МІНІСТЕРСТВО
ОСВІТИ І НАУКИ
УКРАЇНИ

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

ISIT 2019

ПРАЦІ
Міжнародної науково-практичної конференції

19 – 24 серпня 2019 року
Одеса, Україна

Міністерство освіти і науки України;
Одеська Міська Рада
Одеський державний екологічний університет
Одеський національний університет імені І.І. Мечникова,
Одеська державна академія технічного регулювання та якості
Харківський національний університет радіоелектроніки
Економічна академія "Д.А.Ценов", Болгарія
Інститут спеціального зв'язку та захисту інформації КПІ ім. Ігоря Сікорського
AGH науково-технологічний університет ім. Ст. Сташіца, Польща;
Університет Бєльсько-Бяла, Польща;
Університет Північ, Республіка Хорватія;
Представництво "Польська академія наук" в Києві
Лодзький університет, Польща
Лодзький Технічний університет, Польща

«ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

праці
міжнародної науково-практичної конференції
19 – 24 серпня 2019 року
Одеса, Україна

«INTELLECTUAL SYSTEMS AND INFORMATION TECHNOLOGIES»

proceedings
of the International Scientific and Practical Conference
2019, August, 19th to 24th
Odesa, Ukraine

Одеса

ТЕС

2019

УДК 004.89.03(062)
И 730

Наукові редактори: д.т.н., проф., Гунченко Ю.О. (ОНУ імені І.І.Мечникова)
к.т.н., доц., Фразе-Фразенко О.О., (ОДАТРЯ)

Матеріали статей опубліковані в авторській редакції

И 730 «Інтелектуальні системи та інформаційні технології»; матеріали статей міжнародної науково-практичної конференції, м. Одеса, 19 – 24 серпня 2019 року./ Одеський державний екологічний університет.– Одеса: ТЕС, 2019 –260 с.
ISBN 978-617-7711-43-7

Збірка містить праці Міжнародної науково-практичної конференції з інформаційних технологій, систем та засобів штучного інтелекту, обчислювальних машин, систем, мереж та їх компонентів, автоматизації систем та процесів керування, систем захисту інформації, кібернетики, управління проектами, електротехніки та телекомунікацій, інтелектуальних приладів та систем.

УДК 004.89.03(062)

ISBN 978-617-7711-43-7

Software for Automated Design of Network Graphics of Software Systems Reengineering

Stanislav Velykodniy
Department of Information Technologies
Odessa State Environmental University
Odessa, Ukraine
velykodniy@gmail.com; ORCID:
ttps://orcid.org/0000-0001-8590-7610

Zhanna Burlachenko
Department of Information Technologies
Odessa State Environmental University
Odessa, Ukraine
7035373@ukr.net; ORCID:
https://orcid.org/0000-0001-8451-5527

Svitlana Zaitseva-Velykodna
Department of Informatics
Odessa State Environmental University
Odessa, Ukraine
svetlana.zaytseva@gmail.com;
ORCID: https://orcid.org/0000-0001-7453-8821

***Abstract*-The subject of the work is the construction of a raphical network model of reengineering of the software ystem. Purpose of the paper is development of software for ncreasing the level of automation of designing network charts or the organization of production by reengineering software ystems in the framework of project management. The task efore creating a software tool is the ability to work with all ypes of network charts with the possibilities of their omprehensive transformation. Project decisions proposed by he authors are the results of the article. The content of the esign part is determined, firstly, by the specifics of the lanning of software projects reengineering, and secondly, by he features of specific technical proposals for a project that is nanageable. Conclusions. The architecture is developed in the orm of several structural and behavioral diagrams, namely: ise case diagram, which provides an analyst with a detailed dea of the software field of application; sequence diagram that s designed to create a programmer's imagination on how to erform actions when working with a future program tool; tatechart diagram that is required for a visual representation f those states in which the software can be at different times; lass diagrams that are used to design the main form filling of he future software; component diagram that is designed to xamine the composition of the components of the future oftware and indicate the sequence of compilation and assembly f individual modules. The numerical and temporal estimation f the planning parameters is based on the data obtained from he Gantt design charts.**

Keywords–project management; graph; network schedule; oftware; reengineering; CASE-tool; UML-diagram

I. INTRODUCTION

The process of a project creation, prototype, preimage of e future object, condition and methods of its production is alled design. In engineering, design is understood as the evelopment of project, construction and other technical ocumentation designed to provide for the creation of new pes and patterns. In designing a systematic approach is sed, which consists in the establishment of the structure of e system, such as links, defined attributes, and the analysis f the effects of the environment. In the design process, chnical and economic calculations, charts, graphs, explanatory notes, estimates, calculations and descriptions are made.

One of the main components of design is planning, defined as a pre-scheduled procedure or an optimal allocation of resources necessary to achieve the goal. Network planning is one of the forms of graphical representation of the content of work and the duration of implementation of strategic plans and long-term complexes of project, planning, organizational and other activities of the enterprise.

Along with linear charts and table calculations, network planning methods are widely used in the development of perspective plans and models for the creation of complex production systems and other objects of long-term use. Network plans of enterprises to create new competitive products include not only the total duration of the whole complex of design and production and financial and economic activities, but also the duration and sequence of the implementation of individual processes or stages, as well as the need for the necessary economic resources.

II. DETERMINATION OF PREVIOUSLY UNSETTLED PARTS OF THE GENERAL PROBLEM

The general content of the graphs [1] is as follows: linear or ribbon graphs on a horizontal axis at a selected time scale, postponed the duration of work in all stages and phase of production, the content of the work cycles depicted on the vertical axis with the required degree of their division into separate parts or elements.

At domestic enterprises, cycles or linear charts, as well, are used in the process of short-term or operative planning of production activities [2]. The main disadvantage of such plans-schedules is the lack of the possibility of close interconnection of individual works in a single production system or the overall process of achieving the planned end-goals of the company (firm). Network graphs serve not only to plan a variety of long-term jobs, but also for their coordination between executives and project implementers [3], and network charts are needed to determine the necessary productive resources and their rational use. Automated Enterprise Resource Planning (ERP) systems typically include computer programs [4] that automate some of the steps in drawing up and updating network graphs in one form or another, but such programs have a fairly high licensed price and are not suitable for a domestic producer [5].

Thus, the purpose of the work is to development of software for increasing the level of automation of designing network charts for the organization of production by reengineering software systems in the framework of project management. The object of work is the network planning of the production process. The subject of the work is the construction of a graphical network model of reengineering of the software system. The task before creating a software tool is the ability to work with all types of network charts with the possibilities of their comprehensive transformation.

III. MATERIALS AND METHODS

The article deals with network planning for the PERT methodology, use of elements of graph theory and Gantt chart method as an accounting method for project management. The simulation of the system software architecture is carried out within the UML (Unified Modeling Language) 2.5 methodology using the Enterprise Architect 14 CASE.

IV. RESEARCH RESULTS

The development of the system architecture of a software tool for network planning management of a software project reengineering (hereinafter referred to as software) shall be started with design of a Use-Case Diagram (UCD). UCD's are used to provide an analyst with a detailed picture of the application development industry. With the help of UCD, it becomes clear what the product is intended for, subsystems and modules it operates, links which are the elements and the entity in it. The central element of the projected UCD is a network graph (NG), depicted as an actor with a stereotype "business actor". The actor "NG" has a variety of connections, most of which associations with the stereotype "uses", but also there are dependencies. It can be seen from UCD that NG's are used in all applications associated with the list mentioned.

Let's turn to the consideration of the essence, which is connected with the NG type of connection dependence "dependency". On the diagram of use case it is a package "Software Development", the implementation of which is really based (depends) on NG. The composition of the package includes many entities, united by links, the collection of which is the enclosed designed subdigraph of the diagram of use case, which has a graph similar to the topology.

Software engineering and computer science in general are one of the areas where NG is most often used, therefore the package is submitted and presented as a separate dependent structure. The complexity and the large number of modules and protocols in modern software products greatly complicates the understanding of their work, management and optimization. Therefore, NG programs are often compiled, and most often it is done automatically by assemblers, compilers, or parsers.

Sequence Diagrams (SD) are designed to create a programmer's imagination on how to perform actions when working with future software, which are formed by the system architect. There are, as a rule, two types of SDs: the SD for displaying programmer actions when developing the software and the SD for displaying user actions while continuing to work with future software. SDs of the second type are used to create user instructions that are an integral part of software and methodological complexes. This type of SD is designed in the given article, since for the implementation of the reengineering of a software project, it is necessary to understand the principles of the software work.

State Machine Diagram (SMD) is required for a visual representation of those states of software in which it can be at different times. In other words, we can say that states describe the behavior of software, which, in turn, can depend both on user instructions and on the computational procedures of the software itself. As can be seen from the name of the chart – the key points in it – are state (state), representing, of its kind, the point of stopping the work of the software or the collection of statistical and technical information (logging). Each state has entry points and exit to this state, and there may be several such points (both for inputs and outputs) – precisely for fixing these positions within the state – there is such a concept as the "state history", which optionally turns on in the specification of each state and has the designation: the letter "H" (history), which is presented in a circle.

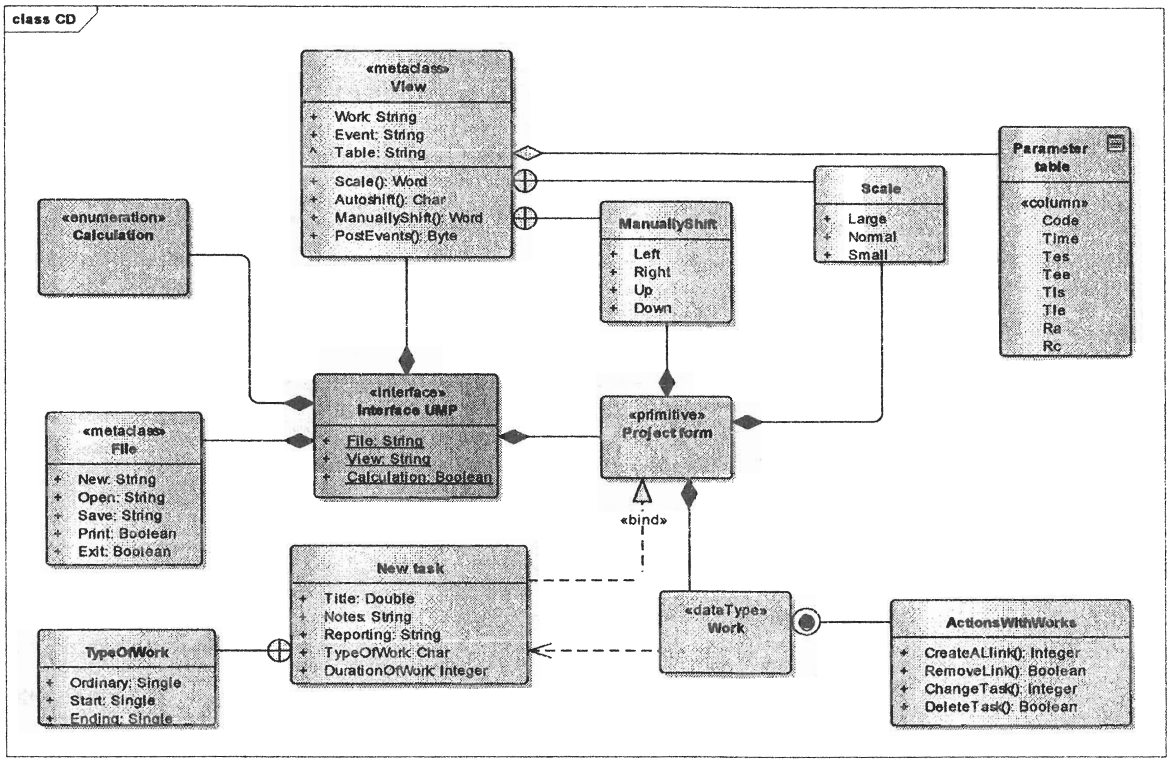
In addition, important for the SMD, as well, there are transitions, which are the relationships between states. Each transition has a syntax that reveals the mathematical or physical process that caused this transition. In turn, the syntax, in addition to the name of the transition, may contain a limiting condition (it is given in rectangular brackets [ ]) – in case of failure of which – the transition is impossible. The SD, as shown in the figure, is a cascade model of the design of the software, where, as a gradual stage of the implementation of the states, the main diagonal of the diagram with many branches (depending on the points of exit from the concrete states) is in different directions.

Activity diagram (AD) is intended to reflect the activity of a particular object, during the analysis of the action of interest. With the help of AD, you can research and protocol the flow of information flows that are required for the creation of further software. AD emulate some aspects of the SD and SMD, but only in AD is the ability to design complete algorithmic cycles, based on which the blocks "Decision" will stand. The main essence of AD is the so-called activity (Activity), which is the essence similar to the state of the SMD. Activity shows the specific actions of the software process. Also, on AD there are transitions that have the same nature as the transitions to the SCD, with the same limiting conditions. In addition, there are activity paths on AD, which indicate the allocation of boundaries and activity zones of each of the objects. Designed AD for the main information flow ("I-Flow") of software design of NG is shown as activity paths "User", "Interface" and "Compute and

Iemory Area". Their names are in the upper part of AD, and ie activity limits are separated by solid lines.

Class diagrams (CD) are used to design the main form of ontents of future software. As for UML notation, classes ontain attributes (self-encapsulated data of different origin) nd operations (actions on these or other data). Each specific ttributes and class operations that are designed in the rovided software are shown in Fig. 6. In addition to the lasses on the CD, connections or relationships are also nportant for analysis (this is a more precise definition for :D). In general, entities or relationships, in relation to the methodology of CD – there is a rich diversity, but let us dwell only on those that are present in the designed CD of software (Fig. 1) and need additional clarification.

The central class on the CD is a class interface ("Interface UMP"), which is a graphical user interface software. This class first appears to the user in the form of a bootable primitive class "Project form", which has a compositional relationship with the class "Interface" and defines the primary geometric dimensions of the interface window. Also in front of the user will appear a menu bar, whose elements – are attributes of the class "Interface UMP".



ig 1. Class Diagram

Calculation class (enumeration) "Calculation", which is ɘlated to the compositional relationship with the "Interface IMP" class, implements the numerical (tabular ɘpresentation parameters) and graphical (critical path) of the IG parameters. The "New Task" class attached (relation Bind") to "Project Forms" is intended to create new and ıbsequent edits of an existing data type "Work", which is a raphical representation of the "Work" contents. The nesting ass "Type of Work" is attached to the "New Task" class. It ırms the types of work that differ in the presence or absence f incoming and outgoing links.

Component diagram (CPD) is intended to study the ımposition of future software components and to indicate e sequence of compilation and assembly of individual odules. The main requirement for CPD is standardized – ere are no cycles, that is, the consistency of the components ould be clear and transparent. The programmer works with e components that may be available to him – in reverse order. Designed by Component diagram software is shown in Fig. 2

Detailed study shall be applied to the structure of Component diagram. The first component that the programmer deals with is the artifact or the shortcut for the "Link for UMP (* . lnk)" attached to the executable "UMP.exe" that launches the new project through the exposed interface components "Interface Visibility". This component serves as the interface to the "Interface UMP" , which includes port help and port driver. Port help has connected to it document aircraft "Help index . Gid", which may be optionally called by the user in case of a complicated situation with the project. Through the Port Driver – an artifact containing special data is connected – Device Driver Profile "Device Driver . ddp". From this driver, the mode and support of the solving ability of the entire software interface depends.

Also, the "Device Driver ddp" component depends on the form of the Delphi Module "Module Form .dfm" and the package component of the "Delphi Compiled Module dcu", which also depends on "Module Form . dfm" Thus, the package component "Compiled module Delphi (*.dcu)" has a double arrangement (Fig. 2). The package component "Compiled module Delphi (*.dcu)" is main for the range of the components, namely for: executable file "UMP exe", described above; document artifact "Status Information", which contains technical information on the current and final state of the project; object "Default project dpr," which is downloaded as a template when creating a new project and which (if necessary) can be modified and supplemented.

Cumulative database component "Dbase resource . res" has double arrangement and is compiled each time on addition or change of information in document artifact "Status Information" and object "Default project . dpr". Executable file "UMP . exe" is main for: artifact or contact shortcut "Link for UMP (* . lnk)" described above; document artifact software options and parameters "Options . ini", which contains technical information on the latest geometric size of the project form set by the user; Web-document artifact "Graf . htm", which reflects the final formed topology of the placement of tasks on NG; thus Web-document artifact "Graf . htm" forms (realize) the object itself "Graf . bmp" as an independent (standalone) bmp-file that can be saved (printed, sent, etc.) and edited. Thus, document artifact "Options . ini" and Web-document artifact "Graf. htm" are arranged as abstraction dependency, which means creating them as essences only for the needs of the user and with the necessity of his participation (making changes in size, printing teams or Web-conversion.

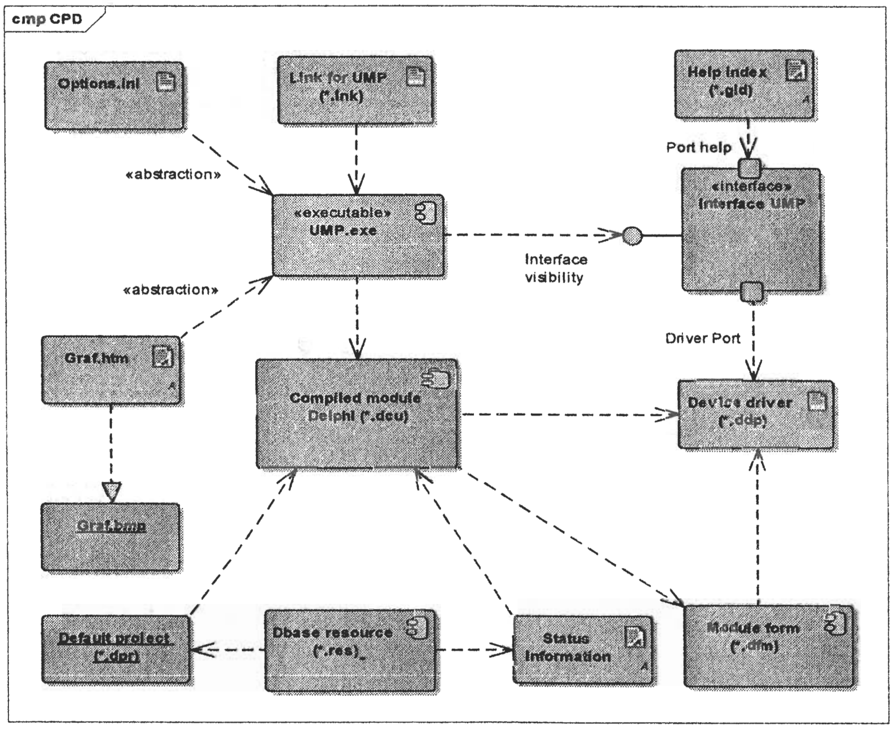


Fig 2. Component Diagram

V. Discussion

When analyzing and discussing the results of the project, system architects expressed a lot of thoughts about the means of implementing the software. These thoughts related to the technology and implementation language, the number of encoders and testers involved, etc., but all of this is part of a new phase in the implementation of the software. At this stage, regardless of the choice of means, languages and software coding technologies, its architecture has already been developed and ready. When you select programming languages, you will only need to process the component diagram.

VI. Conclusions and Perspectives of the Further Development

It should be noted that although in today's paid specialist packages of computer programs of planning and operational management, the type of "Activity-on-arrow network" graphs is used mainly – the software designed is suitable for all types of network graphs with the possibilities of their comprehensive transformation. The result of this article is a project decision suggested by the authors. The content of the design part is determined, firstly, by the specifics of the planning of reengineering of software projects, and secondly, the features of specific technical proposals for a project that is manageable.

In the given article the designed of the software for management of network planning of reengineering of the software project has been designed. The architecture is developed in the form of several diagrams of various nature, executed with the observance of UML 2.5 notation using the CASE toolkit Enterprise Architect 14. The development of the basis includes the methods of network planning for the PERT methodology and the use of the elements of the theory of graphs. The numerical and temporal estimation of the planning parameters is based on the data obtained by the Gantt chart method, as an account for the management of software projects.

The prospects for creating this software consists in the product encoding for a final industry user (software project manager), which is important to know only the sequence of reprogramming work on the software system and the duration of each stage, and it does not matter how the graph is formed, that is, which it type, since the type of network graphics itself can be mutually converted.

For work with the software will be created a program and methodical complex, which will be developed by user instructions on the application of the necessary software, supplemented by comments on the work of the software designed.

REFERENCES

[1] J. A. Bondy, U. S. R. Murty, Graph Theory. San Francisco : Springer, 2008, 655 p., doi: https://doi.org/10.1007/978-1-84628-970-5.

[2] D. Jungnickel, Graphs, Networks and Algorithms. 4th ed., Berlin: Springer, 2013, 677 p., doi: https://doi.org/10.1007/978-3-642-32278-5.

[3] S. Velykodniy S., Zh. Burlachenko and S. Zaitseva-Velykodna, “Software for automated design of network graphics of software systems reengineering” in Scientific Journal Herald of Advanced Information Technology, vol. 2 (03), 2019, pp. 20–32, doi: https://doi.org/10.15276/hait.02.2019.2

[4] I. Nevlyudov, S. Velykodniy, M. Omarov, “Using CAD / CAM / CAE / CAPP when forming control programs for CNC machines” [“Ispol'zovanie CAD/CAM/CAE/CAPP pri formirovanii upravlyayushchikh programm dlya stankov s ChPU”] in Eastern-European Journal of Enterprise Technologies, vol. 2, iss. 2 (44), 2010, pp. 37–44 (in Russian).

[5] R. W. Selby, Software Engineering: Barry W. Boehm's Lifetime Contributions to Software Development, Management and Research. John Wiley & Sons, New Jersey, 2017, 818 p.

Наукове видання

«ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

праці
Міжнародної науково-практичної конференції
19 – 24 серпня 2019 року
Одеса, Україна

«INTELLECTUAL SYSTEMS AND INFORMATION TECHNOLOGIES»

proceedings
of the International Scientific and Practical Conference
2019, August, 19th to 24th
Odesa, Ukraine

Підписано до друку 5.08.2019. Формат 60х84/8. Папір офсетний
Ум. друк. арк. 30,23. Наклад 70 прим. Замовлення № 0696
Видавництво та друкарня «ТЕС». (Свідоцтво ДК№ 771) Одеса,вул.Канатна 81/2
Тел: (0482)42-90-98, (0482)42-89-72
Надруковано з готових оригінал-макетів

Одеський державний екологічний університет
65016, Одеса, вул. Львівська, 15
Одеський національний університет імені І.І. Мечникова,
Одеська державна академія технічного регулювання та якості