Project-competent paradigm of knowledge representation of the three-level engineering education system

Bulat Kubekov[1], Leonid Bobrov[2], Elena Savelyeva[3], Vitaliy Naumenko[1], and Anar Utegenova[1]

[1] Institute of Information and Computational Tehnologies, Kazakhstan
b.kubekov@mail.ru

[2] Novosibirsk state university of Economics and Management, Russia
bobrov@nsuem.ru

[3] Kazakh-German University, Kazakhstan
savelyeva@dku.kz

Abstract. In many universities of the world, modernization of engineering training programs for engineering activities is carried out using the CDIO approach [1,2]. This approach has become widely used in the countries of near and far abroad and in particular at the Tomsk Polytechnic University of the Russian Federation. As noted by head of the Department of engineering pedagogy of the University, Doctor of Technical Sciences, Professor A.I.Chuchalin [3,4], the approach provides a framework for the design of multi-level programs of engineering education, is relevant in connection with the requirements of international standards for learning outcomes in higher education institution and the competences of professional engineers[5]. The CDIO approach provides the opportunity to form optimal structures, content and technologies of implementation and quality assessment of level engineering education programs.

It is known that any profession is based on a certain set of theoretical knowledge and practical experience. If this framework is formalized and properly documented, such a body of knowledge can be a starting point for developing layered training programs for specialists, as well as for accrediting academic programs and professional certification.

Recently, the term ontology has become common in the information technology literature. In the works Gavrilova T.A., Subetto A.I., Khutorskoy A.V., Tsukanova N.I. [6,7,8,9,10,11], applications of ontology, as domain models, the development of a knowledge base of intelligent systems, the use of an ontological approach to knowledge management based on the competence approach, and also methods for developing educational and methodical complexes based on ontology are considered.

Keywords: Paradigm of knowledge representation · Competence model · Ontology model · Project teaching method · CDIO initiative .

1 Introduction

As noted in the article L.V. Borovikova [12], the main, constantly escalating contradiction of education, lies in the dialectical interconnection of its main sides - between the renewed content of education offered by the increasingly complex social life

Fundamental and constantly escalating contradiction of education lied in the dialectical interconnection of its main principles - between the renewed content of education, which offered by the increasingly complex social life and its existing form that cannot satisfy the growing needs of society. According to L.V. Borovikova [12] this contradiction, should be resolved, both through the modernization of society, and the introduction of new forms of its organization, and new educational technologies.

In our point of view, in this connect, it should also be noted the relevance of the information education base conversion into a new paradigm of knowledge representation that correspond with the needs of digital economy, with the focus on the skills developing in analyzing information and thinking creativity. Such information base conversion will contribute to increasing the efficiency and quality of teaching, diversifying the methods of electronic pedagogy and the range of educational services, as well as the development of new educational, including Smart technologies.

The problem of the correspondence between the increasingly complex social life and the need for express-update of the education content must be resolved through the proactive modernization of curricula and engineering training plans, and through the development of the epistemological function of the competence approach [13]. The need for a conversion to the new representation of knowledge paradigm, requires the application of knowledge engineering, in particular, ontological engineering methods, as well as the introduction of techniques for intensifying the learning-cognitive activity of students.

Taking into account the analysis of the state of engineering education and the tasks relevant in the light of the implementation of the program for digitalization of the higher school, there were revealed some information:

– there is a general tendency of the need to switch to a new paradigm of the information base of education, since the existing base does not provide, as far as is necessary, the opportunities for diversifying educational technologies and services;

– the opinion of researchers from near and far abroad is reduced to the need for intellectualization of the educational content presentation, the creation of intelligent software systems in education and the transition to Smart-teaching;

– according to the learning and teaching support material, especially for engineering specialties, there is clearly a tendency to switch to a project teaching method, as the basis for the methodology of forming a graduate's competence model;

– the problem of proactive planning of curricula and the design of training programs become more serious, in accordance with the demands of the labor market and the educational processes management, with a focus on individualizing teaching.

2 Concepts and functions of interaction between levels of engineering education system

The solution of the current problems has to see in the context of a systematic presentation of concepts and mechanisms of interaction and in the context of each level of modern education system. Let's show our vision of the hierarchical structure of the three-level education system, using the example of the specialty "Software Engineering".

Only competent scientific-educational and educational-methodical composition of these levels of engineering education, under conditions to integration with the design and competence paradigm of knowledge representation, and the mandatory participation of IT industry experts, will be able to provide high-quality and competitive training of specialists of appropriate levels in modern areas and areas of the digital economy, and to justify the expectations of the labor market.

The highest level in inheritance hierarchy for a given chart is the level of Doctoral degree which should be associated with learning the future, within the framework of the fundamental disciplines of the scientific areas of Computer Science and Software Engineering.

The term "learning the future" presupposes disciplines that have a long-term and slowly aging character or special aspects. It can also be noted that the disciplines of learning the future are disciplines based on the laws of the development of the industry of information and telecommunication facilities, technologies and systems, taking into account the forecasting of such development.

The competence model of the PhD level is a composition of universal competencies in the areas of Computer Science and Software Engineering, the fundamental and professionally oriented disciplines of which constitute the PhD-level educational and research base, and regulate problem-oriented learning of master's level.

The second level is the level of magistracy, which have to follow nuanced approach when implementing scientific and educational trends of fundamental and professional-oriented disciplines of Computer science and Software Engineering direction and PhD level.

The competence model of magistracy level is the composition of competencies of the direction of the Software engineering and problem-oriented Software_skills, and forms the basis for planning curricular and master's degree programs.

These disciplines have to be adapted to real needs of IT-industry and, first of all, be oriented to the modern development trends and issues of problem areas of Computer science and Software Engineering. Therefore, basic and profiling disciplines of magistracy have to define curricular trends of bachelor degree and regulate problem-oriented education, and the corresponding trend from the disciplines of the master's program is to determine the educational and methodological base of the bachelor's level.

The first level of three-level education system is the baccalaureate level. General, basic and profiling curricular subjects of bachelor specialty "Software Engineering" have to realize curricular trends of basic and profiling magistracy disciplines, and competence bachelor model have to be composition of competencies in SWEBOK area of expertise and subject-oriented Software_skills.

Disciplines of the curriculum at the bachelor's level should be connected to the acquisition of engineering knowledge by specialty, the experience of project and innovation activities, the development of their professional skills and subject-oriented Software_skills, the development of software products and systems in their professional activities

Thus the represented model of three-level education system in the context of specialty "Software Engineering" presupposed the following:

– mandatory continuity of curricula and programs between the levels of the training system;

– determination of competence models of all levels of the education system;

– definition of scientific and educational and educational methodical trends of the PhD and Masters levels, respectively;

– the need to move from the traditional information base of education to knowledge engineering, on the basis of the design and competence paradigm of knowledge representation, and using ontological engineering methods;

– development of mechanisms for proactive planning and adaptation of the content of curricula, programs and subjects taught, in accordance with scientific and technological challenges and labor market needs;

– the use of knowledgeable components of repeated use and their reuse for the design of educational programs and curricula;

– transition to the provision of educational services, using Smart technologies;

– the need for open management of educational processes and individualization of education.

3 Project-oriented and competence paradigm of knowledge representation

Paradigm is a term widely used in modern systems design and designating a way of constructing systemic abstractions, based on commonness and variability [3]. Under the paradigm in education, we will understand the basic model of a specific method of organizing educational information, based on the properties of community and variability, and, as a whole, as a leading approach to the choice of content and forms of organization of educational resources.

As a base model of knowledge mapping, in our studies, a network model is adopted - an information model of the domain that has the form of an oriented graph whose vertices correspond to objects in the domain and the arcs define relations between them. Formally, the network can be specified in the following form: $H =< I$, C, $G >$, where I is the set of information units; C - the set of types of links between information units; G is a mapping that specifies specific relationships from existing types (relationships) between elements. In the semantic network, the role of vertices is played by the concepts of the knowledge base, and the arcs (and directed ones) set the relations between them. Thus, the network model reflects the semantics of the domain in the form of concepts and relationships.

To concretize the network model and further practical usage, there was accepted the ontology model that is defined as the triple of the following sets: $Om =< C$, R, F>. Where C is the set of concepts (terms) of the subject area. R is the set of relations between concepts. F is the set of interpretation functions whose definitions are defined on the relations between concepts in ontology.

The ontology might be represented visually by oriented graph G, peaks of which are conceptions, and edges are relations between them. There is the ontology model represented by an oriented graph, the root vertex of which is a basic notion of the subject area in our constructions. The identification of a support concept is limited to no more than two levels of child hierarchy concepts, in which any optional concretizing concept is a hanging vertex, that is, a vertex that does not have its child concretizing concepts.

Thus the ontology model is represented as hierarchical conception structure, root vertex of which is main concept, the second level is the identifying concept of main concept and, last of all, third level incudes concretizes concepts of each identifying ontology concepts [14,15,16,17].

Next, consider the formalisms of each of the elements corresponding to the sets of the ontology model used in the specially developed language of the knowledge specification. This language allows you to describe the basic concepts of the domain in an unambiguous and formalized form. The language of the knowledge specification is, firstly, endowed with rich possibilities for semantic reflection of declarative knowledge of the subject domain, and secondly, it allows to store them in the repository, as well as to accompany and change.

Declarative knowledge is knowledge about the world of the problem, that is, knowledge that describes the properties of the subject area in which the task is solved. Declarative knowledge, laid in the basis of the competence model of the learner, determines a dynamic set of knowledge, skills and skills that are necessary for his future professional activities and personal development.

As example lets take into account SWEBOK area of Software Engineering knowledge, which is connected with software requirements [18, 19]. Declarative knowledge of the field of knowledge "Software requirements" might be represented as a set of seven main concepts: C_1, C_2, C_3, C_4, C_5, C_6, C_7, each of them have two levels of identifying and concretizing concepts. For example, for the main concept C_2 - "User requirements", identifying concepts are C_{21} - user functional requirements, $*C_{22}$ - non-functional user requirements and $*C_{23}$ - script of events.

Identifying concept is the first level concept of main ontology concept with the help of which are determined semantic and distinctive properties of given main concept against the other main concepts of the subject area.

Concretize concept is the second level concept of main ontology concept, which in the context of its main identifying concept with the help of typical, or all possible combinations of child concepts, which have a clear and unambiguous description of the supporting concept.

In the expression of knowledge, the signs: '*', '+' and '~' define the relations "composition", "aggregation" and "alternative choice" on concepts of ontology, denoting accordingly the properties of generality (obligatory) and variability (optional). At the same time, the relation "composition" is a relation in which the property of generality for the child concept is reflected, and the obligatory presence of it in all instances of the parental notion of ontology; the "aggregation" relation is the relation in which the community property for the child concept is reflected, and the optionality of its presence in instances of the parental notion of ontology; the relation "alternative choice" is the relation in which the property of the variability of the child concept is reflected and the optionality of its presence in the instances of the parental concept of ontology.

With the help of the expression of knowledge, we can formulate an assertion about any main concept in the form of a predicate, which is always true. A predicate is a narrative sentence containing subject variables, the replacement of which by constant values transforms the sentence into a statement - true or false.

The accepted types of concepts and relationship on concepts, we show as example on software engineering SWEBOK subject area: "Software requirements" declarative knowledge of which we represented by ontologies of seven main concepts:

C_1 – Business requirements.

$*C_{11}$ – business-requirements;

$*C_{12}$ – restriction on system behavior;

$*C_{13}$ – concept document.

C_2 – User Requirements.

$*C_{21}$ – functional user requirements;

$*C_{22}$ – non- functional user requirements;

$*C_{23}$ – script of events.

This identifying concept is the semantic context of the second concretize level, which includes following concepts: $*C_1$– usage variant, $+C_2$ – the diagram of data stream or $+C_3$ – the diagram state transition or $+C_4$ "event and answer" table.

C_3 - Subject area requirements.

$*C_{31}$ - functional requirements;

$*C_{32}$ - non- functional requirements.

This identifying concept is the semantic context of the second concretize level, which includes following concepts: $*C_1$ - project targets, $*C_2$ - attributes of quality, $+C_3$ - project structure and restrictions of outward.

C_4 - System requirements

$*C_{41}$ - system requirements organization;

$*C_{42}$ - metrics of the quality of system requirements;

$+C_{43}$ - backward requirements.

C_5 - Non-functional requirements.

$*C_{51}$ - Project requirements. This identifying concept is the semantic context of second concretize level, which includes associated concepts: $*C_1$ - operation requirements, $*C_2$ - efficient requirements, $*C_3$ - reliability requirements, $+C_4$ - endurability requirements, $*C_5$ - error management, $*C_6$ - interface requirements.

$*C_{52}$ - Organization requirements. This identifying concept is the semantic context of the second level of concretize, which includes the following associated concepts: $*C_1$ - production requirements, $+C_2$ - requirements for realization, $*C_3$ - requirements for standards.

$*C_{53}$ - External requirements. This identifying concept is the semantic context of the second level of concretize, which includes the following associated concepts: $*C_1$ - requirements for cooperation, $+C_2$ - ethic requirements, $+C_3$ - juristic requirements.

C_6 - Requirements validation.

$*C_{61}$- authenticity;

$+C_{62}$- consistency;

$*C_{63}$- completeness;

$*C_{64}$- realizability.

C_7 - Requirements management.

$*C_{71}$- management planning;

$*C_{72}$ - change management.

For example for main concept C_5 "Non-functional requirements", expression will have the following view:

$$C_5 <= *C_{51}(*C_1 * C_2 * C_3 + C_4 * C_5 * C_6) * C_{52}(*C_1 + C_2 * C_3) * C_{53}(*C_1 + C_2 + C_3) \quad (1)$$

The Figure 1 below shows the expression as oriented graph. Solid spheres determine concepts, which have community properties, and empty spheres have the property of variability.

Our research also uses a characteristic model that has a similar syntax and relationships, but allows one to map the relationship of the support concept to characteristics whose values belong to certain types of information [20].

A characteristic is a distinguishable essential property, as well as a significant and accessible aspect of the concept. Characteristic, from the point of view of conceptual modeling, is an important property that allows us to identify similarities and differences between instances of a supporting concept.

The characteristics are indispensable in formulating a brief description of the properties of the concept and for identifying the differences between instances of the concept.

By the characteristic model Fm we mean a model with a generalization of characteristics and variable parameters of the concept, as well as a logical justification of the composition and aggregation of characteristics for specification of the properties of the concept and its instances. The characteristic model captures

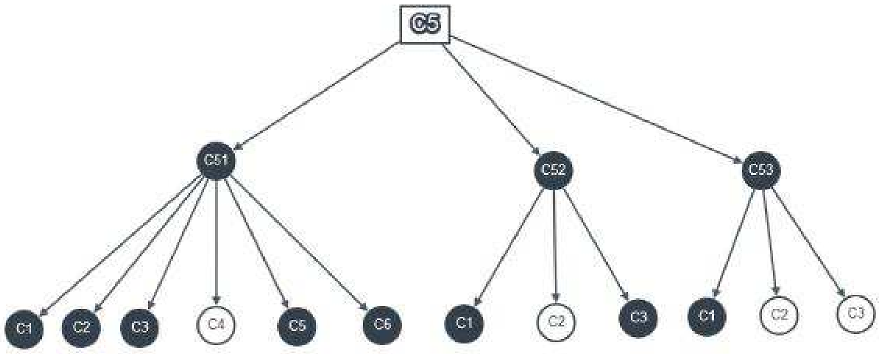

Fig. 1. Graph of main concept C_5 - "Non-functional requirements"

all properties that relate to the possibility of configuring reusable components and reusing them.

Figure 2 shows the characteristic model Fm of the concept of C_i.

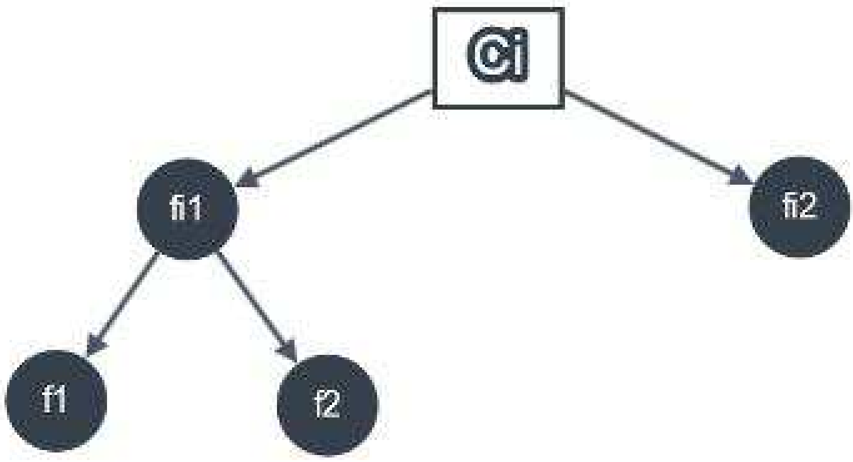

Fig. 2. Characteristic model of the concept C_i

As we see the concept C_i has the composition of obligatory concretizing characteristics f_{i1} and f_{i2} and the characteristic f_{i1} has its own concretize level as composition of associated obligatory characteristics f_{i1} and f_{i2}. Thus, instances of the concept C_i can be admissible sets of the following mandatory characteristics: f_{i1}, f_1, f_2, f_{i2}.

The specification of the reference concept C_i by the characteristic model will be called the specification expression, which will have the following form:

$$C_i <= *f_{i1}(*f_1 * f_2) * f_{i2} \quad (2)$$

Let's consider an example of the characteristic model of the reference concept C - "Pointer to the program object", the topics "Index in C programming language", discipline "Programming technology", read in the first year to students of the specialty "Software Engineering".

Identifying concepts of the ontology of the reference concept C are: C_1 - "Pointer to a static program object" and C_2 - "Pointer to a dynamic program object", which can be represented by the following expression of knowledge: $C <= +C_1 * C_2$;

The characteristic model of the reference concept "Pointer to the program object" is described by the following characteristics:

– f_1 - the value of the variable of the index type;
– f_2 - direct access to the program object;
– f_3 - indirect access to the program object;
– f_4 - operations on pointers.

Then the expression of the specification of the characteristic model will have the following form:

$$C <= *f_1 + f_2 * f_3 * f_4 \quad (3)$$

In the form of a composition of mandatory identifying characteristics: f_1, f_3, f_4, and aggregation of the optional characteristic – f_2.

Figure 3 shows the characteristic graph of the main concept C – "Pointer to the program object":

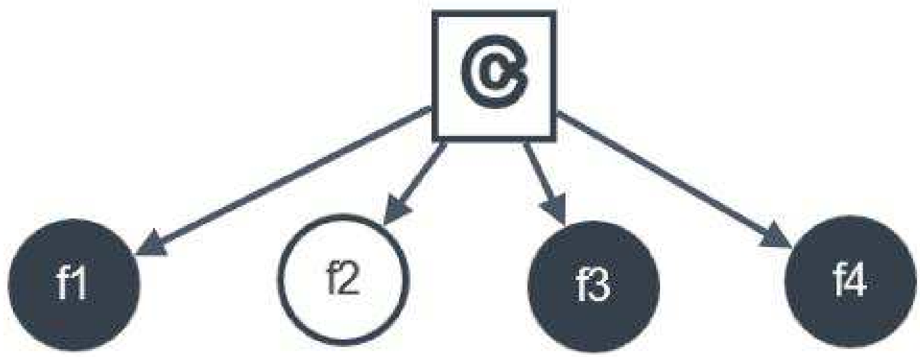

Fig. 3. Characteristic model of the concept C_i

For identifying concept C_1 - "Pointer to the static program object" was the following characteristics were singled out:

– f_{11} – unary operator '';
– f_{12} – unary operator '*';
– f_{13} – constant pointer;
– f_{14} – constant pointer.

Then the characteristic model of the identifying concept C_1 – "Pointer to a static program object", can be represented by the following specification expression:

$$C_1 <= *f_{11} * f_{12} + f_{13} + f_{14} \tag{4}$$

As composition of obligatory f_{11}, f_{12}, and aggregation of non-obligatory associated characteristics f_{13} f_{14}.

4 Conclusions

Thus, in this article, a systemic vision of a three-tier higher engineering education is considered, with the example of the specialty "Software Engineering". Such a vision presupposes the mandatory continuity of curricula and programs between levels, the determination of competence models of all levels of the engineering education system, and on their basis, the definition of scientific and educational and educational-methodical trends.

The solution of the problem of transition to a new educational information base based on the design and competence paradigm of knowledge organization and ontological engineering is presented. The transition from the traditional information base of education to the engineering of knowledge at all levels of the engineering education system presupposes the introduction of innovative models of knowledge mapping, the design of which must be adopted by the project teaching method, according to the CDIO initiative, and the presentation of each stage of the initiative by the relevant competence model. Declarative knowledge, as the basis of the competence model of each of the CDIO stages, seems to be a necessary and sufficient set of ontologies of the basic concepts of the semantic content of educational resources.

The developed language of the specification of knowledge is endowed with sufficient opportunities for analyzing the structure of information and the relevant mapping of the semantics of educational resources; allows to take into account the properties of the generality and variability of concepts, which makes it possible to configure knowledgeable components of repeated use, and their use for the design of educational disciplines and individual educational programs.

This work was supported by the grant of the Ministry of Education and Science of the Republic of Kazakhstan (project No. AP 05134973 "Research and development of models and methods for the presentation and organization of knowledge using the ontological approach and tools of Smart technology, in the implementation of educational programs and processes").

References

1. Crowley E.F. : The CDIO program: Description of the goals and objectives of undergraduate engineering education. The CDIO Report No.1 ed. MIT, http://www.cdio.org. Last accessed 4 Oct 2017
2. E. Crawley, J. Malmgvist, S. Ostland, D. Brodeur, K. Edstrom: Rethinking Engineering Education, the CDIO Approach. 2nd edn. Springer,(2014)
3. Chuchalin A.I.: Modernization of bachelor's degree in the field of technology, taking into account international engineering education standards // Higher education in Russia,(10), (2011) .
4. Chuchalin A.I.: About application of the CDIO approach for design of level programs of engineering education // Higher education in Russia. 2016. No.4(200). pp.17-32
5. IAE Graduate Attributes and Proffessional [Electronic resource]. http://www.ieagreements.org/IEA-Grad-Attr-Prof-Competencies.pdf
6. Gavrilova T.A.: Ontological approach to knowledge management in the development of corporate automation systems / News of artificial intelligence.-2003.-No.2.- pp.24-30.
7. Gavrilova T.A., V.F.Khoroshevskyi: Knowledge bases of Intellectual systems. Textbook. - SPb.: Peter, (2000).
8. Subetto A.I.: The integration model of a university graduate on the basis of a system-specific and competence-based approach. - Kostroma: Publ.house of KSU n.a.N.A. Nekrasov, pp.28,(2005).
9. Elina E.G., Friesen M.A.: Educational technologies and methods of higher education in the United States and European countries. g. Educational technology. Publishing House "National Education",(1) pp.31–37, (2013).
10. Tsukanova N.I.: Ontological model of knowledge representation and organization. Textbook for universities.- Moscow: Hotline-Telecom, pp.272, (2015).
11. Tsukanova N.I.: Ontology of an educational and methodical complex. / Tsukanova N.I., Strakhova Z.V. / / Bulletin of the Ryazan State Radio Engineering University.(1), (2013).
12. Borovikova L.V.: Innovation in education through integration ontological and approaches. Leningrad institute of education development OGU Newsletter **2**(121)/February, pp.84, (2011)
13. Khutorskoy A.V.: Competence approach in training. Scientific and methodical manual. - .: Publishing house "Eidos", pp.73 ("New standards" series), 2013.
14. B.Kubekov, I.Utepbergenov.: The use of multiparadigm approach to knowledge modeling.- // 7th International Conference on Education and New Learning Technologies".- Barselona (Spain), 6th-8th of July, pp.136–141, (2015).
15. B.Kubekov, J.Kuandykova, I.Utepbergenov, A.Utegenova: Application of the conceptual model of knowledge for formalization of concepts of educational content //9th International Conference on Application of Information and Communication Technologies AICT2015.- Rostov-on-Don, 14-16 Octoube, . 294-306, (2015).
16. B.Kubekov: Educational components formation technology for the planned CDIO SYLLABUS education. 9th annual International Conference of Education, Research and Innovation - ICERI2016, Seville (Spain), 14-16 November, (2016), pp.6139–6145 ISBN: 978-84-617-5895-1 / ISSN: 2340-1095.
17. Kubekov B.S., Ditmur Beyr, Utegenova .U., Zhaksybaeva N.N.: Innovative paradigm of education of knowledge - competency form based on ontology. Journal of theoretical and applied information technology 15th November (2017), Vol.95. 21, 2005-ongoing JATIT@LLS (E-ISSN 1817-3195 / ISSN 1992-8645). pp 5859-5868.

18. Vigers Carl: Development of software requirements. Trans. with English-M .: Publishing house - trading house "Russian Edition", pp. 576, (2004)
19. Kubekov B.S.: Technologies of software development: Textbook - Almaty: Economy, - pp.307, (2011).
20. Czarniecki K., Eisenecker U.: Generative Programming: Methods, Tools, Application. For professionals. - St. Petersburg: Peter, pp.731, (2005)