Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DEVELOPMENT OF THE METHODS AND TECHNOLOGIES FOR THE INFORMATION SYSTEM DESIGNING AND IMPLEMENTATION

[1]BORANBAYEV S.N., [2]NURBEKOV A.B.
L.N. Gumilyov Eurasian National University, 010008 Kazakhstan, Astana city, Satpayev st., 2,
phone: +7 (7172) 709500, Fax: + 7 (7172) 709457
E-mail: [1]sboranba@yandex.kz, [2]nurbekoff@gmail.com

ABSTRACT

The article is devoted to the development of methods and technologies for the designing and development of the information systems. The implementation of the developed methods and technologies forinformationsystem designing and creation will enable to regulate the modeling process and collect in the course of its implementation formalized information to plan the stages of the designing, to provide functionalcompleteness and logical integrity of their results.

Keywords: *Model Method, Designing, Information System, Technology, Network*

1. INTRODUCTION

In contemporary world there is an urgent need for science-based methods and technologies to develop information systems that would enable to plan the parameters of the system and guaranteedly receive a high-quality and reliable product. It is essential to reduce the dependence of the quality of the development results of the information systems on such subjective factors as the qualifications of administrators and their experience. In this regard, the establishment of science based methods and technologies for the designing information systems are an important scientific and technical challenge.

2. SYMBOLIC-FORM MODEL

In the works [1-3] a model for the information system designing was proposed. Here are some of the results of these studies for the purpose of their application for the further study. Lemma and theorems are given without proof.

As a model for designing an information system, it is recommended to use an associated oriented network reflecting the progress of the project and the analysis of the logical structure of the project. The network has a single input and a single output node. Each vertex is the work in the project.

Simple network is a home network portion having a single input and a single output top vertex and consisting of a linear sequence of project work. All works included in a simple network were implemented sequentially from the first to the last. We use the notation from [1-8].

Source network is considered as a set of finite number of simple networks. Sequence performing of simple network displays the progress of the project.

Network project progress is an oriented connected network S = (M, R, m_0) with a single initial vertex m_0 without incoming arcs in which a finite set of vertices M={m_q}, $q = \overline{0, N}$ is a simple network, and a finite set edges R = {rj}, logical connections between simple networks. Vertices m_0 and m_t can be artificially introduced.

By the a sequence it is understood the vertices of the network (m_0, m_1, m_2,…, m_q,…, m_t), that for any value of q, $0 \le q \le t-1$, couple (m_q, m_{q+1}) is an edge. If mt= m0, then the path is called a loop.

The vertex m_q is called precursor of the vertex m_γ, if there is a path from the vertex m_q to the vertex mγ. If (m_q, m_γ) is an edge, the vertex m_q is called an immediate predecessor of the vertex m_γ.

If the vertex m_q is a predecessor to the vertex m_γ, then the vertex m_γ is called a follower of the vertex mq. If (m_q, m_γ) is an edge, then the top m_γ is a direct follower of the vertex mq.

Simple Network SP_q predecessor is called simple network SP_γ, if there is a way out of SP_q simple

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

network in an SP_γ simple network. If there are no other simple networks among them, a simple network SP_γ is called an immediate predecessor of a simple network SP_q.

If simple network SP_q is a predecessor of a simple network SP_γ, then SP_γ is called a predecessor of a simple network SP_q. In case there are no other simple networks among them, a simple network SP_γ is called an immediate predecessor of simple network SP_q.

The vertex m_q is called ancestor of m_γ vertex, if every way from the initial vertex m_0 to vertex m_γ includes vertex m_q. If the vertex m_q is the predecessor for vertex m_γ and no other predecessors for the vertex m_γ in the ways from the vertex m_q to the vertex m_γ, then the vertex m_q is called the direct ancestor m_γ.

Lemma 1. Initial vertex m_0 of the network S=(M, R, m_0) is a predecessor over each network vertex $m_q \in (M - \{m_0\})$.

Lemma 2. If the vertex m_q is the predecessor for vertex m_γ, then the vertex m_γ in this turn over the vertex m_k, then vertex m_q is the predecessor for vertex m_k.

Lemma 3. If the vertex m_q is the predecessor for vertex m_γ, then the vertex m_γ cannot be the predecessor for vertex m_q.

Lemma 4. If the vertex m_q и m_γ are the predecessors for vertex m_k, then either vertex or vertex m_q are the predecessors for the vertex m_γ, or the vertex m_γ is the predecessor for vertex m_q.

Lemma 5. For each vertex $m_k \in M, k \neq 0$, there is only one immediate predecessor.

A number of predecessors and successors (both immediate and simple), as well as many of the predecessors and immediate predecessors can be linked to each vertex $m_q \in M$ of network S=(M, R, m_0).

We shall denote MNP_q^+ – set of immediate predecessors for the vertex m_q; MNP_q – many immediate predecessors for the vertex m_q; MNS_q^+ – many immediate predecessors for the vertex m_q; MNS_q – many followers for the vertex m_q; MP_q – many predecessors for the vertex m_q; MP_q^+ – an immediate predecessor for the vertex m_q.

It is evident that $MNP_q^+ \subset MNP_q$, $MNS_q^+ \subset MNS_q$, $MP_q^+ \subset MP_q$.

$$MNP_q = \{m_\gamma \in M \mid m_q \in MNS_\gamma\}$$

$$MNS_q = \{m_\gamma \in M \mid m_q \in MNP_\gamma\}$$

$$MP_q = \{m_\gamma \in M^* \mid M^* \subset M, M^* = \bigcap_i Q_q^i\}$$

where Q_q^i – i- way from the initial vertex m0 to the vertex m_q.

Suppose that the network is given S=(M, R, m_0) implementation of the project and a vertex g.

By subnetwork T(g) it is understood a part of the network with the input vertex g meet the following criteria:

1) Vertex is the only input subnetwork T(g), that is, any path from the input vertex m0 of S network to the vertex g does not contain a vertices from the variety T(g)-{g}; the vertex g will name the subnetwork root;

2) Each vertex of the subsets T(g)-{g} is the predecessor of the vertex g, that is $(T(g) - \{g\}) \subset MNS_g$;

3) If there are closed paths, in the subnet T(g) then all of them contain the vertex g, and a subset of vertices T(g)-{g} contains no cycles.

Theorema 1. Root g of subnetwork T(g) is a predecessor for the vertices of subsets T(g)-{g}.

Theorema 2. Each vertex of the network S=(M, R, m_0) is included in the only one subnet of the network.

Theorema 3. Assume that T(g) – subnet of the network S=(M, R, m_0). Some vertix $m_q \in M$, $m_q \neq g$ can be in subnet T(g), only if all of its immediate predecessors are on the same subnet.

Result. If all the immediate predecessors of a vertex $m_q \in M$ of network S are not the part of the same subnet, then the vertex m_q is the root for the other subnet of the same network.

Technologically allowable sequence of vertices of subnet T(g) we can call ordered sequence of vertices (m1,…, mj,…, mq), m_1=g, meet the following conditions: if the vertices of the subnet to process in this sequence, for each vertex m_j, $1 < j \leq q$, all of its predecessors in the subnetwork achieved along the paths from the root subnet

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

containing no cycles are to be processed up to the vertex m_j.

Network S=(M, R, m_0) can be "splitted" on a finite set of subnets I={T(g_1), T(g_2),…, T(g_j),… }. If each subnet is represented as a single vertex, then the subnets can be set the same ratio as for the vertices of the network S.

Assume that I is a a finite set of network subnets S=(M, R, m_0). Network we call an integrated network of the second rank for the network S, if the following conditions are satisfied.

1) Multitude M^2 of vertices of the network S^2 is a set of network subnets S, that is M^2=I, wherein each vertex of the network S^2 represents only one subnet work S, but each network subnet S in the network S2 is represented only by one vertex. For convenience, we assume that the subnet T(g_j) in the network S^2 is represented by the vertex m_j^2.

2) A plurality of edges R^2 represents the logical connection between subnets network S. That is $\left(m_j^2, m_i^2\right) = r_\mu^2 \in R^2$, if there exists a vertex $m_q \in T\left(g_j\right)$, that $\left(m_q, m_\gamma\right) = r_f \in R$, where $m_\gamma = g_i$.

When in the subnet T(g_j) there are several vertices that meet the criteria, the edge $\left(m_j^2, m_i^2\right) = r_\mu^2 \in R^2$ represents all the edges emanating from the vertices the subnet T(g_j) and included in the root of the subnet T(g_i). For instance, fot the network illustrated in figure 3, in the subnet T(3) two vertices exist 5 and 6, of which the edge of the subnet goes in the root T(7).

3) Vertex m_0^2 represents the one subnet network S, for which the input vertex is the root m_0 of its network.

Hereafter a network S=(M, R, m_0) we will call the network of first rank $S^1 = \left(M^1, R^1, m_0^1\right)$, and its subnets - subnets of the first rank I^1={$T^1(g_1)$, $T^1(g_2)$,…, $T^1(g_j)$,…}.

Similarly, for the network S=(M, R, m_0) can be defined as integral network of the second, third, fourth, etc. ranks.

In general case, the network $S^n = \left(M^n, R^n, m_0^n\right)$ we will call as integral network of n–rank, $n \geq 2$, for the network S=(M, R, m_0), if to meet the following conditions.

1) Multitude M^n of vertices represents a set the subnets I^{n-1} of integral network (n-1)–rank S^{n-1}, that is $M^n = I^{n-1}$.

2) Multitude of edges Rn represents a set of logical relationships between subnets of (n-1)–rank, $\left(m_j^n, m_i^n\right) = r_\mu^n \in R^n$, if there exists a vertex $m_q^{n-1} \in T^{n-1}\left(g_j\right)$, that $\left(m_q^{n-1}, m_\gamma^{n-1}\right) = r_f^{n-1} \in R^{n-1}$, where m_γ^{n-1} – root subnets $T^{n-1}(g_i)$.

3) Vertix m_0^n represents the exact network subnet $S^{n-1} = \left(M^{n-1}, R^{n-1}, m_0^{n-1}\right)$, for which the root is the vertex m_0^{n-1}.

From the network S=(M, R, m_0) integral networks can be obtained until an integrated network of some of m–rank.

It should be noted that not all the networks S=(M, R, m_0) can be reduced to a single vertex in the process of building integrated networks.

An example is cited in Figure 1 and Figure 2 in which for the network S=(M, R, m_0) only integral network of the second rank S^2 exists as well as it includes several vertices, not the only one. Each subnet of integral network S^2 includes the only vertex. Therefore when trying to build an integral network S^3 of the third rank, we will obtain the same network S^2.

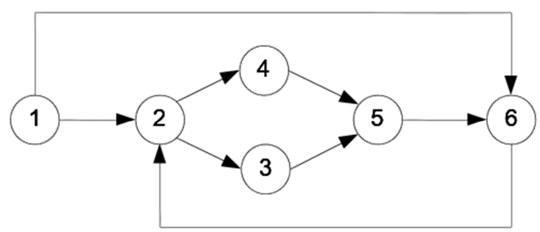

Figure 1: Network S

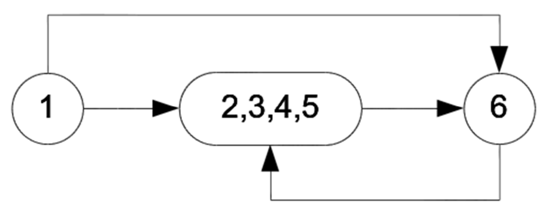

Figure 2: Integrated Network S^2

3. SYMBOLIC MODEL PROPERTIES

We investigate the properties of a mathematical model of the information system designing. We introduce the following definition.

Definition: Network S=(M, R, m_0) is called integrated, if the sequence of integrated networks exists for it S^1, S^2,…, S^n, n>1, S^1= S, such that the

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

integrated network of the highest rank, contains a single vertex.

Theorema 4. Assume that the network S=(M, R, m_0) is an integrated, then an integrated network of (n-1)–rank is nonperiodic, (that is oriented by the network without loops).

Provement: By definition, the initial vertex m_0 of the network S does not have any predecessor. This means that if the integrated network S^{n-1} comprises at least one closed path, there is subnetwork of (n-1)–rank. This contradicts the condition on which the network Sn contains a single vertex. Therefore, the integrated network (n-1)–order cannot contain loops (closed paths).

Definition: The network, which is not integrable, we shall call not integrable.

Theorema 5. Assume that the network S=(M, R, m_0) is not integrable. Then the integrated network of the highest rank S_n comprises at least three vertices for it.

Provement: Assume that the S_n integrated network for non-integrable network S contains two vertices. Then the vertix in any case will be a part of the same subnet, it is possible to form an integrated network of S^{n+1}, consisting of a single vertex. Thus, if the network S^n contains only two vertices, i.e. S is necessarily integrable.

Theorema 6. Suppose that a network *S=(M, R, m_0)* is given. If there is such an integer $n \geq 1$ that the of integral network comprising more than two vertices, each subnet of a single vertex, then the network S is not integrable.

Provement: Since each subnet of *n*–rank in this case it represents one single vertix, thus when forming integrated network of the next higher *(n+1)*–rank where the vertices can be the subnets of *n*–rank, there will be an S^n network, that is the network *S* is not integrable.

Theorema 7. Suppose that network *S=(M, R, m_0)* includes three vertices, that is *M={m_0, m_1, m_2}*. it appears that in order the network *S* to be not integrated it is necessary and sufficient, aimed to meet the following conditions:

1) $m_1 \in MNS_0^+$;

2) $m_2 \in MNS_0^+$;

3) $m_1 \in MNS_2^+$;

4) $m_2 \in MNS_1^+$.

Proof of necessity. If at least one of the conditions is not met 1)-4), that is if $m_i \in MNS_j^+$, $i \neq j$, *i=1,2, j=0,1,2*, at the vertex m_i There is only one immediate predecessor, which allows the network to divide the network *S* at least in two subnets (see Theorema 3). This means that the network S an integrated network can be obtained consisting of at least two vertices. Then, based on Theorema 5 it should be considered that the network of S is integrable.

Proof of sufficiency. When conditions 1) -4) are satisfied, each vertex of network S is the subnet. Consequently, by Theorem 6, the network is not integrable.

Network consisting of at least four vertices, will be called a trivial subnet, if for some of its three vertices m_α, m_β, m_γ the following conditions are satisfied:

1) $m_\gamma \in MNS_\alpha$;

2) $m_\beta \in MNS_\alpha$;

3) $m_\beta \in MNS_\gamma$;

4) $m_\gamma \in MNS_\beta$.

Theorema 8. Network *S=(M, R, m_0)* It is not integrable then only if it contains a trivial subnet.

Provement: if network S contains a trivial network subnet, then the formation of a integrated t subnets at best some the integrated network of S^n, *n>1*is obtained, the vertix of which satisfy the conditions of Theorema 7. Therefore, a network of S will be not integrable.

Now consider the case when the network S does not contain trivial subnet. Then there is no integer $n \geq 1$, that in S^n integrated network any three vertices satisfy the conditions of Theorema 7. In other words, each forming integrated network of the next (n + 1) - rank is required to reduce the number of vertices compared to integrated network of n- grade (see the proofment of Theorema 7). Therefore, as a result of the formation of a network of integrated networks for network S is obtained the integrated network consisting of a single vertex.

Comment. In the cases when network *S=(M, R, m_0)* is not integrable for equivalent integrable network vertices can be splitted. For example, for non-integrable network shown in Figure 3, the equivalent integrable network obtained splitting vertices m_2, is shown in Figure 4.

Journal of Theoretical and Applied Information Technology
20th December 2015. Vol.82. No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

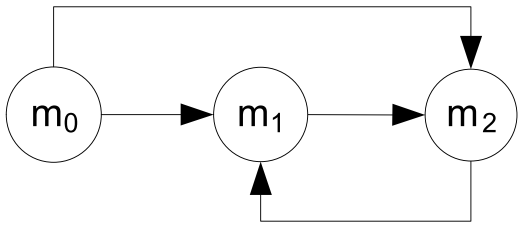

Figure 3: An Example Of Non-Integrable Network

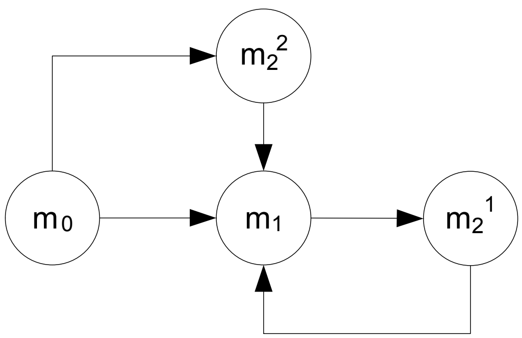

Figure 4: Equivalent Integrable Network

4. METHODS AND TECHNOLOGIES OF INFORMATION SYSTEMS BASED ON A NETWORK MODEL

In paper [3] architecture was proposed and the prototype software and information for the design of complex information systems is described. In our work, based on modification of the model and the methods proposed in [1-3], the technology and full-featured software package for the creation of information systems has been developed. The technology is based on the representation of the IS work designing stages in the form of network model. Network models of work stages can be integrated and can get some subnet of higher k-rank. Thus, the network of work stages can be divided into ("divide") on the subnet to the required level of detail. Representation of work stages in the form of network model provides a rigorous and clear description of the stages and types of work enables the integration of multiple networks into a larger network, to further their application in planning and allocation of work between the performers, allowing more effective management of project implementation.

The developed technology is implemented as a software package. The software package is designed for the management and maintenance work stages, starting with the initialization of the project and ending up with the receipt of the prototype information systems, as well as to monitor the progress of work on the analysis and design phases of the life cycle of information systems.

Work with the software package begins with the start of the main window - "Start screen". At the top of the screen is the main menu, where the user can apply to all windows software package. At the bottom of the window is a status bar that displays the current number of working directories stored in the software package.

The main menu of software: File; Projects; Data domain; Work Stages; Setting; Window; Reference.

"File" menu contains the item "Exit". Menu "Projects" consists of items: Register of projects; Reports on the projects. Menu "Data domain" consists of items: staff units; Organizational structure; Glossary facilities. Menu "Work Stages" contains no sub-items. "Settings" menu consists of items: Reference Service; Methods and modeling tools; Life cycle standards and document templates; Shared repository; User management. Menu "Windows" contains a list of open windows. "Help" menu consists of items: Description of the stages of technology; Assistance under the program; About the program.

In the "File" menu when selecting menu item "Exit", program complex will open a confirmation dialog box out of the program, with the following query "Do you really want to exit the program?" ("Yes/No"). Closing software modules is performed with a positive response.

Menu "Projects" contains a register of projects. Project register consists of a tabular representation of all existing projects in the software package. Each line of the registry contains basic information about the project: Short name; Customer; Developer; Project start date; project completion date.

The register of projects includes the following steps on the projects: Change the information on the project; New Project; Remove project.

When changing or creating a new project dialog window " Project Card ", which serves to create a project or change information about the project. When you create a new project, specify the data of the people associated with the project. The card designing includes all the information from the register of the project and additionally the following information: Supervisor to the Customer; The project manager from the Developer; Purpose

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of the system; Information on the project team of developers; Information about the working group of the project; The reference to "Project Charter"; The reference to the "Agreement"; The reference to "Project Plan".

Information about the project and working groups can contain multiple values. The reference to "Project Charter", "Treaty" and "Project Plan" contains links to the latest versions of documents from the repository.

Lookup function and information change on the project and working groups is envisaged in the Project card design and references to the following documents: Treaty; Project charter; Plans: Basic Plan; Work plan.

Window "Glossary of objects" contains links between information objects. Information objects are selected from the directory. Communication between the object created with type "Document" and subject to the type of "Essence". Communication between objects is determined for a specific project. The information defined in the menu "Data domain", except in the "Glossary of objects" is not attached to the project, and this information can be used for any project.

Window "Work Stages" is the main work carried out with the project. It contains information about the sequence of work required for the designing the information systems. Information about the stages of work consists of two vertical frames. In the left window part - select and move through the list containing the name of the milestones in the right part of the window is information on selected stage. The software package includes work with the stages of work in two modes: standard and stepwise. Mode selection is provided to the user in the form of standard tabs in the "Work Stages". The list of works is displayed to the user in standard mode: Selecting the life cycle standards of the system; Description of business processes; Choice of system architecture; Identify methods combined set; The choice of tools for modeling and designing; Development of models of activity and data models; Decomposition system modules and functions; Identification of system users; Checking the quality of working models; Development of project documentation; Design of the database structure; Development of a prototype of the system; Introduction of a repository project.

In step mode, work stages are phased. Initialization phase: Development of the documentation for the project initialization; Choosing Standards of LC system. The phase of the survey: Description of business processes; Identify methods for working models "as it is"; Selecting the tools for modeling patterns of activity "as it is"; Development of models of operations "as it is"; Checking the quality of working models "as it is"; Development of documentation pre-survey. Phase analysis: decomposition of modules and functions; Identification of system users; Identify methods for working models "As it should be"; select of tools for modeling patterns of activity "as it should be"; Development of models of activity "as it should be"; Checking the quality of working models "As it should be"; Identify methods for data models; Select the tools for modeling data models; The development of data models; Development of techno-working documentation. Stage design: Selecting the system architecture; Selecting the tools for the design of a database; Designing the database structure; Selecting the tools for the design of user interfaces; Designing user interfaces; The development of a prototype system.

To work with the work stages following options are available: Save changes; Cancel changes.

Window "Corporate Directory" - contains a list of directories used. The following directories: Organizational Units; Positions; Information objects; Types of information objects; DBMS; Application Servers; Modeling techniques; Types of modeling approaches; Types of modeling techniques; Types of models; CASE-tools; Assignment types of CASE-tools; Group types of functions; Signs of readiness of the project document; Types of documents repository; Symptom document repository; Updates of work stages; Types of users. A review possibility and adjust the reference data is given.

Window "Methods and tools for modeling" - describes the modeling techniques and communication tools. This defines the connection kinds of models and types of modeling techniques. It provides the ability to change this information.

Window "life cycle standards and documents templates " - contains a description of the life cycle of communication standards and document templates. The standards and document templates are stored in the repository.

Shared repository is the repository of all documents. It provides viewing and updating references to documents. The repository is stored as summary containing information on database management systems, application servers, modeling, CASE-tools; standards governing the establishment of the life cycle system; document

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

templates. When a document is opened from the repository of links, the documents themselves should be open external applications, which create these documents (provided that does not provide specific applications for certain types of documents). When placing the document in the repository, you must enter the following information: name of the document (used as the title of reference); Document type; Link to the document; Date and time of placing (the current date and time for reference); Note (optional information).

The type of the repository document is selected from the directory and it is displayed in a hierarchical tree. For each kind of document more links to documents in the repository can be determined and stored for several documents with one kind or with no reference, if not placed in the repository for the documents of this type of document. The list of document types in the repository can be changed and expanded. The software package provides the opportunity to present the work Drafts as a network with the desired level of detail. This makes it possible to see the stages of readiness of the project. Stages can be performed sequentially and in parallel. Each phase may be in one of the following states:

1. Has not started - this corresponds to the red color of the top of the net. All phases of the project possess this state by default when you create the project.
2. In the work - this corresponds to the blue color. The given state may possess one or more stages, depending on the users.
3. Pass - green.

5. DISCUSSION

Many problems exist related to the different stages of the software life cycle. At the same time the analysis and design stage, the purpose of which is the identification, classification and formalization of information about all aspects of the subject area, affecting the properties of the final result is the initial stage of the "Development" and has a decisive influence on the quality of the results of the project. Hence, a particular importance of the tasks relating to this stage.

The works in this domain have been conducted for several decades by various scholars: C.Gane, T.Sarson [9-10], T. DeMarco [11], E.Yourdon [12], J.Rumbaugh [13], G.Booch [14-16], I. Jacobson [17-18], A.A.Shtrick [19], B.A.Pozin [20-21], I.Yu. Tuder [22-23], G.N.Kalyanov [24-25], E.Z. Zinder [26], A.M.Vendrov [27-28] etc. However, the currently existing methods and technologies of software development is not enough formalize the modeling domain. In modern methods is not enough formalized criteria and procedures to ensure the functional completeness and logical consistency of the results of building information models. At the same time the development of information systems has its own characteristics that must be reflected in the special events to maintain the logical integrity of the results throughout the project.

The developed technology and software package allows for analysis and planning of all necessary work in the design of information systems. Allow to assess the extent of the information systems are the basis for planning and resource allocation in the design of information systems.

6. CONCLUSION

The software package is formed as a distributed (multi-user) system with client-server architecture. The database system is a centralized, database access is performed using a two-tier architecture.

For user authentication, a user name and password is required when you log in, access to the functions of the system is determined by the user authentication procedures. The software package provides two groups of users: administrators and users.

In the long view, the developed technology can include a combined method based on the methods of structural and functional as well as object-oriented analysis. In this case, the developed technology can present the tools, enabling to describe the stages of the designing work, which offers the essential guidance at every stage. Available blank documents (project documentation templates) provided for a fast and high-quality document creation. The methods of structural and functional analysis will ensure a rigorous and clear description of the structure developed by the information system, multi-level decomposition of the process into the separate stages and types of works, presenting the logical organization and relationships, separate display of input, output and control information, as well as methods and means for the implementation of life cycle processes of the information system. On the other hand, the object-oriented methods, being a major figure in the system analysis and design, based on the decomposition of the domain object is represented as a set of objects that interact with each other by sending messages. This approach does not

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

contradict to the structural approach, moreover, the methodology fragments of structural analysis (basic models DFD, ERD, STD) are used in object-oriented analysis to structural modeling and behavior of the objects themselves. Furthermore, in the structural principles of the approach objective approach can be considered, for example, IDEF4. Therefore, a combined method based on the methods of structural and functional and object-oriented analysis can be included in the developed technology. In accordance with the selected methods, testing the methods of combined selection can be done and provide recommendations in the case of error detection. The selection of methods can be represented in the form of manuals with a brief description of the application and characteristics of the methods.

REFRENCES:

[1] Boranbayev S., Boranbayev A., Altayev S., Nurbekov A. Mathematical model for optimal designing of reliable information systems. Proceedings of the 2014 IEEE 8th International Conference on Application of Information and Communication Technologies-AICT2014, Astana, Kazakhstan, October 15-17, 2014, pp.123-127.

[2] Boranbayev S., Baidyussenov R. Development of the methods for designing information systems using the network model. Vestnik ENU. 2010, №4, p.211-216.

[3] Boranbayev S., Baidyussenov R. Development of software and information center for designing information systems. Proceedings of the II International Scientific and Practical Confer ence Informatization of society-Astana, 2010, p.140-142.

[4] Boranbayev S. Development methods of information systems. Astana, "Software master", 2012. - p. 256.

[5] Boranbayev S. The theory of information systems. Astana, "Elorda", 2006. – p. 212.

[6] Boranbayev S. Mathematical Model for the Development and Performance of Sustainable Economic Programs / / International Journal of Ecology and Development, Vol. 6 , No. W07, 2007 , p.15 -20.

[7] Boranbayev A., Boranbayev S.Computer-Based Automation In Medicine And Proposed Methodology And Component-Based Architecture of a Computer-Assisted Posting and Processing of Medical Billing Claims. Proceedings of the 2010 International Conference on Internet Computing - ICOMP'10, Las Vegas, Nevada, USA, July 12-15, 2010, P.116- 122.

[8] Boranbayev A., Boranbayev S., Defining Optimal Approaches And Methodologies for Concatenating Necessary Features of Various Java-Based Frameworks and Efficient Technologies for the Effective Development of Enterprise Information Systems. Proceedings of the 2010 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government - EEE'10, Las Vegas, Nevada, USA, July 12-15, 2010 , P. 385-390.

[9] Chris Gane, Trish Sarson. Structural system analysis: tools and methods.// перевод с translated from English, M., "Atex" 1993.

[10] Chris Gane, Trish Sarson. Structured System Analysis. Prentice-Hall, 1979.

[11] De Marco T. Structured Analysis and System Specification. Englewood Cliffs, New Jersey: Prentice Hall, 1979.

[12] Yourdon E. Modern Structured Analysis. // Englewood Cliffs, New Jersey: Yourdon Press, 1989.

[13] Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorensen W.. Objectoriented modeling and degisn. Englewood Cliffs, NJ. Prentice Hall, 1991.

[14] Booch G. Object-Oriented Analysis and Design with Application examples C++/ translated from English M.: BINOM SPb. Nevsky Dialect, 1998.–p. 560.

[15] Booch G. Object Solutions. Managing the object-oriented project. // CA, Addison-Wesley Publishing Company Inc, 1996.

[16] Booch G. Object-oriented development. IEEE Transactions on Software Engineering 12, 2 (Feb.1986), 211-221.

[17] G. Booch, J. Rumbaugh, I. Jacobson. The Unified Modeling Language Reference Manual – Second Edition, Addison-Wesley, 2004.

[18] Jacobson I., Christerson M., Jonsson P., Overgaard G. Object-Oriented Software Engineering. A Use Case Driven Approach. // Addison Wesley Longman Limited, England, 1996.

[19] Strick A., Ossovetskiy L., Messikh I. Structural design of reliable embedded computer programs. – L.: Mashinostroeniye, 1989 –p. 296.

[20] Lapin S., Fominskiy E., Kozlov G., Fominskaya E., Pozin B. Experience in the application of RAD and DATARUN

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

methodologies in specific projects. // Proceedings of the conference "Corporate database 97"1997.

[21] Pozin B. Modern methodologies and automated technology design and application software development of information systems. // STI. Series 1. The organization and methods of outreach.1999. №11.

[22] Tuder I. The collective analysis of the subject area // Banking technologies, M. "Business and computer"№5 May 2001 p. 32-38.

[23] Tuder I Development of standards based on a DFD as the de facto standard // All-Russian Conference "Standards in projects of modern IS": Proceedings of the II-nd All-Russian conference, Moscow, 2002, p.97-100.

[24] Kalyanov G. Consulting at the automation of the enterprises (approaches, methods, tools). M.: SINTEG,1997.

[25] Kalyanov G. Methods and tools for system structural analysis and design. M .: NICC MSU, 1995.

[26] Zinder E., Belokon A. Personalization of information technology and tool support in the design.//Tahkekeha elektroonika elementide projekteerimise ja kat- setamise numbrilised meetodid ja vahendid. Vabar. noup. ettek. teesid. K.II. – Tallinn: TTU, 1989.

[27] Vendrov A. CASE-technologies. Modern methods and means for designing information systems. // M.: "Finance and Statistics"1998, p.175.

[28] Vendrov A. Software design of economic and information systems: Textbook. -M. Finance and Statistics, 2000. – p.352.