Linguistic Cause and Effect Model of Functioning of Formation of Functional Refusal

N. R. Yusupbekov¹, Sh. M. Gulyamov², B. M. Temerbekova³, D. A. Mirzayev⁴
¹,²,³,⁴Department of Electronic and Automatic, Tashkent State Technical University
Address: Universitetskaya st. 2, 100095, Tashkent City, Republic of Uzbekistan

Abstract—The logical-linguistic model of process of functioning of the software in the mode of education functional in which is considered the structure of model of process, expediency of its representation in the form of indistinct display of a set of functional refusals to a set of entrance sets is considered. The software of the computing systems which are a product of intellectual activity of the person in which the illegibility in statement and the formulation of the main objectives, criteria and restrictions, and also impossibility of application is inherent in the analysis of absolute estimates is offered.

Keywords — computer systems, software, complex of technical means, execution of programs, fuzzy mathematics, input-output, fuzzy arithmetic, fuzzy automaton, algorithm for evaluating the reliability, probability.

I. INTRODUCTION

Questions to improve the performance of computer systems (CS) is currently very topical. In this context, the reliability of software (SW) as one of the components of quality of functioning of particular importance. Sun includes a complex of technical means (CTM) and software. Of greatest interest is the reliability of the whole system, but given that the physical nature of the CTS and the software is different, it should be assumed that a unified evaluation of reliability of functioning of the entire system can be formulated with regard to a number of restrictions.

Reliability CTM determined by conventional methods of analysis used in the theory - reliability. Reliable operation of the software, as will be shown below, it is advisable to evaluate the methods based on the ideas of fuzzy logic. Software as a product of human intellectual activity, has all the attributes that are inherent in this type of activity: blurred productions major challenges not formalized criteria and constraints, the need to make a decision in the case of insufficient (or unreliable) source of information, etc. On the other hand, as a product for industrial use, the software must possess all the attributes of this product: the metric for comparing performance, etc.

It should be noted the dual nature of the term “reliability” as applied to software. If the concept of failure in the technical means of uniquely formulated, the physical reasons for the "technical failure" clear and usually unambiguous, then the concept of "software failure" is to some extent subjective. For example, the flashing of the display screen from one air traffic controller is considered functional failure, but another manager in this situation is working properly. In addition, if it is possible to eliminate the situation in which the function is performed by software, the failure of the software is not fixed and the reason is not detected. Thus, as software upgrades during its accompanying software failure concept is modified.

Fig. 1 Area of input vectors: A region of IV, in which the software has been tested; In - IV region that were specified in the TOR; C - area real input vectors (IV); D - area real IV included in the TOR, in which the software was tested.
Difficulties in trying to use of traditional theory to software reliability due to the fact that in the real world functional Sun failures because the software is relatively rare, so reliable statistics on the basis of which can be made objective conclusions can be made only after many millions of hours of sun, and although the significance of these failures is extremely high, to apply their analysis and probabilistic methods impractical, furthermore, these estimates may give an erroneous result basically. Nevertheless, the assessment of the reliability of the software (fault tolerance) so far performed using indicators that are probabilistic nature [1, 2].

For example, in [1] that the probabilistic characteristics activities as operating time to first failure in the execution of programs (EPR) in some forms can be considered as the criteria of reliability programs. It is necessary to further knowledge about the operation of the software after the failure in the recovery process.

Consider Fig. 1, contained in [1]. As can be seen, a plurality of input vectors (IV) can be expressed in four subsets. It is easy to see that if in the case of "a" failure in the execution of the program takes place at BH belonging to a subset of D, and in the case of "b" in LN of a subset of C, then the estimates of reliability can not be compared. In the first case described by BH subjective (linguistic) probability (very likely), the second (not very likely) or (ALMOST INCREDIBLY).

From the practice of the software to function real-time sun can cause a lot of examples where a single system, which had, for example, "in a time T ten functional failures, the software functioned objectively better than the other system, which had in the same time, say, five functional failures . It depends on what the IV were in the first and second cases.

Reliability analysis software is currently based solely on theoretical concepts inherent reliability of the vehicle. This is due to the fact that among the entire set of functional failure (FF) has a rather large subset of those that appear "hardware", ie. E. Can be presented clearly. For such refusals traditional reliability assessment seems justified, although in this case insufficient statistics significantly shows- However, it should warn against ignoring the "ambiguous" FF, which though are much smaller (compared to the "hardware") subsets but nevertheless can be fatal. Oc-new features using the classical notion of security with respect to software:

- Fundamentally different physical nature of the software reliability and TM in the software “aging” fundamentally missing, so the "time to failure" is used as the main and essentially only indicator - reliability software requires a separate study;
- Important concepts FF subjectivism leads to dialectical contradiction - it is necessary to collect "objective" statistics comparably subjective factor;
- Inability to obtain exact distributions of probabilistic parameters which determine the nature and quality of the computational process in the formation of functional failure.

When analyzing and assessing the reliability SW of the software there is a significant need to develop new concepts and new principles of analysis. As is known, the formulation of new principles and new concepts possible if additional information on the process of the software to function SW during the formation of FF. The software is so complex product production and technical purposes, the quality of its operation can not be described by only one scalar- indicator of what is, for example, the time to failure. Obviously, the challenge is to synthesize a vector indicator of quality such that in particular the TM FF would be one component of the index.

Provides an integrated measure of quality of functioning, software that takes into account the causal relationships that exist in the software during the formation of FF, and allows to have a traditional assessment in the form of SW CS, state aircraft showing quality of functions performed, it is suggested to characterize the tuple (XI, X4, k, T), X4, XI- fuzzy finite sets functional failures and input vectors, respectively ranging values of the membership functions (boost factors k, - a function of XI, X4 (generally fuzzy) T - the time during which the identified XI and X4 - scan interval entry and exit.

Separately, we must examine the definition.

Given that k, rationally define as an arithmetical operation on X4, XI, which, in turn, being the ranked values of the membership functions are fuzzy variables (number), or it is necessary to apply the existing methods and algorithms fuzzy mathematics (FM) or develop new ones. The desirability of developing new algorithms FM (or rather, on new principles) is due to the fact that the existing algorithms have several drawbacks and, in addition, not enough constructive. We point out some shortcomings of existing algorithms:
- A high degree of uncertainty of the result, in some cases exceeding the maximum uncertainty component of the operation; when multiple transformations result virtually "disappear", i.e., impossible to decide which subset of the plurality considered decision;  
- "Diversity" of operations of the same type, such as the operation of the "simple" and "additional" division, subtraction, and others, without rational applications significantly increases not determine the result.

An algorithm is proposed for operations FM, based on the so-called "matrix principle." Both components of the binary operation represented as a row vector and a column vector, respectively, and the operation is carried out "line on the column" membership functions are selected by the maximin principle is implemented as a multi-step procedure. At each step of the procedure using heuristic rules "specified" result is performed "truncation" FM-dimensional component of the operation result, and thus the uncertainty of results when multiple transformations practically does not increase. Decision in this case can be greatly formalized.

Structural work includes an introduction, five chapters and Appendix. * In the introduction the reasons which led to the search for a new indicator of the quality of the software to function, the possible ways and conditions of the synthesis of such a measure, the need to construct an algorithm of fuzzy arithmetic (FA), which has compared to traditional higher constructive.

The second section discusses the logical-linguistic model of the functioning of the software in the functional mode of education which considers the structure of the process model, the appropriateness of its presentation in the form of fuzzy mapping of a set of functional failures in the set of input vectors.

In the third section, in order to determine the type of process mapping software to function in a mode of education FF represented as a process of functioning of fuzzy automaton (NSES).

In the fourth section, based on the results, obtained in the third section (representation of the process software to function as a NSES), an algorithm for evaluating the reliability of the software by the ratio of "total number of states - the number of functional failures" as the implementation of the algorithm of fuzzy mathematics.

The fifth section provides an example of evaluating the reliability of the software based on Sun Bank data of the real system.

The annex contains a fragment of a data bank, which collects the state of the system from the inputs and outputs, complemented by system messages, which can significantly reduce the arbitrariness in the appointment of the membership functions.

II. EXPOSITION

In [3] proposed a model of the software to function in the formation of functional failure (note that the term "functional failure" to more fully reflect the refusal because of the software, as a consequence of failure is the partial or complete failure to fulfill the functions assigned to the software). Proposed in [3] model is a scenario that simulates the operation of the software and the process based on the use of relations, more objectively and adequately (in comparison with the probability metric) reflecting the process. Scenario takes into account the causal relationships that exist in the software in the formation of functional failure, and has the form:

X1: custom input set (CIS) => X2: fault situations (FS) => X3: job means to prevent the transition fault situations in a functional failure (FF) => X4: a functional failure or normal operation - fig. 2.

Explain Fig. 2 in more detail. XI => X4 little means you can move the input set in a functional failure; XI => A2 => X4: input set => fault situations => functional failure: X1 => A2 => A3 => X4 - Inward Dial => fault situations => work means compensation transition fault situations in functional failure => functional failure (if the work means compensation were unsuccessful).

The possibility of developing a computational process for some branches can be described by a system of statements using linguistic, subjective probabilities or clear, for example, if VL is a class (probably), the possibility (probability) that the process will go on branches XI => X4 (LOW). If the terms (probably) (LOW) can be replaced with clear numbers, it is possible to come to the traditional statistical conclusions. Unfortunately, an unambiguous definition of these and similar terms are not feasible, although in some cases this fact to ignore, the law of probability distribution is assigned (as rigorously prove the applicability of a law distribution is virtually impossible), and of these conditions obtain reliable (?) statistical inference.
The advantage of the proposed scheme (scenario) is that it allows you to get more information, consisting in the account of linguistic probabilities (frequencies) of the input set, fault situations, operation of the means of compensation, functional failures. Studies show that even the process of building a model of the "input - input" - "input set \( \Rightarrow \) functional failure" makes it possible to obtain a more objective model of software reliability compared to models that take into account only the failures without identifying their causes.

Functioning quality Pts based on a scenario described by a system of statements, for example: (if the frequency of the IV set B (see Fig. 1) IV = high frequency and functional failures = LESS IV, the quality of the functioning of the high P). To describe the operation of the software based on the script used fuzzy type approval (frequency X1 = MALA, GREAT, CLOSE TO ...), fuzzy relations \( R_j \) between the elements of the sets XI, X2, X3, X4, for example, (the absolute difference frequencies in NVN j th type and FF type "F" is small, large), and so on.

Fuzzy approval and fuzzy relations in the general case are presented in the form of fuzzy sets:

\[
\omega_{ij} = \frac{\omega_{ij}^{(i)}}{\mu_{\omega_{ij}}^{(i)}}, \quad j = 1, 2, \ldots,
\]

Where \( \omega_{ij}^{(i)} \) - the possible value of the parameter belonging to, for example, the class frequencies described by fuzzy statement (= small);

\[
\mu_{\omega_{ij}}^{(i)} \quad \text{Membership function, which determines the degree of probability (subjective, linguistic) or opportunities to which the value relates to the class.}
\]

\[
\mu_{x_i}^{(i)}(x) \rightarrow [0,1]
\]

It should be noted that the possibility of assigning the frequency of events (eg, FF) to a particular class of frequencies can increase the information content of the measurement, since as additional information can be viewed all the properties held by the class [4].

Note that a similar approach (use of statements to evaluate the software) used in [5] for a description of the two systems ((BEST) system with (high) quality and (high) maintainability having software e (high enough) initial quality and other characteristics of the medium, has a number of system failures: 1st year - 200, 2nd year - 40, 3rd year - 20, and so on. D.) {Worst} system with (low) quality ... has a number of system failures: 1st year - 400, Eu-50, 3rd year 40, and so on. D.

Based on the scenario, you can formulate a system of statements for software that determine the quality of operation, is ranked high, medium, low) (or any other aggregate). By taking these (or any other) state as standards on the basis of the principle of reference [6, 7] to determine the system preferences on the basis of which to assess the reliability of the components of the quality of the software to function. Application of the principle of reference to the evaluation of the quality of the functioning of the software discussed in [7].

The process of the software to function in accordance with the cause and therefore the model will be considered as a mapping of the set IV FF, t. E. F: \( X_I \Rightarrow X_4 \). IV ranked membership function from a position of assignment to a particular class of probabilistic linguistic terms are written off (very likely, likely, not very likely, almost unbelievable), and so on. D. Similarly, we can rank the set of FF, selecting a subset of FF, which took place at the (very likely IV), (probably) etc. Accordingly \( X_I \) and \( X_4 \) have the form

\[
X_1 = \bigcup_i x_1^{(i)} / \mu_{x_1}^{(i)}, \quad j = 1, 2, \ldots
\]

\[
X_4 = \bigcup_i x_4^{(i)} / \mu_{x_4}^{(i)}, \quad j = 1, 2, \ldots
\]

Since the values of fuzzy or linguistic variables (LV) \( x_1^{(i)} \in X_1, x_4^{(i)} \in X_4 \), fuzzy subsets correspond to the membership functions \( \mu_{x_1}^{(i)} \in F(X_1), \mu_{x_4}^{(i)} \in F(X_4) \), then the mapping F: \( F(X_1) \Rightarrow F(X_4) \) can be generally assumed fuzzy. It can be obtained as a fuzzy match for all functional failures and input vectors:

\[
\hat{O} = \bigcup_{j=1}^{J} \mu_{x_1}^{(j)} \times \mu_{x_4}^{(j)}
\]

\[
\mu_{x_1}^{(j)} \in F(X_1), \mu_{x_4}^{(j)} \in F(X_4)
\]

Thus, we can pose the problem of determining (derivability) fuzzy vector values \( x_4^{(j)} \in X_4 \) with a new set of fuzzy vector values \( x_1^{(j)} \in X_1, j = 1, 2, \ldots, \) eg \( \mu_{x_4}^{(j)} = \mu_{x_1}^{(j)} \otimes \hat{O} \iff X_4 = X_1 \ast \hat{O} \). These issues are discussed in [7-10, 11].
B [11] the possibility of constructing a model of a fuzzy system based on logical structure. This model includes the definition of the system with some operator F * such that:

a) H4i- H1i * = F for a given pair of "input-output" X1i ⇒ H4i;

b) the logical structure defined by, for example, models; ML - if X1 (large); that X4 (the Great); M2 if X1 (the Great), the X4 (MAPA). Applicability of this approach to solving problems is limited essentially by the fact that for the fuzzy output (if X1 *, then the X4 *) must prove that X1 * is (between) and H1j H1j + 1 (for MI or M2). This sluchae, defining H4j X1 = j * F and H4j +1 = X1 j + 1 * F, it can be argued that the X4 * is (between), and X4 H4j j +1 and fairly X4 = X1 ** F. In this paper we study the rational construction of the mapping F in order to develop an algorithm based on F evaluating the reliability of software.

The process of the software to function in the formation of FF can be regarded as the achievement of such a state (under the influence of IV and internal reasons), which does not run the scheduled function, manifested residual errors, and other factors that lead to the output state classified as a functional failure. V denied the concept of "state ID" naturally leads to the problem to try to formalize it and, if possible, obtain the corresponding equation of state. Leaving aside the question of how can be fixed these states, we note that the assumption that the number of "I" (input states) can be mapped to the number of "O" (output states), is justified. Practice confirms the legitimacy of such assumptions.

To formalize the subsequent arguments, we note that the statements of the type (probability = HIGH, MEDIUM, LOW), etc. compared probability scale [0, 1], adjusted for the specifics domain.

Displaying [1, 2] that a functional failure is due, as a rule, non-standard set (t. E. Not included in the test set IV) j-th type. In real-time systems, such as automated air traffic control systems, the input sets of continuous flow, identify where the ends set of j-type, belonging to, for example, a plurality of sets in which the software tested, and starts dialing the j + 1-type, in which the software is not tested, it is practically impossible.

Previously noted the feasibility of the model "input-output". In our case, the model of the "input-output" includes a definition of finite fuzzy sets of input and output states, the output state is determined by the implementation of a basic (or more) of the functions of the system.

If a plurality of inputs:

\[ X_i = \bigcup_j x^i_j / \mu_{x^i_j}, j = 1,2,..., \]

where \( x^i_j = \{x^i_{1j}, i = 1, n; x^i_j \in X1 - - IV \) component \( \mu_{x^i_1} \rightarrow [0,1] \). then the set of output states must have identical structure, ie. where \( \bigcup_j x^4_j / \mu_{x^4_j}, j = 1, 2,..., \) where

\[ x^4_j = \{x^4_{1j}, i = 1, m; \mu_{x^4_{1j}}(x) \rightarrow [0,1] \}, x^4_j \in X4 - - \)

a component output. This means that the components of input and output are determined synchronously at time instants \( t_j \in X4; \) we have a set of pairs.

\[
\begin{align*}
X1^1 & \Rightarrow X4^1 \\
X1^2 & \Rightarrow X4^2 \\
& \ldots \ldots \\
X1^n & \Rightarrow X4^n
\end{align*}
\]

Continuous sets of inputs and outputs are presented in the form of finite fuzzy sets in which the membership functions and the mean probability (subjective, linguistic, etc.) IV assignment to a particular class and the frequency of the output state - to a class of full, partial or FF good state and others.

One method of sampling and IV output states is that an interval scan inputs and outputs, and to obtain information about the frequency class, which may belong to the IV and output state. The value should be determined rate of change of IV. Questions sampling a continuous set of input vectors to determine the class of linguistic (subjective) probabilities are wealthy value for solving such a problem has not yet proposed a satisfactory algorithms. One can assume that if the IV consists of several components, the update rate of IV should be taken not less than the update rate of the fastest components.

If the changes are very "fast" components of IV, over time the number of states \( T^X_p \) software that could potentially carry out a functional failure, ie number of pairs X1 X4 does not exceed \( mx = T^X_p / \Delta_T \). Consequently, the number of components of fuzzy finite sets X1 and X4 power \( mx \).

Can specialize the input sets, attributing all IV, in which the software has been tested to the standard class (SK), the rest of IV, which can potentially occur naturally related to non-standard (NS).
International Journal of Emerging Technology and Advanced Engineering  

So SVN rationally describe linguistic probabilities (always, very often, very probably often) (FEW probably not often, rarely, almost unbelievable, very rarely, etc.).

In this case $\mu_{X^4}^f(x)$ Membership function reflects the "significance" of IV in terms of its influence on the formation of a functional failure; IV belonging to the class described by the probability (RARE) affects the formation of functional failure is stronger than IV in the class frequency (probability) (ALWAYS), i.e.

$$\mu_{\text{Seldom}}(x) \geq \mu_{\text{Always}}(x).$$

In a series of tests determined the number and type of input vectors from the set $X1$, the number of output states qualified as FF,

$$\hat{0}_4 = \bigcup_j x^j/\mu_{X^4}, j = 1,2,....$$

Where $\hat{b} / j$ - the number of functional failures $\mu_{X^4}^f$ - "Importance" of the denial of $j$-type, determines the value of AF $\mu_{X^4}^f(x) \rightarrow [0, 1]$.

The number of tests (runs), you can restrict the output yield a state in which you want to reboot the system. Thus, the term "time to failure" in this case can be replaced by the concept of "working hours" (the number of fault situations or functional failures, the importance of which $\mu(x) < 1$) on the condition that requires a system reboot - $\mu(x) = 1$ when applying for entrance standard and / or non-standard HV.

The process of the software to function in a mode of formation of the Armed Forces functional failure will be represented as the process of functioning of odd limited automaton [10], which is described as a tuple

$$A = \langle X1, S, X4, S_0, \delta, \delta_1 \rangle,$$

Where $X1 = \bigcup_j x^j/\mu_{X^4}, j = 1,2,....$ - a finite set of fuzzy inputs (input sets, ranked value of the membership function, which determines the frequency of the set class, which owns BH);

$$S = \{ S_i, i = 1, m \} - \text{finite set of states (generally fuzzy)} \text{comprising normal and "reject" condition, potentially causing FD:}$$

$$X4 = \bigcup_j x^j/\mu_{X^4}, j = 1,2,.... - \text{a finite set of outputs (functional failures), each FF has its "importance" } - \mu_{X^4}^f(x) \rightarrow [0, 1]; \text{if } \mu_{X^4}^f(x) = 1, \text{the system reboots and goes to clear the initial (zero) condition; }$$

$$\delta: S \times X1 \times S \rightarrow L - \text{The transition function; }$$

$$\delta_1: S \times X4 \rightarrow L - \text{The output function; }$$

$S_0$ - initial state, generally $S_0 \in F(S), S_0: S \rightarrow L$.

Suppose that the set of input vectors - $X1$, the transition function is $\delta: S \times X1 \times S \rightarrow L$ determined from the expression

$$\delta_{SS}(\lambda) = \delta(S, \lambda, S') = \begin{cases} 1 & \text{when } S' = S, \\ 0 & \text{when } S' \neq S; \end{cases}$$

$$\delta_{SS}(\theta) = \delta(S, \theta, S') = \max \min \left[ \delta(S, x^{(1)}, S'), \delta(S, x^{(2)}, S_0), \ldots, \delta(S, x^{(f)}, S') \right],$$

$$\theta = \{ x^{(1)}, x^{(f)} \} \subset X1, \theta \neq \lambda.$$

If the transition matrix of the output at the $T_\theta = T_{x^{(1)}} \circ T_{x^{(2)}} \circ \ldots \circ T_{x^{(f)}}$, when the input sequence of $\theta$ is the form $X4 \Leftrightarrow \mu_A(\theta) = S_0 \circ T_\theta \circ \delta_1$. If we limit the analysis of the state, in which there is a single output Nav $X4 = \{ x^4 \}$ - functional failure, the fuzzy output is determined by the vector $\delta_2 = \{ \mu_{\delta_2}^j, j = 1, f \}$, where $\mu_{\delta_2}^j \not\in \delta_2$ which indicates the degree of possibility of obtaining FF (x4) from the state $S_j \in S, j = 1,2,....$, and respectively of $x^{(i)}$.

Can be even more specialized NSES defining with many final states $S_F \subset S..$ For such an automaton output function

$$\delta_1 = \{ \mu_{\delta_1}^j, j = 1, n; \mu_{\delta_1}^j = \begin{cases} 1 & \text{when } S_j \in F, \\ 0 & \text{when } S_j \not\in F. \end{cases}$$

Assuming that after a system reboot is required to return to the original zero (clear) state that $S_0 \in S$, i.e.

$$\delta_1^0 = \{ \tau, j = 1, n; \tau_j = \begin{cases} 1 & \text{when } S_j = S_0, \\ 0 & \text{when } S_j \not= S_0. \end{cases}$$

The values of the output function defined as the

$$X4(\theta) \Leftrightarrow \mu_{X^4}^f(x) = \max \delta_{SS}(\theta),$$

where $\mu_{X^4}^f(x) = \max \delta_{SS}(\theta).$
where \( \theta = \{x_1^{(1)}, x_1^{(2)}, ..., x_1^{(m)} \} \)

III. CONCLUSION

Analysis of this expression shows that the process of the software to function in a mode of formation of functional failure when introduced assumptions, do not violate the fundamental physics of the computational process, mainly determined by the function (matrix) of the transitions from state to state. So it makes sense to choose as an indicator of the reliability parameter based on the ratio of the total number of "distinguishable" states and the number of functional failures.

The software of computing systems is a product of intellectual activity of the person in which the illegibility in statement and the formulation of the main objectives, criteria and restrictions, and also impossibility of application is inherent in the analysis of absolute estimates.

When determining an indicator of reliability of functioning \( S \) it is necessary to consider the relationships of cause and effect existing in \( S \) at formation of functional refusal which formation of such state is expedient to treat \( S \) at which the functions assigned to \( VS \) are carried out partially. It is necessary to consider that procedure of definition and calculation of an indicator of reliability of functioning \( S \) is connected with incompleteness of information, unauthenticity of the applied mathematical models, a weak formalizovannost of the main objectives. The main sources of incompleteness of information are:
- the test set of entrance sets which isn't coinciding with the valid set of IV which take place in the conditions of real operation of CS;
- subjectivity of definition of the concept "functional refusal";
- unauthenticity and/or insufficiency of statistics concerning quantity like IV and FF.

S in the normal mode and at formation of functional refusal it is expedient to present functioning in the form of the indistinct limited machine gun.

In the assumption that NSES has the only exit - X4 functional refusal, a matrix of transitions 0 can be interpreted by \( \{X_4^*\} \) as the relation of an indistinct final set of entrances of X1 to an indistinct final set of exits of X4, and quality of functioning of the software can be presented in the train form \( (X_1, X_4, T) \).

REFERENCES