computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

FAILURES DETECTION METHODOLOGY IN NON RECOVERY COMPUTER SYSTEMS BASED ON DIVERSITY MODELING

George Popov

Technical University – Sofia, Computer Science Department,
Sofia 1756, Bulgaria, email:popovg@tu-sofia.bg

Abstract: *Diversity is a known approach for increasing reliability of computer systems. The goal of this work is to present quantitative criteria for measure of diversity in non recovery computer systems. For this purpose, the model of diversity-based system with two failure types: detectable and undetectable is presented and a formula to calculate it is proposed.*

Keywords: *diversity, dependability, computer system, embedded system, fail-safe, fault-tolerance*

1. FORMULATION OF THE PROBLEM

As is well-known, diversity is a method of solving a problem (mathematical, logical, technical or other) in two (A and B) different ways (paths) with identical input data, by virtue of which a criterion of the solution being perfect is the correspondence (in this particular case- identity) of the obtained output results [1]. The assumption is that there exist at least two ways of solving it.

The input data (Fig.1) are processed in two ways (A and B) and are compared in terms of their correspondence. When the system performs perfectly well, the comparison of the obtained results shows a positive output (OK). That is a condition for the normal work of the system which continues until there is a failure

Error consequences are activated with certain input data and flow of algorithm. If along one of the two ways of solving the problem (in one of the programs, e.g. processing A) an error or defect is activated, there will not be a result or the result will be incorrect at the respective exit. But as at the other channel the result is correct, the output results will not correspond and the agreement (OK) is removed from the exit. The system passes on to a mode of detected Failure. This event is visualized through the diagnostic information.

An analogous result is obtained when a Fault is activated on the other program. When errors or defects are activated on both programs, we get different output vectors as the causes and the processing channels are different. The probability of getting one and the same wrong result is quite negligible.

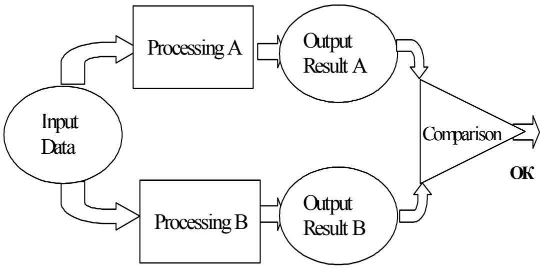

Fig. 1 – A principle of diversity processing

The difference between the output vectors under comparison is an indication that the processing is incorrect and the work of the system is terminated, we search for the cause and it is removed. In this way;

- we can identify errors and defects in an off-line mode (in the testing period);
- we can terminate producing of an incorrect guiding or control signal in the on-line mode of the Real-time Control Systems and create a compulsion for the removal of the causes of failure.

The principle of diversity processing has long been known not only in scientific literature, but also in the practice of the Real-time Systems, used to control technological processes of great importance. In our country such systems are used in rail transport.

The aim of this paper is to model diversity in such a way as to make it possible to determine the factors which influence the identification of failures. Then we will proceed by exploring the influence of the different factors and will suggest methods of enhancing the identifiably of failures.

2. AN OVERVIEW OF THE PROBLEM

Microcomputer systems are a basic element of modern technologies in such important areas as industry, military science, nuclear power engineering, transport, communications, medicine, etc. In this case we speak about safety critical control systems. Some authors use the terms as survivable [2,3], ultrareliability[4] systems.

The problems in the development of these systems arise from the tight schedules for their development and the requirements for reliability [5, 6]. At the level of hardware [1,7,17,22] and software [2,8,9,10,11,12,14,15,16] surplus and diversity are the most common method of enhancing reliability.

Avizienis and Laprie [12,13] define a few aspects of reliability; readiness, good working order, safety, confidentiality, integrity and capability to be maintained and upgraded. A basic approach for enhancing reliability, according to them, is diversity in the architectural system.

Strunk [2] draws attention to the fact that the different variants of one and the same algorithm (program) must be developed independently and a test must be run at different stages of the algorithm so that the results are compared. The benefit from such an approach is that the errors appear in the different versions at different times in the running of the programs. Avizienis [13], Horning and Sha[20, 21] are of the same opinion regarding software diversity.

From what we have said so far it becomes clear that diversity is one of the main approaches of enhancing the reliability of the embedded microprocessor systems. In spite of its indisputable potential, in the literature there are scanty ideas of quantitative evaluation and modeling of diversity [16, 22], hence, conclusions about its effectiveness (economic, technical) in their particular realizations. There is a difference between the two versions, yes, but how big is it, from what point of view, what metrics is used to measure it? And how does this difference, identified using the adopted metrics, affect the ability to discern (to detect) the causes of failure?

An attempt for solving this scientific problem has been made in [1, 7, 21, 22]. We should mention, that in [22] a minor incorrectness is found, which affects the final result.

It comes down to the following. Two types of failures reflecting these facts have been introduced:

α - failures: detectable by comparing the output results

η - failures: undetectable through comparison as they bring about one and the same mistakes in the compared results.

The division of the failures into these two classes is based on the presumption that despite all attempts to make it complete, diversity in practice is not absolute. An absolute diversity would mean that the failures in the two versions are absolutely independent (uncorrelated) and that there are no common causes for failure along the two paths which will lead the same wrong result. In practice these causes present in the common components of the systems: when entering information from a common source, in the only comparator for comparing the results, in synchronizing the work along the two channels, in the common power supply and others, i.e. where incorrectness after failure of error is introduced in one and the same way in the two channels.

On this basis in [1,7,22] a measure of diversity is introduced:

$$\Omega = \frac{\lambda_\alpha}{\lambda_\alpha + \lambda_\eta}, \tag{1}$$

where $\lambda_\alpha, \lambda_\eta$ is the intensity of the two types of failures. If all failures are due to the same cause, for example if the two programming versions A and B are the same and the errors in the two copies of the single program duplicate, then the results will be wrong, but corresponding, and the detectability at comparison **Ω** is brought to zero. The deeper the diversity is the closer **Ω** is to one and the bigger the detectability of the errors and defects. Undetected, although detectable, will remain only failures which by accident cause one and the same output result (vector- i) from the two processings.

3. MODELING DIVERSITY

In the context of probability logic we can assume that there will be inability to identify the failure in two cases:

1. If α_a and α_b accidentally cause one and the same wrong results;
2. If a η failure has happened.

We introduce the Boolean function F_{ni} (non identification) which expression is given with equation (2) and illustrated in Fig 2

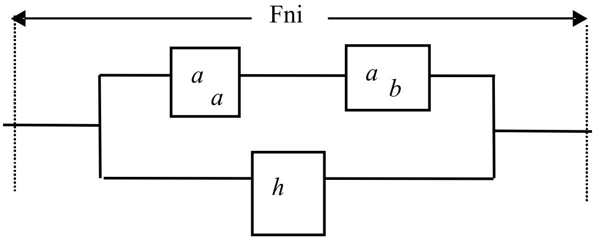

Fig. 2 – Boolean function F_{ni} (non identification)

$$F_{ni} = z^1_{\alpha_A} z^1_{\alpha_B} \vee z^1_{\eta} \tag{2}$$

Where z_i^0 is the logic variable of the statement that «something» has not happened, and e z_i^1 - that the “something» has happened.

In order to model, in terms of probability, the effect of diversity on the ability to identify failures, a logic-probability transition has to be carried out [7]:

$$\mathrm{F_{ni}} = z^1_{\alpha_a} z^1_{\alpha_b} \vee z^1_{\eta} = \overline{\overline{z^1_{\alpha_a} z^1_{\alpha_b}}.z^0_{\eta}} \tag{3}$$

After applying the theorem of De Morgan we have arrived at a non-recurrent Boolean function in a basis “conjunction - negation“. When we make replacements in this function using the rules of the logic-probability transitions, we arrive at the probability that the failures will not to be identified

$$Q_{ni}(t) = 1 - \left\{1 - \left[Q_{\alpha_{A\Sigma}}(t) Q_{\alpha_{B\Sigma}}(t)\right]\right\}[1 - Q_{\eta}(t)] \tag{4}$$

where:

- $Q_{ni}(t)$ is the probability for non-identifiable;
- $Q_{\alpha A\Sigma}(t)$ and $Q_{\alpha B\Sigma}(t)$ are probabilities for an identifiable failure to arise in both channels, which will result in accidentally equal but wrong output signal in the two processing’s;
- $Q_{\eta}(t)$ - is the probability of failure because of a common cause, which generates unidentifiable through comparison output results from both channels.

During A- and B- processing, the probability to get a coincidence of wrong and unidentifiable through comparison output vectors as a result of failure may occur as:

- coincidence of the first vectors;
- coincidence of the second vectors;
- and etc., when a coincidence of any similar-range vectors is found.

Then, the component in (4) $Q_{\alpha A\Sigma}(t)$,$Q_{\alpha B\Sigma}(t)$ could be written as a sum of probabilities:

$$Q_{\alpha_{A\Sigma}}(t) Q_{\alpha_{B\Sigma}}(t) = Q_{\alpha_{A1}}(t) Q_{\alpha_{B1}}(t) + \ldots + Q_{\alpha_{A2^{w-1}}}(t) Q_{\alpha_{B2^{w-1}}}(t), \tag{5}$$

where $Q_{\alpha_{ai}}(t)$ and $Q_{\alpha_{bi}}(t)$ are the probabilities for coincidence to happen through i-th vector of both processing, $i \in \{1,2,...2^w\}$.

The presumption to make sum of the products of probabilities to get the overall probability when unidentifiable failures could occur is based on impossibility to arise more than one coincidence of output vectors in one and the same moment (or to coincidence the first, the second, or …or 2^w th). As these logical relationships are orthogonal, they could be presented as a sum:

$$Q_{\alpha_{A\Sigma}}(t) Q_{\alpha_{B\Sigma}}(t) = \sum_{i=1}^{2^w} Q_{\alpha_{Ai}}(t) Q_{\alpha_{Bi}}(t) \tag{6}$$

When all wrong post-failure vectors are equally probable: $Q_{\alpha_{Ai}}(t) = Q_{\alpha_A}(t)$,i.e $Q_{\alpha_{Bi}}(t) = Q_{\alpha_B}(t)$, and taking into account the equally probabilistic distribution for eq.(6), we can write:

$$Q_{\alpha A\Sigma}(t) Q_{\alpha B\Sigma}(t) = 2^w \left[\frac{Q_{\alpha A\Sigma}(t)}{2^w} \frac{Q_{\alpha B\Sigma}(t)}{2^w}\right] \tag{7}$$

In every single moment, there is one identical to the functional vector among the 2^w wrong ones. The probability to get it is:

$$Q_{\alpha}(t) = \frac{Q_{\alpha A}(t)}{2^w} \frac{Q_{\alpha B}(t)}{2^w} \tag{8}$$

As we are searching for a probability for failure, it should not be attached to the failure behavior and must be subtracted from their total number. So, we have:

$$Q_{ni}(t) = 1 - \left[1 - \left(2^w - 1\right) \frac{Q_{\alpha A}(t)}{2^w} \frac{Q_{\alpha B}(t)}{2^w}\right] \left[1 - Q_{\eta}(t)\right] \tag{9}$$

If we have an equal probability for failure in both channels $Q_{\alpha_a}(t) = Q_{\alpha_b}(t)$ we get from (9) the following:

$$Q_{ni}(t) = 1 - \left[1 - \left(2^w - 1\right)\left(\frac{Q_{\alpha}(t)}{2^w}\right)^2\right] \left[1 - Q_{\eta}(t)\right] \tag{10}$$

When the distribution for failure is exponential, the failures’ intensities λ_{α} and λ_{η} are constant and time-independent values and the probability for failure is:

$$Q(t) = 1 - e^{-\lambda t}, \tag{11}$$

After a substitution of (11) into (9), we obtain:

$$Q_{ni}(t)=1-\left[1-\left(2^{w}-1\right)\left(\frac{1-e^{-\lambda_{\alpha}t}}{2^{w}}\right)^{2}\right]e^{-\lambda_{\eta}t} \qquad (12)$$

When we take into account (1) instead of failure intensities λ_{α} and λ_{η}, the "depth of diversity" parameter Ω could be introduced in (12):

$$\lambda_{\alpha}=\Omega\lambda, \qquad (13)$$

$$\lambda_{\eta}=(1-\Omega)\lambda \qquad (14)$$

Following substitution, the formula (12) could be rewritten as:

$$Q_{ni}(t)=1-\left[1-\left(2^{w}-1\right)\left(\frac{1-e^{-\Omega\lambda t}}{2^{w}}\right)^{2}\right]e^{-(1-\Omega)\lambda t} \qquad (15)$$

Obviously, if the diversity has a maximum value, and **Ω**=1, based on (15), we can obtain that the probability for wrong identification is minimal:

$$Q_{ni}(t)=1-\left[1-\left(2^{w}-1\right)\left(\frac{1-e^{-\lambda t}}{2^{w}}\right)^{2}\right]= \\ =\left(2^{w}-1\right)\left(\frac{1-e^{-\lambda t}}{2^{w}}\right)^{2} \qquad (16)$$

When the diversity is missing: **Ω**=0, the probability of wrong identification is maximum and gets equal to that of single-channel system which has failure intensity of **λ**.

$$Q_{ni}(t)=1-\left[1-\left(2^{w}-1\right)\left(\frac{1-1}{2^{w}}\right)^{2}\right]e^{-\lambda t} \qquad (17)$$

i.e $$Q_{ni_{\max}}(t)=1-e^{-\lambda t} \qquad (18)$$

In the case of ***w***=1 each of the two channels, the output signals of which is under comparison, has one binary physical output. Then, ***w*** = 1 and if **Ω**=1 the minimal probability for wrong identification of the failure will have a maximum value:

$$Q_{ni\,\min_{\max}}(t)=\left[\frac{1-e^{-\lambda t}}{2}\right]^{2}=\frac{1}{4}Q(t)^{2} \qquad (19)$$

It means, that the both diversity described channel have equal intensity of failures ***λ*** and the diversity is absolute, the probability for unidentified failure of the single-output system would be ¼ of the probability for failure powered by two in each independent channel.

4. THE EFFECT OF DIVERSITY ON THE IDENTIFIABILITY OF FAILURES

Obviously, the probability of unidentifiable (dangerous) failure $Q_{ni}(t)$ in diversity systems depends on:

- depth of diversity **Ω**;
- the number of bits ***w***, with which the output result from the processing in the diversity channels is represented;
- the total intensity of failures in system ***λ***;
- the working out of system ***t***.

In formula (15) some calculations have been made for a fixed time 1200 months, and ***λ*** = 0, 0013 and a fixed number of bits ***w***=8 of the input vector. In Fig. 3 the obtained results have been interpreted graphically.

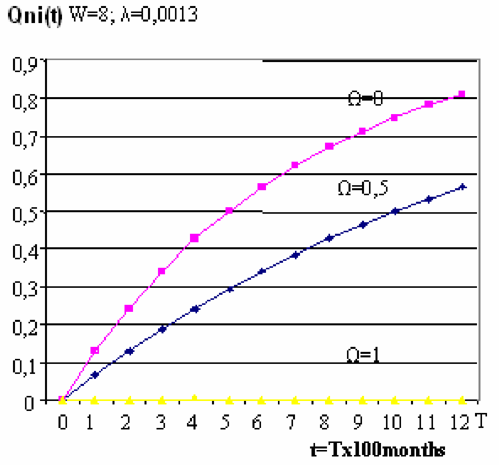

Fig. 3 – Q_{ni}(t)=f(t)

We can see that with the increase in the depth of the diversity **Ω** (Fig. 4), the probability of not identifying the failure decreases significantly the longer the input vector ***w*** is and the bigger the intensity of the failures of the microcomputer system. With **Ω** = 1, which is practically unattainable, we can achieve a hundred of times greater identifiability.

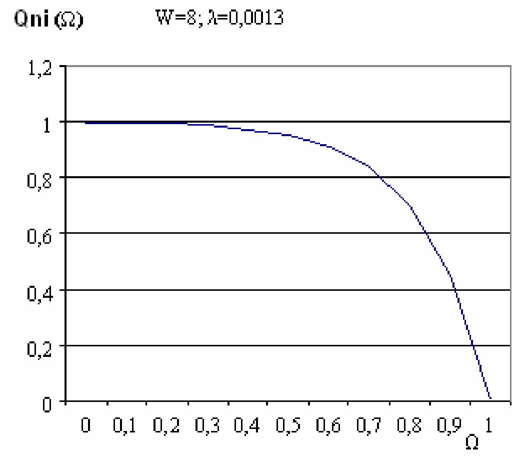

Fig. 4 – Q_{ni}(t)=F(Ω)

Fig. 5 shows 3D graphics of function $Q_d = f(\Omega,t)$,where $\Omega\in[0,1], t\in[1,40000]$, λ=0.0001 and $w = 8$. As it has been expected, with the ageing of the system and the increase in the probability of failure increases the probability of not detecting the failure. But with absolute diversity it becomes small.

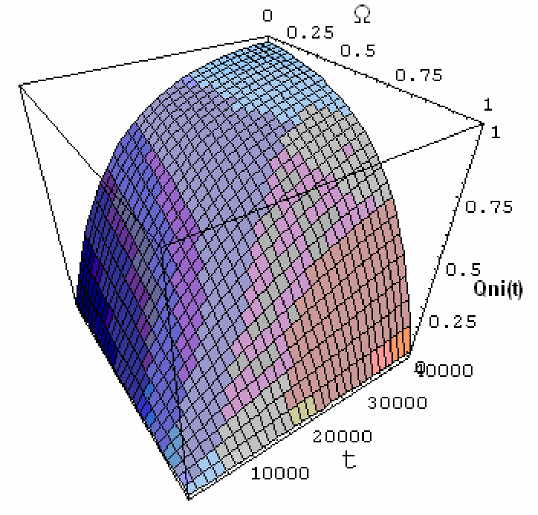

Fig. 5 - Q_{ni}(t)=F(t,Ω)

5. CONCLUSION

This paper states and solves the issue of modeling diversity in computer systems. Adopting as metrics the depth of diversity, variable Ω as it is known in the literature, it has been proven that it increases, on the one hand, with the independence of the information processing channels of the hardware and the software and, on the other hand, with the decrease of the common diversity systems programming and apparatus components. On that basis we have obtained the following new results:

1. We have found a new correlation between the effect of diversity depth on the ability to identify failures which result from defects in the hardware and errors in the software;

2. We have done a study of the quantitative values of the effect of diversity depending on the bits of the input vectors and the intensity of the failures of the system.

3. We have put forward ideas for schematic – technical solutions of redundant systems with increased identifiability evolving from the conclusions drawn from the present study.

6. REFERENCES

[1] Hristov H., *The Problem about Reliability of Electronic Safety Systems*, D.Sc. Dissertation, Technical University-Sofia, Bulgaria, 1988 (in Bulgarian)

[2] Strunk E. Survivability in Embedded Systems, Ph.D. Dissertation, Sept. 12, 2003

[3] Knight, J. C., E. A. Strunk and K. J. Sullivan. Towards a Rigorous Definition of Information System Survivability,DISCEX 2003, Washington, DC, April 2003.

[4] Butler, R. W., and G. B. Finelli.*The Infeasibility of Experimental Quantification of Life-Critical Software Reliability*. ACM SIGSOFT '91 Conference on Software for Critical Systems, New Orleans, LA, December 1991.

[5] Karakehayov Z., K.S.Kristensen, O.Winther, *Embedded Systems*, Technical University of Denmark, Department of Applied Electronics, 1995.

[6] Isaksen U., J. P. Bowen, N. Nissanke. *System and Software Safety in Critical Systems*, The University of Reading, Department of Computer Science Whiteknights, PO Box 225, Reading, Berks RG6 6AY, UK, December 1996

[7] Hristov H.A., V.Trifonov, *Safety and Reliability of Communications*, Book, Novi Znania, Sofia, 2005 (in Bulgarian)

[8] Martin Törngren and Jan Torin. *Conceptual Design of Dependable Embedded Control Systems.* 7.Oct 1998.

[9] Burns A., Wellings A.J. *HRT-HOOD: A Structured Design Method for Hard Real-Time Systems,* J. of Real-Time Systems, Vol. 6, No. 1, January 1994.

[10] Rivera J.G., Danylyszyn A., Winstock C.B., Sha L., Gagliardi M.J. *An architectural description of the Simplex Architecture.* Technical report CMU/SEI-96-TR-006 ESC-TR-96-006. Carnegie Mellon University, Software Engineering Institute, 1996.

[11] Törngren and Wikander (1996). *A Decentralization Methodology for Real-Time Control Applications*:*Control Engineering Practice*, Vol. 4, No. 2, pp. 219-228, February 1996

[12] Avizienis, A. *The N-version approach to fault tolerant software:IEEE Transactions on Software Engineering* 11(12):1491-1501, December 1985.

[13] Avizienis, A., J. Laprie, and B. Randell. *Fundamental Concepts of Computer System Dependability*., IARP/IEEE-RAS Workshop on Robot Dependability: Technological Challenge of Dependable Robots in Human Environments, Seoul, Korea, May 2001.

[14] Robyn R. Lutz, *Software Engineering for Safety: A Roadmap, The Future of Software Engineering*, ACM Press 2000

[15] Isaksen U., J. P. Bowen, N. Nissanke, *System and Software Safety in Critical Systems*, The University of Reading, Department of Computer Science Whiteknights, PO Box 225, Reading, Berks RG6 6AY, UK, December 1996

[16] Leveson N. G., *Software safety: Why, what, and how*. Computing Surveys, 18(2):125{163, June 1986.

[17] Sandoval M., *"Smart" Sensors for Civil Infrastructure Systems*, A Dissertation Submitted to the Graduate School of the University of Notre Dame, May, 2004

[18] Wilikens M., Masera M., Vallero D. *Integration of Safety Requirements in the Initial Phases of thePorject Lifecycle of Hardware/Software Systems*. Proc.of *SAFECOMP97*, Springer-Verlag, ISBN 3-540-76191-8, (1997)

[19] Redell O. *Modelling of Distributed Real-Time Control Systems: An approach for design and early analysis.* Licentiate thesis, Department of Machine Design, Royal Inst. of Technology, Stockholm, (1998).

[20] Horning J. J., H. C. Lauer, P. M. Melliar-Smith, and B. Randell. *A program structure for error detection and recovery.* Symposium on Operating Systems 1974: 171-187.

[21] Sha, L. *Using Simplicity to Control Complexity: IEEE Software* 18(4):20-28.

[22] Popov G. *Modeling Diversity as a Method of Detecting Failures in non Recovery Computer Systems :Information Technologies and Control*, 2005, N#2

George Ilinchev Popov received his PhD in Technical University, Sofia, Bulgaria in 2006. He is working now with Department of Computer Science as an assistant professor. His research interests include Recognition Systems, Embedded systems, Dependability systems and etc.