Proposal of System Testing Integration into Safety Critical System Design Process Supported by SysML

Spendla Lukas, Hrcka Lukas
Faculty of Materials Science and Technology
Slovak University of Technology
Trnava, Slovakia
e-mail: lukas.spendla@stuba.sk

Abstract—This paper focuses on system testing in designing and developing the process of safety critical systems. The proposal aims at identifying the requirements for system testing of safety critical systems and connects them with system model defined in the SysML language. The design and development process is based on analysis of standards and guidelines carried out in our previous work that focused on system testing process in terms of execution. The proposed model is captured using appropriate SysML diagrams.

Keywords—safety system; safety critical system; system testing; UML; SysML

I. INTRODUCTION

During the development and maintenance of standard systems are constantly used new development environments, languages and approaches for software development. The development of safety critical systems is however specific area in which are mainly used time-tested procedures, processes and approaches. This is mainly due to the fact that new approaches used in standard system do not guarantee that the new system will be error free.

Therefore testing is one of the most important phases of the safety critical system life cycle. Testing must verify not only the system requirements, but also all requirements related to safety as well as real time operation.

In the recent years are for design and development of standard systems used SysML language (system modelling language), supplemented by appropriate methodology. Therefore, our proposal focuses on connection between design and development process of safety critical systems and system model in SysML language. The advantage of this link is clearer and more precise definition of requirement in different phases of design and development process.

A. Safety critical systems

To define the safety critical systems, we must first define terms safety and system safety. Standards and guidelines define safety as freedom from those conditions that can cause death, injury, damage to or loss of equipment or property, or damage to the environment. This concept of safety is inclusive of human safety, which includes workers directly involved in system interactions, workers not directly involved in system interactions, as well as members of the general public [1].

Although this definition is broad, it focuses exclusively on physical consequences, not on operability. However, for systems such as nuclear power plants, damage to or loss of equipment may be meaningful only insofar as it translates into damage to environment or human lives [1].

System safety is usually defined, based on the previous definition as the application of engineering and management principles, criteria, and techniques to optimize safety. These steps must be within the constraints of operational effectiveness, time, and cost throughout all phases of the system life cycle [1].

The term system refers to one integrated entity that performs a specified function and includes hardware, software, human elements, and consideration of the environment within which the system operates [1].

Based on these definitions, safety critical systems can be defined as systems that apply the system safety engineering practices. Typically these systems are used in oil, gas and nuclear power plants, chemical industries, and every industry where some of the safety must be maintained on specific safety integrity level [1].

B. System testing

System testing as a part of testing process is performed after integration resting. The system is tested as a whole for functionality and fitness of use based on the system / acceptance Test Plan. Systems are fully tested in the operating environment before acceptance resting occurs. The sources of the system tests are the quality attributes that were specified in the software quality assurance plan or non-functional system requirements [2].

System testing is a set of tests to verify these quality attributes and ensure that the acceptance test occurs in a relatively trouble-free manner. System testing verifies that the functions are carried out correctly. It also verifies that certain non-functional characteristics are present. Some examples include usability testing, performance testing, stress testing, compatibility testing, conversion testing, and documentation testing [2].

C. System modeling language (SysML)

SysML is a general-purpose graphical modelling language that supports the analysis, specification, design, verification, and validation of complex systems. These systems may include hardware, software, data, personnel,
procedures, facilities, and other elements of man-made and natural systems. The language is intended to help specify and architect systems and specify their components that can then be designed using other domain-specific languages such as UML for software design and VHDL and three-dimensional geometric modelling for hardware design. SysML is intended to facilitate the application of a model based system engineering approach to create a cohesive and consistent model of the system that yields the benefits [3].

The SysML language is a critical enabler of model based system engineering. Effective use of the language requires a well-defined model based system engineering method. SysML can be used with a variety of engineering methods and enables representation of a system from multiple perspectives. Each of the individual perspectives may be complex in their own right, but ensuring a consistent model that integrates across the different perspectives introduces additional challenges [3].

II. SYSTEM MODELING LANGUAGES METHODOLOGIES

SysML is visual modelling language that provides semantics and notation, but not provides methodology or tools. Therefore to use SysML in model based system engineering field, we need to support it by appropriate methodology. When searching for the appropriate methodology for system engineering two methodologies arise to the fore. Specifically we can talk about ICONIX methodology a SYSMOD methodology.

The chosen methodology is important in terms on inputs and outputs of individual activities. The outputs will be used to connect the SysML models with requirements of safety critical systems in various phases of their design and development.

A. ICONIX methodology

The ICONIX methodologies provide the methods for developing SysML models across multiple fields. One of the methodologies focuses on process for embedded systems development by using SysML modelling language [4].

Despite the fact that this methodology focuses mainly on embedded system, it process is straightforward enough to be used also in more general system engineering [4]. The overview of the ICONIX roadmap is captured as SysML activity diagram on Fig. 1.

The process starts with definition of functional and non-functional requirements. Based on gathered requirements is defined problem domain and block system structure. Simultaneously with this activity are created use cases and defined system states that are used to model the system behaviour. Constraints and scripts are defined in appropriate blocks after these two parallel activities are finished. In these steps are also created parametric SysML diagrams that are used in subsequent simulation.

The simulation activity contains also required configuration steps, like definition of input and output parameters and their values. Subsequently the simulation results are verified and validated. If the results pass, the process moves to implementation activities. If they do not pass, system structure and behaviour must be modified to successfully pass this step. The implementation of hardware and software are defined as separate parallel activities. In the Implement software activity is the source code generated using functional code from activity and state machine SysML diagrams. The result of Implement Hardware activity is HDL code based on defined system states.

The last activity in this process is testing of hardware and software, after which the designed and implemented system can be delivered.

B. SYSMOD methodology

The SYSMOD methodology is a pragmatic approach to model the requirements and the functional and physical architecture of a system. It provides a toolbox of tasks with input and output work products, guidelines and best practices. The process was introduced in [3] by Tim Weilkiens, that provides details about the process and the SysML language used [3].

The methodology uses the OMG Systems Modelling Language (SysML). Despite this fact slightly modifications to various elements were introduced to simplify the system engineering process [3].

In contrast with the ICONIX methodology, SYSMOD focuses on system engineering as a whole, and focused more on design of the system that on its implementation. The overview of the SYSMOD process is captured on Fig. 2. The SYSMOD process can be divided into two phases, system analysis phase and system design phase.
The first phase starts with identification of stakeholders and gathering of system requirements. After this activity is created system content based on identified actors, interaction points and information flows. After the system content is created are modelled use cases. These are supplemented with description of system procedures, use cases and object flows. The next steps are activities that can be performed in parallel. These activities define domain knowledge for selected system and create glossary based used requirements.

![Figure 2. Overview of SYSMOD system design process.](image)

When these steps are completed, the process moves to second phase, where the activity Realize Use Cases is performed. In this activity are created system interactions and system interface and is defined system structure.

III. PROPOSAL OF SAFETY CRITICAL SYSTEM DESIGN PROCESS IN TERMS OF SYSTEM TESTING

One of the most important parts in the design and development process is testing. However, before we can design, perform and evaluate testing process, we need to integrate it into process of design and development of safety critical systems. In our proposal we are going to focus on system testing.

A continuity of steps from [5] is illustrated as the UML activity diagram captured on the Fig. 3. This diagram represents an approximate time continuity of particular basic steps of the design process of safety critical computer system.

These steps serve as basis for model capturing process of design and development of safety critical systems presented in our previous work in [6] that focuses on the system testing process from test execution perspective. Among other things, it focuses also on identification of requirements for system testing from multiple perspectives. These identified requirements were used also in our current proposal.

![Figure 3. Basic steps of design and development of safety critical systems.](image)

Particular activities of this diagram can be divided into two phases. Steps illustrated on the left show a design of standard computer control system for required process (or in general equipment under control) without safety functions. The following steps (shown on the right) are needed to secure a system safety which was proposed in the previous step.

Figure 4. System model structure according to SysML.

This process presents a connection between a design of standard and safety critical computer systems. The system testing must be performed on fully integrated system, after the design evaluation step. Therefore the requirements for system testing are built during whole design and development process.

A. SysML design specification structure

As defined in SysML notation the system model is organized into four sections. These sections, captured as SysML use case diagram on Fig. 4, represent the overall system model. For greater clearness and simplicity, this diagram captures only the main section.

For two of the most used SysML methodologies (described in previous chapter), we have identified inputs and outputs for each activity as well as most important internal activities. After this step we have categorized each activity output into appropriate section.

253
The requirements section contains functional requirements, which represent capabilities of the designed system, non-functional requirements that cover areas such as performance and reliability and system context that covers the area of intended system use.

The behaviour section describes the system behaviour from multiple perspectives. This section contains use cases that describe interactions between users and the system, defines important system states and captures the system processes.

The representation of hardware and software structure is defined in structure section. In addition to the block structure, these blocks define also internal parts, ports and connectors. Important part is also definition of problem domain of designed system.

Content of parametric section allows to define physical laws and constraints on system blocks. Thanks to this, it is possible to simulate how the system will behave. This is specifically important to verify that the designed system meets the requirements.

**B. Safety critical system design process in terms of system testing and SysML**

For the proper system testing realization, it is important to connect particular specific requirements and properties of these systems to integrate system testing into the process of design and development of safety critical systems. Requirements and specific steps needed for processing of particular activity were identified for individual steps of process of design and development (shown in Fig. 3).

Specific steps were displayed as the UML sequence diagram and divided into two parts for greater simplicity. We have focused on a static representation of process of design and development. Therefore this diagram is divided into two parts. The first part (in Fig. 5) reflects an identification of functional and operation requirements, and design of basic computer system.

The functions represent basic steps of design and development process and these steps are invoked by the development team. The SafetySystemDesign and SafetySystemSpecification phases represent actions that are required to perform system testing, according the standards and guidelines for safety critical systems.

The last four phases (namely Requirements, Behaviour, Structure and Parametrics phase) represent SysML categories which bind together outputs of appropriate Iconix and Sysmod steps. Self-actions defined on SafetySystemSpecification phase represent data created by previous steps on this phase. It should be also noted, that many of steps defined in this diagram could be also executed in asynchronous order, due to their overlapping.
We have chosen the synchronous model over the asynchronous model, mainly for greater clearness and simplicity.

The first step gatherFuncAndOpReq in a design of safety critical system is specification of functional and operation requirements containing requirements for software as well as hardware.

The Hardware requirements for the controlled process (or equipment under control) defined in system context must be analysed, as they present outputs for containing requirements. The next step analyseHardwareRequirements covers analysis of all requirements related to hardware, based on functional and non-functional system requirements. For the subsequent steps it is also necessary to analyse basic block structure of the designed system obtained by blockStructure step.

The software requirements represented by analyseSoftwareRequirements in Requirements Phase must analyse also specific tasks processed by software and which are responsible for various parts of software functions. Another important part is identification of particular states or system modes represented by analyseStates step. Even if they are not represented by any step directly, requirements from the perspective of running in a real time must be considered, too. Except of definition of particular system states or system modes, the identification of system tasks using the step analyseSystemProcesses is also required.
Because our design of testing is focused on software testing the existing hardware is considered also in the next main step createBasicComputerSystem. The subsequent step createSoftwareDesign contains step required for the system software design. The design is based on hardware and software requirements defined in previous main step, represented by steps hardwareRequirements and softwareRequirements. The next steps focus on analysis of use cases and their processes. Important step is to integrateConstraints and parameters restrictions.

The basic computer system will be represented by software program created by createSoftwareProgram step implementing all previous requirements. Particular software programme must come from previously created design gained through softwareDesign step.

The second part of the diagram (in Fig. 6) reflects the identification and risk analysis, requirements and specific aspects of an architecture design of safety critical computer system, an approval of its design and system testing.

The next main step in the design and development process is defineSafetyComputerSystemArchitecture step, which implements the safety functions into the designed computer control system. From the perspective of a controlled process the architecture of fail-safe or fail-operate types must be used. Except for these, some standard architecture of computer system should be applied in a practice (e.g. simplex, duplex, triplex architecture ...). This approach is captured as selectStandardArchitecture step in our proposal.

The design of final computer system architecture must be based on hazard analysis performed in previous steps, specifications of safety requirements obtained through analyseSafetyRequirements step and identified failure modes based on identification of system states in analyseStates step.

The next main step evaluateDesign, carries out the design and evaluation of a specific safety critical computer system. In this step we have focused more on a concrete design of this system because an area of design evaluation is very specific. The design is influenced by system architecture through the step systemArchitecture and reflects results of previously performed hazard identification. Particular design of safety critical computer system must be based on a proposal of basic computer system and it must implement safety requirements defined through analysis of safety requirements.

This design of safety critical computer system must be evaluated in evaluateSafetyComputerSystem step. We have decided not to capture this step more specifically, due to its complexity and multiple methods which can be applied.

The processing of a system testing must be done after an evaluation of safety critical computer system. An integration test must be done before it, which will secure that particular parts work correctly. The system testing verifies if the designed system works correctly as a whole. A concrete testing process must implement requirements for a testing of particular safety critical systems.

A concrete system testing covers a wide test spectrum. Each of these tests has certain specific aspects which are used by the testing. In the proposed sequence diagram is captured as a simple step, mainly to for clearness of the whole model. System testing requires fulfillment of multiple demands that are created through the whole design and development process of safety critical system. Therefore the complex system testing step would contain mostly all specific steps performed in previous stages. The previous identification presents a summary of the main requirements which must be applied by the system testing of safety critical systems.

IV. CONCLUSION

This article focuses on system testing of safety critical system from the design and development process point of view. The proposal aims on system testing requirements in individual phases of design and development process. The system specification is represented by system model defined as SysML diagrams.

Based on our model proposal, system model created using SysML provides all necessary information on the requirements required by system testing of safety critical system by standards and guidelines for safety critical systems. The most commonly used SysML methodologies provides outputs represented by model that could be used in the subsequent design and development of safety critical system and their system testing.

In the terms of standards and guidelines for safety critical systems it is still needed to create specific outputs required by standards and guidelines for these systems, to satisfy specific processes used in the design, development and testing od safety critical systems.

ACKNOWLEDGMENT

This publication is the result of implementation of the project: “Testing model proposal for control software of safety critical systems” supported by the Support Program for young researchers funded by the Slovak Technical University

REFERENCES


