

The software platform for creating and conducting artificial intelligence competitions with a visualization subsystem

A.A. Zakharova, N.S. Silchenko, R.A. Krylov, V.I. Averchenkov
zaa@tu-bryansk.ru|silchenko.nk@gmail.com|oktopy@gmail.com|aver@tu-bryansk.ru
Bryansk state technical university, Bryansk, Russia

The article describes the solution to the problem of teaching programming skills using modern techniques and additional software. It is proposed to use simulation modeling of interaction developed by users as part of the training of intellectual agents in a competitive form that implement various algorithms as the main approach to the solution. Intellectual agents are presented in the form of artificial intelligence developed by platform users, which interacts with other intellectual agents following the rules of the developed scenario for the competition. Scenarios provide a set of capabilities for intelligent agents, an interaction environment, and a set of constraints that participants follow. To support this simulation, it is proposed to use a specialized software platform. The platform allows organizers to develop scenarios with a unique set of rules, and additional platform tools speed up development and allow organizers to implement visual display to users that can be used to show competitive process. A set of built-in platform tools allows organizers to focus directly on the rules of the competition, since the platform provides communication with participants and additional tools for calculating the results of the competition. In addition there is a set of basic competition systems on the platform. However, if necessary, the organizers can present their own competition format and implement it separately. The article describes the developed platform for teaching and holding competitions in artificial intelligence. The article also examines a number of scenarios and intelligent agents.

Keywords: artificial intelligence, intellectual agent, artificial intelligence competitions.

1. Introduction

At present, technologies are developing at tremendous speed. One of the most promising areas today is the development of methods and technologies based on artificial intelligence. Artificial intelligence permeate deeper and deeper into various spheres of human life, both in the field of science and industry, and in everyday life. In addition, the training of IT specialists who are able to create new and maintain software products, analyze big data, administer complex technical systems and etc. is especially relevant.

The acquisition of new professional competencies in a dynamically changing world requires the creation and application of new technologies in the learning process. Currently, there are different software platforms. However, these are specialized platforms for implementing either educational programs or competency assessment activities (competitions, hackathons, etc.), including in remote form.

For the quick creation of unique training tracks (educational and methodological support adapted to the individual learning path) a tools are needed. Current trends are also aimed at the gamification of the learning process in order to improve its quality, and the competitive aspect is much better motivates for independent study of materials [1]. It is important to implement active pedagogical methods involving the student in the design of his own educational path, increasing his degree of motivation for the learning process.

There is the introduction of meta-design into the educational process [2], which allows you to organize a space for effective education for the required competencies, taking into account the individual characteristics of students.

In that case, it can be argued that the creation of a new educational platform that is adaptable and implements the above-mentioned modern approaches to education is relevant.

2. Analogues

Currently, there are platforms aimed at artificial intelligence competitions among user's intellectual agents. Among the most famous, the following platforms can be distinguished: Russian AI Cup, Mini AI Cup and Google AI Challenge [3-5]. In the process of participating in the competition, users are provided with additional information about working with the platform, and a set of additional training materials is created for participants.

However, these platforms can hardly be used to direct training for the following reasons.

1. Closure. Users are provided with limited functionality and are not allowed to implement their own scenarios. As a result, it is not possible to build a training program based on them.
2. Narrow focus. Focus on a fixed set of subject areas. Often, for each subject area, a new separate platform is created, completely duplicating the functionality of existing platforms, making only minor changes.
3. Lack of control. There is no ranking of users by levels of knowledge and skills. All users are forced to be in the same group: both novice developers and professionals.
4. The same type of competition system. There are a huge number of competition systems, however, modern platforms often use the most popular or basic approaches (for example FFA, double elimination, etc.).
5. Completeness. The platforms do not provide the opportunity to conduct a series of competitions with subsequent monitoring in order to collect statistics and evaluate the results. Support for the competition's scenario is over.
6. Decentralization of information. There is no supporting methodological assistance and additional information materials. The organizers provide only materials directly related to the competition.

Given the above, we note that there are tools for creating and conducting competitions with a number of significant drawbacks. Modern technologies are

developing at a rapid pace, new techniques, approaches to learning and tools that can implement them appear. Based on the existing shortcomings, it was concluded that there is a need to develop a universal educational platform on artificial intelligence with the following basic properties:

1. Openness. The users should be provided with the full functionality of the platform, which will allow not only to participate in the competition, but also to act as the organizer of the competition, as well as use the platform in training.
2. Broad focus. The platform should be universal, which will allow it to be used in training in any IT field (using additional materials and methodological support), to conduct competitions and test algorithms in the user-developed scenarios [6].
3. Flexible control. The platform should be able to rank users according to various criteria. Both basic criteria (level of skills and competencies, age), and any other criteria defined by the scenario developers can be used.
4. The flexibility of the settings. The platform should contain not only the basic conducting systems built into it by default, but also allow users to implement their own, if this is necessary.
5. Renewability. This property allows you to collect statistics and can serve as a universal tool for assessing the knowledge of participants. This is especially useful for monitoring the overall level of educational skills.
6. Centralized information. The presence of additional training material on the platform on various aspects of programming will allow new users to always have high-quality and up-to-date information from trusted

sources and to develop much faster in the direction they are interested in.

The platform under development is based on openness. Openness implies that any user of the platform can be not only a participant, but also the creator of the educational program and the organizer of the competition. The platform also has an ever-expanding knowledge base necessary for working with intellectual agents, which is also developed at the expense of platform users, through which the platform is also excellent in educational activities. The presence of high-quality and verified material will allow faster to find high-quality information needed in the subject area of interest [7-8], and as a result to get better information than in open sources.

Moreover, the platform allows you to create your own scenarios and visually display them [9]. At the moment, to organize a competition, the user needs to develop a software product that implements the logic of the competition and establish communication between its participants. This platform already has the basic mechanisms for that, and the organizer can focus only on conditions of the competition and the tasks of the scenario.

Separately, it is worth noting that platform users can be united into a group and learn how to develop in competitions with each other among opponents with equal knowledge.

3. Description of the approach to building a platform for modeling the interaction of intellectual agents

To solve this problem, the following architecture is proposed (Fig. 1).

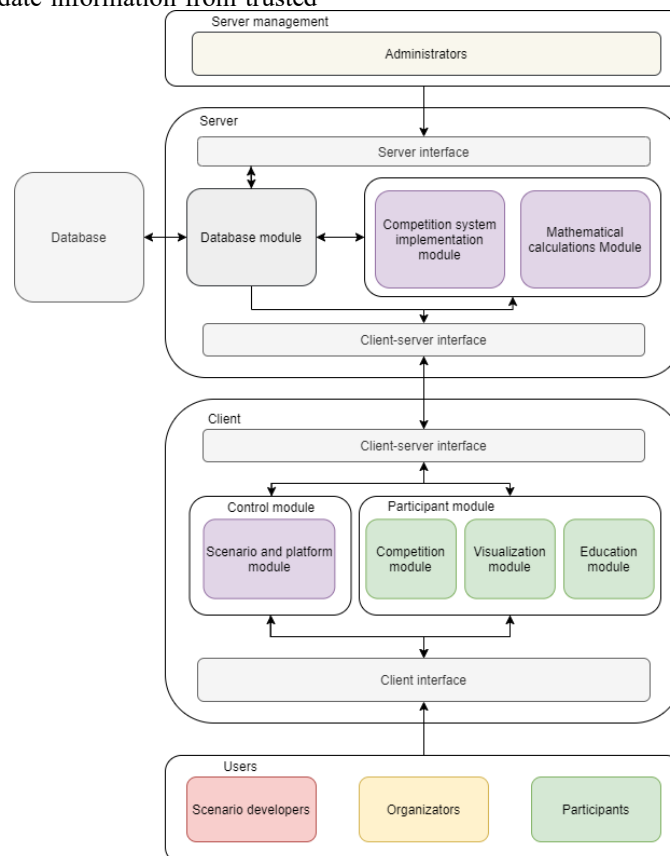


Fig. 1. Software platform architecture

Platform users are represented as scenario developers, organizers and participants. It means that the same person or group of people can act as the developer of scenario and organizer of the competition.

Organizers and scenario developers work in the control module, and participants interact with the participant module.

In the control module, organizer configure the competition parameters and load the scenario.

The participant module contains both general training materials and specific data for the scenario. The competition module and the visualization module are interconnected. Competitions using the module can used to tests by the developer, and if the competitions are presented in a graphical representation, the user can see the results in a visualization module.

The competition is held on the server, which is necessary for data protection. The server has a module for mathematical calculations and a module for implementing the system. The first module is responsible for the outcome of the competition, and the second for its format. Then the data gets into the database, and from there the organizers and participants can get it.

Administrators provide technical support and resources for computing.

The platform implements all the principles described above. Openness is achieved due to the lack of a global separation of roles. Any user can be either a participant or an organizer of one or several competitions at the same time. And also there is no local separation, which provides flexibility in management.

The platform allows you to develop in several directions at once, since it is universal and supports the development of scenarios, both competitive and training, and the complexity of such scenarios depends only on the task set by the authors. The module for providing a conductive system is also flexibly configured, which allows you to choose a conductive system from those already implemented on the platform, or implement your own.

There is also a module that allows students to get methodological and informational software for training.

The module for working with educational material allows you to centrally store information and use it in the learning process. At the same time, any competition can be resumed for a specific local group, which corresponds to the renewability property.

The platform has a built-in visualization module, which can significantly reduce the time for developing a visual display. Users can use both the tools built into the platform and, if necessary, connect their own.

4. Description and testing of developed scenarios

On the developed platform, three test competitive scenarios were created: “Sea battle”, “Snake” and “Virus war”. Each scenario has its own set of rules.

Sea battle (Fig. 2) is presented in its classic, desktop version, with the possibility of the initial arrangement of ships and directly the game process.



Fig. 2. Sea battle

The snake (Fig. 3) is presented in the form of a game for four participants, where the main goal of all agents is

to be the last among those who have reached a dead end created by the enemy's snake or natural obstacles.

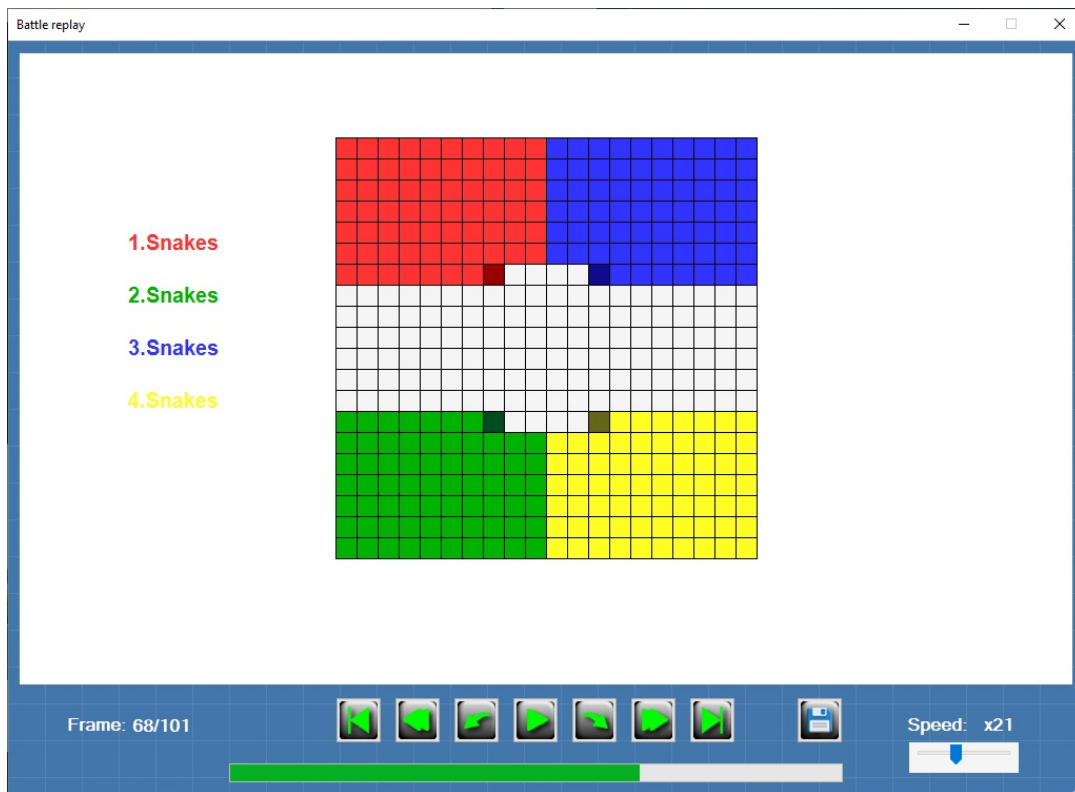


Fig. 3. Snake

The virus war (Fig. 4) is presented in the form of capturing a field with special conditions: an agent can capture only cells adjacent to those already captured, and

if the agent attacks the cell of another agent, it becomes impassable and free.

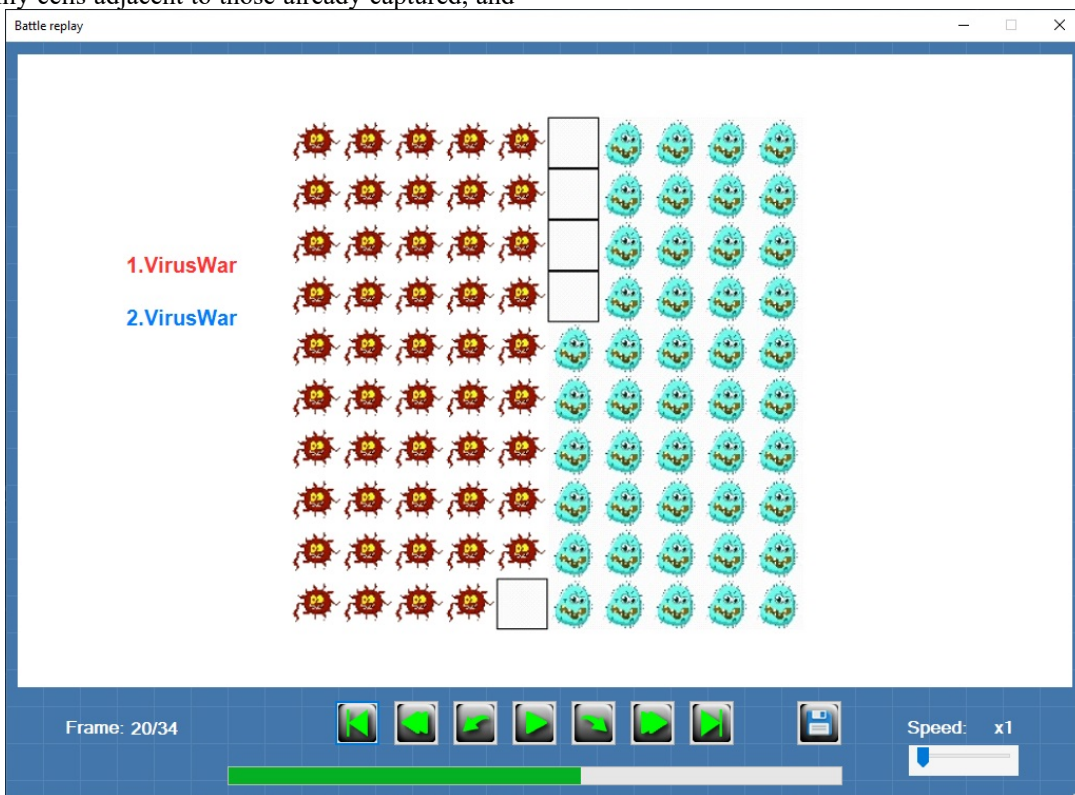


Fig. 4. Virus war

The platform also supports custom single-user scenarios. The idea is that the user receives a task to implement a specific algorithm. At the end of

development, the user agent is tested by the platform using various input data to verify its operation. For testing, two

task scenarios were created: matrix transposition (Fig. 5) and array sorting (Fig. 6).

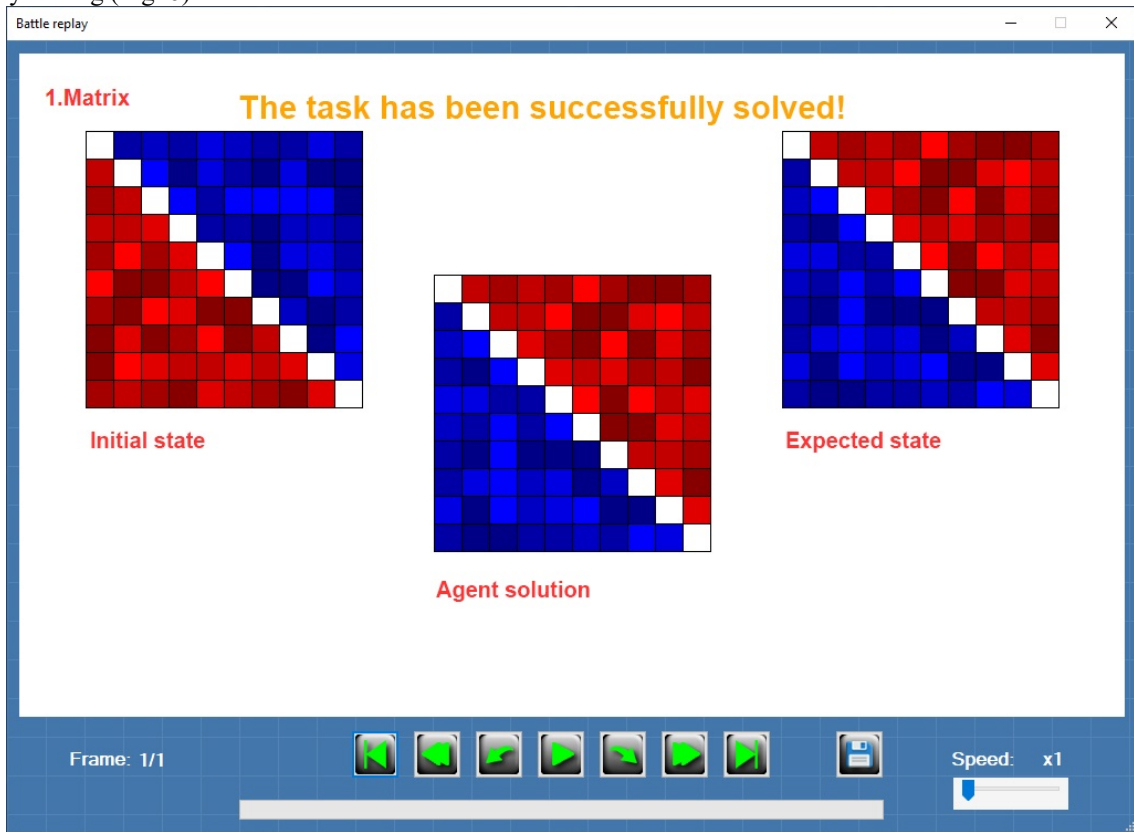


Fig. 5. Matrix transposition

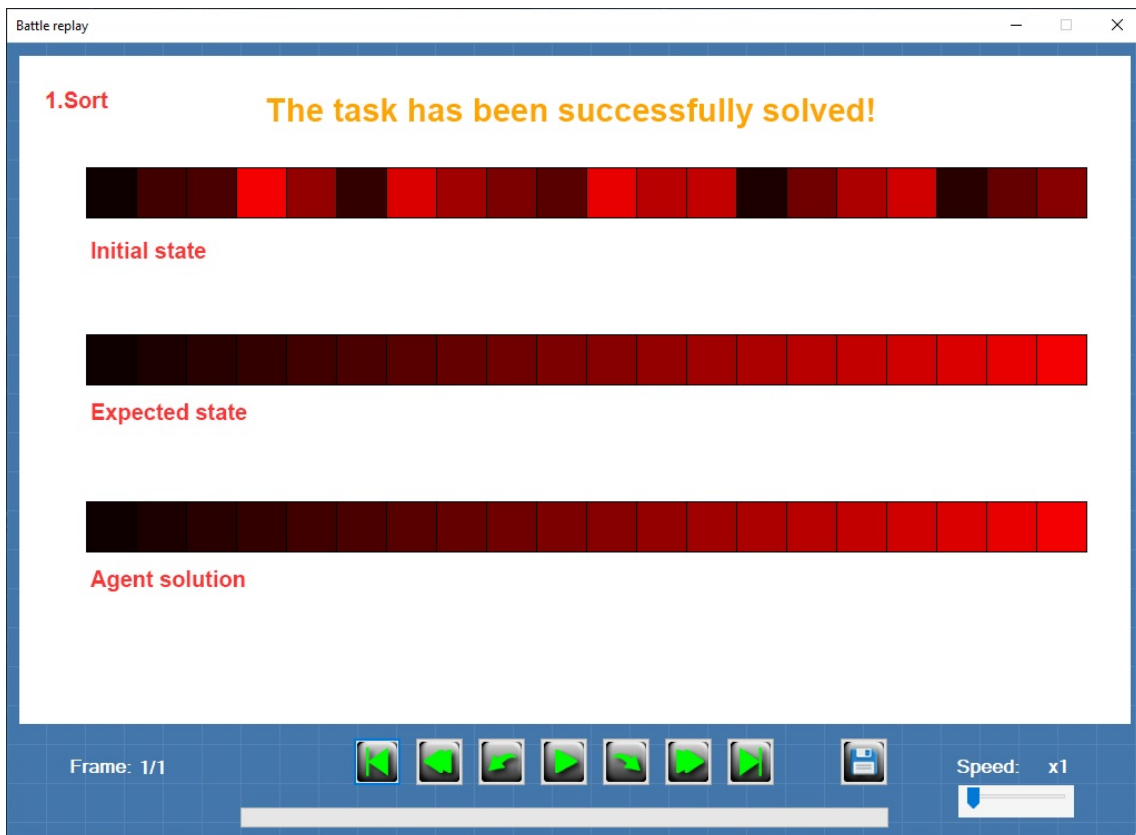


Fig. 6. Array sorting

A few intellectual agents have been developed for all scenarios. They solve the problem in various ways. Series of tests were carried out.

For competition scenarios were used intellectual agents that implement various strategies. Using the “sea battle” scenario as an example, consider the results of three different strategies:

- A. Algorithm based on random events;
- B. Algorithm with analyzing field;
- C. An algorithm that bases its choice on the experience of previous parties.

As a tournament system, it was circular, i.e. each agent will play with each other.

Table 1. Performance characteristics of various algorithms for the scenario "Sea battle"

Victories, % Defeats, %	A	B	C
A		71,7%	93,4%
B	28,3%		70,8%
C	6,6%	30,2%	

As follows from the table, algorithm B and C have a significant advantage over algorithm A, and algorithm B is significantly inferior to algorithm C.

Consider a similar table in which copies of these agents participate.

Table 2. Performance characteristics of various algorithms for the scenario "Sea battle" with copies of agents

Victories, % Defeats, %	A1	A2	B1	B2	C1	C2
A1		50,1%	70,9%	72,1%	90,2%	93,1%
A2	49,9%		71,3%	70,8%	92,7%	91,9%
B1	29,1%	28,7%		49,5%	69,5%	70,9%
B2	27,9%	29,2%	50,5%		71,1%	70,5%
C1	9,8%	6,3%	30,5%	28,9%		49,8%
C2	6,9%	8,1%	29,1%	29,5%	50,2%	

Based on the second table, we can conclude that algorithms of the same type on average always have a 50% probability of winning against each other, and the conclusion based on the results of the first table is fully consistent with the conclusion of the second.

5. Conclusion

The developed system allows you to create scenarios with different conditions, as well as set them your own unique display and conduct an unlimited number of tournaments according to the created scenarios. It is also possible to use it as a training system due to special task scenarios in which the user needs to implement the algorithm and send the result to the server for verification.

The article analyzes the main analogues - Russian AI Cup, Mini AI Cup and Google AI Challenge, identifies the main shortcomings and formulates the properties that the platform should have, and provides the platform architecture that meets all the requirements.

Distinctive features of the developed system compared to other platforms is its versatility and immutability both in supported scenarios and in visualization methods for the interaction of intellectual agents. Also, the presence of a flexible system for visual reproduction of scenarios allows more efficient development and debugging of algorithms from various areas of artificial intelligence.

As an example of implementation, a number of scenarios were demonstrated, including “sea battle”, “snake”, “virus war”, “array sorting” and “matrix transposition”. For the “sea battle” scenario, a set of agents was developed that implements a number of algorithms aimed at solving the problem, two experiments were conducted using different numbers of agents, and the results of these algorithms on all systems are presented.

The developed system was used to conduct an artificial intelligence competition as part of the Bryansk Regional IT Festival.

Acknowledgments

The reported study was funded by RFBR, project number 19-07-00844.

References

- [1] Chesani F., Galassi A., Mello P., Trisolini G. (2017) A Game-Based Competition as Instrument for Teaching Artificial Intelligence. In: Esposito F., Basili R., Ferilli S., Lisi F. (eds) AI*IA 2017 Advances in Artificial Intelligence. – AI*IA 2017. Lecture Notes in Computer Science, vol 10640. – Springer, Cham.
- [2] Zakharova, A.A., Vekhter, E.V., Shklyar, A.V. (2019) The of visualization tools in the meta-design of an educational environment. European Journal of Contemporary Education, Vol. 8 no. 1, pp. 43-51.
- [3] Russian AI Cup – artificial intelligence programming competition. – Access mode: <http://russiaaicup.ru>
- [4] Mini AI Cup – artificial intelligence programming competition. – Access mode: <https://aicups.ru/>.
- [5] Google AI Challenge. – Access mode: <http://ants.aichallenge.org>.
- [6] Zakharova A.A., Korostelyov D.A., Fedonin O.N. (2019) Visualization Algorithms for Multi-criteria Alternatives Filtering. Scientific Visualization, Vol. 11, no. 4, pp. 66-80.
- [7] Jones, M. T. Artificial Intelligence Programming in Applications / M. T. Jones. – M.: infinity science press llc, 2018. – 312 p.
- [8] Norwing, P. Artificial Intelligence: A Modern Approach. – UK: Glivice, 2014. – 1408 p.
- [9] M.B. Mihailuk, P.U. Timohin. (2019) Memory-effective methods and algorithms of shader visualization of digital core material model. Scientific Visualization, Vol. 11, no. 4, pp. 1-11.

About the authors

Alena A. Zakharova, Sc.D. in Technique, professor of Informatics and Software Engineering Department at Bryansk State Technical University. E-mail: zaa@tu-bryansk.ru

Nikita S. Silchenko, graduate student of Informatics and Software Engineering Department at Bryansk State Technical University. E-mail: silchenko.nk@gmail.com

Rostislav A. Krylov, graduate student of Informatics and Software Engineering Department at Bryansk State Technical University. E-mail: oktopy@gmail.com

Vladimir I. Averchenkov, Sc.D. in Technique, professor of Computer Technologies and Systems Department at Bryansk State Technical University. E-mail: aver@tu-bryansk.ru