Software requirements for the control systems according to the level of functional safety

D. GABRIŠKA

Abstract

The article describes the main requirements of the software subsystems management development. Standard IEC 61508-3 provides an overview at all stages of the life cycle of all security systems, including E/E/PE of a security system from initial concept, design, and implementation to operation maintenance. In this paper we analyzed set out requirements for the drafting of a software architecture that is consistent with the hardware architecture while meeting specified requirements for software safety.

General Terms: Functional Safety, Safety Integrity Level, Software Security Systems
Additional Key Words and Phrases: Process Safety Management, Lifecycle, Safety Functions, Standard IEC- 61508, Control Systems

1. INTRODUCTION

In most the cases safety of process control systems is defined by the reliability of separate subsystems of management. Their reliability is reached due to the use of several systems in which various technologies are applied.

The programmable electronic E/E/PE elements (electrical/electronic/programmable electronic) are most often used. The strategy of safety must then include not only all elements which are a part of separate systems (for example, the sensors, operating and actuation mechanisms) but also the reliability of programs of systems of safety [Michaľčonok and Korytár, 2014].

Standard IEC 61508-3 considers all stages of the life cycles of the entire security system, including the E / E / PE safety software system from initial concept, design, and implementation to operation maintenance [Standard IEC 61508, 2010]. The purpose of this article is to define the main requirements for development of the software of subsystems of management, which would correspond to the necessary level of functional safety of SIL (Safety integrity level)[Korytár and Gabriska, 2014].

10.1515/jamsi-2016-0002

2. CLASSIFICATION OF THE SOFTWARE FUNCTIONAL SAFETY

A model like V-model of the life cycle of software security systems E/E/PE is used as standard for the integrity of software safety. If in the certain E/E/PE safety system life cycle is used, then it must be specified as a part of control function safety activities, a part of all the objectives and fulfillment of requirements [Michaľčonok and Korytár, 2015].

According to the IEC 61511-3 software are divided into three types:

- application software;
- utility software, i.e., the software tools used to develop and verify the application software;
- embedded software, i.e., the software supplied as part of the PE;

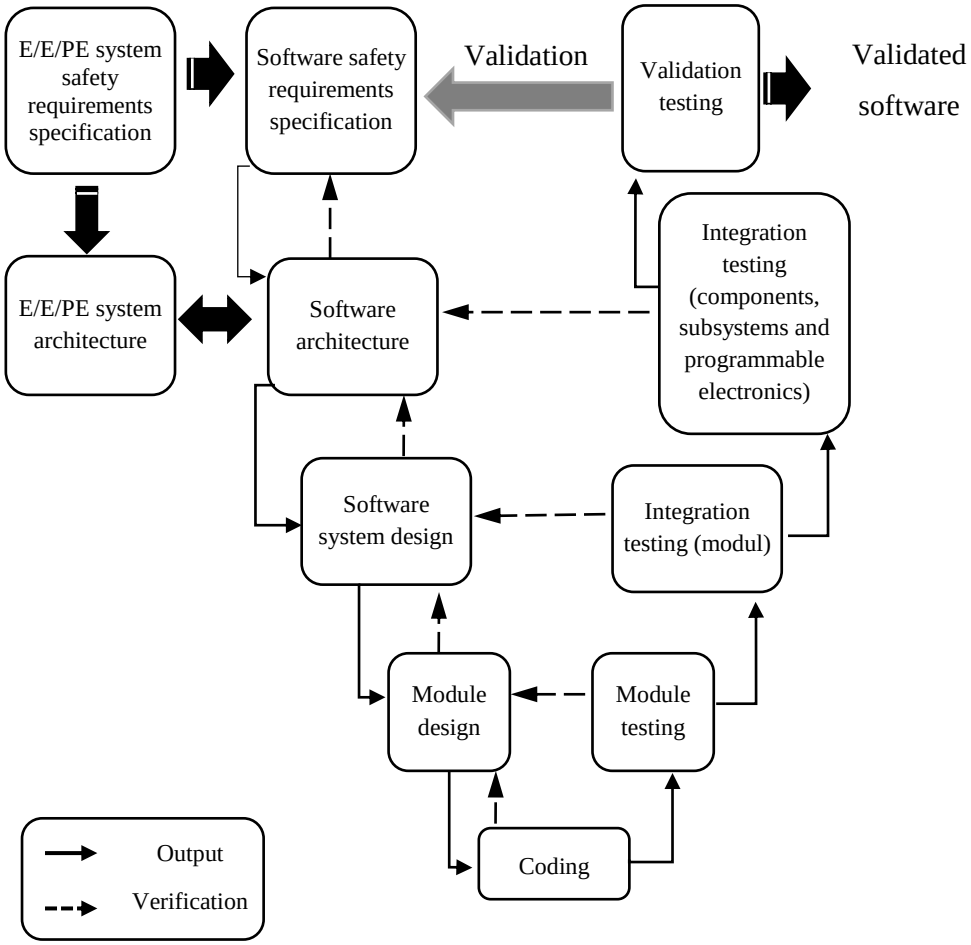

Fig. 1. Software systematic capability and the development lifecycle (the V-model) [Standard IEC 61508-3, 2010]

Software development standard defines three types of languages [Standard IEC 61511-1, 2003]:

- fixed program languages (FPL);
- limited variability languages (LVL);
- full variability languages (FVL).

3. REQUIREMENTS FOR DEVELOPMENT OF APPLICATION SOFTWARE

The following information must be available before beginning of detailed software application design:

- specification of software safety requirements;
- a description of the application software architecture, including the identification of the application logic and functionality descriptions of potential errors, a list of input and output data, software module and general support instruments to be used, and the procedures for programming application software.

Hazardous function of systems with safety levels higher than those defined in the safety integrity level 4 are converted into the instrumentation. Applications that require the use of a single instrumental function with the fourth level of safety integrity are rare in industrial processes. It is necessary to avoid such a system and maintain a high level of performance throughout the life cycle of the safety system if is it possible. In the case of the introduction of such systems into practice throughout the duration of the safety lifecycle a high level of capability of all participants is required. If results from the analysis of the 4-th security level are transferred to the instrumental function, it is necessary to either consider changing the design of the entire process such that the process becomes safe or add the next security layer to ensure safety. These improvements should reduce the requirements for safety integrity level for instrumentation functions.

Safety functions of safety integrity level 4 shall be permitted only if the following criteria in either a) point or simultaneously both b) and c) points are met.

(a) There has been an explicit demonstration, a combination of appropriate analytical methods and tests fulfilled from the target case safety integrity failure measure.

(b) Safety function were carried out from extensive operational experience components. It can be used as part of the instrumentation features.

(c) There is sufficient information about the hardware failure obtained from components used in the instrument safety functions. The data must be such as to

ensure the necessary confidence in the integrity of the security hardware. Hardware is focused on the measures failure that must be applied.

4. SPECIFICATION OF THE SAFETY REQUIREMENTS OF APPLICATION SOFTWARE

The main aim of the requirements is to create a design of software architecture that is compatible with the hardware architecture while meeting specified requirements for software safety. Another task demands a review and evaluation of the various requirements imposed on software from hardware to software architecture. For example, side effects of the behavior of the hardware or software applications inherent margin of error, the interaction between hardware and embedded software, application software architecture for safety. It is important to select and determine the appropriate set of tools (including software tools) to develop application software. Finally, it is necessary to determine whether the software safety requirements in terms of required safety of software have been achieved.

Development, testing, verification and validation of the application program using a fully variable language must be done in accordance with IEC 61508 as well as the method of construction shall be in accordance with the development tools and the restrictions on the use of the subsystem. The chosen design method and the language should have modularity features. Software should be built on the proven software modules. The modules can include various user-defined library functions and rules for linking software modules. If the application software has implemented safety instrumented function and different levels of safety integrity, such a software belongs to the highest security level.

The actual standard IEC 61511 describes the application software developed by FPL and LVL. Requirements for the development and modification of application software are suitable for up to SIL 3. Therefore, this standard does not distinguish between levels SIL 1, 2 and 3. The development and modification of application software by FPL or LVL up to SIL 3 must comply with this standard. Development and adaptation of SIL4 application software must comply with IEC

61508. The development and modification of application software using FVL must be in compliance with IEC 61508. The initial specifics of software safety requirements for each subsystem must include [Standard IEC 61511-1, 2003]:

- Set safety requirements SIF (Safety Instrumented Function);
- Requirements arising from the very architecture;
- All safety requirements planning.

Specification of the safety requirements of application software must be sufficiently detailed, so that the design and implementation to achieve the required safety integrity enables the assessment of functional safety. The specification must describe:

- functions supported by the application software;
- the capacity and response time performance;
- individual equipment, operator interfaces, and their operational capability;
- all relevant operating modes;
- measures to eliminate incorrect values, i.e. sensor value out of range;
- control tests and diagnostic tests for external devices, i.e. sensors;
- software for self-monitoring i.e. validation of data range;
- monitoring other devices within the architecture, i.e. sensors;
- the regular testing of security features of the device in use;
- characteristic functions of the device in use;
- references to the entry documents.

Specified software safety requirements should be expressed and structured in such a way that it will be apparent to all parties involved in any phase of the safety lifecycle. Great emphasis is placed on the use of terminology and descriptions that are clearly understandable for operators and device administrators and for application programmers.

Specification of the safety requirements of the application software must provide information to enable the correct choice of equipment:

- functions that facilitate the process of reaching and maintaining a safe condition;

- functions related to detecting, reporting bugs in the management of the subsystems;
- functions related to the regular monitoring of security features on-line;
- functions related to the regular monitoring of security features off-line;
- the capacity and response time of performance;
- safety integrity level for each of the above features.

The overall design of application software architecture should be based on the required safety specifications within the constraints of the system architecture. At the same time, the design must comply with the requirements of the chosen design of the subsystem with a set of tools. The actual description of the architecture must provide a comprehensive description of the internal structure of the subsystem and components. Also, it should defines all found elements, including the connections and interactions between the identified components (software and hardware). It should describes the order and function logic processing data with respect to input respectively output subsystems.

CONCLUSION

The main role in satisfying the requirements is designing a software architecture that is compatible with the hardware architecture while it meets specified requirements for software safety. Another necessary task is a review and evaluation of the various requirements imposed on software from hardware to software architecture. Development, testing, verification and validation of the application program using fully variable language must comply with the standard IEC 61508. The overall design of application software architecture must be based on the required safety specifications within the constraints of the system architecture. At the same time, the design must comply with the requirements of the chosen design of the subsystem with a set of tools.

REFERENCES

Gulland, W.G.: Methods of Determining Safety Integrity Level (SIL) Requirements - Pros and Cons, Proceedings of the Safety-Critical Systems Symposium, 2004, Springer pp. 105-122

Janota, A., Spalek, J., Brtková, Z., Hrbček, J.: Úrovne integrity bezpečnosti v iniciatíve eSafety. [online]. Available on internet: http://kris.uniza.sk/janota/dokumenty/NavAge06-023l.pdf

Korytár, M., Gabriska, D. Integrated security levels and analysis of their implications to the maintenance. In *Journal of Applied Mathematics, Statistics and Informatics*. Vol. 10, No. 2 (2014), pp. 33-42. ISSN 1336-9180.

Michaľčonok, G., Korytár, M., Tanuška, P. The use of Petri Nets in safety - critical control systems. In *Applied Mechanics and Materials : Novel Trends in Production Devices and Systems II. Special topic volume with invited peer reviewed papers only*. Vol. 693 (2014), pp. 104-109. ISSN 1660-9336.

Michaľčonok, G., Korytár, M. Integration of principle of functional security to the process control system. In *IV International scientific – practical conference „Actual problems of modern science", Alushta 27.-30.4.2015*

STANDARD IEC 61508-3 2010. *Functional safety of electrical/electronic/ programmable electronic safety-related systems. Part 3: Software requirements.*

STANDARD IEC 61511-1 2003: *Functional safety — Safety instrumented systems for the process industry sector. Part 1: Framework, definitions, system, hardware and software requirements.*

Darja Gabriška
Department of Applied Informatics and Mathematics,
University of SS. Cyril and Methodius, 917 01 Trnava,
Slovak Republic
email: darja.gabriska@ucm.sk