FORMAL METHODOLOGY FOR SAFETY-CRITICAL SYSTEMS ENGINEERING AT CERN

F. Valentini, T. Hakulinen, L. Hammouti, T. Ladzinski, P. Ninin, CERN, Geneva, Switzerland

Abstract

A safety-critical system is a system whose failure or malfunctioning may lead to an injury or loss of human life or may have serious environmental consequences. The Safety System Engineering section of CERN is responsible for the conception of systems capable of performing, in an extremely safe way, a predefined set of Instrumented Functions preventing any human presence inside areas where a potential hazardous event may occur. This paper describes the formal approach followed for the engineering of the new Personnel System of the PS accelerator complex at CERN. Starting from applying the generic guidelines of the safety standard IEC-61511, we have defined a novel formal approach particularly useful to express the complete set of safety functions in a rigorous and unambiguous way. We present the main advantages offered by this formalism and, in particular, we will show how this has been effective in solving the problem of safety function testing leading to a major reduction of time for the test pattern generation.

INTRODUCTION

In environments where safety of human life is a major operational constraint, the engineering and the validation of *safety-critical systems* represents always a big challenge for the engineers in order to ensure that the system will behave correctly even under extremely unlikely conditions. The risk assessment for a site like CERN highlights both industrial and radiological risks [1]. The host states' regulatory bodies ensure that adequate measures are taken to guarantee personnel and environmental safety. To avoid severe sanctions that might even lead to the closure of the site, adoption of a strictly formal methodology for the phases of system design, development, and validation is highly recommended.

For the phases of design and development of the new Personnel Protection System (PPS) of the PS complex, we closely referred to the safety standard IEC-61511 [2]. The IEC-61511 covers in detail all aspects related to the preliminary risk analysis, identification of all protection and mitigation barriers including Safety Instrumented Functions (SIF) specification, allocation of a target Safety Integrity Level (SIL) to safety functions and probabilistic reliability analysis of the system architecture. Additionally, it provides essential guidelines for the future system operation and maintenance periods. The close adherence to these guidelines ensures that the developed protection system fully accomplishes all its safety objectives and that its architecture conforms to the reliability level determined by the risk analysis.

For the final system verification and validation task we defined a specific methodology for *black box* testing that relies on the *Model-Based Testing* [3] approach, which can improve the safety validation process.

The problem of testing is a crucial step for the commissioning of safety systems and it represents always a big effort in terms of execution time and cost. Only a high degree of test coverage may guarantee detection of all major errors affecting the behavior of the safety functions. However, an exhaustive testing campaign, ensuring verification of the complete system input state space, may not be feasible for most parts of PLC-based applications (typically handling hundreds or thousands of inputs). The testing state space, in fact, grows exponentially with the number of system inputs considered.

In 1972, in a debate about the establishment of programs correctness, *Dijkstra* claimed that '*software testing can be used to show the presence of bugs, but never their absence*'; this for highlighting the fact that a formal approach for the testing problem was essential in order to increase the quality and the level of confidence of any validation test plan [4].

The Model-Based Testing idea consists of automatically defining and generating only a strictly minimum set of **relevant** test patterns starting from a formal model of the system's technical specification. A given set of tests can be considered as *relevant* if it has a high probability of spotting a certain family of errors. The central point at the base of the Model-Based Testing approach is based on the formal definition of the so-called *test criterion*. A test criterion **C** formally defines what constitutes a relevant test set and it allows measuring its efficiency. It can be defined as the function:

$$C: S \times F \times T \rightarrow [true, false]$$

Where S is the domain of the systems under validation, F the domain of the formal system specifications and T the domain of all possible tests for the system. The function (**C**), determining whenever a test t is adequate, is then expressed as follows:

$$C(s,f,t) = \begin{cases} true : \text{test is adequate} \\ false : \text{test is not adequate} \end{cases}$$

In this paper we present a practical guideline to implement a Model-Based Testing strategy for validation of the safety functions in PLC based safety systems. We will show that relevant test patterns for a predefined test criterion can be automatically derived if a proper formal language is adopted for the specification of the safety functions.

ISBN 978-3-95450-139-7

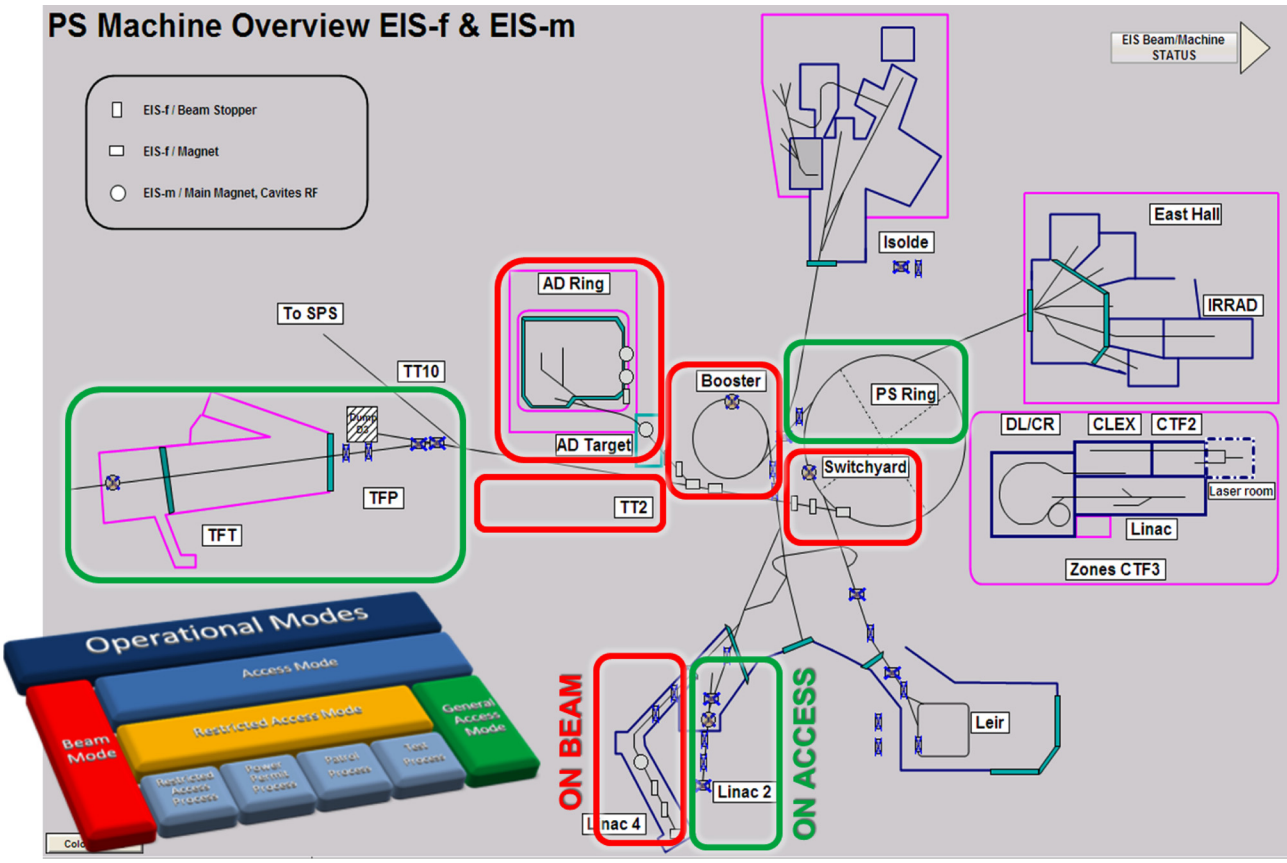

Figure 1: The PS Complex.

CASE STUDY: PS-PPS

As a practical case study to demonstrate our methodology we will refer to the new safety system developed for protecting the personnel accessing the PS Accelerator Complex at CERN [5].

The PS Complex, shown in Figure 1, includes several particle accelerators and is the starting point of every particle physics experiment conducted at CERN. Today it is composed of 17 independent zones, each one dedicated to a particular research activity or used to deliver beams to other CERN particle accelerators, the SPS and the LHC.

The renovation of the PS safety system was motivated by the obsolescence of the existing system and by the objective of rationalizing the personnel protection systems across the accelerator complexes at CERN to meet the latest recommendations of host state regulatory bodies. The project engineering required particular attention to the following aspects:

- **Operational Independence of Zones**: every PS zone shall operate independently from the others. Consequently, a beam may circulate in a certain zone while other adjacent zones are in access mode, allowing presence of personnel.
- **Global Safety Consistency**: the risk assessment highlights different and specific risks for every PS zone that must be covered by the protection system. Furthermore, if a certain hazard appears in a given zone, in addition to a specific set of safety actions performed *locally*, the PPS may need to perform more *global* safety actions having a potential impact on all other zones of the complex. Such scenarios are treated by a system *global hazard mitigation action*.

In order to guarantee independence from operation and hardware failures of the different zones of the complex, the PPS architecture is based on a highly distributed design. A safety PLC, Siemens series S7-400, runs a specific set of *local* safety functions for its application zone and, in addition, exchanges information (wired signals) with a number of neighbouring PLCs to undertake *global* mitigating actions.

In such operational context there were two major challenges: finding a convenient formalism to express clearly, concisely and completely both *local* and *global* safety functions; and defining an efficient validation criterion capable of spotting all major errors related to interactions between triggering events of the local and global safety functions.

SAFETY FUNCTION MODELLING

In the following sections we show that a proper language, *formal* and at the same time sufficiently *friendly* to be understood also by non-specialists in formal methods, may be defined and directly employed as an

ISBN 978-3-95450-139-7

entry point to a "Model-Based" testing approach to validation of the system safety functions.

In the domain of safety systems, according to the norm IEC-61511, a Safety Instrumented Function is defined as a:

"*function to be implemented by a Safety Instrumented System, other technology safety-related system or external risk reduction facilities, which is intended to achieve or maintain a safe state for the process, in respect of a specific hazardous event*".

Return of experience in developing safety-critical systems at CERN has taught us that a non-rigorous specification of the SIF may have dramatic consequences during the whole life cycle of a system. In particular, use of natural language (often characterized by long textual descriptions) may introduce ambiguities, incompleteness, and misunderstanding directly affecting development of safety software and essential parts of the system architecture. It must also be considered that problems, deriving from an ambiguous specification of the system safety functions, may be very difficult to detect in the final validation test plan, which is typically prepared using the SIF specification document as an entry point.

Due to these observations we have paid special attention to the definition of a formal template, which has been adopted for the specification of all safety functions of the PS Personnel Protection System. This template consists of three sections fitting on a single page:

SECTION 1: INFORMAL PRESENTATION

Here a first informal snapshot of a function is given highlighting its main safety features, such as the required SIL, the type of redundancy, the operational constraints, the major mitigated hazards and summarizing, via a synthetic textual description, its functional scope.

SECTION 2: I/O INTERFACE DESCRIPTION

In this section all the interactions of a function with the outside world are identified, coded, and formally listed in terms of input / output Boolean variables.

SECTION 3: FORMAL DESCRIPTION

Here each safety function is expressed via Boolean logic formalism. *Boolean formulae* are built on the base of the function's input variables from the previous section and identify all events responsible for triggering a specific function interlock action.

A typical example of our modelling approach is given by the following formula, describing the system interlock actions to be taken in all cases where a risk of circulating beam in a zone accessed by personnel is detected:

$$\mathbf{F_1} = ((MODE\_Acc \mid MODE\_TFA \mid MODE\_Tra) \,\&\, \neg ACC\_Tst \,\&\, \neg ACC\_TfT \,\&\, \neg EISb\_Pos) \mid (\neg MODE\_Acc \,\&\, \neg EISa\_Safe)$$

The Boolean formalism allows expressing the whole set of SIF as a convenient system of equations in which each SIF action may be given as a function of the result of other SIF computations:

$$Y : F_a(X_{a1}, \ldots, X_{an}, F_b(X_{b1}, \ldots, X_{bm}))$$

This schema is particularly efficient for describing the deep logical relations between many local and global SIFs of the system.

AUTOMATIC TEST CASE GENERATION

The first steps to be taken in order to define a qualitative validation test plan for any given software or electronic system consists of the definition of a:

1. **Test criterion**: representing the strategy according to which all *relevant* tests (the ones having higher probability of spotting a certain family of errors) are derived.
2. **Test generation algorithm**: a mechanical procedure allowing counting and derivation of *all and only* possible tests related to a chosen test criterion.

The validation test plan of our set of SIF was based on the following simple test criterion:

Verify all PLC output values for every possible event triggering a SIF action.

The following *test generation algorithm* allowing derivation of the set (T) of tests (t), for a given SIF (φ), was adopted for the above test criterion:

$$T = \{t \mid \varphi(t) = true\}$$

At this point all possible tests, satisfying our test criterion and representing an interlock triggering event for the function (φ), could be automatically generated using MATLAB by simply finding all solutions satisfying the Boolean formula (φ).

A qualitative validation test plan shall, however, consist of a minimum number of tests capable of spotting most of the critical errors in order to reduce execution time and cost. Following this principle we formally identified and excluded every system input combination resulting in an impossible or redundant output. Then, for every SIF, we formalized all *non-relevant* tests (system events that we do not want to test) via a specific group of Boolean formulae that we call *restriction formulae*. As an example, a restriction formula for the expression $\mathbf{F_1}$ from the previous section is given by:

$$\mathbf{R_3} = (\neg MODE\_Acc \,\&\, (ACC\_Tst \mid ACC\_TfT))$$

The above formula describes an impossible configuration of the access modes for the system.

The restricted set of only the **relevant** tests (T) for the safety function ($\mathbf{F_1}$) can at this point be automatically generated by finding all solutions satisfying the following, more complete, formula (ψ):

ISBN 978-3-95450-139-7

$\Psi(t)$: ($F_1(t) = 1$) & ($R_1(t) = 0$) & ($R_2(t) = 0$) & ($R_3(t) = 0$)

The above procedure allows a major reduction of the number of possible tests generated by the Model-Based Testing approach, and it permits formally documenting what has been effectively verified and what has been excluded from the test plan.

For the function (**F_1**), this procedure reduced the *total input state space* from 128 possible tests to simply 10 relevant tests as shown in Figure 2.

	MODE_Acc	MODE_TFA	MODE_Tra	ACC_Tst	ACC_TIT	ElSb_Pos	ElSa_Safe	RESULTS
Test 1	0	0	0	0	0	1	0	
Test 2	0	0	0	0	0	0	0	
Test 3	0	0	1	0	0	0	1	
Test 4	0	0	1	0	0	0	0	
Test 5	0	0	1	0	0	1	0	
Test 6	0	1	0	0	0	0	1	
Test 7	0	1	0	0	0	0	0	
Test 8	0	1	0	0	0	1	0	
Test 9	1	0	0	0	0	0	1	
Test 10	1	0	0	0	0	0	0	

Figure 2: Relevant test case generation.

RESULTS

The testing strategy presented in this paper has been adopted for the validation of all the safety functions computed by the PLC controllers of the first 9 zones of the PS Personnel Protection System, which is under commissioning.

The development team responsible for all the PLC safety software defined and performed (manually) an initial phase of tests. The final validation campaign was conducted once the development team was confident with their software. In spite of this, a consistent number of errors on every PLC controller were spotted by our test cases. In addition to a number of errors at the human-machine interface level, critical errors related to an incomplete or erroneous implementation of certain critical safety functions were identified.

Furthermore, thanks to the restricted number of relevant tests generated by our approach, the complete SIF validation procedure did not take more than 2 days for every zone controller, while the testing strategy of the development team required an entire week.

CONCLUSIONS

Return of experience from engineering and commissioning of Personnel Protection Systems at CERN have showed us that, after the risk analysis study, the documentation of system safety functions is a critical entry point for all subsequent system development and commissioning activities. For this reason, adoption of a proper formal specification language can be an essential tool to help safety engineers to comply with the high safety integrity levels required by the safety standard guidelines.

A formal specification language can be considered as **proper** if it is not too abstract (a major criticism against formal methods) and, at the same time, if it is adequate for expressing completely and without ambiguities the behavior of the entire system. It can bring direct advantage in terms of communication between the engineering and development teams and represent the premise for the adoption of a Model-Based testing approach.

REFERENCES

[1] P. Ninin, "IEC61508 Experience for the Development of the LHC Functional Safety and Future Perspectives," ICALEPS 2009 Conference Proceedings, Kobe 12-16 October 2009, Japan.

[2] IEC-61511 *Functional Safety – Safety Instrumented Systems for the process industry sector*; http://www.iec.ch

[3] H. Zhu, P. Hall, J. May, "Software Unit Test Coverage and Adequacy," ACM Computer Surveys, Vol. 29, December 1997, United Kingdom.

[4] F. Valentini et al., "Safety Testing for the LHC Access System," EPAC Conference Proceedings, Genova 23-27 June 2008, Italy.

[5] P. Ninin et al., "Refurbishing of the CERN PS Complex Personnel Protection System," MOPPC059, this conference, ICALEPCS'13.

ISBN 978-3-95450-139-7