ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse144952020.pdf
https://doi.org/10.30534/ijatcse/2020/144952020

The problem of structuring indicators of quality of decision software support system

Oleksandr Turinskyi[1], Hennadii Pievtsov[2], MaksimPavlenko[3], Serhiy Osievskiy[4], Sergey Herasimov[5], Volodymyr Djus[6]
[1]Ivan KozhedubKharkiv National Air Force University, Kharkiv, Ukraine, red.hnups@gmail.com
[2]Ivan KozhedubKharkiv National Air Force University, Kharkiv, Ukraine, info@hups.mil.gov.ua
[3]Ivan KozhedubKharkiv National Air Force University, Kharkiv, Ukraine, bpgpma@ukr.net
[4]Ivan KozhedubKharkiv National Air Force University, Kharkiv, Ukraine, gsvnr@ukr.net
[5]Ivan KozhedubKharkiv National Air Force University, Kharkiv, Ukraine, gsvnr@ukr.net
[6]Ivan KozhedubKharkiv National Air Force University, Kharkiv, Ukraine, vdjus@ukr.net

ABSTRACT

The article addresses improving the performance of man-machine systems (MMS) by improving the quality of decision support system software (DSSS). The proposed solutions are based on existing models and methods, which have been analysed in detail in a scientific study, in particular the application of methodical static application debugging in certain solutions obtained. It is proposed to use the methods for solving the problem of MMS control according to the vector criterion as the basis for solving this problem. A mechanism for synthesizing a comprehensive plan of operations to improve the quality of the DSSS is proposed, and the possibility of analysing the step-by-step implementation of a plan to obtain an analytical assessment is shown. For this purpose, a comprehensive plan is presented in the form of a relevance tree, built in accordance with the provisions of graph theory and the rank approach. The time complexity of the algorithms implementing the relevance tree tasks was evaluated. The solutions obtained meet the requirements of DSTU ISO / IEC 9126, ISO DSTU / IEC 14598, and take into account the requirements of the Software Quality Requirements and Evaluation series of standards as the value of the vertices of the event tree graph. In the process of solving the problem, the specificity of the functioning of the MMS was taken into account, the possibility of formalizing various aspects of knowledge (aesthetic, disizional, causal, diontic) and ensuring a given level of efficiency in finding solutions.

Key words :software, decision support system, graph, relevance tree.

1. INTRODUCTION

Comprehensive expansion of decision-making automation capabilities in poorly formalised areas, through the use of expert knowledge, leads to an increase in the number of errors in DSSS. And as a consequence, there is an increase in possible sources of failures (both software and hardware). An analysis of DSSS errors on the operation of software complexes in general showed that such errors caused up to 70% of their failures [1-5]. On the other hand, the intensification of the development of knowledge processing methods in DSS actually provokes a sharp increase in their intellectualization, that is, the volume of knowledge bases increases, and requires the parallel development of effective quality control methods for special DSSS. From the point of view of the fact that the European Electrotechnical Commission has made adjustments and additions to industry standards RAMS-requirements for software decision-making software (which contain recommendations for the application of a comprehensive software quality criterion), the relevance of quality control issues for special DSSS should be discussed.

2. LITERATURE REVIEW

The problem of developing methods for assessing the quality of DSSS and debugging the knowledge bases has been described in some detail in the writings of scientists such as V. Lipaev, S. Zykova, A. Narinani, Scott, Suva, Nguyen, Perkins, Tepandi, Kragun, Stendel [6-12] and others. It is clear from the analysis of these works and publications that the greatest success has been achieved in the development of static debugging methods, those methods that do not require the launching of an expert system for execution[13, 14]. However, there is currently no uniform approach to formalizing the so-called structural errors detected by static debugging methods: redundancy, incompleteness, contradiction[15, 16]. This is why the statically correct knowledge bases do not guarantee the quality of decision-making at the expense of errors in the knowledge, often related to the complexity of the individual subject matter, which, for example, allows for duplication of reasoning logic. The most difficult for detecting errors of type "forgetting to delete" and "critical combination of events", which leads to errors in decision-making. Testing is the way to detect knowledge errors. Today there are only isolated attempts to automate the construction of tests[17, 18]. Despite the development of hybrid DSSS combining different types of knowledge presentation, most modern systems are based on the product approach. Another developed way to create DSSS is to use trilayer personatron as a variant of artificial neural networks for which there is no single approach to debugging at all. The ever-increasing complexity of the functionalities implemented by the software in intelligent systems increases their volume and labour-intensive development[19, 20]. In line with the changing complexity of the software, the number of detected and undetected defects and errors is increasing, which affects the operation of the software complex as a whole[21, 22]. Complex millions of lines of code DSSS with tens of thousands of rules in the knowledge base cannot fundamentally be unmistakable. In [15] it is shown that "the problem of detecting and fixing errors intensifies as the complexity of the tasks to be solved by the programs, and threatens to catastrophe in systems performing critical functions of controlling large, expensive and especially important objects or processes".

This is why the task of assessing the quality of the DSSS with a view to improving the performance of MMS is relevant.

Based on the aforementioned existing solutions and theoretical provisions of the theory of Data Base (DB) and Knowledge Base (KB), the solution to the main task is seen in the development of a practical mechanism, it consists in the synthesis of a comprehensive plan of operations to improve the quality of software and analysis of the possibility of implementing this plan under various control influences and states of the external environment, followed by construction of models and algorithms for the synthesis of this plan in the form of an event tree (based on elements of Boolean algebra and the rank approach), which will allow finding ways to reach the top of the first rank (root) of the event tree and determine the minimum number of combinations of events that can cause a defect in the DSSS.

3. MAIN MATERIAL

The solution to this problem is proposed to be based on methods for solving the problem of MMS control according to a vector criterion. This is primarily related to the ability to analyse and manage the performance characteristics of a dedicated special DSSS such as performance, design and operation costs, flexibility, upgrades and others. In addition, the mathematical formulation of this problem allows a more reasonable approach to the development of a methodology for achieving the goal, which is to increase the efficiency of the functioning of complex of MMS, from the standpoint of system analysis and to show the influence of local tasks implemented in the special DSSS modules on solving the problem of improving the functioning of man-machine systems and due to this to develop more effective models and methods that can significantly improve the quality of special DSSS.

Formally, the problem of MMS control according to a vector criterion is reduced to the development of an algorithm for the synthesis of a vector of control actions $\vec{u}(t) \in \{\overleftrightarrow{U}(t)\}$, the implementation of which under any admissible conditions of the environment $\vec{x}(t) \in \{\overleftrightarrow{X}(t)\}$ and allows maximizing the criterion $\vec{K} = (K_1, K_2, ..., K_m)$, which characterizes the efficiency of the functioning of MMS at a given time interval $[t_s, t_f]$:

$$K_1 = \int_{t_s}^{t_f} K_1^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right] dt\,;$$

$$K_2 = \int_{t_s}^{t_f} K_2^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right] dt\,;$$

$$...$$

$$K_m = \int_{t_s}^{t_f} K_m^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right] dt\,. \qquad (1)$$

Under functional restrictions

$$F_i^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right] \geq 0, i = \overline{1, n_1^{'}}\,; F_i^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right] < 0, i = \overline{1, n_2^{'}}\,. (2)$$

Boundary conditions:

$$F_i^{'(t_s)}(\vec{x}^{'}(t), \vec{u}^{'}(t)) \geq 0, i = \overline{n_2^{'} + 1, n_3^{'}}\,;$$
$$F_i^{'(t_f)}(\vec{x}^{'}(t), \vec{u}^{'}(t)) = 0, i = \overline{n_3^{'} + 1, n_4^{'}}\,, \qquad (3)$$

related to the specific functioning of the MMS. In expressions (1)–(3): $n_1^{'} - n_4^{'}$ is known constants determined by the specifics of the functioning of a MMS; $\left\{\vec{X}^{'}(t)\right\}$ and $\left\{\vec{U}^{'}(t)\right\}$ is areas of admissible values of vectors $\vec{u}^{'}(t)$ and $\vec{x}^{'}(t)$, respectively; $K_1^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right]$ is subintegral function characterizing the quality of the software of the special purpose intelligent systems of the man-machine system; $K_2^{'}\left[\vec{x}^{'}(t), \vec{u}^{'}(t)\right]$ is subintegral function characterizing the security of data systems). Consideration of the specifics of the MMS functioning is mandatory and justified in [16]. In particular, the specific features of the man-machine system are as follows:

taking into account the possibility of formalizing different aspects of knowledge (aletic, dissident, causal, diontic);

consistency property;

ensuring a given level of completeness in the description of the subject area;

ensuring a predetermined level of speed in the search for solutions;

providing the ability to obtain many alternative solutions.

Therefore, taking of expressions(1) – (3) into account, in order to solve the given problem it is necessary first to develop mathematical models, numerical methods, algorithms and software complexes allowing to optimize one of the basic scalar components of $\vec{K} = (K_1, K_2, ..., K_m)$ is target function $K_1'\left[\vec{x}'(t), \vec{u}'(t)\right]$ describing the quality of the DSS software.

In view of the above, the objective of improving the quality of the DSS software is outlined.

The production of the proposed solutions is oriented towards meeting the requirements of ISO DSTU / IEC 9126 Software Engineering. Product quality (Parts 1–4) [23] and DSTU ISO / IEC 14598 Information technology for software product evaluation (Parts 1–6) [24] which complete the quality model and place the evaluation process in a separate group of ISO / IEC standards. In addition, the requirements of the Square Standard Series (Software Quality Requirements and Evaluation) (Figure 1) are taken into account, which defines the software quality performance model as consisting of the following quality criteria:

internal quality criteria, that is, requirements for the quality of the code and for the internal architecture;

external quality criteria, i.e., capability requirements;

quality in use criteria, that is, those that are established not only for software but for the entire information system.

This is due to the fact that the SQuaRE provides a specification of software quality requirements and reduces uncertainty when organizations work together to develop, implement and maintain software, for example, between developers, developers and independent appraisers [19, 25].

Section on quality requirements 2503n	Section of quality model 2501n	Section of quality assessment 2504n
	Section of quality management 2500n	
	Section of quality measurement 2502n	
Section of further expansion 25050 – 25099		

Figure 1: Organization of a series of SQuaRE standards

According to these documents, the quality of the software, including the DSSS of the systems, is the set of properties that characterize the ability to maintain its level of performance under the specified conditions for a given period of time.

Based on this definition of quality, the subintegral expression of the performance criterion of a problem is function $P_Q\left[\vec{x}'(t), \vec{u}'(t)\right]$, characteristic of the probability of maintaining at a given time interval $[t_s, t_f]$ the required level of software quality under unfavourable circumstances. The adversity refers to the occurrence in the operation of an intellectual system of such a sequence of events, each of which individually has no significant impact on the quality of the software, the combined effect is a significant reduction in quality.

In view of the above, the objective of improving the performance of software in the design and operation of the special DSSS is to develop an algorithm for determining such control effects as $\vec{u}'(t)^* \in \left\{\vec{U}'(t)\right\}$, The implementation of which, under any permissible environmental conditions $\vec{x}'(t)^* \in \left\{\vec{X}'(t)\right\}$, would maximize the $K_1 = \int_{t_s}^{t_f} P_Q\left[\vec{x}'(t), \vec{u}'(t)\right]dt$ criterion, Characteristic of the probability of maintaining the required performance of the DSSS at the $[t_s, t_f]$ time interval.

In a formalized form, the problem of improving the quality of DSSS has the following formulation:

$$K_1 = \int_{t_s}^{t_f} P_Q(\vec{x}'(t), \vec{u}'(t))dt \to \max, \quad (4)$$

under restrictions

$$F_i\left[\vec{x}(t), \vec{u}(t)\right] \geq 0, \; i = \overline{1, n_1};$$
$$F_i\left[\vec{x}(t), \vec{u}(t)\right] < 0, \; i = \overline{1, n_2}. \quad (5)$$

boundary conditions

$$F_i^{(t_s)}\left[\vec{x}(t), \vec{u}(t)\right] \geq 0, \; i = \overline{n_2 + 1, n_3};$$
$$F_i^{(t_f)}(\vec{x}(t), \vec{u}(t)) = 0, i = \overline{n_3 + 1, n_4}, \quad (6)$$

arising from the specificity of the control object specified in [16], where $n_1 - n_4$ are known constants; $\left\{\vec{X}(t)\right\}$ i $\left\{\vec{U}(t)\right\}$ is the domain of the valid values of the vectors $\vec{u}(t)$ and $\vec{x}(t)$ respectively.

A general approach to problem (4) can be formed based on the assumption that (4) is a variable calculus class for conditional extremum. Its solution using the classical mathematical apparatus of the theory of variational calculus, based on the use of Euler's equations, involves considerable difficulties, the main ones being:

high task dimension;

the uncertainty of a number of scalar constituents of environmental state vectors $\vec{x}(t) \in \left\{\vec{X}(t)\right\}$ at $[t_s, t_f]$ time intervals of considerable length (due to the specificity of the control object);

the presence of both quantitative and qualitative variables in the $\vec{x}(t) \in \left\{\vec{X}(t)\right\}$ and $\vec{u}(t) \in \left\{\vec{U}(t)\right\}$ vectors, which are also unclear;

The need to solve in real time a complex system of high-order nonlinear differential equations for synthesis of $\vec{u}(t)$, etc.

Due to this circumstance, the developed algorithm of the vector synthesis of control effects $\vec{u}(t)$ is based on the statement, according to which, in order to achieve the goal of maximizing the $K_1 = \int_{t_s}^{t_f} P_Q\left[\vec{x}(t), \vec{u}(t)\right]dt$ criterion, it is sufficient to develop and implement a detailed comprehensive plan of operations $PL(t)$ to improve the performance of the DSSS [26].

In implementing this approach, the main challenge is to develop a formal procedure for testing the feasibility of a $PL(t)$ plan at different $\vec{x}(t) \in \left\{\vec{X}(t)\right\}$ values.

This would reduce the task (4) procedure to a periodic test of feasibility PL(t), including under unfavourable circumstances.

According to the terminology used in the development of operation research models and methods, an operation is understood as a system of action that shares a common vision and aims to achieve its objective [21]. If the plan is executed at a given point in time, the quality of the DSSS under consideration has been maximized, that is to say, the task has been completed. Otherwise, it is necessary to identify the obstacles to PL(t) implementation and also to develop and perform a set of measures to address them.

To implement this heuristic approach to solving problem (4), it is necessary to develop models and algorithms for checking the feasibility of the plan of operations based on the use of:

apparatus of Boolean functions;

productive models;

models and methods of system dynamics.

In the event that all of the above-mentioned stages of verification at the established sites confirm the feasibility of the plan of operations, PL(t) shall be considered feasible and, for each operation of the plan, the values of the optimal set of control effects $\vec{u}(t)^*$ shall be determined, which need to be realized for its implementation.

If, at one stage or another of the verification, the plan of operations $PL(t)$ turns out to be impossible, then the reasons for the impossibility of its execution are determined and the control actions are specified, the implementation of which will allow $PL(t)$ to be executed.

If under the circumstances it is not possible to ensure the feasibility of a PL(t) plan, task (4) has no solution, and optimizing its efficiency criterion requires the development of another solution.

The implementation of individual elements PL(t) is proposed to be represented by an event tree G_{et}, a connected acyclic graph $G[u(t), e(t)]$ of whose vertices $u(t) \in \{U(t)\}$ correspond to the individual operations of plan $G[u(t), e(t)]$, and an edge of graph $e(t) \in \{E(t)\}$ characterizes the hierarchical cause effect relationship given on the set of its vertices $(\{U(t)\}, \{E(t)\})$ are the set of vertices and edges of $G[u(t), e(t)]$ respectively.

As events, individual operations of the PL(t) plan are considered, which $\forall t \in \text{const}, t \in [t_s, t_f]$ may or may not be executed (partial execution is not taken into account). In essence, graph $G[u(t), e(t)]$ is a formalized description of the plan of operations $PL(t)$.

Constructing an operation plan to improve the quality of software functioning is a rather complex and time-consuming task, the solution of which must be carried out for each DSSS, taking into account the peculiarities of its functioning, operating experience, and expert opinion. In this case, it is advisable to use known models and methods of management of programme projects to increase their efficiency and reduce the cost of implementation [27].

4. RESULTS AND DISCUSSIONS

On the basis of the above, a modified heuristic procedure for constructing the PL(t) plan based on the requirements of ISO / IEC 9126 and ISO / IEC 14598 standards is proposed. It consists of the following main steps.

1) A graph of the hierarchical cause effect relationships G_{cl} between the individual elements of the DSSS is constructed, for which the assigned task, the structure of which is given in (Figure 2) [28] is performed. The main software modules of the given system $N_i, i = \overline{1, m}$, which influence the quality of its functioning are defined.

2) For each programme module $N_i, i = \overline{1, m}$, according to [23, 24], the characteristics, sub-characteristics and properties (indicators) of the quality of its performance shall be selected. As a result, a graph of hierarchical cause effect relationships between the characteristics and performance indicators of program modules $N_i, i = \overline{1, m}$ is formed. To construct this graph, it is proposed to use positions in graph theory using a rank approach where the characteristics, sub-characteristics, and properties have corresponding rank. The basic principles for the use of rank approach in graph construction are described in [29].

Graph G_{cl} characterizes the complex of cause effect relationships that exist between the quality indicators of $N_i, i = \overline{1, m}$ software modules and the most common types of DSSS errors that affect the quality of its operation. The most common types of software errors as well as their impact on the quality indicators of the software modules are discussed in detail in [30, 31].

From the point of view of the specificity of the tasks to be addressed by DSSS, it is necessary to supplement the general list of errors with specific types of errors inherent in the DSSS.

The event tree graph is defined by merging the graphs G_1 and $G_m : G_{cl} = G_1 \cup G_2 \cup ... \cup G_m$, the fragment of the graph is given in Figure 3.

The edges of this graph are conjunctive edges.

A formalized description of the vertices of the fragment of the graph shown in Figure 3 (events) is given in Table 1.

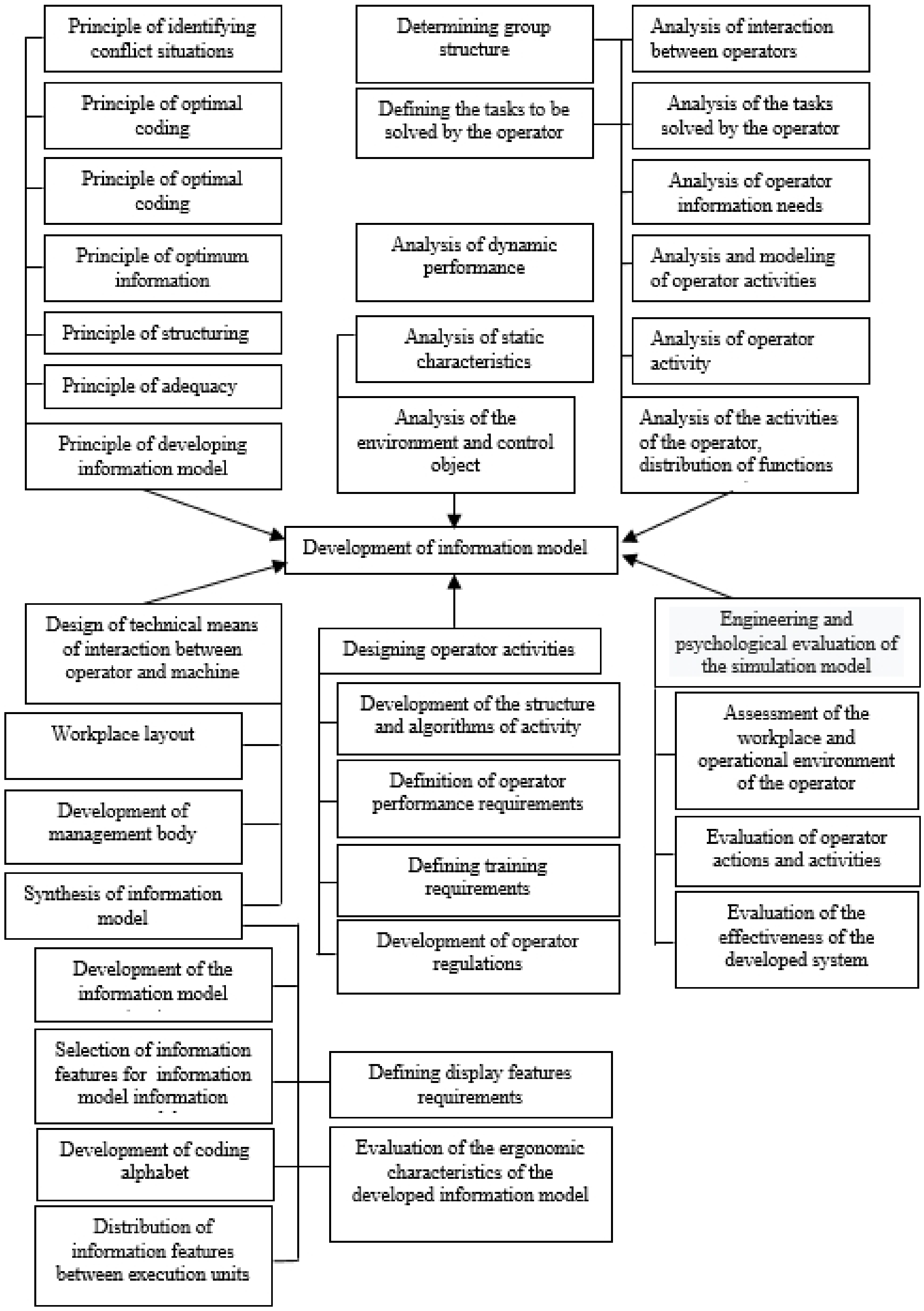
Principle of identifying conflict situations
Principle of optimal coding
Principle of optimal coding
Principle of optimum information
Principle of structuring
Principle of adequacy
Principle of developing information model
Determining group structure
Defining the tasks to be solved by the operator
Analysis of dynamic performance
Analysis of static characteristics
Analysis of the environment and control object
Analysis of interaction between operators
Analysis of the tasks solved by the operator
Analysis of operator information needs
Analysis and modeling of operator activities
Analysis of operator activity
Analysis of the activities of the operator, distribution of functions
Development of information model
Design of technical means of interaction between operator and machine
Workplace layout
Development of management body
Synthesis of information model
Development of the information model
Selection of information features for information model information
Development of coding alphabet
Distribution of information features between execution units
Designing operator activities
Development of the structure and algorithms of activity
Definition of operator performance requirements
Defining training requirements
Development of operator regulations
Defining display features requirements
Evaluation of the ergonomic characteristics of the developed information model
Engineering and psychological evaluation of the simulation model
Assessment of the workplace and operational environment of the operator
Evaluation of operator actions and activities
Evaluation of the effectiveness of the developed system

Figure 2: List of tasks to be carried out in the development of the information support system for decision-making

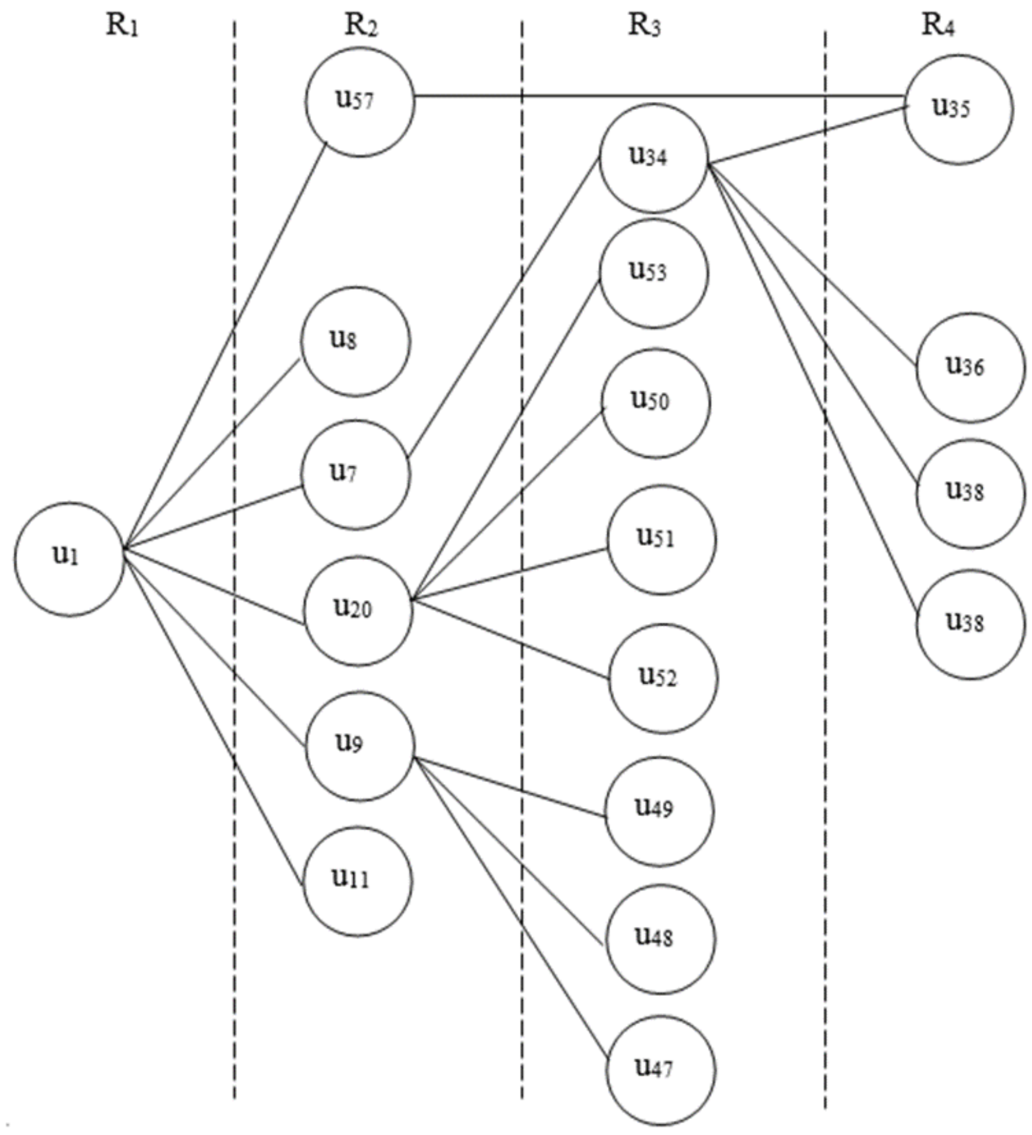

Figure 3: Fragment of an event tree graph G_{cl}

The constructed event tree G_{cl} is the basis for the development of models and algorithms to quantify the ability to execute a plan of PL(t) operations while searching for a solution to the problem (4).

5. CONCLUSIONS

On the basis of an analysis of existing approaches to the task of improving the performance of the man machine system, it has been determined that the most effective solution would be to consider it as a vector optimization problem with limitations in the form of equality and inequality. It has been established that it is necessary to develop mathematical models, numerical methods and algorithms that allow to optimize the target function characterizing the quality of the DSSS at different time intervals. On the basis of the formal objective of improving the quality of the DSSS of the systems under unfavourable circumstances, it has been established that it belongs to the class of variable calculus for conditional extremes. A heuristic approach to solving this problem is proposed and justified. It has been shown that the task of improving the quality of DSSS can be reduced to the task of synthesizing a complex plan of operations for improving the quality of software and analysing its feasibility under different control influences and environmental conditions. The proposed approach to the synthesis of a complex plan of operations for improving the quality of software in the form of an event tree is based on the positions of the apparatus of graph theory, in particular the rank approach.

Table 1: Formalised description of the vertices of an event tree graph

Number of vertices	Process
u_1	Compatibility option
u_7	Adequacy
u_8	Verity
u_9	Capacity for interaction
u_{10}	Security
u_{11}	System consistency as a whole
u_{20}	Consistency of modules
u_{34}	Failure to perform functions specified in the requirement specification
u_{35}	Inconsistency of programme documentation to specifications
u_{36}	Deficiencies in documentation
u_{37}	Lack of software delivery package files
u_{46}	Errors in interaction with telecommunication information exchange software
u_{47}	Errors in the use of data from other software tools
u_{48}	Incompatible data and file formats
u_{49}	Deficiencies in access management
u_{50}	Deficiencies in information management tools and records
u_{51}	Deficiencies in the protection of information against distortions
u_{52}	Deficiencies in the protection of information against distortions
u_{53}	Non-compliance with the requirements of the used cryptographic tools
u_{57}	Reliability errors

6. ACKNOWLEDGEMENT.

This work was supported by the Ivan KozhedubKharkiv National Air Force University(Kharkiv, Ukraine).

REFERENCES

1. D. Besnard, C.Gacek, and B. Jones,**Cliff Structure for Dependability: Computer Based Systems from Interdisciplinary Perspective**, SpringelVerlag London Limited, 2006, 304 p.
2. **Computer Safety, Reliability and Security**,*Proc. 24th Intern. Conf. SAFECOMP 2005*, Friedrikstadt, Norway, September 28-30, 2005, 409 p.
3. E.S. Bakhmach, A.D. Gerasimenko, and V.A. Golovir,**Fail-safe information and control systems based on programmable logic**, *National Aerospace University "KhAI", Research and Production Enterprise "Radiy"*, 2008, 380 p.
4. J.C. Laprie,**Dependability Handbook**,*LAAS Report № 98-346,Toulouse: Laboratory for Dependability Engineering*, 1998, 365 p.
5. S. Herasimov, M. Pavlenko, E. Roshchupkin, M. Lytvynenko, O. Pukhovyi, and A. Salii, **Aircraft flight route search method with the use of cellular automata**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, is. 4, 2020, p.p. 5077-5082, https://doi.org/10.30534/ijatcse/2020/129942020.
6. V.V. Lipaev,**Reliability and functional safety of real-time software complexes**,Moscow, 2013, p. 207.
7. A. Narinyani, and T. Yakhno,**Production systems. Knowledge representation in human-machine and robotic systems**,*Computing Center of the Academy of Sciences of the USSR*, Moscow, 1984,p.p. 136-177.
8. L.A. Zadeh,**Fuzzy Sets**,*Information and Control*,vol. 8,1965,p.p. 338-353.
9. L.A. Zadeh,**The Concept Of A Linguistic Variable And Its Application To Approximate Reasoning**,*Information Sciences*,vol. 8,1975, p.p. 199-249; vol. 9, p.p. 43-80.
10. M.M. Gupta, R.K. Ragade, and R.R. Yager,**An Approach To Fuzzy Reasoning Method**,*Advances In Fuzzy Set Theory And Applications*, 1979, p.p. 137-149.
11. S.P. Miller, M.W. Whalen, and D.D. Cofer, **Software model checking takes off**,*Commun. ACM*,№ 53(2), 2010, p.p. 58-64.
12. T. Nguen, W. Perkins, T. Laffey, and W. Pecora, **Checking Expert System Knowledge Bases for consistency and completeness**,*Proc. of the 9th Int. Joint Conf.on AI*, LosAng., 1985,p.p. 375-378.
13. S. Herasimov, V. Pavlii, O. Tymoshchuk, M.Yu. Yakovlev, D.Ye. Khaustov, Ye. Ryzhov, L. Sakovych, and Yu.A. Nastishin, **Testing Signals for Electronics: Criteria for Synthesis**, *Journal of Electronic Testing*, vol. 35, is. 148, 2019, p.p. 1-9, https://doi.org/10.1007/s10836-019-05798-9.
14. S. Herasimov, E. Roshchupkin, V. Kutsenko, S. Riazantsev, and Yu. Nastishin, **Statistical analysis of harmonic signals for testing of Electronic Devices**, *International Journal of Emerging Trends in Engineering Research*, vol.8, is. 7, 2020, p.p. 3791-3798, https://doi.org/10.30534/ijeter/2020/143872020.

15. V.V. Lipaev, **Problems of quality assurance of complex systems** [Electronic resource], Access mode: http://quality.eup.ru/MATERIALY4/poksps.htm.
16. M.A. Pavlenko, H.S. Stepanov, M.V. Kasyanenko, and V.N Rudenko, **A method of forming features of an information model of conflict situations for decision support subsystems in advanced control systems for special purposes**,*Collection of Research Papers of Kharkiv National Air Force University*,№ 3 (48), 2016, p.p. 101-103.
17. S. Herasimov, Y. Belevshchuk, I. Ryapolov, O. Tymochko, M. Pavlenko, O. Dmitriiev, M. Zhyvytskyi, and N. Goncharenko, **Characteristics of radiolocation scattering of the SU-25T attack aircraft model at different wavelength ranges, Information and controlling systems**, *Eastern-European Journal of Enterprise Technologies*, № 6/9 (96), 2018, p.p. 22-29, https://doi.org/10.15587/1729-4061.2018.152740.
18. S. Herasimov, Y.Belevshchuk, I.Ryapolov, A. Volkov,M. Borysenko, and O.Tokar, **Modeling technology of radar scattering of the fourth generation EF-2000 Typhoon multipurpose aircraft model**, *International Journal of Emerging Trends in Engineering Research*, vol.8, is. 9, 2020, p.p. 5075-5082, https://doi.org/10.30534/ijeter/2020/30892020.
19. A.V. Ryzhkov, N.A. Gerashchenko, R.D. Barvinok, and F.V. Ermolenko, **Using SQuaRE Standards in Software Testing in the Armed Forces of Ukraine**, *Modern information technologies in the field of security and defense*, № 1 (31), 2018, p.p. 89 - 94.
20. S. Herasimov, O. Tymochko, O. Kolomiitsev, G. Aloshin, O. Kriukov, O. Morozov, and V. Aleksiyev, **Formation Analysis Of Multi-Frequency Signals Of Laser Information Measuring System**, *EUREKA: Physics and Engineering*, vol. 5, 2019, p.p. 19-28, https://doi.org/10.21303/2461-4262.2019.00984.
21. H. Tatha, **Introduction to Operations Research**, Moscow, Williams, 2001, 912 p.
22. S. Herasimov, Y. Kozhushko, E. Roshchupkin, V. Dekadin, V. Djus, and Y. Melenti, **Evaluation of surface profile of holographic diffraction reflective coatings on scattering chart using in laser alarm systems**, *International Journal of Emerging Trends in Engineering Research*, vol.8, is. 8, 2020, p.p. 4502-4507, https://doi.org/10.30534/ijeter/2020/74882020.
23. **ISO/IEC 9126**,*Software Engineering. Product quality*, Kyiv, 2014, 20 p.
24. **ISO / IEC 14598**,*Information Technology. Software product evaluation*, Kyiv, 1999, 18 p.
25. Y. Kozhushko, D. Karlov, O. Klimishen, M. Bortsova, S. Herasimov, O.Hrichanuk, and V. Bykov, **Comparison of the Efficiency of Some Images Superposition Algorithms Used in Aircraft Map-Matching Navigation Systems**, *2018 IEEE International Conference on Mathematical Methods in Electromagnetic Theory*, 2018, p.p. 282-285.
26. O.N. Dolinin,**Algorithms and methods of development and debugging of expert systems**, Saratov,Sarat. state tech. un-t, 2015, 226 p.
27. R.T. Fatrell, D.F. Shafer, and L.I. Best man,**Management of software projects: achieving optimal quality at a minimum cost**, Moscow, Williams, 2003, 1136 p.
28. M.A. Pavlenko, V.K.Medvedev, P. Berdnik, and S.V. Mikhasov,**Cognitive approach to the development of information models in decision support systems**,*Science and technology of the Air Forces of the Armed Forces of Ukraine*, № 2 (23), 2016, p.p. 138-141.
29. S.V. Listrova, E.S. Listrovaya, and M.S. Kurtsev,**Ranked approach to solving linear and nonlinear Boolean programming problems for planning and control in distributed computing systems**,*Electronic Modeling*, №. 1, 2017,p.p. 19-38.
30. G.Myers,**Software quality**, Moscow, World, 1980, 360 p.
31. A. Alimpiev, P. Berdnik, N. Korolyuk, O. Korshets, and M. Pavlenko, **Selecting a model of unmanned aerial vehicle to accept it for military purposes with regard to expert data**, *Eastern-European Journal of Enterprise Technologies*, № 1/9 (85), 2017, p.p. 53-60, https://doi.org/10.15587/1729-4061.2017.93179.