{in}(w) = \sum_{\forall p \in D(w)} \frac{\sum\limits_{\forall a \in I(p) \cap A \backslash (D \circ C)(w)} M(a)}{T(p)} . \tag{8}$$

$$B_{net}^{out}(w) = \sum_{\forall p \in D(w)} \frac{\sum\limits_{\forall a \in I(p) \cap (D \circ C)(w)} M(a)}{T(p)} . \tag{9}$$

The introduced notation makes it possible to formulate the problem of optimal allocation of computational resources in PSSD as the problem of combinatorial optimization. It is necessary to find such deployment relation *D**, that gives the maximum to predefined efficiency function *F*.

$$D^* = \arg \max_{D \subset W \times P} F\left(IS\langle D\rangle\right). \tag{10}$$

The following constraints apply

$$B_r(w) \le S_r(w). \tag{11}$$

Let *rem(w)* – the unitless function, that shows average free resources for every node as

$$rem(w) = \frac{1}{|R|} \sum_{\forall r \in R} \frac{S_r(w) - B_r(w)}{S_r(w)}. \tag{12}$$

and the average total of free resources as

$$\langle rem \rangle_W = \frac{1}{|W|} \sum_{\forall w \in W} rem(w). \tag{13}$$

The mathematical problem of HCNUL resource allocation can be formulated as follows. Subject to constraints (14) the optimal deployment at every point of time is

$$D(t) = \arg \min_{D \subset W \times P} \left(\max_{w \in \delta(D)} \left(\sigma_\tau^2 \left[rem(w, \tau) \right] \right) \right). \tag{14}$$

Here:

- $\sigma_\tau^2[\xi(\tau)]$ denotes the dispersion of the stochastic process ξ on the interval $\tau \in (0, t)$,
- $\delta(D)$ denotes the domain of relation D – the set of nodes where system is deployed.

5. Agent-based software architecture for PSSD systems

To create flexible, robust and scalable software that implements a multi-level security system, an agent-based approach is proposed, which involves using a software agent as a basic software object. This allows us to further talk about the software architecture based on agents and implies the presence of some additional elements in the system, such as:

- agent platform is a set of software components that support the life cycle of target agents, the interaction between them in a single namespace and the model of system events;
- agent container is a software environment that provides access to the resources of each compute node and runs software agents.

In accordance with the proposed mathematical models, security system architecture has been developed. It is assumed that the software consists of the following components of the structure:

- set of target agents, each of which is a software object that implements a procedure for processing streaming data;
- set of processing attributes implemented as an in-memory database that provides distributed storage of streaming data and access to them with low latency.

A unified storage space for streaming data is provided by a distributed cached in-memory storage that provides fast and transparent access to attribute values regardless of their physical location. The agent platform is a unified distributed execution space for data processing procedures.

The target agent is the basic software entity in the implementation of the data processing model within the proposed architecture (figure 3). It includes the following components:

- data processing procedure;
- local cache of used attributes;
- local cache of calculated attributes;
- mechanism for the management messages receiving and executing;
- mechanism for the agent initialization and self-destruction;
- agent's state.

The activity diagram of the target agent shows the process of the agent's functioning after creation. In the process of distributed data processing, the target agent can be in the following states:

- “Created” is the initial state in the agent's lifecycle.
- “Initialized” is the calculated attributes are set to the initial values.
- “Running” is the agent's operating state when the attributes are recalculating in cyclic mode.

- “Holding” is the agent's operating state when recalculating of attributes is temporary suspended.
- “Completion” is the final state in the agent's lifecycle (preparation for self-destruction, completion of all recording processes).

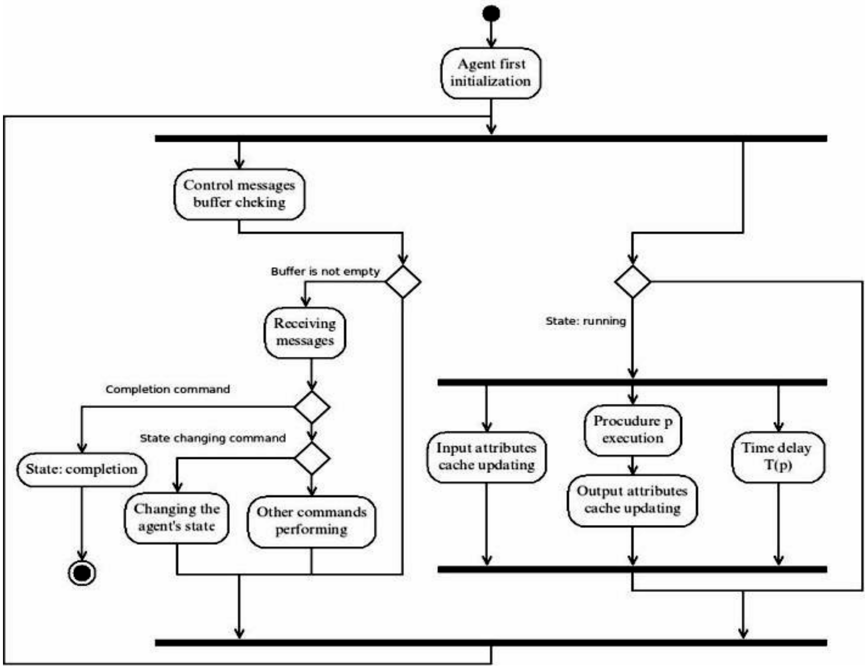

Agent first initialization
Control messages buffer cheking
Buffer is not empty
Receiving messages
Completion command
State changing command
State: completion
Changing the agent's state
Other commands performing
State: running
Input attributes cache updating
Procudure p execution
Output attributes cache updating
Time delay T(p)

Figure 3. The part of activity diagram of the target agent.

A change of the target agent’s state occurs either according to the stage of the lifecycle, or under the influence of management commands.

6. An example of implementation of the agent-based software architecture for PSSD systems

The system is implemented as a simulation model and has a graphical user environment and allows you to use the Java language to develop models. Figure 4 shows the main interface of the program with controls that provide access to each node of the required number of CPUs and Memory for the system to work; flag "Enable/disable crashes"; a slider that sets a threshold at which tasks will be redirected to a more free node; input window required resources to perform tasks; a slider designed to select a node; a flag that allows you to switch between node selection modes; graphics processor and memory usage at each node.

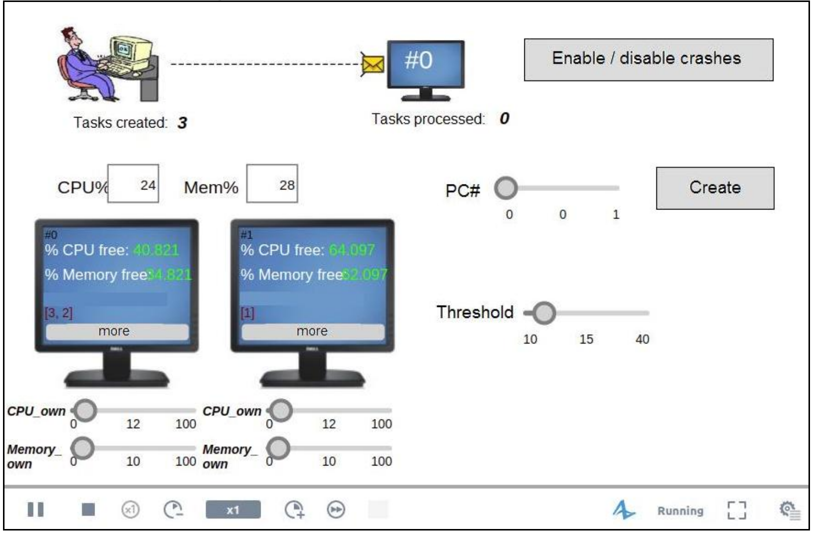
For the convenience of the user, the ability to create and assign tasks has been added. The user must specify the required amount of resource that the task will request from the node on which the tasks will be performed, then you must click the “create” button. The dependence of the expenditure of CPU resource (%) on time for a computer system of two nodes (sec) is presented on figure 5.

The simulation was carried out on the capacity of the computer network, which consisted of ten nodes. The number of tasks in the system is 25 (the number of required CPUs and Memory resources is from 5-30% of the maximum). If on the node the amount of free resource is less than the threshold value (in this case, less than 20%), then the random task will be reassigned to the node with the most resources. Also, periodically, a random failure occurs, which leads to a decrease in free resources by half. After a certain time (in this case, 20 minutes), on a random node (in this example, node number 4), the amount of free resource decreased by half. Task number 13 was reassigned from the first node to the second.

7. Conclusion

The task of the risk management in cyber-physical systems on the basis of multilevel management system is considered. The principals of designing and architecture of multilevel self-learning multiagent system for risk management are discussed. Self-learning ability of the proposed multiagent system is based on application of Big Data algorithms an machine learning algorithms.

As a result of testing software errors were not detected. To assess the quality of the developed software, failures of computing nodes were simulated. One of the handler nodes was forcibly reduced the amount of computing resources by half. At the same time, the system retained its performance; failure did not affect the functioning of other nodes.



Tasks created: ***3***

Tasks processed: ***0***

CPU% 24 Mem% 28

#0
% CPU free: 40.821
% Memory free34.821
[3, 2]

#1
% CPU free: 64.097
% Memory free62.097
[1]

Figure 4. The main program interface.

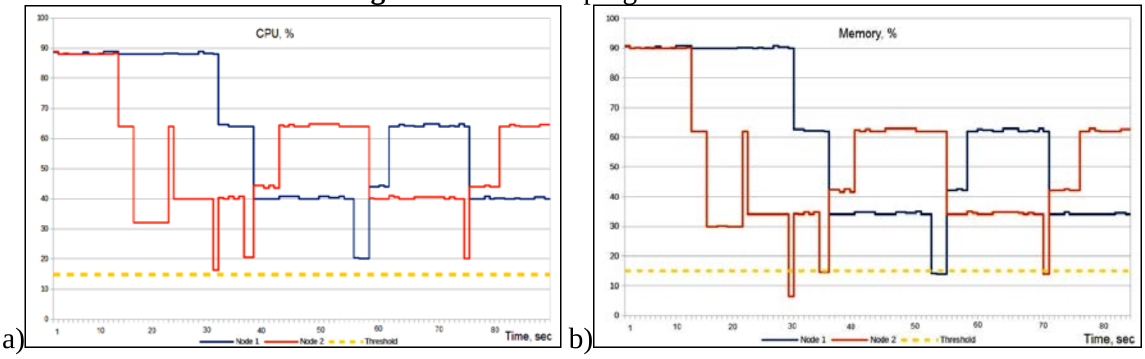


Figure 5. The dependence of the expenditure of CPU (a) and Memory (b) (%) on time for a computer system of two nodes (sec).

8. References

[1] Jiang X, Adeli H 2008 Neuro-genetic algorithm for non-linear active control of structures *Int. J. Numer. Meth. Engng* **75** 770-786

[2] Zack M H 2007 The role of decision support systems in an indeterminate world *Decision Support Systems* **43** 1664-1674

[3] Weisenpacher P, Halada L and Glasa J 2011 Computer simulation of fire in a tunnel using parallel version of FDS *Proc. of the Mediterranean Combustion Symp., Assoc. Sezione Italiana del Combustion Inst.*

[4] Yang P, Li C and Chen D 2013 Fire emergency evacuation simulation based on integrated fire–evacuation model with discrete design method *Advances in Engineering Software* **65** 101-111

[5] Curry D M, Beaver W W and Dagli C H 2018 A system-of-systems approach to improving intelligent predictions and decisions in a time-series environment *SoSE* 98-105

[6] Gosavi A, Agarwal S and Dagli C H 2017 Predicting response of risk-seeking systems during project negotiations in a system of systems *IEEE Systems Journal* **11(3)** 1557-1566

[7] Agarwal S, Pape L E, Kilicay-Ergin N H and Dagli C H 2014 Multi-agent based architecture for acknowledged system of systems *CSER* 1-10

[8] Valeyev S, Taymurzin M and Kondratyeva N 2011 Adaptive data collection system in technical security systems *Automation in Industry* **4** 11-14

[9] Kondratyeva N, Valeev S 2016 Distributed information and control system for emergencies in critical infrastructures *Proceedings of 10th conference on Application of Information and Communication Technologies AICT* 1-4

[10] Kovtunenko A, Bilyalov A and Valeev S 2018 Distributed streaming data processing in iot systems using multi-agent software architecture *18th International Conference on Next Generation Wired/Wireless Networking (NEW2AN), and 11th Conference on Internet of Things and Smart Spaces (ruSMART) conference proceedings* 572-583

[11] Valeev S, Kondratyeva N 2015 Technical safety system with self-organizing sensor system and fuzzy decision support system *Proccedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* 1-6

[12] Valeev S, Kondratyeva N 2017 Infrastructure object management in the conditions of critical situations based on situational models *Proceedings of 10th International Conference Management of Large-Scale System Development MLSD* 1-4

[13] Khakimov A, Muthanna A, Kirichek R, Koucheryavy A and Muthanna M S A 2017 Investigation of methods for remote control iot-devices based on cloud platforms and different interaction protocols *Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*

[14] Ateya A A, Muthanna A, Gudkova I, Abuarqoub A, Vybornova A and Koucheryavy A 2018 Development of intelligent core network for tactile internet and future smart systems *Journal of Sensor and Actuator Networks* **7(1)**

[15] Ramakrishnan R 1998 Database Management Systems *WCB/McGraw-Hill*

[16] Valeev S S, Maslennikov V A and Kovtunenko A S 2012 Design of the middleware based on agent-oriented technologies for the automated control systems of complex technical objects *Proc. of 6th International Conference “Parallel Computing and Control Problems”* **1**

[17] Kovtunenko A S, Maslennikov V A 2014 Creation of distributed control information systems on the basis of agent-oriented approach *Proceedings of XII All-Russian Conference on Problems of Management*

[18] Soulé R, Hirzel M, Grimm R, Gedik B, Andrade H, Kumar V and Wu K L 2010 A unified semantics for stream processing languages (extended) *Technical Report* (New York University)

[19] Nikitin V S, Semyonov E I, Solostin A V, Sharov V G and Chayka S V 2016 Modeling the ‘smartlink connection’ performance *Computer Optics* **40(1)** 64-72 DOI: 10.18287/2412-6179-2016-40-1-64-72

[20] Agafonov A A, Myasnikov V V 2018 Numerical route reservation method in the geoinformatic task of autonomous vehicle routing *Computer Optics* **42(5)** 912-920 DOI: 10.18287/2412-6179-2018-42-5-912-920

[21] Chakravarthy S, Jiang Q 2009 Stream data processing: a quality of service perspective modeling, scheduling, load shedding, and complex event processing *Springer Publishing Company, Incorporated*

[22] Abadi D 2003 Aurora: a new model and architecture for data stream management *The VLDB Journal* **12(2)**

[23] Zdonik S B, Stonebraker M, Cherniack M, Çetintemel U, Balazinska M and Balakrishnan H 2003 The Aurora and Medusa projects *IEEE Data Eng. Bull* **26(1)**

[24] Adi A, Etzion O 2004 AMiT – the situation manager *VLDB Journal* **13(2)**

[25] FIPA ACL message structure specification 2002 *Foundation for Intelligent Physical Agents* URL: http://www.fipa.org/specs/fipa00061/SC00061G.html

[26] Zhao Y B, Sun X M, Zhang J and Shi P 2015 Networked control systems: the communication basics and control methodologies *Mathematical Problems in Engineering* **2015** 639793 DOI: 10.1155/2015/639793

[27] Lin H 2017 Estimation and control for networked systems with packet losses without acknowledgement *Studies in Systems, Decision and Control* **77**

[28] Volkov A, Khakimov A, Muthanna A, Kirichek R, Vladyko A and Koucheryavy A 2017 Interaction of the iot traffic generated by a smart city segment with SDN core network *Lecture Notes in Computer Science*