Journal of Fundamental and Applied Sciences

ISSN 1112-9867

Available online at http://www.jfas.info

Research Article

Special Issue

CONCEPTUAL BASIS FOR DEVELOPING OF TRAINIG MODELS IN COMPLEX SYSTEM SOFTWARE ASSEMBLING GENERATOR

V. O. Georgiev*, N. A. Prokopyev, D. S. Polikashin

Kazan (Volga region) Federal University

Published online: 08 August 2017

ABSTRACT

This paper presents conceptual basis for developing of training models of interactive assembling system for automatic building of application software systems, obtained during practical works over "Design and architecture of software systems" and "Object-oriented analysis and design" courses. The system is intended for practical demonstration of basic stages and operations in development of complex and critical software. This also includes a consideration of solutions for some substantial problems of complex systems software generation, such as: program module compatibility, formalization of computer interaction and choosing of formal model for human machine interface. In addition, several training model implementations are provided.

Keywords: software generator, training models, interactive systems

1. INTRODUCTION

The "Design and architecture of program systems" and "The object-oriented analysis and design" (methodical grant [1]) developed during practical works on training courses conceptual bases of creation of educational and model options of the interactive assembly system which is carrying out in the automatic mode (with preliminary interpretation and the analysis of fragments of program codes) assembly of program systems of applied appointment are presented in article.

Author Correspondence, e-mail: VOGeorgiev@kpfu.ru

doi: http://dx.doi.org/10.4314/jfas.v9i2s.77

In article, at this stage not to be set yet the purpose to provide full and general architecture and the technological principles of creation of generators of program systems. The tasks of systematization of results of the lecture and practical training received during of the mentioned courses, and also, a problem of generalization and use of the practices made earlier of formal models and methods of representation of knowledge of subject domains of interactive systems are set. The solution of these tasks will allow to come to the level of development of conceptual bases of creation of model prototypes of such systems, and, to the subsequent transition to model versions of PO generators, for the purpose of a code illustration of basic provisions of training courses of "Program engineering".

The represented model options of system are intended for evident practical demonstration of maintenance of the main stages and works when developing difficult, critical software. Will be carried to the main stages: design and project management (administrative, automatic or automated), testing of the created system, its documenting, quality assessment, estimation of cost. Assembly is carried out in the interactive interaction with the user going on a communication vernacular. The vernacular of communication is constructed on the basis of a concept of the intelligent interface and is implemented on the basis of a combination of rigidly programmed dialogue (at the "top" level of transition from one stage to another) and "free" scenario dialogue of communication (at the "lower", "intra landmark" levels).

Program realization of scenario dialogue is constructed on the basis of model code representation of the main formal models of the interactive dialogue systems of human-machine interaction presented in the author's course "Design of the Human-machine Interface", the Program Engineering direction of the Kazan (Volga region) Federal University and basic concepts and concepts of the theory of schemes of programs read to students. One of examples of such program realization is presented in [2].

Assembly can be carried out in several modes:

1. Mode of initial codes;
2. The mode of the executed codes;
3. A combination of the initial and executed codes.

Generation of program systems is constructed on the basis of use of concepts of "The theory of schemes of programs", her basic concepts of standard constructional schemes and methodologies of program engineering. Program realization of system is made in several options, depending on that what formal model of the scenario of dialogue is put. As formal models are allocated: graph (automatic) model, model on the basis of functional networks of Petri, model on the basis of semantic networks, model on the basis of relational data, model on the basis of the theory of formal languages, probabilistic model, model on the basis of game theory, tensor model and some other models.

The experience got by students at creation of models of such systems allows to master, also, and the basic concepts of a training course "Theory of computing processes and systems", and also, to seize practical programmer skills on creation of difficult crucial program developments.

Realization of systems is executed on the basis of use of the main tools of the language C# and JAVA environments.

In article (with specification as far as it will be possible on its volume) it is, also, supposed to consider the solution of the following problems arising at generation ON difficult systems:

1. Development and systematization of conceptual bases of the production technology of PO generators of difficult systems and methodologies of their creation.
2. Problems of compatibility of program modules.
3. Problems of formalization of computer interactive interaction.
4. Choice of formal models of the human-machine interface (HMI).

2. SYSTEMATIZATION OF CONCEPTUAL BASES OF TECHNOLOGY OF DEVELOPMENT OF GENERATORS ON DIFFICULT SYSTEMS AND METHODOLOGIES OF THEIR CREATION

At design of it ON as the assembly generator of difficult systems which represents a difficult program system there are numerous tasks demanding assessment of quantitative and qualitative regularities of processes of functioning of system. In a general sense - problems of modeling. At development stages of technical and working projects of systems, models of separate subsystems are detailed, and modeling serves as the effective tool for the solution of specific objectives of design, i.e. the choice of an optimal variant (from a set admissible) solutions of a task of a certain criterion, at the set restrictions. When modeling difficult systems it is necessary to consider the following their features:

- complexity of structure and a complexity of communications between elements, ambiguity of algorithms of behavior under various conditions

- large number of parameters and variables, incompleteness and non-determination of initial information
- variety and probabilistic nature of influences of the external environment.

One of the systematic, disciplined, measurable approaches in program engineering to development of program systems can call the model and model approach based on the technological principles considered in works [3, 4].

2.1. Model and model approach.

The essence of model and model approach * makes use of methodology of the mechanism of the theory of schemes of programs, with allocation of standard schemes of functional interaction of the PO elements with full or partial formalization of their functioning and program realization of model models of the PO elements. At practical program realization the method which essence in performance of the following items is used:

1) Definition of the purpose of a task and deepening in a being of a task.

2) Establishment of local goals and splitting task into subtasks.

3) Detection of specific features of the software developed for the solution of a task.

4) Search of the expert specializing in area to which the task and receiving its consent to the help in system development belongs.

5) Work together with the expert on several experimental examples of applied tasks which the developed program system has to solve.

6) Choice of the software necessary for creation of system. The choice depends on type of a solvable task, financial opportunities and complexity of the software.

7) The choice of technological means at which program tools, acquisition of tools will work.

8) Creation of the laboratory prototype of system allowing coping successfully with examples of the tasks solved in point (5).

9) Design of system with the assistance of the expert. Identification of entities, interrelations between them, types of hierarchies, classes.

10) Performance of necessary number of iterations of system development with test in real practical conditions.

11) Development of documentation to system.

From the first steps of implementation of the project needs to seek for construction at least the model limited by opportunities, but correctly working for improvement of interaction with the expert for what model and model approach is used.

Note. * In article emphasis on generation ON standard systems is placed so far, the questions connected with the custom software will be considered in the following publications.

3. PROBLEMS OF PROGRAM MODULES COMPATIBILITY.

According to the offered approach it is necessary to define originally according to points 1-3 (section 2.1) the main prime problems which we will face.

There is a set of unresolved, now, problems of the difficult systems arising when developing the software:

1) lack of the return compatibility between various versions of the program components and modules which are logging in takes place. What can bring, to:

to violation of integrity and unity of the ecosystem built around this program component and to delay of rates of its development because of dissociation of community of the programmers using incompatible versions of this software in the developments;

to delay of realization and development of the projects using given ON;

to use in development of the system of program components which are outdated that in turn involves problems with safety and overall performance of system;

to economic costs, at implementation of transition of a program system to the new version of a program component.

2) Many program components presented in the form of initial codes are involved in operation of the PO generator of difficult systems. As a part of which there can be two or more components the same module realized in the different versions not compatible among themselves having in the dependences.

Thus, the PO generator of difficult systems has to include a possibility of a solution of the problem of compatibility of program modules, so conform to some requirements which will be provided further. The generator has to incorporate the specialized software aimed at their decision.

3.1. Offers on the solution of problems of compatibility.

During preparation of materials for the publication the research [5] for identification of key factors of implementation of the automated migration of initial codes of appendices from libraries, earlier on more recent versions, was conducted. By results of a research four classes of the problems which are most often found at implementation of manual migration on the new version of library were revealed:

1. Change of a signature of program components with preservation of their semantics;
2. Change of quantity of arguments of functions;
3. Change of requirements to initialization or preliminary control of an environment.
4. Lack of the available description of all nuances of transition to the new version.

Thus, in the course of software development, aimed at a solution of the problem of compatibility, it is necessary to consider a large number of various components of aspects, such as:

Sources of basic data:

Syntax of programming languages;

Frameworks and specialized program libraries;

Formats of submission of documentation;

Classes of the problems leading to loss of compatibility between versions of libraries;

Algorithms of search and elimination of compatibility problems.

Therefore specialized software - appendices have to meet the following conditions:

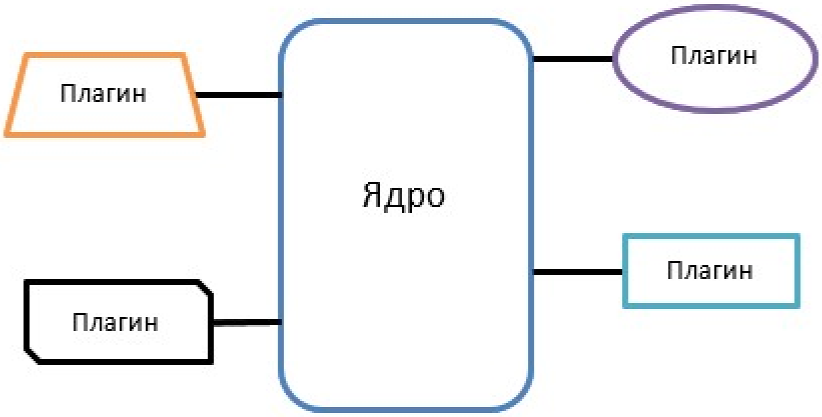
Expansibility that is to have an opportunity to add new entities and functions to system, without breaking its main structure, to provide a simple opportunity to add functionality of support of modern languages of programming and new standards of the existing languages, new ways of search and elimination of problems of compatibility.

- that is to provide to flexibility (Modularity) a possibility of parallel work on the components responsible for various functionality.

Also important aspects are ensuring system effectiveness, its reliability and productivity.

3.2. Architecture of specialized means of the assembly generator.

For satisfaction of the above requirements the micro-nuclear [6] architecture of the appendix was chosen. This architecture consists of two types of components: main system (kernel) and plug-ins. The kernel contains a minimum of logic of the main functionality, but directs loading, unloading and start of necessary plug-ins. Thus, plug-in is the intermediate dependences which are not connected with each other.



Plug in Core

Fig.1. Scheme of micro-core architecture

Productivity of the application created on the basis of such architecture directly depends on the number of the connected and active modules. Also, in need of bigger productivity, such architecture is easily modified in micro-service by removal of the main logic from a plug-in in separate service process. In that case the plug-in is used only for communication of service with a program kernel.

The kernel of system provides the command interface for control and action for search and loading of plug-ins from external storage (function of the manager of packages). The main destination of a kernel is work with the loaded plug-ins:

- Installation, inclusion, shutdown and removal.
- Configuration.
- Coordination of work of plug-ins among themselves.
- Interaction of plug-ins and user.

On the carried-out tasks plug-ins are divided into the following types:

1. The plug-ins focused on identification of problems of compatibility on the basis of the approach considered in articles [7, 8]. On an entrance receive various versions of library between which it is necessary to carry out migration of the target software. This look is subdivided on:

- The plug-ins depending on syntactic designs (language, a framework, a configuration format). Abstract representation of model of the program (is in detail described in [9]) reflecting key characteristics for finding of problems of compatibility is result of work.
- The plug-ins processing specially prepared abstract models. Such plug-ins are specified on strictly certain types of problems. The list of the changes leading to loss of the return compatibility is result of work.

2. The plug-ins responsible for elimination of problems of compatibility in a target code. These plug-ins always depend on syntax of the processed structure [10] and are directed on:

- Reduction of recommendations and instructions for implementation of migration of the program between versions of library the programmer.
- Carrying out the automated transformation of the source code of the target program and configuration files of system of assembly ON agrees to approach [11].

3. Plug-ins responsible for interaction with users.

4. PROBLEMS OF FORMALIZATION OF COMPUTER INTERACTIVE INTERACTION

The present stage of development of software is characterized by wide circulation of the interactive (dialogue) systems providing the solution of various classes of tasks on the basis of optimum division of functions between the user and the computer. One important aspect of dialogue system is the possibility of multiple, flexible and at the same time simple and user-friendly setting up the program. As the quantity of options of use of the PO generator of difficult systems is very broad and there is a set of features of each generated system, the used programming languages, libraries and frameworks, features of its configuration and the target environment for expansion, the possibility of such setup of the generator is priority. Thus, at creation of the human-machine interface of the generator the format of dialogue system was chosen.

Directly the problem of development of the dialogue systems (DS) is multidimensional and incorporates practically all main problems in the field of programming. Difficulties which it is necessary to face at the solution of this task, in particular, are:

- lack of uniform approach to the description of DS (DS uniform abstract model), an illegibility of the existing descriptions;

- lack of the standard terminology;
- a small amount of the generalizing works on methods and construction tools of dialogue systems. **

Note. ** In article the systematization presented in V. O. Georgiev's work is used. Models of representation of knowledge of subject domains of dialogue systems. News of Academy of Sciences. Technical cybernetics, No. 5, 1991 of, page 3-23.

4.1. Models of dialogue systems

Among the main and obligatory stages there is creation of dialogue systems, the essential part is assigned to the choice and development of the formalized model of the system allowing ordering the description of DS, to facilitate a problem of their design and the analysis, to form a conceptual basis of the analysis of their behavior and realization.

The DS model represents the description of schemes of dialogue processes and the operations (functions) performed by system and has to meet the following conditions:

- to be simple and expressive;
- to have potential to abstract from a substantial part of dialogue (from division of roles between the person and DS);
- to be available to formal and mathematical treatment;
- to give an opportunity of transformation to a program system.

The methods applied now and construction tools of the DS models can be divided on informal (or partially formalized) and formal methods.

4.2. Informal methods of the DS models creation

Such models for the definition generally do not use any specialized formal notations. However, by means of informal methods generally it is impossible to describe precisely and unambiguously the general concept of the organization and functioning of DS. As methods and construction tools of informal models are used:

- Flowcharts (scheme of dialogue);
- Charts of states;
- Tabular forms of the description;

- Texts in programming languages of high level;
- Texts in language of some virtual computer;
- Methods of structural programming.

4.3. Formal methods of the DS models creation

When using formal methods of creation of models of dialogue systems certain specialized formal notations are applied.

It is possible to allocate two main directions in such formalizations: models for the description of the organization or functioning and model as means of the analysis, assessment and optimization of DS, and also the approach to a research of interaction of the person and the COMPUTER expressed in definition of types of solvable tasks, in allocation of abstract means of their decision and also in the analysis of different types of interaction of the person and the COMPUTER.

Creation of the DS formal model includes the following stages:

1. Precisely and unambiguously to describe the general concept of the organization and functioning of DS (to connect and present structural, dynamic and information aspects of dialogue interaction from uniform positions);
2. To use model as a basis for design and implementation of the software of DS;
3. To receive means of the analysis and assessment of DS;
4. To act as a methodical basis for creation of tools of the automated development of DS.

It should be noted that development of the DS corresponding formal models is closely connected with questions of development of the formalized models of their subject domains.

Now it is possible to allocate the DS following formal models:

1. Graph (automatic) models;
2. The models using network representation (Petri's network, Woods's network and the generalized networks of transition);
3. The models constructed on the basis of the device of formal grammars;
4. The models using the device of the theory of tasks and ways of their decision;
5. Probabilistic models;
6. The models using a concept of process;
7. The models based on various methods of formal specifications;
8. Relational and frame and relational models;

9. The models constructed on the basis of operator schemes;
10. The models using the principles of game theory;
11. The models based on use of the device of frameworks of the theory of models.
12. Natural language models

5. CHOICE AND PRACTICAL USE OF FORMAL MODELS OF THE HUMAN-MACHINE INTERFACE

The review and detailed consideration of each of models is beyond this article therefore the short description of some most priority models to consideration, their advantages and shortcomings on the basis of which it is possible to choose some of the most suitable models for creation of educational models of the assembly generator is provided. The review is based on sources [12-15].

5.1. Graph models

Such models are the most widespread way of the description of DS now. Process of functioning in these models is represented a transition graph of the finite-state determined machine. A transition graph it is possible to present the focused hierarchical count, the coherent count, a graph with loops, the focused loaded count in the form. In this count each top describes conditions of dialogue in which DS expects input of the message from the user and depending on the entered text passes on the arches displaying possible actions into another (or the same) a state. On a way of the description of dialogue interaction in such models it is possible to allocate, at least, two approaches: or all procedure of interaction of the person with the COMPUTER is described by automatic machine of this or that class, or each act of human-machine interaction is submitted actually automatically.

5.1.1. The models using network representation

Such models are more perfect formal device allowing giving equal display of actions of partners in dialogue and also conditions of performance of these actions.

The models using the office of networks of Petri are the most widespread. Any elementary fragment of dialogue interaction in these models consisting of the description of some conditions (state) of dialogue and action of system in this state is represented the elementary network consisting of some set of tops places ("condition") and the transition of network ("event") corresponding to them. Complex structures of dialogue can be presented as compositions of elementary structures.

Attraction of a mathematical apparatus of networks of Petri allows to investigate structure of DS regardless of concrete language of the user, a dialogue form, its technical means. However essential shortcomings are sharp increase of complexity of model at increase in extent of specification of modeling DS and focus of this device on stationary structure of dialogue.

More powerful and means of expression of representation of the DS models are Woods's networks and augmented transition networks. The DS model is represented as set of two automatic machines - the dialogue monitor and some network interpreter. Distinguish the dialogue models realized by means of augmented transition networks with returns, the formalism combining recursive transition networks of Woods and Petri-Chomsky's grammar from subtypes.

5.1.2. The models using the device of formal grammars

Grammars as descriptors of development of dialogue reflect its linear, cyclic and branching forms, various options of an enclosure of such processes and their recursive interaction. However use of the device of formal grammars demands a certain degree of qualification which is difficult for assuming at applied programmers.

5.1.3. Probabilistic models

Such models are used, ever the further step of dialogue is not defined, or its search is too difficult, or depends on the user. The main mathematical apparatus used in these models includes the theory of sets, the theory of counts, calculation of predicates and the theory of mass service. Research DS as systems of mass service is conducted with use of both analytical, and imitating methods. Imitating modeling is understood as experimenting with mathematical model. The model is under construction from positions of a number of mathematical models with division of time in terms of the theory of mass service and the theory of the regenerated processes.

5.1.4. The models using a concept of process

This direction is presented in the form of approach to representation of behavior of DS in the form of hierarchy of processes of a special look - final processes. Each final process and their relationship in hierarchy are described in a natural language. Other approach is based on a concept of process the model considering actions of the user and describing directive and request and reciprocal dialogue.

5.1.5. The models based on various methods of formal specifications

The approach to representation of the DS model based on methods of formal specifications allows describing concisely difficult behavior of system, without pressing in its structure and details of realization, and allows to have the firm mathematical basis giving the chance of a conclusion of behavior of DS by means of logical deduction. However a problem for these models remains the fact that the DS full specification has to define the functions realized in common by the program and the user.

Now a number of the directions investigating an opportunity to apply to formalization of the DS model of logic of schemes of programs, constructive logicians, a formalism of predicative grammars, imperative program logicians (algebras), Hoare's logicians, SIGMA programming is known.

Concepts and methods of composite programming, in particular the device imperative program the logician, in the annex to creation of the DS model can be on the practical level used. The basis of composite programming is made by the principles of separability and subordination, i.e. use for creation of difficult programs simpler, connected to the help of the generalized structural method of assessment of the person technician systems on the basis of the description of process by the absorbing Markov chain.

In one of the directions it is offered to use stochastic networks for creation of probabilistic models. It is offered to use the stochastic network models considering structure for the analysis of multiterminal DS. In particular, the model of process of the solution of a task in human-machine system presented as the exponential closed stochastic network of mass service, programs are considered as the special functions defined on personalized sets.

5.1.6. Relational and frame-relational models

These approaches in relation to definition of the DS model are based on uniform relational data presentation and a standard set of procedures of data processing including along with operations of relational algebra and operation of calculation of a number of functions and functionalities on attributes of the relations. The tool complex for the automated development of DS can be constructed on the basis of the relational model based on relational algebra and relational calculation. Frame approach is the basis for the description of system of information processing. The frame is presented in the form of the network consisting of knots and communications between them. Top levels of a frame are accurately defined as reflect the situations which are in advance revealed on the basis of the analysis of aprioristic knowledge of problem area.

5.1.7. The models using the principles of game theory

The approaches using for modeling of dialogue in interactive systems the principles of game theory are perspective. Dialogue in this case is considered as a game of two persons with not opposite interests and use of methods of game theory is applied to the solution of a real problem of the choice of the best mode of dialogue in the conditions of uncertainty of characteristics of the user.

5.1.8. The models based on use of the device of frameworks of the theory of models

These models offer the approach to development of interactive systems allowing representing DS as set of the dialogue systems focused on the solution of frequent problems or revealing certain sides of the considered problem area. At statement of a specific objective the solutions required for it components unite, forming uniform DS. The offered approach is based on the concept of consecutive building of model of subject domain by association of several DS in uniform dialogue system.

5.1.9. Natural language models

One of the perspective directions of development of models of dialogue systems are DS based on achievements of computational linguistics in the field of work with natural languages. Key feature of such model is drawing up dialogue in a natural language. In such system the user does not choose from a limited set of the offered actions, and enters inquiry or the answer to a system question in a natural language in the form of the text, or, what is even more preferable, by means of voice set. The system analyzes the received inquiry and passes into a necessary state, makes necessary action.

Naturally, success of realization of such approach depends on the linguistic direction used for creation of system. Well-developed direction is the methodology of the analysis of natural language inquiries used in search engines. Its methods include themselves the machine training, use of neural networks and other methods based on the statistical analysis of language.

The direction of the semantic analysis of language is a little less developed. The main developments in this area are connected with the dialogue training systems, for example, systems of electronic testing. Dialogue in this case is based in a look "a system question - the answer tested", and action of system is assessment of the answer and transition to the following question. In this direction for dialogue systems the pragmatically focused approach to the semantic analysis of a natural language is perspective. Such approach is based on the assumption that the question is asked by some context of the answer, limits its form and variations. It allows to avoid the full, unlimited analysis of the natural language text and to simplify a task to the level accepted for realization.

5.2. The choice of models for realization in educational models

Most strengths of some described models represent the following.

- Graph (automatic) models thanks to the graphic notation give a fair idea of organizational structure and functioning of DS.
- Network models allow displaying most adequately process of modeling of communication with the COMPUTER on a limited subset of a natural language and process of modeling of information streams of DS.
- Logical models, despite some complexity of understanding, have, according to the author, two strongest qualities - a logical conclusion and coherence with programming languages. These qualities do them irreplaceable for modeling of processes of planning of calculations and actions in DS and adequate display of model in a program system.
- Relational and frame and relational models allow to display most adequately knowledge of dialogue (knowledge necessary to system for the organization of dialogue process, and means of their representation).
- The probabilistic models and models using the principles of game theory provide to DS developers of means of their analysis, assessment and optimization. Means of these of formalism allow using the model device as the instrument of detection of fundamental properties and laws

of behavior of DS, to estimate the set parameters and resources necessary for realization of system, to optimize the parameters essential to its development and functioning.

- Natural language models represent the natural, convenient and flexible interaction interface in realization with system.

For model realization were chosen: graph, based on game theory and network (it is concrete - tensor) models. The graph model is the classical and simplest example of model suitable for creation of the educational model. The model based on tensor representation is well developed in the form of the theory, however not often meets at practical use therefore creation of the educational model on its basis can form the training basis showing options of its practical application. The model on the basis of game theory is chosen from reasons of its experimental character as attempt to find the most suitable options of its use.

6. PRACTICAL CODE REALIZATION OF MODELS OF GENERATORS ON DIFFICULT SYSTEMS WITH THE CHMI DIFFERENT MODELS.

6.1. The model on the basis of the model using the principles of game theory

The model on the basis of game theory is not the most widespread formal model for the reason that the basic concepts of the theory of dialogue games which in it are used are known only in narrow circles. However it is model it is rather simple in use and has many similarities to the most widespread automatic model. The model of model described further is based on article [16]

The theory of dialogue games is an attempt to consider the ideas of scenario models and dialogue grammars in one structure. It is supposed that dialogues consist of series of so-called games. Each game is made of the sequence of the courses which are possible according to a set of rules (similar to grammars), and all game is planned by the participating agents (as in the model based on scenarios). Thus, agents share knowledge (representations and the purposes) during dialogue, and games can be enclosed (sub dialogues are possible) for achievement of local goals. In this structure the courses are often equated to speech acts. The model quite formally determines the courses, admissible for each of participants of a game at present (by rules and according to the purpose) and thus, dialogues are modeled.

For a task of requirements to dialogue system the following definitions are entered.

The game board is defined by two of its parts: general board and personal boards:

- General board. It is accepted to represent dialogue as interaction on the general board. This board contains the ordered history of the courses, the game identifier with the connected rules which were known before or became known in the course of dialogue.
- A personal board - personal representations of each player, unknown another. It in particular assumes a game with incomplete information.

The course - basic action of the player in a game, and it has the price. A step of a game is called the interval between two courses. The player who has to go - active.

Cognitive context - a condition of personal boards and in common accepted facts on a public board.

With everyone a step of a game its cognitive context because of addition or removal of the facts on a game board changes. In this regard this concept can be carried to the act of dialogue. Formally the act of dialogue is defined as the communicative function applied to substantial statements. Also the empty course is entered. This course does not influence a cognitive context of a game, but increases the counter of the courses as any dialogue action.

In addition, the active player can make the logical course. Such course changes a board, applying the logical rule. The logical course offers nothing and does not increase the game counter.

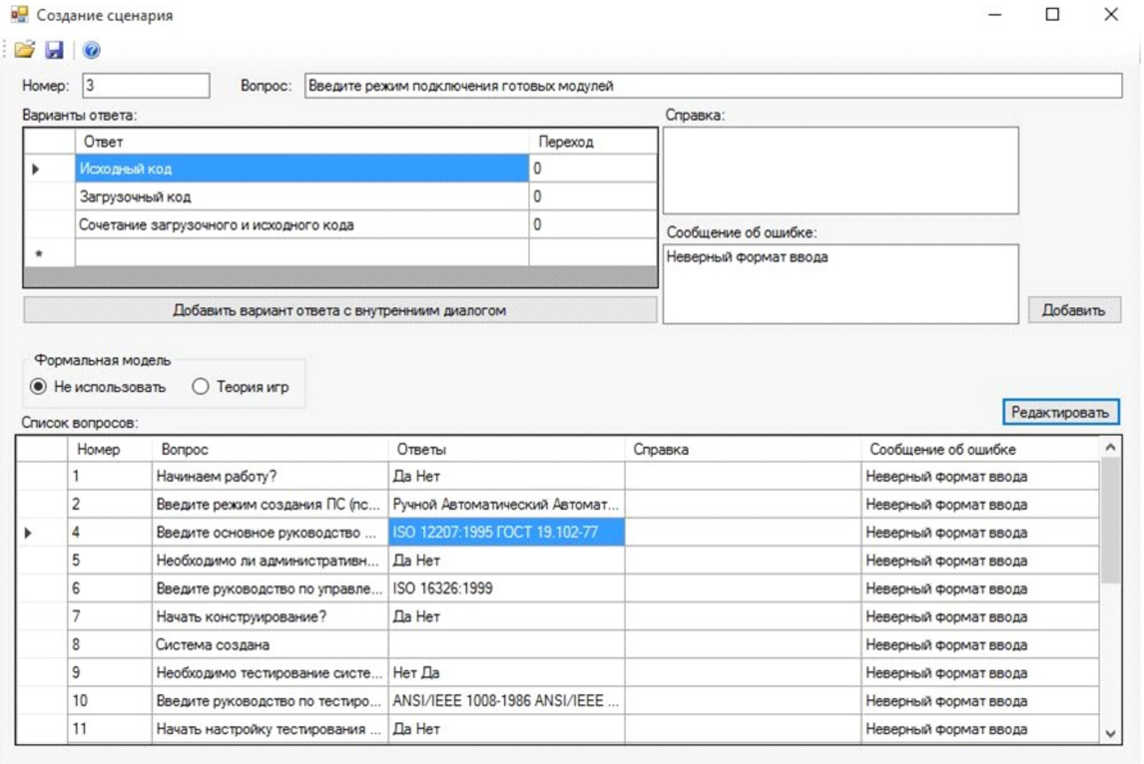
Purposes. Even if intentions are directly not realized in a game, players have the purposes when play a game. The purposes are written down in this model by structure of a game.

Rules. The concept "rule" is strongly connected with essence of a game. There is no approach which could give a pure idea of various rules in dialogue games. Initial rules of the game can just be defined as the function defining the available courses. Such rules are called effective - having effect.

The structure of a complex is made by two parts: generator of scenarios and dialogue system. In practical realization the model looks as follows:

- There is a dialogue history in which all phrases of system and answers of the user are stored.
- There is a function accepting history of dialogue as an argument and returning a set of the admissible courses. This function defines so-called rules of games.
- Each speech act is brought in history. If any step does not satisfy to rules of the game, then the condition of a game is rolled away on a step backwards.

Creation of the scenario constructed on the basis of model of game theory begins addition of questions and other information and comes to the end with creation of the configuration document of the scenario. The expert fills in the text of a question, the rule in the table, transition to other scenario of dialogue, the reference and the error message. If to leave the field "Transition" empty, then at the answer corresponding to it the system automatically will turn to the following question (the player in that case makes "the empty course").



Номер: 3

Вопрос: Введите режим подключения готовых модулей

Варианты ответа:

	Ответ	Переход
▸	Исходный код	0
	Загрузочный код	0
	Сочетание загрузочного и исходного кода	0
*		

Справка:

Сообщение об ошибке: Неверный формат ввода

Формальная модель

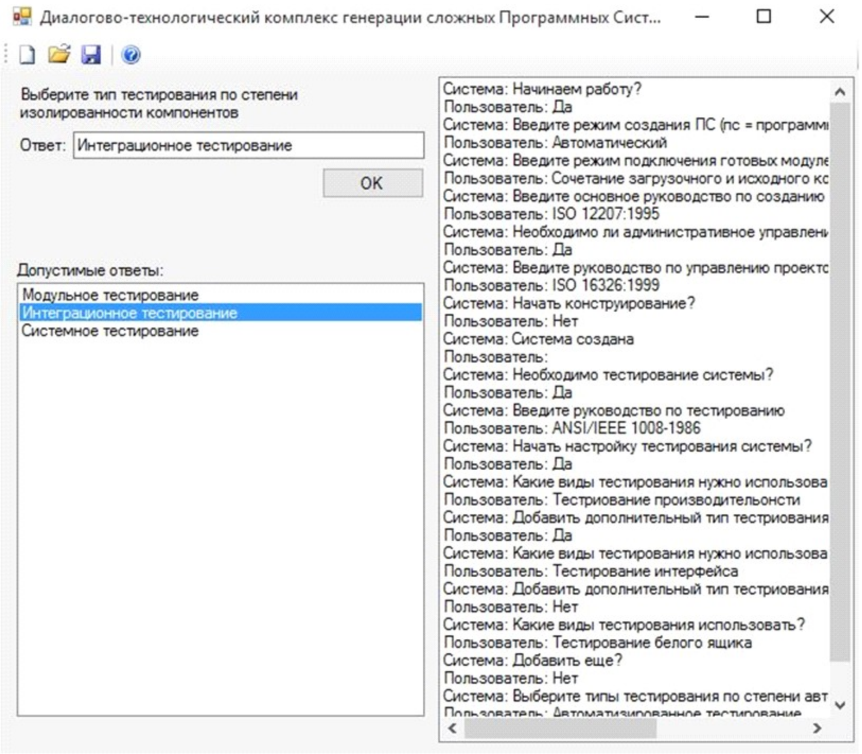
● Не использовать ○ Теория игр

Список вопросов:

	Номер	Вопрос	Ответы	Справка	Сообщение об ошибке
	1	Начинаем работу?	Да Нет		Неверный формат ввода
	2	Введите режим создания ПС (пс...	Ручной Автоматический Автомат...		Неверный формат ввода
▸	4	Введите основное руководство ...	ISO 12207:1995 ГОСТ 19.102-77		Неверный формат ввода
	5	Необходимо ли административн...	Да Нет		Неверный формат ввода
	6	Введите руководство по управле...	ISO 16326:1999		Неверный формат ввода
	7	Начать конструирование?	Да Нет		Неверный формат ввода
	8	Система создана			Неверный формат ввода
	9	Необходимо тестирование систе...	Нет Да		Неверный формат ввода
	10	Введите руководство по тестиро...	ANSI/IEEE 1008-1986 ANSI/IEEE ...		Неверный формат ввода
	11	Начать настройку тестирования ...	Да Нет		Неверный формат ввода

Fig.2. The model on the basis of game theory: interface of creation of the scenario.

After performance of these actions the most dialogue system can open the created file. The user is offered to walk according to the scenario of dialogue and to answer questions. Proceeding from the answer of the user, the system passes into a necessary state and writes down results.


Диалогово-технологический комплекс генерации сложных Программных Сист...
OK

Выберите тип тестирования по степени изолированности компонентов

Ответ: Интеграционное тестирование

Допустимые ответы:

- Модульное тестирование
- Интеграционное тестирование
- Системное тестирование

```
Система: Начинаем работу?
Пользователь: Да
Система: Введите режим создания ПС (пс = программ
Пользователь: Автоматический
Система: Введите режим подключения готовых модуле
Пользователь: Сочетание загрузочного и исходного кс
Система: Введите основное руководство по созданию
Пользователь: ISO 12207:1995
Система: Необходимо ли административное управлени
Пользователь: Да
Система: Введите руководство по управлению проектс
Пользователь: ISO 16326:1999
Система: Начать конструирование?
Пользователь: Нет
Система: Система создана
Пользователь:
Система: Необходимо тестирование системы?
Пользователь: Да
Система: Введите руководство по тестированию
Пользователь: ANSI/IEEE 1008-1986
Система: Начать настройку тестирования системы?
Пользователь: Да
Система: Какие виды тестирования нужно использова
Пользователь: Тестриование производительонсти
Система: Добавить дополнительный тип тестриования
Пользователь: Да
Система: Какие виды тестирования нужно использова
Пользователь: Тестирование интерфейса
Система: Добавить дополнительный тип тестриования
Пользователь: Нет
Система: Какие виды тестирования использовать?
Пользователь: Тестирование белого ящика
Система: Добавить еще?
Пользователь: Нет
Система: Выберите типы тестирования по степени авт
Пользователь: Автоматизированное тестирование
```

Fig.3. The model on the basis of game theory: interface of dialogue with the user.

6.2. The model on the basis of graph model

The graph (automatic) model is one of available and clear models for realization of program systems. The count - the main object of studying of the mathematical theory of counts, set of a nonempty set of tops and sets of pairs of tops (communications between tops).

The structure of the model is made by two parts: realization of model with the set properties with the help, so-called, the generator of the scenario and application of the constructed scenario in dialogue system. As prototype practical realization of graph (automatic) model the model will look as follows:

- To each top of the count there corresponds the question of dialogue;
- To each transition in the column there corresponds one of possible possible answers on a question that in fact gives us an opportunity to move to any available states;
- Similar to the count's top, some state is appropriated to each question.

Creation of the scenario constructed on the basis of graph (automatic) model begins with creation of the empty document in the .xml format. For data recording of a state (top) of the count the field "Question" to which the text of a question registers is used. The count is realized by the table with the fields "Possible Answers", "Transition to a State". After filling of a form it is necessary to press the Add a Question button. By such consecutive actions with the indication of number of transition to a certain state for each possible answer on a question it is possible to create the file in which the scenario constructed on the basis of model is stored.

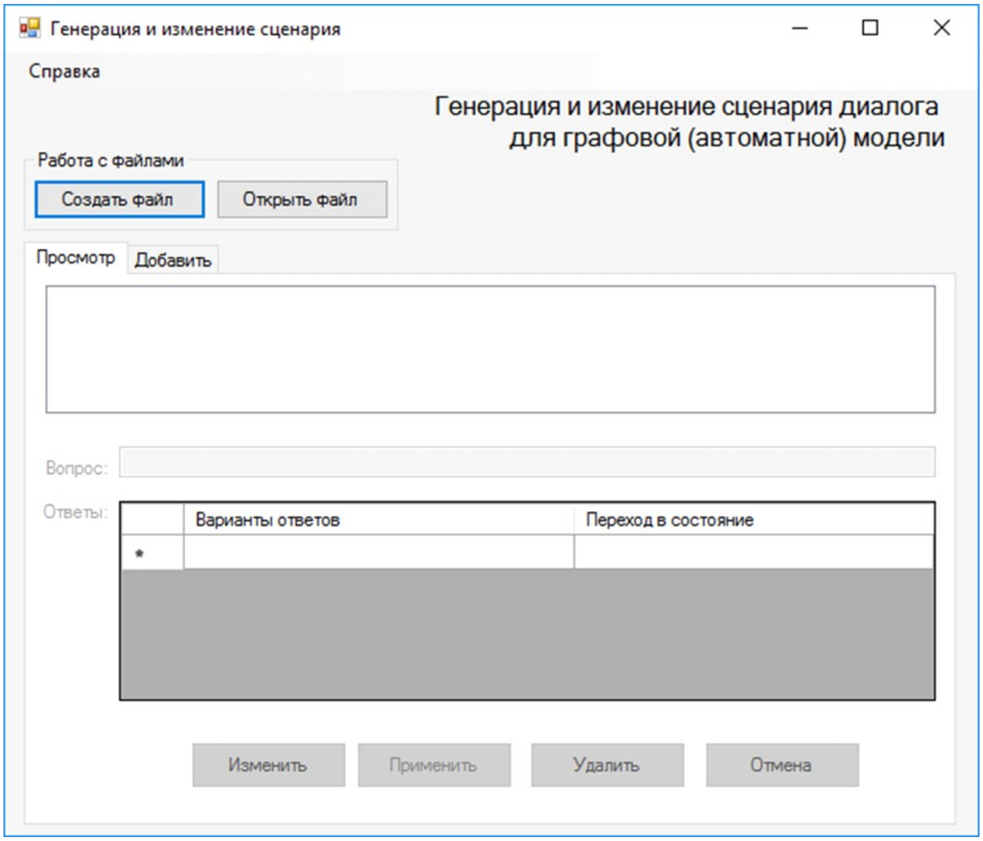


Fig.4. Model of graph model: interface of creation of the scenario.

After performance of these actions it is possible to open the completed file of the scenario in a main window of the program. After such manipulations the user is offered to walk according to the scenario of dialogue and to answer questions. Proceeding from the answer of the user, the system passes into a state necessary for us and writes down result in "The dialogue course". During dialogue there is an analyzer of mistakes which displays in the field "Error message" information to the user that it entered the incorrect answer and system again asks it to answer a question, having chosen one of the offered possible answers.

6.3. The model on the basis of the model using network representation in the form of tensor network

The model using tensor network representation was described in the book [17] in due time, and this description is taken as a basis for creation of the model of model. The task of creation of an information system is set as a task of network. There is a network consisting of set of one-dimensional simplex. Influence network some sizes. It is necessary to determine sizes of the responses created by network. The task for networks of various systems of coordinates, transition from one coordinate system to another and displaying of an invariant object in various systems of coordinates gives the chance of management of network.

The possibility of continuous transformations which allow to pass from one systems to others is the basis for the general theory of information systems. In passive interpretation it is possible to consider these transformations as transformation from one generalized coordinates to others, in active - as transformation from one system to another. Therefore studying and design of a difficult information system will come down to studying of simple systems.

Formal definition of network can be given so: on a calculating great number *of M* the binary relation *of R* as a subset *of M ×M is set:*

$$R□\ M \times M, \qquad x□\ M,\ y□\ M,\ (x, y)□\ R.$$

where ***(x, s)*** - the ordered couple. Let's define transformation

$$™(R) = \Sigma \qquad ()\Sigma \qquad ()$$

It establishes univocity between elements r = ($_{xr,}$ $_{yr}$) systems and a nonempty subset of one-dimensional simplex from the ends in points $_{of\ xr}$ and $_{yr}$. Network is called the system (a set of one-dimensional simplex), to tops and to which edges some sizes are delivered in compliance, i.e. functions are defined:

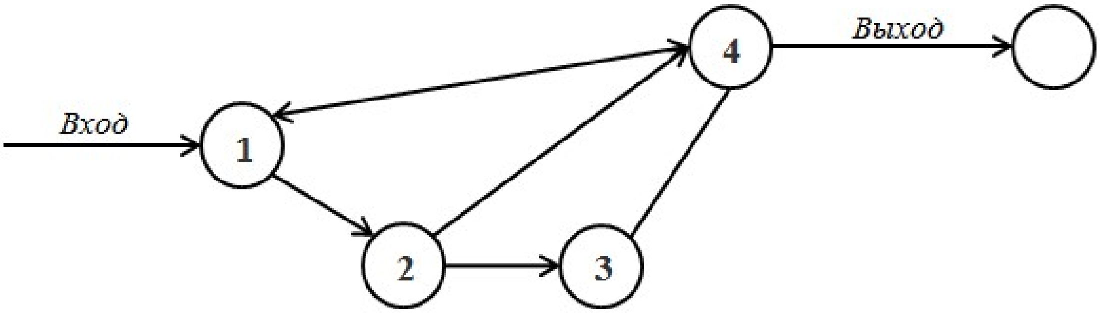
$$_{f1}\text{: } M□\ x, \qquad _{f2}\text{: } S□\ y.$$

The displays keeping mathematical structures on sets allow comparing among themselves various spaces, to find among them isomorphic or coordinated in weaker sense. The set of all displays represents the new set other than initial sets, it is called functional space. Use of functional space allows building sets (system) with the necessary properties or structures.

Let's show how the generalized IS topological model which then, being projected in private systems of coordinates, will give the IS concrete models and ways of transition from one system to another is under construction. The structure of system is set by ways of connection of branches (one-dimensional simplex). Generally as branches the planes, n-dimensional volumes are used. Each branch is considered as independent measurement of n-dimensional space.

The model of an information system is presented generally in the form of network, to tops there correspond domains (a set of data of one type), and to branches - transformations between data. The network represents structure in which there are ways both closed, and opened. This structure formed of branches and knots is considered as a one-dimensional complex.

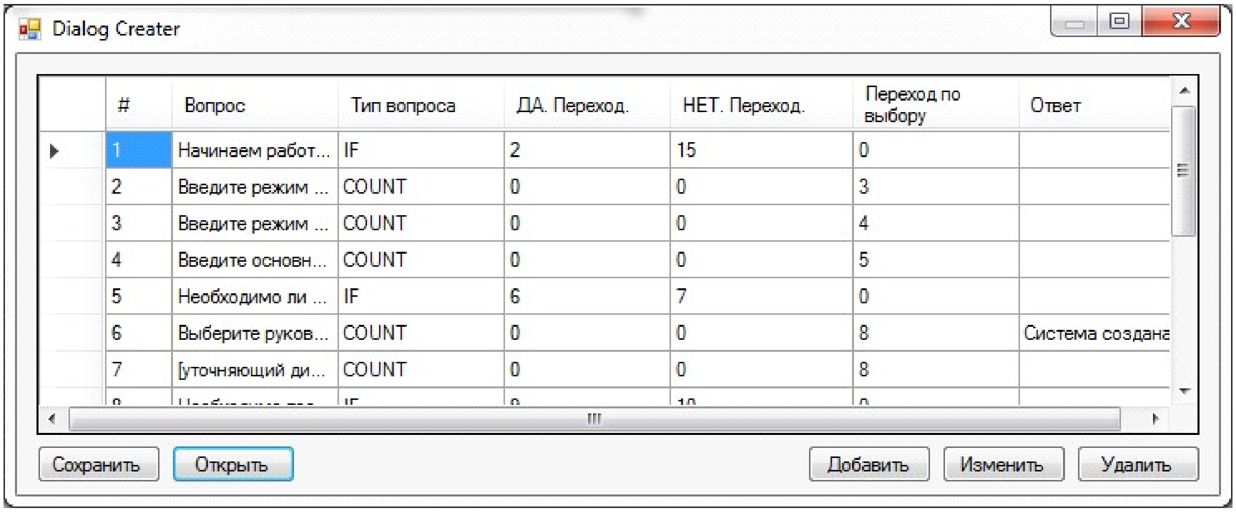
In case of dialogue system questions, and arches transformations of system that is transitions from one system of coordinates to another will be hubs of network. The network is set by a tensor of a databank which to keep in itself all entities and their communications. The task of a question can be compared to function evaluation depending on the answer transition to the following knot of network is carried out and if for this answer there is no transition function, then it is defined proceeding from topology of network. For each question it is necessary to define a set of couples <the answer, transition> if couples are not transition will be defined from topology of network.



Input Output

Fig.5. Scheme of network of dialogue system

During work the complex consisting from was developed: editors of scenarios and dialogue system.

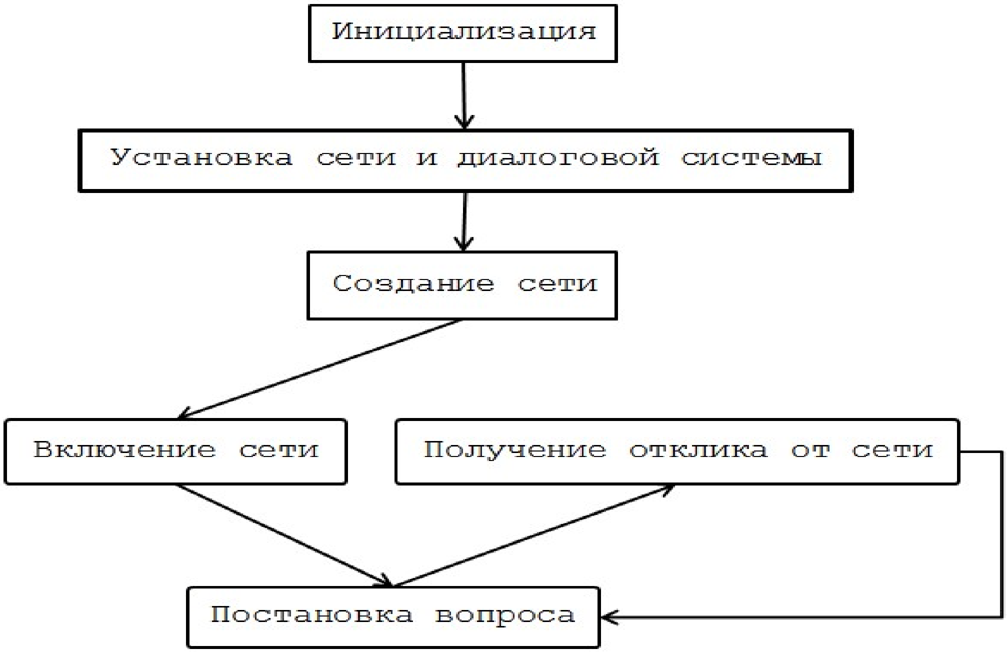


#	Вопрос	Тип вопроса	ДА. Переход.	НЕТ. Переход.	Переход по выбору	Ответ
1	Начинаем работ...	IF	2	15	0	
2	Введите режим ...	COUNT	0	0	3	
3	Введите режим ...	COUNT	0	0	4	
4	Введите основн...	COUNT	0	0	5	
5	Необходимо ли ...	IF	6	7	0	
6	Выберите руков...	COUNT	0	0	8	Система создана
7	[уточняющий ди...	COUNT	0	0	8	
8	[illegible]	IF	9	10	0	

Fig.6. The model on the basis of tensor network: interface of the editor of scenarios

The expert enters a system question, adjusts answer type from the user, and chooses knots in which there is a transition at the choice of this or that option and the response message of system. All entered data are checked for correctness.

The dialogue system consists of one form on which questions change, there can also be emerging messages if in them there is a need.


Инициализация
Установка сети и диалоговой системы
Создание сети
Включение сети
Получение отклика от сети
Постановка вопроса

Initialization

Installing the net and dialogue system

Net creation

Connecting the net

Receiving the net response

Question setting

Fig.7. Scheme of the program work

7. CONCLUSION

Conceptual basics of creation of the PO generator of difficult systems as difficult system with the model and model approach to development allowing receiving the basic model of system suitable for further iterative expansion in short terms for testing and for the educational purposes were covered.

Concepts of creation of a subsystem for a solution of the problem of compatibility of the program modules entering the generated system are defined. The micro-core architecture allowing to make system the most flexible and easy additional functionality for expansion became such architecture. Plug-ins for expansion of functionality are divided into several groups, the obvious agreement on a set of functions of system thereby is entered that facilitates its testing.

The review of various models of human-machine interaction is provided in a format of dialogue system, comparison of their advantages and shortcomings, three models are presented for educational and model realization: network, graph and the model based on game theory.

8. CONCLUSIONS

The received models of models of human-machine interaction will be used further in educational process at the courses "Design and Architecture of Program Systems", "Object-oriented Analysis and Design" and "Designing of the Software".

Further work consists in development of models for other models that it is fuller to investigate their merits and demerits, and on their basis to choose necessary model when developing the assembly generator of program systems. The poorly studied direction of natural language model of dialogue interaction which research is necessary an important task is especially allocated.

ACKNOWLEDGEMENT

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

8. LIST OF REFERENCES

1. Georgiev V. O. A methodical grant according to the "Object-oriented Analysis and Programming" program. Code representation of the basic concepts of OOA and P in different language environments. Part 1. Kazan: Cauldron. un-t, 2016. 230 pages.

2. Georgiev V. O. Educational and model option of interactive system of generation ON difficult systems, with preliminary pre-interpretation of program modules//Materials XXII of the International scientific and technical conference "Information Systems and Technology (IST-2016)". 2016 of page 248-249.

3. Georgiev V. O. Program engineering: the technological principles of development of the software at the level of the formalized description, on the example of the abstracted dialogue system//Materials of works of the international conference TEL'2014 "Language semantics: models and technologies". 2014 of page 40-41.

4. Georgiev V. O., Prokopiev N. A. Model Approach To Interactive System Software Development//International Journal of Applied Engineering Research (IJAER). 2015. v. 10. No. 24. pp. 45208-45213.

5. Polikashin D. S., Enikeev A. I., Georgi V. O. Researching problems of automation of the solution of a problem of compatibility of program systems//Materials XXII of the International scientific and technical conference "Information Systems and Technology (IST-2016)". 2016, page 251.

6. Richards M. Software architecture patterns. Sebastopol: O'Reilly Media, 2015. 47 p.

7. Ponomarenko A., Rubanov V. Automated Verification of Shared Libraries for Backward Binary Compatibility//VALID '10: Proceedings of the 2010 Second International Conference on Advances in System Testing and Validation Lifecycle. 2010. pp. 57-62.

8. Ponomarenko A., Rubanov V. A combined technique for automatic detection of backward binary compatibility problems//PSI' 11: Proceedings of the 8th international conference on Perspectives of System Informatics. 2011. pp. 313-321.

9. Würsch M. Improving Abstract Syntax Tree based Source Code Change Detection. Zurich: University of Zurich, 2006. 78 p.

10. Saukh A. M. The analysis of some semantic aspects of source texts of programs on the basis of formal specifications of syntax and semantics//Applied discrete mathematics. Appendix. 2012. No. 5. c. 110-111.

11. Fluri B., Wuersch M., Pinzger M., Gall H. Change Distilling: Tree Differencing for Fine Grained Source Code Change Extraction//IEEE Transactions on Software Engineering. 2007. v. 33. No. 11. pp. 725-743.

12. Nigay L., Bouchet J., Juras D., Mansoux B., Ortega M., Serrano M., Lawson. J. - Y. L. Software Engineering for Multimodal Interactive Systems//Multimodal User Interfaces From Signals to Interaction. 2008. pp. 201-218.

13. Chatty S. Programs = Data + Algorithms + Architecture: Consequences for Interactive Software Engineering//Engineering Interactive Systems. 2008. pp. 356-373.

14. Camila Cordero Mansilla, ángel de Miguel Artal, Eladio Domínguez Murillo, Ma Antonia Zapata Abad. Modelling Interactive Systems: an architecture guided by communication objects//HCI related papers of Interacción 2004. 2008. pp. 345-357.

15. Sauer S., Breiner K., Hussmann H., Meixner G., Pleuss A., Van den Bergh J. Combining Design and Engineering of Interactive Systems through Models and Tools (ComDeisMoto)//Human-Computer Interaction - INTERACT 2011. 2011. pp. 724-725.

16. Maudet. N., Evrard. F. A Generic Framework For Dialogue Game Impementation//Proceedings of 2nd Workshop on Formal Semantics & Pragmatics of Dialogue. 1998. pp. 185-198.

17. A. E. Armensky. Tensor methods of creation of information systems. Moscow: "Science", 1989. 152 p.

How to cite this article:
Georgiev V O, Prokopyev N A, Polikashin D S. Conceptual basis for developing of trainig models in complex system software assembling generator. J. Fundam. Appl. Sci., 2017, *9(2S), 1018-1043.*