



Co-funded by the  
Erasmus+ Programme  
of the European Union

Міжнародна науково – методична конференція  
«Інжиніринг криз та ризиків транспортних послуг»

Січень 20 – 21, 2021



## **INFORMATION ANALYTICS OF SOLVE A PROBLEM OF SOFTWARE RELIABILITY EVALUATION FOR SOCIOTECHNICAL SYSTEMS AT THE CONCEPTUAL DESIGN STAGE**

*Kosolapov A., Ivin P.  
Dnipro National University of Railway Transport named after  
Academician V. Lazaryan*

*The scientific works proposes a method for evaluation the software reliability of a computer-based process control in real time system at the at the conceptual design stage. The proposed approach makes it possible to evaluate the possibilities of reserving transactions in real time.*

The current stage in the development of approaches to the development and application of computer systems is characterized by a change in the paradigm of computerization: from hardware-centric systems (obtaining maximum performance) to network-centric systems (to provide access to data "here and now") to human-centric systems based on an intelligent interface. Currently, there is a transition to socio-centric systems (socio-technical systems, STS), which are all types of support for automated measurement, control, management systems. STS is an automated system consisting of a technical subsystem, a subsystem of personnel and the external environment. They are represented in the STS architecture by seven types of support, including software. Although initially the idea of STS appeared in the late 1940 and early 1950s, but its practical implementation has become effective only in our time. STS moved to the class of big managed systems, uniting machine complexes and groups of people. It is believed that in a complex (big) system, structure is an essential parameter affecting its effectiveness. This is a dynamically changing "time-migrating" multistructure, functionally and territorially distributed to the "level of things". An important feature of STS is the work in real time (SNS RT). In design such systems, an important step is the stage of conceptual (architecture or early) design, when the main hardware and software solutions of the STS are formed. The complexity of design and research work of architectural stage lies in the reliability of the

source data for decision making. This also applies to the task of evaluating software reliability. The main features of an early assessment of software reliability are described. To solve this problem neural networks, fuzzy logic, Behavior Models are used. But these approaches are difficult to apply in the early stages of design, especially for real-time systems. Therefore, the work proposes a set of mathematical models to describe the operation of the STS, its software under real-time constraints and simple methods for calculating software reliability. We assume that STS RT performs a set of transactions  $R = \{r_i, i = \overline{1, N_r}\}$  on the initiative signals  $S = \{s_i, i = \overline{1, N_r}\}$  received in STS from sensors of ObjW and culminating in the issuance of control actions  $U = \{u_k, k = \overline{1, N_u}\}$  (signals) to actuators of automation and (or) messages to peripheral equipment to the operational personnel of the ObjW  $X_d = \{x_{dl}, l = \overline{1, N_d}\}$ . Example of Transaction  $r_i$  is shown in Fig. 1.

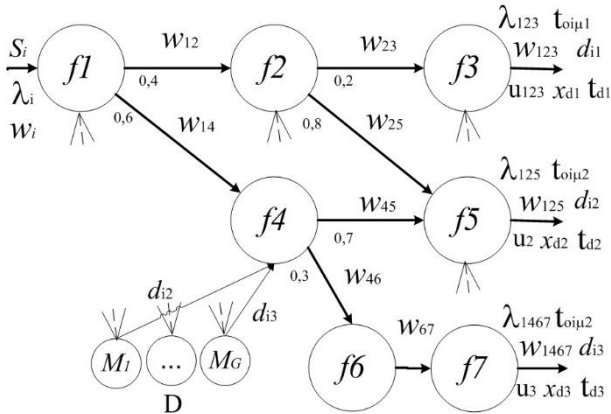


Figure 1 – A graph of the transaction  $r_i$

Let's consider the features of transactions.

1. A transaction in STS RT consists from an indivisible and non-recoverable sequence of tasks (functional-algorithmic and program blocks - FPB)  $F = \{f_j, j = \overline{1, N_f}\}$ .

2. FPBs are characterized by a set of parameters, which are presented in Table 1.

Table 1

| Initial characteristics of the FPB and database |   |                 |     |                     |   |
|---|---|-----------------|-----|---------------------|---|
| Command set                                     | Number of commands in FPB of type $\beta$ |                 |     |                     | Number of cycles and failure rate of commands $\alpha_\eta$ |
|   | $f1$                                      | $f2$            | ..  | $f_{N_f}$           |   |
| 1. mov  | k11                                       | k21             | ..  | $kN_f 1$            | 1 ( $\alpha_{\min}$ )                                       |
| 2. add  | k12                                       | k22             | ..  | $kN_f 2$            | 1 ( $\alpha_{\min}$ )                                       |
| 3. mult   | k13                                       | k23             | ..  | $kN_f 3$            | 18 ( $n3 \times \alpha_{\min}$ )                            |
| 4. div  | k14                                       | k24             | ..  | $kN_f 4$            | 24 ( $n4 \times \alpha_{\min}$ )                            |
| $\beta \dots$                                   | ...                                       | ...             | ... | ...                 | $n\beta$ ( $n\beta \times \alpha_{\min}$ ) ...              |
| $K_{com} \dots$                                 | k1<br>$K_{com}$                           | k2<br>$K_{com}$ | ... | $kN_f$<br>$K_{com}$ | $nK_{com} \times \alpha_{\min}$                             |
| Arrays of DB                                    | Utilization rate of the array             |                 |     |                     |   |
| M1, 256 Mb                                      | Y11                                       | Y21             | ... | $YN_f 1$            |   |
| ...   |   |                 |     |                     |   |
| M6, 8 Gb  | Y16                                       | Y26             | ... | $YN_f 6$            |   |

Each transaction has a time limit for its execution (deadline)

$$T_d = \{t_{dk}, k = \overline{1, N_u}\}.$$

4. The transaction is started at a certain intensity, which is determined by the dynamics of ObjW  $\Lambda = \{\lambda_i, i = \overline{1, N_r}\}$  and the transaction can be estimated execution time (processing) on the microprocessor  $\mu$  for all variants of its completion

$$T_{ou} = \{t_{oiuk}, k = \overline{1, N_u}\}.$$

5. Failure of any transaction (its interruption) leads to production losses  $W = \{w_i, i = \overline{1, N_r}\}$ , which are part of the total possible losses in the event of a system failure  $W = \sum_{i=1}^{N_r} w_i$

$$m_i = w_i / W$$

6. The completion of each transaction has a variability of losses for each output  $w_i = \{w_{ik}, k = \overline{1, N_u}\}$   $w_i = \sum_{k=1}^{N_u} w_{ik}$ . Values  $w_i$  are formed by experts and distributed to the outputs of the transaction  $m_{ik} = w_{ik} / w_i$ .

7. Each transaction uses information from the database of STS RT  $D = \sum_{g=1}^{G_m} M_g$ . When a transaction is executed, each of its FPBs  $F = \{f_j, j = \overline{1, N_f}\}$  accesses to arrays of database and the volume of data used for the  $k$ -th output of the  $i$ -th transaction is  $d_{ik}$  which is part of the total size  $D$ , that is  $d_{ik} / D$ .

#### *Method for estimating software reliability of STS RT*

The reliability of the application software of a STS RT will be evaluated by the probability of no-failure/trouble-free execution of each transaction  $i$  for each variant  $k$  its completion  $P_{ik}(t_{oik}) = 1 - \exp(-\alpha_{ik} \times t_{oik})$  (for the exponential distribution of time intervals between failures). Since this indicator is calculated during the "life" of the transaction, it corresponds to the availability factor  $K_{av}$  (the probability that the object will be in working condition at an arbitrary moment of time). The values  $t_{oik}$  are calculated for the selected microprocessor  $\mu$  with a clock frequency of  $\omega_\mu$ , for which the number of cycles executing a set of command  $K_{com}$  is known (see table 1), when  $t_{oik} = \sum_{\forall f \in r_{ik}} \sum_{\beta=1}^{K_{com}} \nu_{\beta ik} \times K_\beta \times \tau_\beta$  - average transaction processing time  $i$  for output  $k$ ;  $\tau_\beta = n\beta_\tau / \omega_\mu$  - average execution

time of command  $\beta$ ;  $\nu_{\beta ik}$  - branch coefficient in a transaction  $i$  (see Fig. 1 for  $f_1, f_2, f_4$ ).

At the next step, it is necessary to evaluate the failure rate of the FPB chain for each variant of its completion  $\alpha_{ik}$ . The basis is the failure rate of the shortest register-to-register command mov, performed in one clock cycle, which is equal to the failure rate of one chip  $\alpha_{\min}$ . The failure rate of other commands will be calculated according to the clock cycle scheme (Table 1) and are equal  $\alpha_1, \alpha_2, \dots, \alpha_\beta, \dots, \alpha_{K_{com}}$ . For each transaction output, it is necessary to calculate the number of commands of the specified types:

$\theta_{\beta ik} = \sum_{\forall f \in r_{ik}} \nu_{\beta ik} \times K_\beta$  A weighted arithmetic mean of the failure rate for the  $i$ -th transaction by the  $k$ -th output is calculated as

$$\alpha_{ik} = \sum_{\beta=1}^{K_{com}} \alpha_{\beta ik} \theta_{\beta ik} / \sum_{\beta=1}^{K_{com}} \theta_{\beta ik} .$$

Processing FPB in a transaction with database arrays reduces this indicator depending on the amount of data read  $d_{ik}$ . Let's introduce coefficient,  $k_{ik}^{db} = \delta \times (1 - d_{ik}/D)$ ,  $d_{ik} < D$   $\delta$  - coefficient of rationing. Thus, the reliability of the application software is possible to calculate as  $p_{ik}(t_{oik}) = k_{avik} = k_{ik}^{db} \times (1 - \exp(-\alpha_{ik} \times t_{oik}))$ .

As a rule, when designing a STS RT indicates the required values of the availability factors  $\tilde{k}_{avik}$ , at the same time, the condition  $k_{avik} \geq \tilde{k}_{avik}$  must be met. For the entire system (for a microprocessor

$\mu$ ), the following expression is true  $K_{av}^s = \prod_{\forall ik} k_{avik}$ .