

Real-time Sociotechnical Systems: Early Software Reliability Evaluation

Anatolii Kosolapov^{#1}, Pavlo Ivin^{*2}

^{#1,2}Computers Department, National University of Railway Transport named. acad. V. Lazaryan, Dnipro, Ukraine - 49005

Abstract — The article proposes a method for evaluation the software reliability of a computer-based process control in real time system at the early stages of design. Models for the description of automated objects, functioning processes and tasks to be solved are proposed. The concept of a transaction is introduced, a method for evaluation the probability of its smooth operation and availability. For the entire system, the average availability factor is calculated, and the average losses due to program failures during transaction processing. The proposed approach makes it possible to evaluate the possibilities of reserving transactions in real time. The proposed models and method are implemented in the socio-technical real-time systems conceptual design framework CoDeCS.

Keywords — Sociotechnical Systems, STS, Conceptual Design, Software Reliability, Transaction, Availability.

I. INTRODUCTION

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 [1].

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 [2]. 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 [3]. 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 (STS RT) [2, 4, 5].

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

unreliability of the source data for decision making. This also applies to the task of evaluating software reliability [5, 6, 7]. The main features of an early assessment of software reliability are described in [6, 10]. To solve this problem neural networks [9], fuzzy logic [6], Behavior Models [8] 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 [5] and simple methods for calculating software reliability.

II. DESCRIPTION OF BASIC MODELS

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 $X_d = \{x_{d_l}, l = \overline{1, N_d}\}$. Example of Transaction r_i is shown in Figure 1.

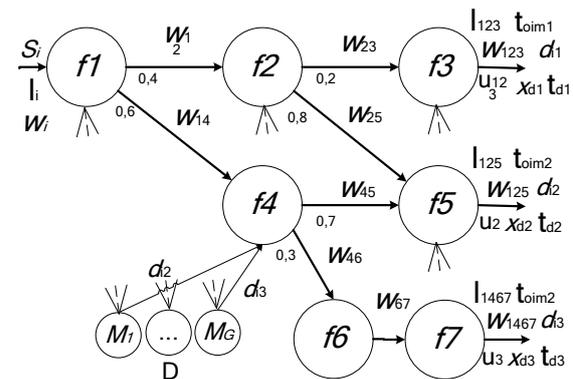


Figure 1: A Graph of the Transaction r_i

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 β				Number of Cycles and Failure Rate of Commands α_η
	$f1$	$f2$...	f_{N_f}	
1. mov	k11	k21	...	$k_{N_f} 1$	1 (α_{min})
2. add	k12	k22	...	$k_{N_f} 2$	1 (α_{min})
3. mult	k13	k23	...	$k_{N_f} 3$	18 ($n3 \times \alpha_{min}$)
4. div	k14	k24	...	$k_{N_f} 4$	24 ($n4 \times \alpha_{min}$)
$\beta \cdot \dots$	$n\beta (n\beta \times \alpha_{min}) \dots$
$K_{com} \cdot \dots$	$k1 K_{com}$	$k2 K_{com}$...	$k_{N_f} K_{com}$	$nK_{com} \times \alpha_{min}$
Arrays of DB		Utilization Rate of the Array			
M1, 256 Mb	Y11	Y21	...	$Y_{N_f} 1$	
M2, 512 Mb	Y12	Y22	...	$Y_{N_f} 2$	
M3, 1 Gb	Y13	Y23	...	$Y_{N_f} 3$	
M4, 2 Gb	Y14	Y24	...	$Y_{N_f} 4$	
M5, 4 Gb	Y15	Y25	...	$Y_{N_f} 5$	
M6, 8 Gb	Y16	Y26	...	$Y_{N_f} 6$	

Each transaction has a time limit for its execution (deadline) $T_d = \{t_{dk}, k = 1, N_u\}$.

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

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

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

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 $F = \{f_j, j = 1, N_f\}$ 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 $\frac{d_{ik}}{D}$.

III. 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 μ with a clock frequency of ω_μ , for which the number of cycles executing a set of command K_{com} is known (see table 1), when

$$t_{oik} = \sum_{\forall f \in r_{ik}} \sum_{\beta=1}^{K_{com}} v_{\beta ik} \times K_\beta \times \tau_\beta \quad - \quad \text{Average}$$

transaction processing time i for output k ;

$$\tau_\beta = \frac{n\beta\tau}{\omega_\mu} \quad - \quad \text{average execution time of command}$$

β ;

$v_{\beta ik}$ - branch coefficient in a transaction i (see Fig. 1 for $f1, f2, f4$).

At the next step, it is necessary to evaluate the failure rate of the FPB chain for each variant of its completion α_{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 α_{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:

$$\theta_{\beta ik} = \sum_{\forall f \in r_{ik}} v_{\beta ik} \times K_\beta$$

A weighted arithmetic mean of the failure rate of the i-th transaction by the k-th output is

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

Processing FPB in a transaction with database arrays reduces this indicator depending on the amount of data read d_{ik} .

Let's introduce coefficient,

$$k_{ik}^{db} = \delta \times (1 - \frac{d_{ik}}{D}), \quad d_{ik} < D$$

δ - coefficient of rationing.

Thus, the reliability of the application software is possible to calculate as

$$p_{ik}(t_{oik}) = k_{av ik} = k_{ik}^{db} \times (1 - \exp(-\alpha_{ik} \times t_{oik}))$$

As a rule, when designing a STS RT indicates the required values of the availability factors $k_{av ik}$, at the same time, the condition $k_{av ik} \geq \tilde{k}_{av ik}$ must be met.

For the entire system (for a microprocessor μ), the following expression is true $K_{av}^s = \prod_{\forall ik} k_{av ik}$.

IV. TO THE QUESTION OF TRANSACTION PROCESSING RESERVATION

If a condition $K_{ik}^{sr} = \frac{t_{dik} - t_{oik}}{t_{oik}} \geq 2$ for a transaction is met (that is, the processor resource for hot duplication or double (or) more transaction miscalculation), then the reliability of the application software is determined

$$p_{ik}^{dub}(t_{oik}) = 2 \times p_{ik}(t_{oik}) - p_{ik}^2(t_{oik})$$

V. THE ESTIMATE OF THE AVERAGE LOSS DUE TO SOFTWARE UNRELIABILITY

The possibility of malfunction/failure at run-time of transaction is calculated as follows $q_{ik} = 1 - p_{ik}$.

Knowing the estimates q_{ik} , w_{ik} it is can to predict of possible weighted average losses of the production due to malfunction/failure in software:

$$W = \frac{\sum_{\forall ik} q_{ik} w_{ik}}{\sum_{\forall ik} w_{ik}} = \frac{\sum_{\forall ik} q_{ik} w_{ik}}{W}$$

Taking into account availability factor of software and by calculating the availability factor of hardware K_{av}^h of STS RT we can estimate the reliability of the entire software-hardware complex: $K_{av}^{stis} = K_{av}^s \times K_{av}^h$.

VI. CONCLUSIONS

The proposed approach to assessing the software reliability of real time STS is used in the Framework CoDeCS for the conceptual design of complex computer systems [12]. Models and the above expressions can be applied autonomously.

REFERENCES

- [1] Kosolapov A., «Changing the Computerization Paradigm», Sat. Scientific Works Sworld. Physics And Mathematics. Computer Science And Cybernetics. Odessa: Kuprienko SV, Vol. 29, No. 2, pp. 45-47, 2014 (in Russian).
- [2] Kosolapov A., and Zhukovitsky I., «Conceptual Design of Real-Time Computer Systems (Models, Methods and Algorithms)», Dnipro. PF Standard-Service View, 2018 (in Ukrainian).
- [3] Gelfand I.M., «On some methods of controlling complex systems», Advances in Mathematical Sciences, Vol. 17, No. 1 (103), pp. 3-25, 1962 (in Russian).
- [4] Grabot B., Mayère A. and Bazet I., «ERP systems and organisational change : a socio-technical insight», (Springer series in advanced manufacturing), Springer-Verlag London Limited, 2008, DOI 10.1007/978-1-84800-183-1.
- [5] Kosolapov A., «Models and Method for Estimate the Information-Time Characteristics of Real-Time Control System», International Conference on Information Systems Architecture and Technology ISAT 2019: Information Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology – ISAT 2019, pp. 58-67, 2019.
- [6] Kumar P.A., and Kumar G.N., «Early Software Reliability Prediction: A Fuzzy Logic Approach», Springer New Delhi Heidelberg, India, 2013.
- [7] Ali A., Jawawi D.N.A., Isa M.A., and Babar M.I., «Technique for Early Reliability Prediction of Software Components Using Behaviour Models», PLOS ONE, journal.pone.0163346 September 26, pp. 1-24, 2016.
- [8] Hu Q.P., Dai Y.S., Xie M., and Ng S.H., «Early Software Reliability Prediction with Extended ANN Model», Proceedings of the 30th Annual International Computer Software and Applications Conference (COMPSAC'06) 0-7695-2655-1/06, 2006.
- [9] Mr.Bibhu Kalyan Mishra, Prof.(Dr.)S.K.Yadav "High Availability Of Resource In Cloud Computing Technology: Review,Issues And Challenges" International Journal of Engineering Trends and Technology 68.1 (2020):34-44.
- [10] Kumar L.S., Kumar A.T., and Vinod G., «Software Reliability Early Prediction in Architectural Design Phase:

- Overview and Limitations*», Journal of Software Engineering and Applications, Vol. 4, pp. 181-186, 2011.
- [11] Kosolapov A. *Methodology for assessing the reliability of fuzzy systems using various types of blurry sets*», Science and transport progress. Newsletter of the Dnepropetrovsk National University of Railway Transport, No. 2 (44), pp. 17-27, 2013 (in Russian).
- [12] Kosolapov A., Loboda D., «*Framework Conceptual Design Of Complex Real-Time Management System (Codecs)*», European Journal Of Advances In Engineering And Technology, 5(8), pp. 559-566, 2018.
- [13] Kalesanwo O., Awodele O., Eze M., and Kuyoro 'S., «*Reliability Assessment of Autonomous Systems: A Systematic Review*», International Journal of Computer Trends and Technology (IJCTT), Vol. 68, Issue 3, March, pp.48-52, 2020.