Bochkov K.A., Sivko B.V.

SELECTION AND DEFINITION OF SAFETY FUNCTION WHEN VERIFYING RAILWAY SIGNALLING AND REMOTE CONTROL COMPUTER-BASED SYSTEMS

The paper considers the issues of definition, formalization and selection of safety function used for the development and correctness demonstration of software of railway signalling and remote control systems. It also provides ways of searching for and changing of safety function based on a specification, limitation of resources, an applied safety strategy and general requirements for system performance.

Keywords: *verification, validation, functional safety, safety function, correctness demonstration, critical computer-based objects.*

Nowadays, new developments widely use computer base in railway signalling and remote control systems, which extends their performance features and makes it possible to implement and provide wider functionality for operating systems. At the same time, however, development, verification and subsequent operation of these systems should correspond and satisfy to the safety level adopted in railway industry. Traditionally, railway signalling and remote control systems used relays, when the construction was based on the principle of hardware implementation of safety functions, while computer-based systems are hardware-software complexes (HSC) in which the majority of functions are implemented in software. At the same time the use of COTS technologies prevails in constructing modern computer-based railway signalling and remote control systems, whereas the development of software (SW) is the most complicated element of these systems. Besides, for computer-based railway signalling and remote control systems there are no uniform, universal and generally accepted methods of safety proof, and in this reference one should apply a complex of methods and means to increase a safety level at all stages of a system life cycle, and an urgent task is the development of new safety case techniques.

One of the possible ways of search for errors and improvement of SW quality within the framework of a used complex of approaches is the demonstration of correctness, which belongs to formal methods [1] and is successfully used for verification of microprocessor devices on the Belarus railway [2, 3, 4]. For the systems associated with safety, standard IEC 61508 has a range of safety integrity levels from SIL 1 up to SIL 4, and railway systems should correspond to the most rigorous level SIL 4, which urgently recommends application of formal methods for critical control systems [5].

As the demonstration of correctness, formal methods can be applied for ready-made SW as well as at early stages of development of the entire HSC, but in any case, one of the first steps of verification is the definition of a safety function subject to the demonstration of correctness [4, 6].

A safety function represents a formalized condition in relation to verified system, whose satisfaction allows us to make the conclusion about a function safety of railway signalling and remote control systems. For the same HSC, a safety function can be defined differently, and a condition to be satisfied can be chosen at different stages of a system life cycle.

The development and application of SW as well as a number of studies show that the later the error is detected, the more difficult is to reveal and correct it, and more problems can be caused by such error [7, 8]. It should be noted that the correction of errors made at the stage of preparing system requirements is ten times more expensive than that of errors made during system implementation [9, 10]. The definition of a safety function, which belongs to formalization of a problem to be solved, is a specification in relation to the demonstration of correctness and possesses the same properties, as the statement of requirements for SW development. The potential errors made at the stage of defining a safety function, negatively influence the quality of verification and can lead to distortion of results of correctness demonstration and, as a consequence, to its full revision.

Fig. 1 shows the sequence of safety analysis stages with defining a safety function.

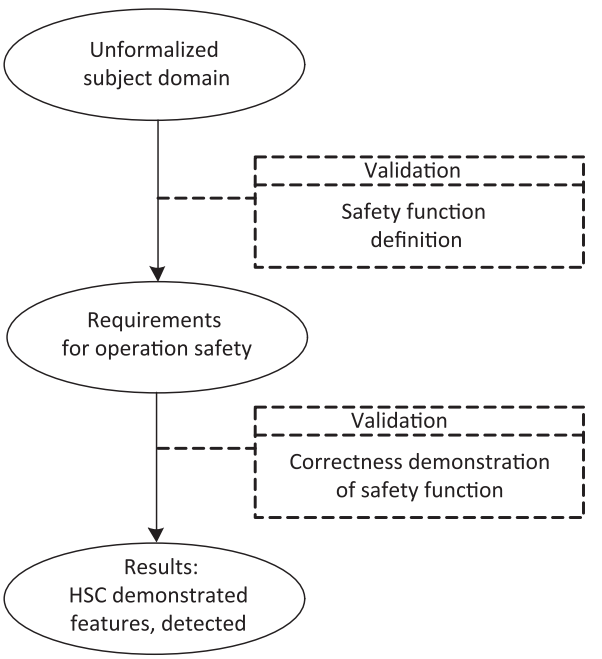

Fig. 1. Sequence of safety analysis stages

Conditions for defining a safety function are defined at the stage of validation based on the characteristics of components used, a variety of methods used, safety strategy and practical experience in the subject domain under consideration [11]. The given process is independent of the subsequent verification: it defines test parameters and forms the initial data, based on which a safety function used in the demonstration of correctness is specified. If errors are made at the validation stage, or behavior features influencing safety are not considered, then it directly influences the quality of a subsequent verification. Also, however effective and diverse methods and means were used during correctness demonstration, they are not capable to reveal and correct problems made during designing as they work according to the same specification, and the end user only can identify an error made at the stage of preparing requirements.

The global experience of application of safety critical systems shows that accidents and disasters occur because of a set of factors, and the significant part of incidents take place due to mistakes made at the stage of preparing system requirements [11, 12, 13]. For example, there are a lot of reasons why a sea ship can be subject to risks: collision with an iceberg, corrosion, explosion of cargo, etc. It is not necessary for engineers to know all sources of risks, but at the same time the formalized decisions for minimization of hazards can be taken during designing: a ship should stay afloat at the specified limit quantity of leaks, saving means should be available, and preliminary actions should be carried out. The steamship Titanic was designed to stay afloat in case of flooding 4 or less first compartments, and this can be named as a safety function. Unfortunately, collision with an iceberg led to flooding of the five first compartments [11]. A safety function was set based on practical experience and knowledge of that time and upon its definition the system was designed on its basis. But, as the history shows, such approach does not mean that all possible hazards are eliminated.

When designing systems, one can accept implicit assumptions, which directly are not related to functional safety but can affect the operation of the whole HSC. For example, an assumption that trajectories of aircrafts will always be above a sea level can lead to SW errors during flights above territories that are below a sea level and to failure of the whole system [14].

Safety conditions of system operation can differ in case of changes of environment or operating conditions that are urgent for railway signalling and remote control systems, and in particular it substantially show itself in case of transition from relays into computer base. For example, when carrying out safety tests of circuits of a route-block relay interlocking, checkout of dependences on point-track sections is carried out once, irrespective of the position of points which are included in the section, and also irrespective of the type and direction of a route set through a section. Independence from the listed factors is conditioned by properties of the first class reliability relay and circuitry for interdependence checkout. However, in case of application of computer base with symmetric failures, HSC does not possess the same properties, and SW verification should be carried out in view of all possible alternatives, which can exist in considered conditions of functioning, with all possible failures of microelectronic elements, according to respective standards, taken into account.

Thus, one of the validation problems is the definition of conditions subject to checkout, and features of the given process are such that after formalization there is no unambiguous criterion and confidence that the adopted function to be proved and demonstrated is necessary and sufficient [15]. During subsequent development or safety analysis it can be found out that the framework set is too strict and it is impossible to carry out correctness demonstration, or on the contrary, the framework is too weak, and thus the probability of detection of SW errors decreases.

Railway signalling and remote control systems possess complexity, due to which strict conditions of accepted safe behavior are complex in formalization, and thus their definition can demand a lot of resources and there is a greater probability to make errors. In order to solve the given problem, one can define such safety function where the given disadvantages are absent. Besides, when developing and demonstrating the correctness, there is no necessity in rigorous selection of a safety function – it can be any function satisfying to the conditions presented in Fig. 2.

The description for the specified areas is:

A – For some reason, the system has not met the condition of a demonstrated safety function, but it has not led to a hazardous failure;

B – The system behavior satisfies to the safety function condition.

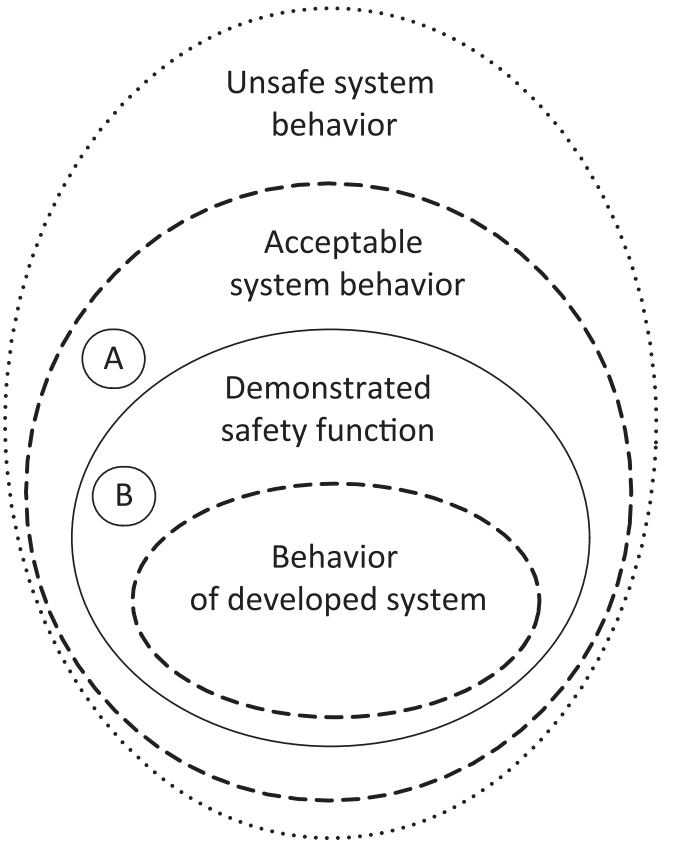

Fig. 2. Selection of safety function for demonstration

Thus, the demonstrated safety function should always be the same or more rigorous than acceptable safe behavior. The behavior of a system under development should satisfy to the condition of a demonstrated safety function.

The experience accumulated by the authors as regards verification of safety critical objects of railway signalling and remote control systems shows that the definition of a demonstrated safety function for developed and existing HSCs should be carried out on the basis of:

– applied safety strategy – for example, during the design stage the strategy of application of logical elements with asymmetrical failures (h_1-reliable elements) can be used and the system should follow it during the whole life cycle [16];

– safety requirements for the whole system – for example, the verified component should stand up to the specified time ranges of signals of interaction with other system components;

– safety requirements for HSC under consideration – for example, the system shall keep invariant and pass into a safe state in case of internal failures.

Thus, the result of verification is the proof that the properties of SW under consideration meet safety requirements for it and for its environment, and also are coordinated with the used safety strategy.

The definition of a demonstrated safety function can be made only on the basis of requirements specification (RS) of a developed HSC, but in this case, safety function formalization may turn out to be difficult and labor-intensive process. In this connection the transition to the demonstration of a more rigorous safety function rather than that was defined based on safety requirements specified in RS and considering the completeness of hazardous failure criteria.

The consideration of requirements for a system as regards a safety strategy, safe internal behavior and coordination of interaction with external components allows us to break down a demonstrated integral

function faster and more effectively into equivalent but simpler functions that reduces the complexity of verification and improves its quality.

The statement of a safety function is not a final decision, and if necessary a safety function can be changed to any other one satisfying to the conditions shown in Fig. 2. The experience of verification shows that transition to other safety function should be carried out when in terms of new consideration the system behavior becomes:

– more determined – It is possible to tell more accurately how the system behaves in those or other situations;

– less complex – there is reduction of costs of safety analysis and time required to understand processes taking place in the system.

The basic methods used by the authors to change a safety function are its extension (easing, weaker definition) and narrowing (strengthening, more rigorous definition). Also, a safety function can be changed not as rigorous easing or strengthening, but in any case it should keep within the framework of acceptable safe behavior (see Fig. 2).

Let us consider three functions of safety f_1, f_2 and f_3, each depending on a vector of arguments $\bar{\alpha}$ and having a range of true/false values. Then strengthening of function f_2 is the transition to such function f_3 when conditions (1) and (2) are met:

$$\forall\ (f_3(\alpha) = true)\quad f_2(\alpha) = true \tag{1}$$

$$\exists\ (f_2(\alpha) = true)\quad f_3(\alpha) = false. \tag{2}$$

The easing of function f_2 is the transition to such function f_1 when the following conditions are met:

$$\forall\ (f_3(\alpha) = true)\quad f_2(\alpha) = true \tag{3}$$

$$\exists\ (f_2(\alpha) = true)\quad f_3(\alpha) = false. \tag{4}$$

Thus, the easing of a function consists in the transition from one function f_2 to another function f_1 in such a manner that always when f_2 is true, then f_1 is also true, but at the same time there are such true values f_1 when f_2 is false. Strengthening is similar to return transition. Graphically relations between functions are shown in Fig. 3.

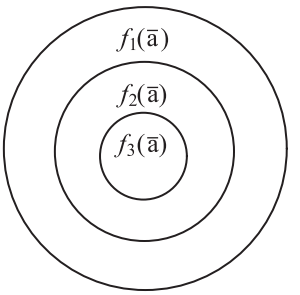

Fig. 3. Strengthening and easing of safety functions

Let us consider an example of safety functions whose choice can affect the demonstration of correctness. We shall assume that there is HSC SW whose whole functionality is carried out in a closed cycle where each subsequent execution should be distinguishable from the previous one and for this

purpose the identifier *id* is stored in the memory. We shall believe that the number of cycles is finite and each of them is numbered sequentially in time from *1* up to *n*, and, accordingly, there is a set of identifiers $\alpha = \{id_1, id_2, ..., id_n\}$.

For the case under consideration we shall introduce several alternatives of a safety function. The first example:

$$f_1(\alpha) = true, \forall\ (i \in N, i < n) \quad id_i \neq id_{i+1}$$

The given function guarantees the difference of the identifier from the previous one and can be used for safe updating of the input information.

The second safety function ensures uniqueness of the identifier for all operating time of HSC from the moment of its operation start and can be used for updating of the information, which takes place not on each turn of a full cycle:

$$f_2(\alpha) = true, \forall\ (i \neq j \in N; i, j \leq n) \quad id_i \neq id_j$$

The following safety function ensures that each subsequent identifier is exactly by 1 more than the previous one and this function can be used for quantity calculation of full cycles between events:

$$f_3(\alpha) = true, \forall\ (i \in N, i < n) \quad id_{i+1} = 1 + id_i$$

Function f_3 is more rigorous than function f_2, which in turn, is more rigorous than function f_1.

To verify a more rigorous function is more complicated than to verify a weaker function as a lot of resources is spent for it and it is not always possible. But, if there is an opportunity, then it is recommended to demonstrate correctness of a more rigorous function as it has the following positive effects:

– obtaining more exact representation of how the system operates – its properties and behavior are defined more strictly;

– reducing complexity of analyzed operation and by that increasing the probability of detection of errors;

– proved or demonstrated functions can be further used for more effective implementation of other correctness demonstrations for HSC under consideration.

However, in case of safety analysis when it is impossible to demonstrate correctness in the offered form or there are no resources for carrying out such amount of works, the easing of a verified function is possible provided that it will allow us to making a conclusion about SW safety.

The definition of a safety function for verification purposes is an important stage of safety analysis and its choice represents a trade-off between available resources and properties to be demonstrated. Practice shows that we can avoid a trade-off only in case when a system is prepared to be verified, when problems of safety function definition are solved before stages of development and designing, which is only possible for HSC to be designed and developed [4, 6].

The described definition of safety function choice has been tested on verification of SW for communication devices with outdoor devices of railway signalling and remote control systems, such as remote control units 16-1 [2], optical LED systems [3] and TU-8B and TC-16B multiprocessing units of “Iputj” computer-based interlocking [6].

Thus, the developed general principles of safety function construction allow us to improve the formalization of specifications of safety cases for HSC SW as part safety critical systems, and the problem of a safety case is simplified if during designing the methods of development of testable SW are used.

References

1. **Butler R.W.** "What is Formal Methods?" NASA LaRC Formal Methods Program, 2001.

2. **Sivko B.V.** Correctness demonstration of the 16-1 remote control unit for "Niemen" centralized traffic control // BelGUT Bulletin: Science and Transport. - 2012. #1 (24). – pp. 18-21.

3. **Harlap S.N., Sivko B.V. S**oftware verification for microprocessor based optical LED systems // BelGUT Bulletin: Science and Transport. - 2012. - #1 (24). – pp. 22-25.

4. **Sivko B.V.** Safe software designing of microprocessor based devices of railway signalling and remote control systems // Problems of safety on transport: Report synopsis, VI International Scientific and practical Symposium, Gomel, November, 29-30, 2012 / Ministry of Education - Belarus, Ministry of transport and communications, Belarus State University of Transport, - Gomel, 2012. – p. 205.

5. **David Smith J.** "Safety Critical Systems Handbook. A Straightforward Guide to Functional Safety, IEC 61508 and Related Standards, Including Process IEC 61511 and Machinery IEC 62061 and ISO 13849" / David J. Smith and Kenneth G. L. Simpson // Elsevier Ltd., 2010.

6. **Sivko B.V**. SW correctness demonstration for multiprocessing devices of communication with facilities of railway signalling and remote control systems // BelGUT Bulletin: Science and Transport. - 2012. - #2 (25). – pp. 27-30.

7. **Fagan M.E.** Design and code inspections to reduce errors in program development, IBM Systems Journal, Volume 15 Issue 3, September 1976, p. 182-211.

8. **Boehm B. W.** Software engineering. IEEE Transactions on Computers 25:1226-1241, 1976.

9. **Telles M., Hsieh Y., Telles M.A.** The Science of Debugging // The Coriolis Group, 2001.

10. **Boehm B.W., Papaccio P.N.** Understanding and controlling software costs // IEEE Trans Softw Eng 14:1462-1477, October 1988.

11. **Nancy G. Leveson,** Software safety in embedded computer systems. Communications of the ACM, 34:34-46, February 1991.

12. **Charles Perrow.** Normal Accidents: Living with High Risk Technologies. Basic Books, New York, NY, 1984.

13. **Ivars Peterson**, Fatal Defect: Chasing Killer Computer Bugs, Times Books, New York, 1995.

14. **Nancy G. Leveson.** Safeware: System Safety and Computers. Addison-Wesley, 1995.

15. **Gerhart S.L., Yelowitz L.** Observations of Fallibility in Applications of Modern Programming Methodologies // IEEE Trans. Software Eng., vol. 2, no. 3, 1976, pp. 195-207.

16. **Sapozhnikov V.V., Century B., Kravtsov Ju.A., Sapozhnikov Vl.V.** Discrete devices of railway signalling and remote control systems // M. Transport, 1988.