Computer Control Systems With Critical Safety Applications: Problems And Some Solutions

Hristo Hristov[1], Mariya Hristova[2]

[1]University of Transport, Sofia, Bulgaria, cac@vtu.bg

[2]University of Transport, Sofia, Bulgaria, mhristova@vtu.bg

Contribution to the state of the art

DOI: **10.7251/JIT1702061H** UDC: **004.934:004.432**

Abstract: Safety Critical Systems (SCS) are defined as systems controlling critical technological processes, on the proper functioning of which depends human safety. The taxonomy of concepts related to SCS is presented as a dendritic classification scheme. The emphasis is on hierarchical relationships between concepts. After studying global scientific literature, international standards and corporate materials, a classification of the scientific issues accompanying the creation of new SCSs was made.

Regarding a part of the broached issues, technical solutions are suggested based on the structural system of the system. In particular, methods and means have been developed to detect and tolerate failures and errors in building the structure and to reduce their adverse impact on the functionality and safety of the systems.

Formal models have been developed, concerning which calculations and studies have been performed. Quantitative dependencies are established between the technical and probability parameters of diversity structure on the one hand and the reliability and safety of the system on the other. Conclusions are drawn as regards the practical application of the methods and models.

Keywords: Safety Critical Systems, risk, security, safety, reliability.

Introduction. Problem Setting

Breaches of regulated functionality (normal operation) due to failures and intrusions in systems controlling Special Critical Technology Processes (*SCTP*) may cause imminence, danger to the health and/ or loss of human life, of large material and/or spiritual values and/or damage to the environment. In various spheres such as aviation, space, defense, rail transport, nuclear power, medical electronics, machine building, etc. there are numerous examples of such technological processes.

Safety Critical Systems (SCS) is a system that controls special critical technology process - SCTP. Human safety within this technological process depends on the designated functioning of the system. In addition to their functional tasks, SCSs are associated with a risk of breach of regulated functionality and are highly critical for the health and life of people, which is why they are rightfully called systems with high moral standards. [1].

Risk, as a concept, combines the assumption of an undesirable event with the magnitude of the foreseeable adverse consequences it entails (threat to life, property damage, loss of natural assets, etc.) and is defined as a probabilistic observation. There are *allowable values* from zero (zero risk) to *limit values - border, accepted risk* (Fig.1). According to the MIL-STD-882D standard, feasibility is determined by the upper limit of the acceptable risk level [2].

The borderline risk in SCS is determined depending on the purpose and application of the system. For the various aforementioned areas, there are

specific standards that define the limit values. In some of them limit values are assigned by groups depending on the size of the supposed losses. For instance, the RAMS [3] standard for tolerable risk in railway safety systems is determined in 4-degrees with probabilities 1.10^{-9}, 1.10^{-8}, 1.10^{-7} and 1.10^{-6}, as for the smaller predictable adverse consequences a higher limit value is set.

Safety is a system property that is measured by its probability of allowing a risk occurrence (both in normal operation and in faults), lesser than the borderline risk.

Figure 1. *Safety and accepted risk*

Nowadays, all SCSs are *computer-based*. Hardware, software, and communications are subject to increased requirements for the reliability and inadmissibility of wrong controlling actions. The system must be so designed that, any hazards, if occurring, can be detected and removed before causing an accident. SCS's failures, often due to software errors, lead to airspace crashes, failed space missions, land transport accidents, inadequate control and accidents in nuclear power plants, military incidents, improper dosage of radiation therapy for patients, significant economic losses, etc.

This publication aims to structure, *within the provided volume, the taxonomy of concepts, to define the main issues in the scientific research of Safety Critical Systems and to propose some solutions.*

TAXONOMY

Taxonomy in general

This is a study of the principles of classification and systematics of complexly organized areas with a hierarchical structure, a "hierarchically structured set of terms that is used for classification and navigation"[6]. It consists of rules, methods and their application. The methodology includes the study of text corpora to identify and annotate the most common concepts, and to define hierarchical relations between classes. The elements and groups of objects and subsystems selected for studying are called taxa. The classification is illustrated graphically with two schemes: dendritic, known in logic as classification, or as circles inserted in each other, representing taxa.

Taxonomy of SCS

Figure 2 shows the taxonomy of concepts suggested by the authors in relation to SCS. A structural classification scheme has been used. At baseline level the terms are: *errors, faults* and *objective external influences*. Three critical features of the systems are accepted as taxa, namely: *safety, security and reliability*.

Error is a deviation from an accepted and validated fidelity (regularity) criterion. It may occur both in the terms of reference (specification) of the system, and in its design, elaboration and programming.

Fault is a condition in which the object under consideration does not comply with the regulatory and technical documentation (terms of reference). It occurs in the course of the work (time or volume of work) and under the influence of external (meteorological, physical, chemical, electrical) and other objective impacts. The object ages, wears out, deteriorates, becomes a subject of impacts and this results in a breach causing at least one parameter to be inconsistent with the system's regulatory and technical documentation.

Security is the ability of the system to defend itself against, resist, counteract to any external destabilizing factors and impacts, as well as internal changes that may lead to danger.

In Figure 2, the hierarchical relationships between the concepts can be traced. Here are some of them.

Errors in the system elaboration and maintenance are subjective: accidental and unintended and non-accidental (ill-) intentioned.

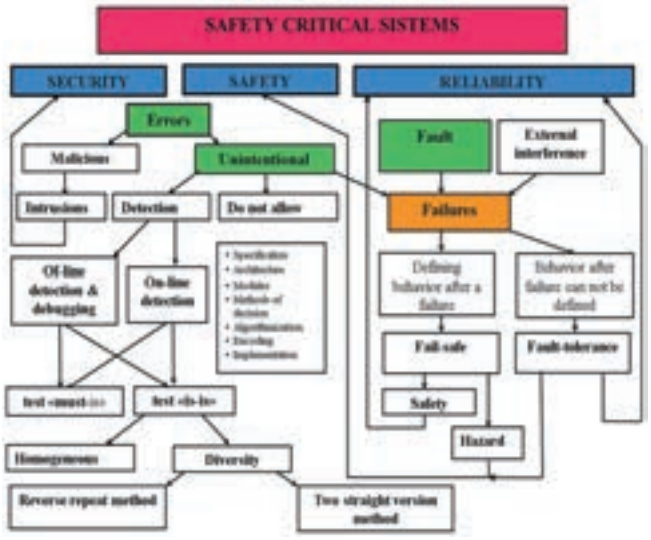

Figure 2. *Taxonomy of concepts in Safety Critical Systems*

Accidental errors (due to lack of qualification and experience of the persons developing and maintaining the system, incompetence, inattention, distraction, negligence, etc.) affect reliability and safety.

Intrusions by outsiders - errors with malicious intent - are unauthorized intrusions into the system in order to block its operation, retrieve and/or change the information in it. Intrusions are concepts of information and network *security* and are related to highly evolving methods and means of detecting external intervention and protecting SCS from malicious impacts.

To prevent accidental unintentional errors, a *perfection strategy* is implemented: a risk-based specification and a set of approaches, specific methods and tools for hardware and software development and structuring of the system.

The detection and correction of faults and errors (in design and technology, algorithmization and programming) is carried out *off-* and/or *on-line*. Off-line testing involves some approaches, methods and tools, most of which are related to the prelaunch phase. And yet, regardless of their efficiency, in a complex system there are many workspaces that may remain untested. It enters into operation with residual errors. Testing in the course of operation (on-line) serves to detect residual errors and/or any emerging faults as well as to provide timely response after identifying these (see below).

Various methods (on/off-line) are used to detect faults and errors, but diversity is the most efficient among them.

Diversity [5, 6, 7] is a method of solving a problem (mathematical, logical, technical, programming, etc.) in two (A and B) different ways (methods, programmes, channels) with one and the same input data. When the two compared solutions are relevant (including identical), there are no breaches of normal operation.

If the two channels are independent, their irregularities are detected because in the case of a breach their outputs differ. Normal operation is deliberately stopped for the purpose of removing the causes or switching to a reserve.

- When the two programs are identical (two copies of one program), their errors are the same, the channels work in the same manner, albeit incorrectly, their outputs are corresponding and the errors are undetectable.
- When the two hardware channels are homogeneous (identical in terms of hardware and software), only the hardware irregularities are identified because they are independent.
- When working on different programs in completely independent (A and B) channels, both errors and faults are detected. This is the genuine diversity method.

The most frequently applied are the two versions that deal with one and the same problem by following different methods, algorithms, and programs developed by different teams in order to be independent.

*Accidental unintentional errors***,** along with hardware malfunctions and external, cause *failures*. Failures are events that lead to a violation of system performance. System responses to failures should be of two types: stopping until the irregularities are removed or switching to a reserve, if any.

When the nature of the SCTP technological process allows, after a failure, the SCS control system may switch to a predefined safe state or a desired fail-safe behaviour. Then, the failures are *safe* and *dangerous (hazard).* If the behaviour of the system contradicts the defined criterion, the failure is hazardous, and if it complies with it – it is safe. Such are the cases in railway signalling, locks, security technological equipment and many others.

When on the other hand SCTP is of such nature that a safe state or behaviour can not be defined (in air transport, life support systems, medicine, etc.),

any failure of the control system thereof is dangerous. For this class of systems, the only approach to achieving safety is fault tolerance. Regardless of its nature, the failure is masked, evasive, tolerated. But the degree of fault tolerance and hence dependability, depend on the presence of redundancy - structural, functional, temporal, etc. When the redundancy due to the failures is exhausted, the system begins to fail and the nature of the failure starts re-emerging. The redundancy exhaustion depends on its depth, and it has functional and economic dimensions.

SCIENTIFIC ISSUES REGARDING SCS

Based on researching global scientific literature, specialized publications, reports of international scientific forums, international standards, corporate materials, etc., an attempt has been made to classify the scientific issues regarding SCS. It is constantly up to date and can be derived from the following hypothesis:

A *risk-based specification,* relevant to the actual SCS application conditions, needs to be established. For this purpose, the potential dangers that may arise from operation/failures of SCS should be studied and examined. Next, a comparative analysis of the principles and structures on which SCS can be based should be performed in order to select a research-based method of developing the designed system.

Optimal technical solutions for individual structural units should be found. A suitable programming language for SCS programming should be selected [5].

No errors should be made at any stage and at any level of SCS development. For this purpose, a class of scientifically sound methods and tools is used. And yet, despite their implementation, errors in complex systems certainly do exist. They must be detected and removed before the commissioning of the system. To that end, another class of scientific methods and tools is applied.

However, some errors still remain undetected and may cause dangerous accidents or may result in limiting functionality. The dangerous impact of the former, and the adverse impact of the latter may be limited if errors are detected in the course of operation (on-line). Then, their consequences are suspended, and the errors are removed or tolerated, so as to avoid their re-emergence. For this purpose, a third class of methods and mechanisms is applied.

Based on this hypothesis, the classification given below is elaborated. Scientific issues of SCS are placed in two areas:

In terms of functionality. The functions and structure of the systems with various intended use are described and formalized in their specifications and other accompanying documents, which must comply with the relevant standards. Scientists, researchers, designers and programmers in the relevant sphere (transport, aviation, energy, medicine, etc.) elaborate specialized SCSs harmonized with these standards. If the authors did not comply with the principles and rules, if not all safety conditions were taken into account, if the correct failure responses were not found, the system may be dangerous also in the course of its normal operation as it has been set. That is, functionality is safety related.

In terms of reliability and safety. It means that we assume that the hypothesis that the specification (terms of reference) as per which the system operates is perfect. If the system functions, it is safe. Functionality is not safety related. Dangers are created when the system becomes incapacitated, ie. after failures only. Problems in this area are related to fail-safe and fault-tolerance principles for structuring and developing SCS, methods to achieve flawless software, methods ensuring the reduction of hazardous operation can be reduced, etc.

Based on the studies and conclusions made so far, a structure and summaries of SCS's scientific issues can be made:

1. Creating a *risk-based specification* that is complied with the potential dangers under actual application conditions
2. *Principles for developing and structuring* SCSs relevant to the field of application and safety standards;
3. Selection of *SCS programming language and development software* that not only provides the necessary functions but also ensures that the system is safe to operate and safe in the event of failures
4. Methods and tools for *detecting faults and errors* in SCS.
5. Methods and algorithms to *tolerate SCS fail-*

ures and errors and to reduce their impact on the functionality and safety of the systems.

6. Quantitative *assessment of the impact of failures* on the reliability and safety of SCS and comparison with the default values.

Some Solutions Proposed

Here are some proposed solutions stated under i.4, i.5 and i.6 of the SCS problem area defined as such.

Error detection through diversity in dual channel control systems

Dual channel structures 2Ú2 *are studied* [7]**.** The entered input vector *X* is processed in two channels 1 and 2 (Fig.3), and the output vectors *Y1* ($y_1, y_2 \ldots, y_v$) and *Y2* ($\boldsymbol{y_1, y_2 \ldots, y_v}$) with identical length are compared following the principle "is – is". Their correspondence is an availability criterion, whereupon the comparator gives OK for execution of the impact *Y1* on the controlled object CO.

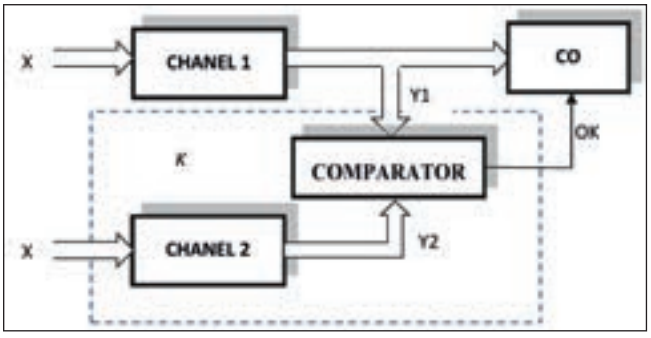

Figure 3. *Dual channel diversity control system*

A metric is proposed to measure the difference ϕ (0 ÷ 1) between the channels. Formulas for quantification of the effect ξ of diversity on the two channels are determined by this metric [6]. In this authored publication the effect is determined upon exponential distribution of the system until failure (λ = const.): where ***t*** is work (aging) and ***v*** is the length of the compared vectors in bits.

$$\xi = \frac{1-e^{-\lambda t}}{1-\left[1-(2^v-1)\left(\frac{1-e^{-0,5\varphi\lambda t}}{2^v}\right)^2\right]e^{-(1-\varphi)\lambda t}}$$

It is calculated at different ϕ values and at different values of the parameters involved in the derived formulas. Graphical results are shown in Fig. 4.

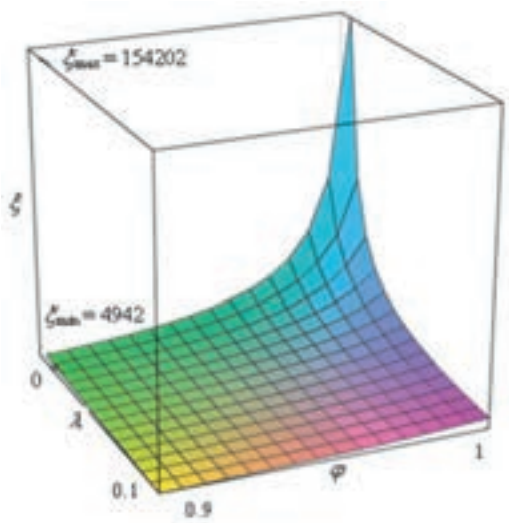

Fig.4 $\xi = f(\varphi, \lambda)$

It is established that:

- The maximum probability of non-identification of errors and faults, and hence the lowest safety, occurs in the absence of diversity which is equivalent to a one-channel system.
- The safety of the system is the greatest at a minimum probability of non-identification of errors and malfunctions. It is attained in the presence of full diversity when the two channels are absolutely independent. Then, the probability of safe operation is reduced to several orders of magnitude (hundreds of thousands of times) as compared to a one-channel system.
- The effect of diversity is more significant proportionally to the greater intensity of failures resulting from errors, and the fewer number of failures due to faults.
- The probability of dangerous failures is very sensitive to the intensity of failures caused by software errors. By reducing software errors, the danger is reduced by almost two orders of magnitude.

Error detection using the reverse iteration method

One of the problems of the error-detection method outlined above is that the structure still implies the existence of at least two independent, effective methods for solution (hardware and programming), which is often not the case. In addition, in order to achieve complete independence of the A and B pro-

grammes, they must be designed by different teams, which renders the decision more expensive.

These considerations stimulate the search for an approach that uses principally different methods for solution, but the distinction must be obtained by compulsion on a mandatory basis. As such, a *reverse iteration method - RIM* is proposed [8]. The application of this method spares the necessity to search for principally different algorithms. This option is embedded in the very nature of the method itself.

A prerequisite for RIM is the ability to solve "straight" and "reverse" problems (Fig.5). As per the input data X by the "straight" algorithm, problem A is solved, the output result which controls the controlled object CO. But under condition: available permit (OK) by the comparator, which compares input data ***X*** to the result of the reverse problem B. In turn, it has as input the result Y and by the algorithm that is „reverse" to A it has to calculate the input data x.

A simple example illustrates the idea. A solution to the algebraic problem $y = x^2 + 1$. The reverse problem is $x = \sqrt{y-1}$. If we set an input value $x = 2$ for the straight task the output result will be $y = 5$. The input data $x = 2$ are memorized and submitted for execution to programme A. The solution result $y = 5$ is memorized and submitted as input data for the reverse problem B. The result obtained by the processing of B given these data should be the same as for the input data of A. If the compared vectors (2 ↔2) coincide, an OK permission is obtained. If not, then an error has occurred. "OK" is cancelled.

Certainly, in large systems solving complex problems it is not that simple.

In digital circuits such as microprocessors, code vectors $X\ (x1\ x2\ ...,\ x_w)$ are entered at the input A of the circuit with a length of w bits. After processing the information flow by the programme, a vector comes out in the form of combinations of ones and zeros $Y\ (y_1 y_2 ..., y_v)$ with a length of v bits (Fig.6).

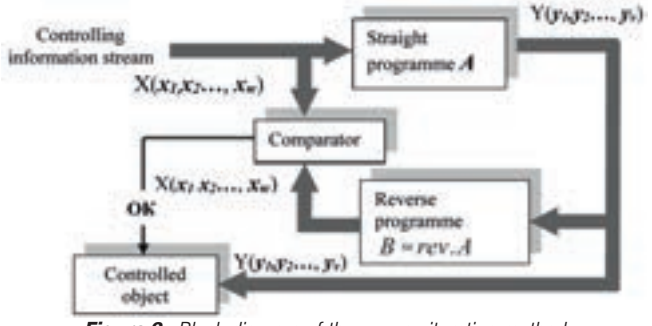

Figure 6. *Block diagram of the reverse iteration method*

$Y\ (y_1\ y_2\ ...,\ y_v)$ is a control signal to the process (object), but it is given with the condition of having a match in the comparison after the solution of the reverse task. In the case of equivalence, the comparator (comparison device) gives OK. The controlled object obtains the right to accept the *Y* signal $(y_1 y_2 ..., y_v)$ and perform the command.

The advantages of the reverse iteration method are that the reverse programme naturally creates an algorithmic diversity by overcoming the difficulty of finding two independent effective methods for solving the problem. The safety of the *reverse iteration method* is based on the following: the error activation in programme A will trigger the wrong output vector A ,to result, which will entail incorrect end result B'. When comparing A ↔B ,the error is detected. A failure in the reverse programme, despite the correct B result, will result in an incorrect end result B. These are effective methods for solving the problem, and that means that it can be solved by a single programmer.

Error tolerance

Now the problem is the opposite. No identification is sought, but just the opposite - suppression, tolerance of errors.

As per the algorithm shown in Fig. 7, the two programmes are connected in a reliable way in parallel - it is sufficient for one of them to function in order to attain operability of the system.

In dynamic programme redundancy, only the main program B functions normally. The backup R is switched on when a failure occurs causing its activation. When B is activated by an error and an incorrect result is obtained, the backup program tolerates it. The system will not operate only if hardware faults or software errors are activated in both programs. The system's quick response under this method is greater because the backup programme is switched on only when needed.

In order to establish graphical dependencies of reliability from the depth $\boldsymbol{\varphi}$ of the diversity, formulas are derived in which the time is marked by t, and the intensity of the failures by $\boldsymbol{\lambda} = const$. The diversity depth is determined by the degree of independence of the two programs, that is, by the degree to which they generate independent failures. When $\boldsymbol{\varphi = 0}$ failures are common, and when $\boldsymbol{\varphi = 1}$ failures of each

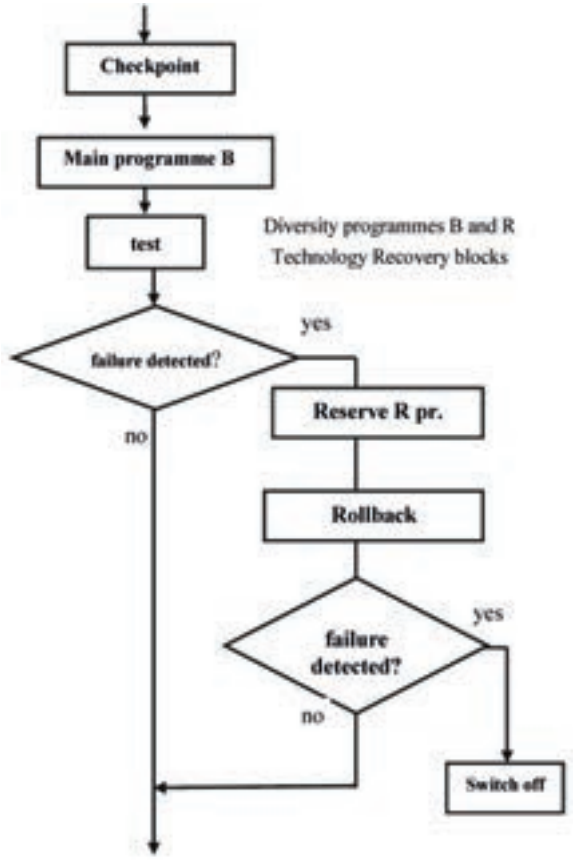

Figure 7. *Software fail safe through dynamic software redundancy*

programme are separate and independent from each other.

Let us assume that the two programmes, albeit being diversity programmes, have equal reliability, i.e., that as a result of errors each of them generates one and the same flow of failures. The following may be written about the reliability of a diversity programme:

$$P_s(t)_{\lambda=const} = e^{-(1-\varphi)\lambda t}\left[1-(1-e^{-\varphi\lambda t})^2\right]..$$

It is obvious that when diversity is maximal, upon full independence of both programmes (φ=1), the reliability of the programme system is the highest:

$$P_s(t) = 1-(1-e^{-\lambda t})^2 = 2e^{-\lambda t} - e^{-2\lambda t}$$

When the two programmes ***B*** and ***R*** are identical, i.e., there is the absence of diversity (φ = 0)

$$P_s(t) = e^{-\lambda t},$$

which should have been expected.

For $\lambda t = 0,1$ in the same programmes a probability of failure Qs = 0,0952 may be expected, in fully independent Qs = 0,0091, i.e., reliability increases 10.5 times.

Calculations for the two-channel cases are made. Some results are graphically illustrated in Fig. 8. From the curves it can be seen that with the increase of the depth of diversity the reliability of the system Ps (λt) improves considerably. For example, with λt = 0,1 and depth φ = 1,0, the probability of failure *Qs(λt) = 1-Ps (λt)* decreases as a result of diversity from 0,1 to 0,01, that is, by 10 times. Based on the models so derived and the performed research as shown above, the following important summaries can be presented:

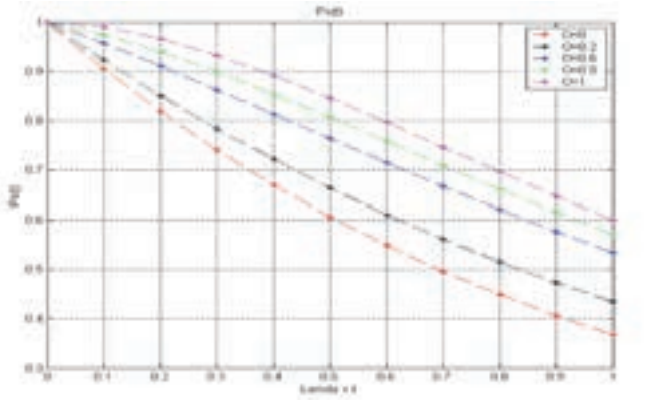

Figure 8. *Graphical results regarding the impact of diversity on the reliability of a dual programme system*

The deeper the software diversity is, the greater the reliability will be. By varying the depth of diversity from 0 to 1, the reliability changes as in a transition from a coherent to a parallel in terms of reliability system.

In order to determine the factors on which the depth of diversity depends, it is necessary to examine the particular scheme for the particular case by searching for the general and local causes of failures and their intensity.

Conclusion

Safety Critical Systems are an important class in the area of real-time control systems. Their essential distinction from the other classes within this area is the mandatory requirement not only to provide the necessary functions as intended, but also to ensure that the system is safe to operate and safe in the event of failures.

In order to resolve SCS-related issues, it is necessary to strictly define the concepts and dimensions involved in the study of this class. On this basis, a structure and classification of the scientific problems is provided herein, a main part of which is related to the errors in the development and maintenance of the system.

Some diversity-based solutions are proposed, through which timely detection can be attained. Formal models are derived and calculations are made allowing for the provision of recommendations for their practical application.

REFERENCES:

[1] Caia H., Ch. Zhanga, W. Wub, T. Hoa, Z. Zhangb. Modelling High Integrity Transport Systems by Formal Methods, International Conference on Traffic & Transportation Studies (ICTTS'2014) 729 - 737, 2014.

[2] Lyu M. R., Handbook of Software Reliability Engineering, McGraw-Hill, 1996.

[3] Standard: CENELEC - EN-50128 Railway applications - Communication, Signalling and Processing Systems - Software for railway control and protection systems, European Committee for Electrotechnical Standardization, Brussels, 2011.

[4] Algirdas Avizienis, Jean-Claude Laprie и Brian Randell. Basic Concepts and Taxonomy of Dependable and Secure Computing IEEE Transactions on Dependable and Secure Computing, vol.1, 2004.

[5] Христова, М., Софтуер за критични по безопасност системи, София, ВТУ „Т. Каблешков", 2016.

[6] Popov G., Failures Detection Methodology In Non Recovery Computer Systems Based On Diversity Modelling, ISSN 1727-6209, International Scientific Journal of Computing, Vol. 6, Issue 3, 46-51 Nov., 2007.

[7] Hristov H., W. Bo, Safety Critical Computer Systems: failure Independence *and software diversity effects on Reliability of dual channel structures,* Information Technologies and Control, № 2, pp. 9-18, 2014.

[8] Христова М. Обнаружение ошибок программного обеспечения посредством метода реверсного повторения. Proceedings of the II[nd] International Scientific and Practical Conference "Methodology of Modern Research", Dubai, UAE, "World science", ISSN 2413-1032, 4(8), vol. 4, 2016.

Submitted: November 11, 2017.
Accepted: December 4, 2017.

ABOUT THE AUTHOR

Hristo Hristov, DSc, is Professor at the Technical University of Sofia. He graduated from the Mechanical and Electrical Institute of Sofia, MSc programme in Telecommunications in 1962. He defended a PhD thesis in Moscow (1972) and a thesis for the degree of "Doctor of Sciences" at the Technical University of Sofia (1988). He has developed and is a primary teacher of various courses related mainly to critical safety systems, which is also the subject of his textbooks. Eleven dissertations have been defended under his scientific supervision. Prof. Hristov is the author of over 330 scientific papers including 32 books (textbooks, manuals, and monographs), 31 inventions and more than 50 scientific projects. He was awarded Honorary Gold Medal of the Technical University of Sofia and was elected a member of the Transport Academy of Russia, "Doctor Honoris Causa" of the St. Petersburg State University of Railways.

Mariya Hristova graduated the Sofia University "St. Kliment Ohridski" as a Mathematician. She received Ph.D. degree in Automated control systems and information processing .The current position is Professor in Informatics and Computer Science in Department of Mathematics and Computer Science and Vice Dean of Faculty of Telecommunications and Electrical Equipment in Transport in Todor Kableshkov University of Transpor, Sofia. Her field of interest includes: mathematical modeling in engineering science, safety critical sistems, information and communication technologies, expert systems and artificial intelligence; information systems and databases; E-learning; information and management components of higher education, e-bisiness. Mariya Hristova is the author of over 130 scientific papers, textbooks, manuals and monographs and participant and manager of more than 30 research projects.