Automated construction of competence-oriented models of specialists in the field of software engineering

Galina V. Rybina
Doctor of Technical Sciences, Professor
National Research Nuclear University MEPhI, professor
115409, Russia, Moscow, Kashirskoe shosse, 31
galina@ailab.mephi.ru

Andrei Y. Nikiforov
Ph.D.
National Research Nuclear University MEPhI, docent
115409, Russia, Moscow, Kashirskoe shosse, 31
AYNikiforov@mephi.ru

Abstract: The scientific-methodical and technological experience of the development and use of tutoring integrated expert systems and the creation of a single ontological space of knowledge and skills for the automated construction of competence-oriented models of specialists in the field of methods and technologies of artificial intelligence and software engineering obtained at the Department of Cybernetics of NRNU MEPhI is analyzed.

Keywords: Software Engineering, Tutoring Integrated Expert Systems, Problem-Oriented Methodology, Intelligent Software Environment, ATTECHNOLOGY Workbench, Intellectual Tutoring, Knowledge Engineer, Student Model, Competence-Oriented Specialist Model, Ontology of Courses/Disciplines, Professional Competence.

1 Introduction

The purpose of this work is to show some methodological aspects of software engineering (SE), reflecting the departmental view that has developed over the years on issues of training specialists in the field of programming technology (PT), and also highlight and briefly characterize modern innovations that are currently used for practical implementation of the educational process on the basis of the ontological approach and the use of tutoring integrated expert systems (IES) to support basic courses / disciplines.

The origins of the term “Software Engineering” in our country has long been preceded by the term “Programming technology”, which denotes methods, means and tools that ensure the process of creating software systems [1]. Modern understanding of SE was summarized in 2004 in SWEBOK (Software Engineering Body of Knowledge) - as a system of methods, tools and disciplines for planning, developing, operating and maintaining software systems, including readiness for implementation. In 2014, SWEBOK v3.0 was published, which received international recognition as ISO / IEC Technical Report 197596: 2015.

These definitions are so close that, in the opinion of many experts [2-4] and others, SE can be considered a natural development of the PT, which is very important both for the preservation and further development of the national experience of the PT as a whole, and in terms of training specialists in "Software Engineering". The Department of Cybernetics of the National Research Nuclear University MEPhI has accumulated many years of experience in teaching PT courses / disciplines as part of the specialization of “Computer Software” (CS) of the basic specialty “Applied Mathematics”. The creation and active development of CS belonging to the periods 1971-1990 and 1991-2014 The current stage, which began in 2015, continues as Software Engineering degree programs (bachelor and master).

Historically, the priority research areas of the Department of Cybernetics of the National Research Nuclear University MEPhI and the basic fundamentals of training of mathematicians in the fields Artificial Intelligence Systems, Computer Software, Information and Mathematical Systems of the Automated Control System, within the specialty Applied Mathematics, were related with the creation in the early 1970s. Science School of L.T. Kuzin [5] in the field of artificial intelligence (AI). The main provisions of the scientific school of L.T. Kuzin and the research program until the mid-1980s focused on the creation and evolutionary development of cybernetic models associated with the release from the cybernetics the new area of knowledge, AI, and the development and research of a wide range of models and knowledge representation languages in the context of intellectual databanks.

A new stage in the formation of the scientific and educational appearance of the department in the field of AI began at the turn of the 1970s and 1980s, when from the cybernetic scientific school of L.T. Kuzin emerged several separate areas related to the development of methods for constructing intelligent systems of various architectural typologies as the main products of AI, as well as the creation of appropriate software, taking into account the need of training system programmers and specialists in programming technology.

Therefore, for the educational process, in conjunction with courses on methods and technologies of AI, there were developed and tested at the base enterprises (NICEVT, INEUM, Keldysh Institute of Applied Mathematics, etc.) and in the educational and scientific laboratories of the department specialized courses on programming languages, fundamentals of translators, programming technology, modern software development tools, software verification and certification, basic systems modeling, etc. Basic education of programming and programming technology has become an integral part of training specialists in AI and a "calling card" of the graduates of the department.

For the educational process, author's courses and programs were developed, original textbooks and tutorials on the basics of designing and programming technology of intelligent systems of various architectural typologies, studying problem-and subject-oriented languages for AI (PROLOGUE, LISP), as well as specialized tools (INTER-EXPERT, ECO, Level 5 Object, G2, AT-TECHNOLOGY, etc.).

In general, the formation of the modern scientific and educational appearance of the department in the field of artificial intelligence is naturally associated with the historical, conceptual and technological integration of all three of the above specializations within the basic specialty “Applied Mathematics”, and the creation and active formation refers to the periods of 1971-1990 and 1991-2014. The current stage of development, starting in 2015, has been continuing in the framework of the direction of training “Software Engineering” (bachelor and magistrate programs).

Consider some modern innovations in the field of AI technologies, which are currently used for the practical implementation of the educational process at the Department of Cybernetics. A new approach is the development and use of tutoring integrated expert systems (IES) [3,4] and the creation of a single ontological space of knowledge and skills for the automated construction of competence-oriented models of specialists in the field of AI in the area of "Software Engineering".

2 Integration of methods and technologies of AI with software engineering in educational process

The main criterion for future success of students in the field of SE (PT) and AI is the inculcation of skills in computational, algorithmic and logical thinking within the framework of basic courses in programming, modern tools and design technologies for information and intelligent systems of various architectural typologies.

Among the languages traditionally studied today in introductory programming courses dominate Python, Java, as well as C and C++. There are many studies [6, 7] and others, where comparisons are made of the features of programming languages by such criteria as simplicity, integrity, capabilities of basic data types, syntax, degree of data abstraction, expressiveness, etc. In particular, in accordance with [6], according to the aggregate of these criteria, Java turned out to be the best language, followed by Python, and then the rest. Starting from the first course, when learning C and C ++ languages, students learn the basics of algorithmization, building data structures, understanding abstract and user-defined data types and classes, and acquire skills in documenting programs. However, for trainees it is important to get not only knowledge of the specific language features, but also skills of using this language for solving problems, therefore the choice of language directly depends on the direct subject of study, degree of audience (bachelor, masters) and design features of a particular class of software systems.

In particular, the use of object-oriented programming paradigms adds another level of abstractions that require an understanding of the basics of class design and the skills to integrate them into existing software products. Based on this, the most important component of the curriculum of “Software Engineering” is the consideration of various programming paradigms so that in the future, graduates can choose the most appropriate type of programming language to solve specific practical problems. For example, the study of functional programming languages (Common Lisp, Scheme, Haskell, and others) and Prolog-type logic programming languages allows students to solve problems in the field of AI. Familiarity with the basics of scripting languages such as Python, used when working with web applications, as well as with the principles of event-oriented programming, allows you to understand how to create applications that interact with the user.

It is important to note that the development of universal applications for solving complex interdisciplinary tasks (in particular, the creation of intelligent systems for particularly important sectors of the economy) makes it possible to discover in practice the advantages of various languages and programming tools. Mastering universal programming languages (C, C ++, etc.) allows you to gain experience that can be used to solve problems in the field of system

programming and create various problem-oriented and subject-oriented specialized tools to support the development of a wide class of intelligent and informational systems (like CASE-tools, intelligent packages, Workbench-systems, etc.).

In general, based on the experience of building competence-oriented models of graduates (bachelors and masters) in the field of PT, it is necessary to provide solutions to the following important tasks in the organization of the modern educational process:

- conceptual understanding of the programming process and instilling computational and logical thinking skills for the implementation of specific tasks in the form of a program and / or a software system;
- selection and use of various programming paradigms, including OO-oriented, functional, logical, environment-oriented, and various types of programming languages, as well as the development of competence in the use of a particular language;
- application of new approaches from the field of AI by presenting typical knowledge from the field of programming in the form of plans for solutions [8, 9], to help students who have difficulty in moving from a method or algorithm for solving a problem to its software implementation.

A significant role in solving the above problems is currently assigned to the organization of the educational process on a single conceptual and methodological basis, which is carried out by attracting methods of ontological engineering and the technology of teaching IES [3, 4] and others.

For example let us consider some features of professional and universal competences represented in professional standards [10], which are formed taking into account the scientific and educational appearance of the department in the field of SE (PT). The main focus is traditionally placed on the ontologies of such disciplines as “Cybernetic Systems Programming Technology (Software Project Management)”, “Software Systems Design and Architecture”, “Designing of knowledge based systems”, “Dynamic Intelligent Systems”, etc., within which students receive basic theoretical knowledge and practical techniques, typical for the development of traditional software systems and intelligent systems of various architectural typologies, including: life cycle and methodology design and development of software systems; various software system architectures; modeling in languages such as UML, testing, verification and certification of software; CASE-tools, workbench-systems and other types of tools that allow to automate the process of developing software systems, etc.

It is important that within the framework of these courses and research projects students implement individual educational projects, during the course of which the following professional competencies are acquired. The table 1 shows example of professional competencies in SE (PT) field.

Table 1 – Example of professional competencies in SE (PT) field

ID	Professional competencies
PC-1	requirements elicitation (for software, information software and user interface)
PC-2	selecting of a life cycle model
PC-3	using metrics to measure project properties
PC-4	selecting of the style (approach) to the software implementation and the rules (standard) of the programming style
PC-5	estimation of the project functions coverage by its program implementation
PC-6	designing of user interface and interface specifications
PC-7	justification of the interface compliance specification to the needs of the potential user
PC-8	documenting model, information, operational, functional, architectural and other aspects of the project
PC-9	documenting solutions to test cases and selected real-world problems
PC-10	documentation of trial operation and commercial implementation

3 Practice of development and use of tutoring integrated expert systems in the educational process of NRNU MEPhI

The use of the most modern scientific results in the field of AI, including teachers and staff of the Department is a long-term tradition of the organization of educational process on methods and technologies of AI at the Department of Cybernetics. In this case, we are talking about the theory and technology of construction of IES of different architectural typology, based on problem-oriented methodology [3, 4] and intelligent software environment of the AT-TECHNOLOGY workbench [3, 4], providing a total automated support of the processes of development and maintenance of a wide class of static and dynamic IES, including tutoring IES and tutoring web-oriented IES (web-IES).

The problems related to the application of problem-oriented methodology and tools of the AT-TECHNOLOGY tutoring for the purposes of intellectual tutoring [3] through the development of intelligent tutoring systems (ITS), created on the basis of the architecture of tutoring IES and web-IES, were widely covered in the press of different years, for example [11, 16 - 18], etc.

Since 2008, tutoring IES and web-IES, developed in the Intelligent Systems and Technologies laboratory of the Department of Cybernetics of the National Research Nuclear University MEPhI, have been actively used for automated support of basic courses / disciplines in the direction of training of Applied Mathematics and Information and Software Engineering including: Introduction to Intelligent Systems, Models and Methods of Knowledge Representation and Processing, Intelligent Dialog Systems, Dynamic Intelligent Systems, Designing the Knowledge-Based Cybernetic Systems, Modern Intelligent System Architectures, Intelligent Simulation Modeling Tools, Intelligent Information Systems and ect. For all these courses and disciplines, using the basic tools of the AT-TECHNOLOGY workbench, the applied ontologies [3] are implemented and dynamically developed, which together form the "Intelligent Systems and Technologies" generalized ontology, which allowed to create a single ontological space of knowledge and skills through integration with ontologies of basic courses/disciplines on programming technology and to implement, practically, a full set of functional tasks typical of intellectual technology, including [3, 11]:

- *individual planning* of the methodology for studying a specific training course (specifying on the basis of ontologies of courses / disciplines of a personal trajectory / tutoring strategy, individual control and identification of "problem areas" in students' knowledge and skills, optimization of individual learning, taking into account the psychological portrait of the learner, etc.);
- *intelligent analysis* of learning tasks (modeling the reasoning of students solving learning tasks of various types, including using non-formalized methods, identifying types of errors and causes of their manifestation in knowledge and skills instead of their finding, feedback through dynamic updating of students' knowledge and skills, forecast grades on exams, etc.);
- *intelligent decision* support (using technology of traditional expert systems (ES) and IES for intellectual assistance at each stage of solving educational problems, including extended explanations like "how?" and "why?", the choice of solutions, a hint of the next stage of the decision, etc. d.).

These capabilities of tutoring IES and web-IES fully correspond to the functional and technological aspects of modern ITS, in particular [12 – 14, 19], and adaptive tutoring systems [15], and also create prerequisites for further research on the implementation of promising approaches in the form of intelligent monitoring and intelligent collective tutoring, as well as for the semantic integration of individual tutoring IES with their parallel use in the educational process.

Currently, the possibility of effective practical implementation of such an approach to the construction of ITS is provided by two factors.

1. Active development of the conceptual foundations of the problem-oriented methodology, which allows on the basis of scalable architecture of IES [3,4] to implement a fairly powerful functionality necessary for modern ITS (construction of developed student models, adaptive tutoring models, models of problem areas, models of explanation, teacher models, models of applied ontologies of courses/disciplines/specialties, etc.).

2. The use of intelligent software technology (based on the means of the ATTECHNOLOGY workbench) for automated support of the processes of constructing IES at all stages of the life cycle, providing archiving of the unique expert-methodological experience of subject teachers, reducing the intellectual load on knowledge engineers (knowledge analysts), reducing the development time of tutoring IES and web-IES [3, 4].

In modern conditions, the implementation of effective practical implementation and further functioning of the ITS of any architectural complexity is impossible without instrumental software support for the design and maintenance of the ITS at all stages of the life cycle. However, currently there is no accepted as a standard technology for the development of intelligent systems, including ITS, therefore, for these purposes, either general-purpose tools and platforms are used, or specialized tools are created.

4 Automated construction of competence-oriented models of specialists

The ontology model in accordance with [3] is presented in the form of a semantic network in which vertices reflect various elements of courses / disciplines (sections, topics, subtopics, definitions, etc.), including a model of targeted competencies and a set of models of various learning influences on the identification of knowledge and the skills of the students, and the edges show the different types of connections (relationships) between course elements, competencies and learning influences.

Automated support for the construction of applied ontologies for each course / discipline based on this model is carried out using special tools that function as part of the AT-TECHNOLOGY workbench, closely described in [17, 20] and other works.

From a methodological point of view, it is important to note that by creating the ontology "Intelligent Systems and Technologies" (which currently has about 900 vertices from 8 courses / disciplines) it was possible to build a single ontological knowledge and skills space, which allows:

- implement the training cycle in accordance with [10], the curriculum and teaching materials (lectures, practical exercises, etc.);
- provide a fully functional construction of competence-oriented students models for the entire period of study (bachelor, master), and as a final result the formation of models of future professionals, including individual psychological portraits);

- compare the current competencies of the students with the target ones, identify the so-called “problem areas” in knowledge and plan the training effects in the form of solving specific practical tasks for each student in order to achieve a higher level of competencies, etc.;
- show great intellectual learning opportunities, namely: individual planning of the methodology for studying specific training courses; intellectual analysis of learning tasks; intellectual decision support, etc.

As a basic information and methodological resource for building models of professional competencies, in particular, for such professions as "software engineer", "system analyst", " IT system specialist ", "software architect", etc. the professional standards for the Information Technology industry [21] were used quite effectively.

The main focus is traditionally placed on the ontologies of such disciplines as “Cybernetic Systems Programming Technology (Software Project Management)”, “Software Systems Design and Architecture”, “Designing of systems based on knowledge”, “Dynamic Intelligent Systems”, etc., within which students receive basic theoretical knowledge and practical techniques, typical for the development of traditional software systems and intellectual systems of various architectural typologies, including: life cycle and methodology design and development of software systems; various software system architectures; modeling in languages such as UML, testing, verification and certification of software; CASE-tools, workbench-systems and other types of tools that allow to automate the process of developing software systems, etc.

At present, there is no universal classification of competencies, however, the generally accepted point of view is the allocation of professional and universal competencies. Further specification depends on the specifics of the profession, the traditions of the university that trains specialists in this field, and other feature.

In this case, taking into account the long-term traditions of the Department of Cybernetics of NRNU MEPhI, it is, in fact, the integration of system and software engineering with methods and technologies of AI, so according to the Federal State Educational Standard 3+ [10], the two following competencies are used as the base ones for training knowledge engineers: *the formalization capacity in his or her subject area, with the view to the limitations on study methods in use; the ability to use methods and instrumental means of study of professional business items*.

The achievement of these target competencies is facilitated by the common ontological space of knowledge and skills, which is formed by the applied ontologies of courses / disciplines of several tutoring IES and web-IES. It is important to note that the general competency model, which is a component of the ontology base model in the form of a semantic network, is applied in applied ontologies of courses / disciplines as a hierarchy of subject / problem-oriented private competences (with weights) reflecting the methodology of teaching specific courses.

As for the information necessary for the formation of social and personal competences (from the group of universal competences), taking into account the personal characteristics of the students, here you can partially use the information presented in the professional standards in the job description “self-development” for each specialty [10]. In addition, to identify personal characteristics, there are a large number of psychological tests, surveys, there are special sites, etc. For example, in the context of tutoring IES and web-IES [3,4] for building models of students, the possibility of identifying about 20 personal characteristics and their correlation with an individual learning model. The main problem here is the search and selective selection of expert information, signaling the degree of manifestation of specific competence for each of the personal characteristics.

As the experience of the development and use of tutoring IES and web-IES in the educational process showed, the main problems in the formation of professional and universal competences are:

- selective selection at each stage of training (bachelor, master) of the knowledge, skills and abilities that students should acquire (applied ontologies of courses / disciplines, generalized ontologies of individual areas of training are used);
- improvement of methods of control and testing, conducted both with the purpose of forming current competence-oriented models of students, and upon completion of training (using the web-testing of students with the generation of variants based on a genetic algorithm);
- effectively taking into account the personal characteristics of students when selecting and shaping learning strategies and influences, including the development of special corrective learning influences aimed at developing individual student personal characteristics (the results of psychological tests of students are used together with various types of learning interactions);
- the use of additional (repeated) training on the basis of the identified gaps in knowledge and skills, etc. (the sets of learning interactions are used for different clusters of students).

In general, based on the experience of building competence-oriented models of graduates (bachelors and masters) in the field of Software Engineering, it is necessary to provide solutions to the following important tasks in the organization of the modern educational process:

- conceptual understanding of the programming process and instilling computational and logical thinking skills for the implementation of specific tasks in the form of a program and / or a software system;
- selection and use of various programming paradigms, including object-oriented, functional, logical, environment-oriented, and various types of programming languages, as well as the development of competence in the use of a particular language;
- application of new approaches from the field of AI by presenting typical knowledge from the field of programming in the form of plans for solutions [9,16], to help students who have difficulty in moving from a method or algorithm for solving a problem to its software implementation.

The monitoring of the functioning of tutoring IES and web-IES in this case is associated with “tracking” and analyzing all the processes of building for each student an personalized model of the student in the relevant discipline by identifying the current level of knowledge / skills using web testing and other methods, as well as the formation of a psychological portrait of the student's personality as an important component of the student model.

It should be noted that, in accordance with the problem-oriented methodology, the basis of the approach for constructing the current competence-based model of the student is a dynamic comparison of the results of web testing with the corresponding fragment of the applied ontology of the course / discipline. The result is the so-called "problematic areas" [3] in the knowledge of students in individual sections / subsections and the construction of current competencies, jointly reflecting the state of the student model not only in terms of knowledge level, but also providing a conceptual and technological connection with the processes of identifying skills solve some types of educational non-formalized tasks recommended in [3] or training in knowledge engineering.

It is also necessary to constantly form lists of students (contingents) with high and / or low indicators of knowledge / skills, conduct systematic statistical data processing, as well as ensure the generation of current and final reports (statements) for departments and deans.

The final term logs that reflect the students the competence-oriented students models contain complete information about the students - assessments obtained during the control measures related to the identification of knowledge and skills, the current level of professional competence, information about passing psychological testing, information about independent work, the final forecast grades, as well as a real grade obtained in the exam (statements are formed for all students enrolled in a particular course / discipline).

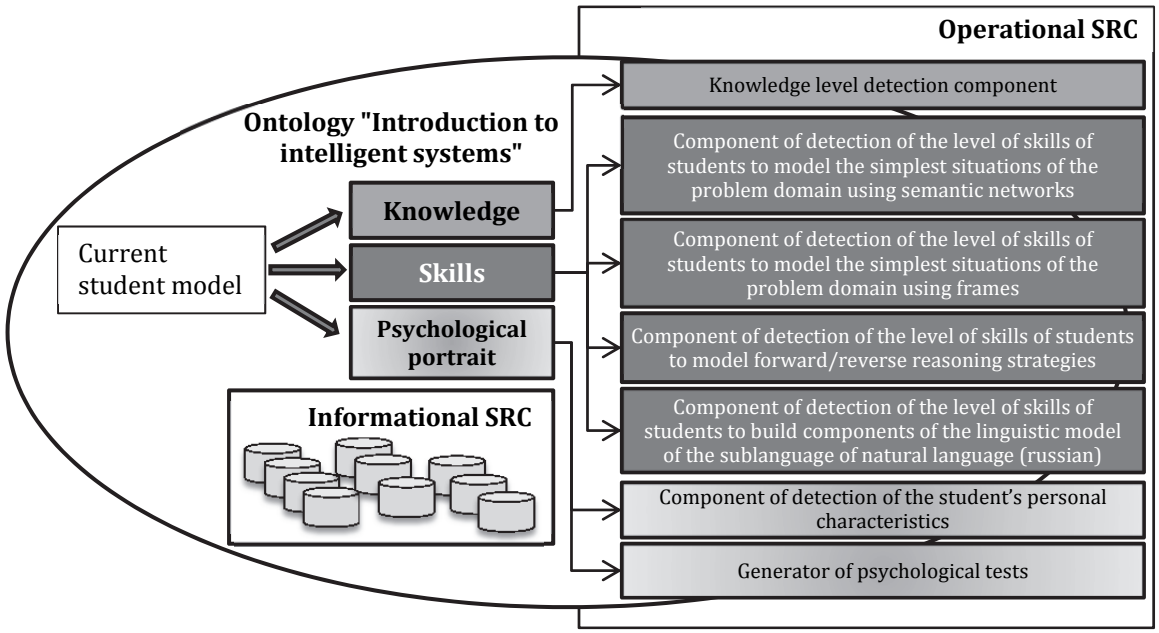

Figure 1 – The technology of automated construction of competence-oriented models

The figure 1 shows the technology of automated construction of competence-oriented models (where SRC – standard reusable components). An example of the implementation of the processes of automated construction of competence-oriented models of specialists in the field of knowledge engineering based on the ontology "Intelligent Systems and Technologies" is given in [22].

Conclusion

Thus, the methodical and technological experience in the domain of automated designing using tutoring IES and web-IES of competence-oriented models of specialists in the field of SE (PT) and AI showed, that big opportunities exist today, that enable to promptly and efficiently review, adjust (focusing on the most modern innovations in the professional sphere) and predict the level and quality of the graduate professionals’ cohort. This approach lays the foundation for not only in relations with employers and potential customers, but also allows to plan targeted training of specialists in various areas, starting with junior courses.

Acknowledgements

The work was performed with the Russian Foundation for Basic Research support (Project No. 18-01-00457).

References

[1] I.V. Vel'bitskiy. Tekhnologiya programmirovaniya. Kiev: Tekhnika, 1984.
[2] E.M. Lavrishcheva. Programmnaya inzheneriya. Paradigmy, tekhnologii i CASE-sredstva / Uchebnik dlya vuzov. M:Izdatel'stvo Yurayt, 2016.
[3] G.V. Rybina. Intellektual'nye sistemy: ot A do Ya. Seriya monografiy v trekh knigakh. Kniga 1. Sistemy, osnovannye na znaniyakh. Integrirovannye ekspertnye sistemy. M.: Nauchtekhlitizdat, 2014
[4] G.V. Rybina. The Theory and technology of integrated expert systems construction. Monograph. M.: Nauchtekhlitizdat, 2008.
[5] Kibernetika: lyudi, gody, sobytiya. Kollektivnaya monografiya / Pod red. G.V. Rybinoy. M.: Radiotekhnika, 2013.
[6] M.S. Farooq, S.A. Khan, F. Ahmad, S. Islam, A. Abid. An Evaluation Framework and Comparative Analysis of the Widely Used First Programming Languages. PLoS ONE 9(2): e88941, 2014.
[7] R. Sebesta. Concepts of Programming Languages, 11th Edition, Pearson, 2015.
[8] S. Lucci and D. Kopec. Artificial Intelligence in the 21st Century, 2nd ed. Dulles, Virginia: Mercury Learning & Information, 2015.
[9] C. Schweikert. Programming Languages with Plan Knowledge Representation for Learning // In Proceeding of the Int'l Conf on Artificial Intelligence (ICAI 16): CSREA Press, 2016.
[10] Portal Federal'nykh gosudarstvennykh obrazovatel'nykh standartov vysshego obrazovaniya www.fgosvo.ru.
[11] G.V. Rybina. Intellectual technology of construction of training integrated expert systems: new opportunities // Open education. Vol.21. No. 4: 43-57, 2017.
[12] B.D. Nye. Intelligent tutoring systems by and for the developing world: A review of trends and approaches for educational technology in a global context // International Journal of Articial Intelligence in Education. №25: 177-203, 2015.
[13] D. Bonner, J. Walton, M.C. Dorneich, S.B. Gilbert, E. Winer, R.A. Sottilare. The development of a testbed to assess an intelligent tutoring system for teams. Workshops at the 17th International Conference on Artificial Intelligence in Education, AIED-WS 2015. CEUR Workshop Proceedings, 2015.
[14] A.A. Rahman, M. Abdullah, S.H. Alias. The architecture of agent-based intelligent tutoring system for the learning of software engineering function point metrics. 2nd International Symposium on Agent, Multi-Agent Systems and Robotics, ISAMSR 2016: 139-144, 2016.
[15] S. Sosnovsky, A. Mitrovic, D. Lee, P. Brusilovsky, M. Yudelson. Ontology-based integration of adaptive educational systems. 16th International Conference on Computers in Education (ICCE 2008): 11-18, 2008.
[16] G.V. Rybina, Yu.M. Blokhin. Methods and software for intelligent planning for the construction of integrated expert systems. Artificial intelligence and decision-making. Vol. 1: 12-28, 2018.
[17] G.V. Rybina, V.M. Rybin, Yu.M. Blokhin, E.S. Sergienko. Intelligent support of educational process basing on ontological approach with use of tutoring integrated expert systems. Proceedings of the Second International Scientific Conference «Intelligent Information Technologies for Industry» (IITI'17), Vol. 2. Advances in Intelligent Systems and Computing Volume 680. Berlin: Springer: 11-20, 2018.
[18] G.V. Rybina, E.S. Fontalina. Automated construction of young specialists models with the use of tutoring integrated expert systems. Proceedings of IV International Conference on Information Technologies in Engineering Education: 41-44, 2018.
[19] Yu. F. Telnov, V. A. Kazakov. Ontological modeling of network interactions of organizations in the information and educational space // fifteenth national conference on artificial intelligence with international participation of CII-2016. Proceedings of the conference. Vol. 1/ Smolensk: Universum: 106-114, 2016.
[20] G.V. Rybina, Yu.M. Blokhin, L.S. Tarakchyan. Intelligent planning and control of integrated expert systems development process. Advances in Intelligent Systems and Computing, Vol.848: 266-271, 2018.
[21] Development of professional standards for the information technology industry. The Committee on education in the it field. The website APIT http://www.apkit.ru/default.asp?artID=5573
[22] G.V. Rybina, E.S. Sergienko. Nekotorye podkhody k avtomatizirovannomu postroeniyu kompetentnostno-orientirovannykh modeley spetsialistov v oblasti inzhenerii znaniy s ispol'zovaniem obuchayushchikh integrirovannykh ekspertnykh system, 15(7): 3-15, Informatsionnoizmeritel'nye i upravlyayushchie sistemy, 2017.