

Model Proposal for Performance Testing of Safety-Critical Systems

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

Abstract—The main focus of article is the process of performance testing for safety-critical systems. The proposal is focused into safety-critical systems as a part of information systems. The contribution demonstrates the usage of performance test in the test system, with incorporated essential requirements for the safety-critical systems testing. Requirements are based on the analysis of basic standards and guidelines for safety-critical systems. To visualize the process of testing, the model was outlined using UML sequence diagrams.

Keywords—system testing; performance testing; safety critical systems; UML

I. INTRODUCTION

Testing as a part of verification and validation is a very important process. It belongs to one of the most important phases of life cycle and each system must be tested. Software testing approaches for standard systems are well known and widely used in praxis.

However the testing of software aspects of safety-critical systems is specific area, which needs different methods, approaches and strategies as standard systems. Failure of the software in safety-critical systems can have extensive effects on the environment, process or human health. Therefore it is necessary to verify (and also test) security features of the safety-critical systems.

Testing of safety-critical systems is very broad area that cannot be covered with any generic test model. Therefore we had to focus on a particular area. Our proposal is focused on testing communication subsystem and various issues related to communication. These requirements were also implemented in the proposed performance testing model.

A. Performance testing

Performance testing, as part of system testing, is used to determine end to end timing of various time critical business processes and transactions, while the system is under low load, but with a production sized database. This sets 'best possible' performance expectation under a given configuration of infrastructure. It also highlights very early in the testing process if changes need to be made before load testing should be undertaken [1].

The best practice is to develop performance tests with an automated tool, so that response times from a user perspective can be measured in a repeatable manner with a high degree of precision. These test scripts can later be used

again in load testing. This way the results can be compared back to the original performance tests. The performance test is not valid until the data in the system under test are realistic and the software, including configuration, is production like [1].

B. Safety-critical systems

There are many definitions available for safety-critical systems, but the important aspect is the correlation between the system and the prevention of danger. A safety-related system contributes to the assurance of overall safety. It is intended to achieve, on its own or with other safety-related systems, the level of safety integrity (SIL) necessary for the implementation of the required safety functions. Therefore all work activities must be checked for any contributions they may make towards the safety functions of a safety-critical system. With the increasing complexity of such systems, this task is very onerous [2].

The term safety critical usually brings to mind nuclear plants, oil refineries, airlines and other high-visibility application where loss of safety can kill or injure in large numbers. However the term safety-critical applies to wider family of applications in which failure can lead not only to injury or death but property and environmental damage as well [3].

1) *Development of safety-critical systems.* All safety related systems have to be developed, implemented and maintained according to existing national and international regulations, standards and guidelines. The standard IEC/EN 61508 is applicable for all aspects of electrical, electronic and programmable electronic systems, with safety relevant functions and applications. In general, the standard can also be applied on all safety related E/E/PES, even if no specific safety standard exist for this particular area. Therefore IEC/EN 61508 is considered as one of the fundamental standards for the safety-critical systems. This standard provides a systematic and risk based approach for safety relevant problems [4]. The different chapters of the standard are presented in Table I.

2) *Testing of safety-critical systems.* The testing of safety-critical systems is necessary condition for the verification of their safety functions. Standards for the safety-critical systems mainly focus on integration testing. System testing, under which falls also the performance testing, is mentioned only as one of the test that needs to be

carried out. Therefor in 1994, Good Automated Manufacturing Practice (GAMP) Forum partnered with the ISPE organization to publish the first GAMP guidelines [5].

TABLE I. CHAPTERS OF IEC/EN 61508

Chapter	Content
IEC/EN 61508-1	General requirements
IEC/EN 61508-2	Hardware requirements
IEC/EN 61508-3	Software requirements
IEC/EN 61508-4	Notation and abbreviations
IEC/EN 61508-5	Example to calculate the different safety integrity levels (SIL)
IEC/EN 61508-6	Application guidelines for chapters 2 and 3 of this standard
IEC/EN 61508-7	Overview of techniques and actions

Their main focus was to promote understanding of how computer systems validation should be conducted in the pharmaceutical industry. GAMP's guidance approach defines a set of industry best practices to enable compliance to all current regulatory expectations. More than simply a strict compliance standard, GAMP is a guideline for life sciences companies to use for their own quality procedures [5].

The approaches defined in GAMP can be also used for testing any safety-critical systems, since the requirements basis is the same.

C. Standards for software testing

Over the years a number of types of document have been invented to allow the control of testing. They apply to software testing of all kinds from component testing through to release testing. Every organization develops these documents themselves and gives them different names. To provide a common set of standardized documents the IEEE developed the 829 Standard for Software Test Documentation. This standard applies to all software based testing and therefor it can be easily used for any desired test type. The purpose of this standard is to [6]:

- Establish a common framework for test processes, activities, and tasks in support of all software life cycle processes
- Define the test tasks, required inputs, and required outputs
- Identify the recommended minimum test tasks corresponding to integrity levels for a four-level integrity scheme
- Define the use and contents of the test plan for various test types
- Define the use and contents of related test documentation

This White Paper outlines each type of the document contained in this standard and describes how these documents work together. There are eight document types in the IEEE 829 standard, which can be used in three distinct phases of software testing [6].

II. PROPOSAL OF PERFORMANCE TESTING MODEL

The starting point for our performance testing model is the testing standard IEEE 829. It will provide the basis for

our proposal of the performance testing steps. Individual steps of the performance testing were mapped for each phase of the testing standard IEEE 829. This gave us overview on the performance testing process for standard systems.

However for testing the safety-critical systems we need to cover their specific requirements. For their identification we have analyzed the IEC61508 standard and the GAMP guidelines for the safety-critical systems.

These gave us the baseline for further steps. Based on this analysis we have defined specifics of the safety-critical systems in terms of testing. Their generalization serves as the source for the testing requirements that needs to be implemented in our model. Each requirement was assigned to the relevant phase of the testing methodology IEEE 829 and their required actions were identified. Subsequently, these actions were implemented into the performance testing model for standard systems. Given the number of activities, our model had to be divided into smaller views which focus on the specific testing activities. The main view which captures initialization and iterations of the performance test is captured on Fig. 3.

The initial phase Test Plan includes a procedure that describes how the testing will proceed and captures the performance testing process as a whole. In this model we have merged phases Test Log and Test Incident Report to the Test Execution phase, mainly because of model complexity. Their specific task will be further described in partial diagrams.

In the first step of our proposal the test system have to obtain system specification required for creation of the performance model. This performance model is based on analysis of all functionalities provided by the system and also rate of use for these functionalities. These steps are captured with certain degree of abstraction, because it is very large and complex process. They will be more detailed modeled in the next iteration of our proposal.

The next step is analysis of system that is going to be tested. The aim of this analysis is to gather and identify all relevant data required for the testing process. These data were divided into specific steps to provide more details. First of these steps must specify the test objectives through which we should be able to determine expected results of the test and identify support tools and software, that will be needed to carry out the test. After that the current state of the system needs to be analyzed and described for further test evaluation. Next steps will specify the test load (at which the test is performed), reduces the combination of inputs for the test (using mathematical analysis) and states condition when the test stops.

Subsequently, the test system begins performing cycle. Individual sessions are simultaneously created in such quantity as is required by definition of distribution. This distribution represents the rates of use for different functionalities as described in the performance model. For each individual session is generated test script, which consists of various end-user activities.

In the next step, injectors are assigned to the individual sessions. Their number corresponds to the required load. This means we can accurately increase the load of idle

system to the desired level in order to ensure test repeatability. Every injector must load and generate data which are used for the system load in sessions. The data needed for the test evaluation must be recorded for each session. This step is captured with certain degree of abstraction and will be clarified in the separate diagram. If the number of sessions is lower than is required (e.g. one session ended) a new one must be created and the whole process is repeated until the desired load is reached.

It should be noted that our proposal captures only most important steps for performance testing and requirements for the safety-critical systems.

A. Evaluation of performance data

Important steps for evaluation of the individual session were modeled in detail and captured as UML sequence diagram on Fig. 1.

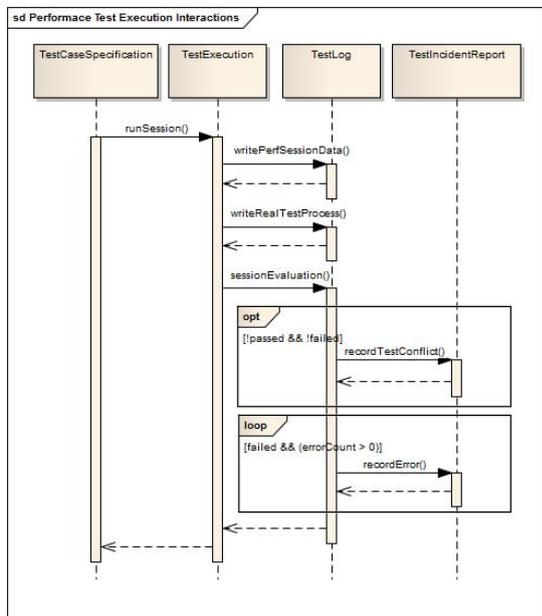


Figure 1. PROPOSAL OF PERFORMANCE TEST EVALUATION STEPS

For unambiguous test evaluation and subsequent analysis it is necessary not only to record the performance data, but also the real process of testing. If the test fails it is necessary to record all identified errors into one or more test incident reports. Due to the complexity of this process, this step was simplified to use only one incident report. If the evaluation of the test is not explicit it is necessary to record all relevant information about this conflict.

It must be said, that this detailed model captures mainly requirements of the safety-critical systems for the test evaluation. Its focus is more on the support activities than on the evaluation itself.

B. Evaluation of performance test

Another important step is to analyze the test results. This process was enhanced with specific steps for the testing of

safety-critical systems and captured as UML sequence diagram on Fig. 2.

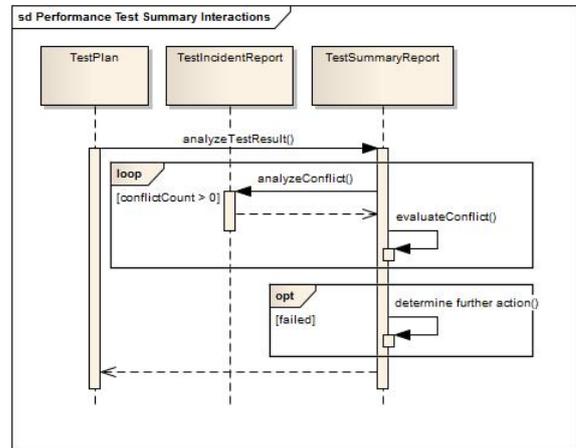


Figure 2. PROPOSAL OF PERFORMANCE TEST RESULT ANALYSIS

According to requirements for the safety-critical systems, each test must pass or fail. Therefore it is necessary to assess all conflicts that arose during the evaluation process. Also if the test failed it is necessary to determine further actions. The test may be repeated once again, the system can be modified using modification management tools or the test (or group of tests) may be excluded.

III. CONCLUSION

The aim of this article is to design the performance testing procedure for safety-critical systems. Our proposal is based on the basic steps of software performance testing with implemented requirements for testing the safety-critical systems, which are captured in our modeled diagram. Proposal is based on the software testing standard IEEE 829. Requirements for the safety-critical systems are based on the IEC/EN 61508 standard and GAMP guidelines. Our design was captured by multiple UML sequence diagrams in UML 2.0.

ACKNOWLEDGMENT

This publication is the result of implementation of the project: "Increase of Power Safety of the Slovak Republic" (ITMS: 26220220077) supported by the Research & Development Operational Programme funded by the ERDF.

REFERENCES

- [1] RPM Solutions Pty Ltd (2004), Load and performance testing of any technology. [Online]. Available: <http://loadtest.com.au/>
- [2] W. F. Bates, "Safety-related system design in power system control and management," in Power System Control and Management, Fourth International Conference on (Conf. Publ. No. 421) , pp.15,20, 16-18 Apr 1996
- [3] W. R. Dunn, Practical Design of Safety-Critical Computer Systems (Book style). Virginia, Reliability Press, 2002, p. 358.
- [4] J. Burcsuk, "Development of safety related systems," in Strategic Technology, 2007. IFOST 2007. International Forum on 3-6 Oct. 2007, pp.564,569, 3-6 Oct. 2007

- [5] J DeSpautz (2008), GAMP Standards For Validation Of Automated Systems, [Online]. Available: <http://www.pharmpro.com>
- [6] L. Špendla, O. Víkovič, P. Tanuška, "Accelerated stress testing automation," in Process Control 2010 : 9th International Conference.

Kouty nad Desnou, 7.-10. 6. 2010, Pardubice : University of Pardubice, 2010.

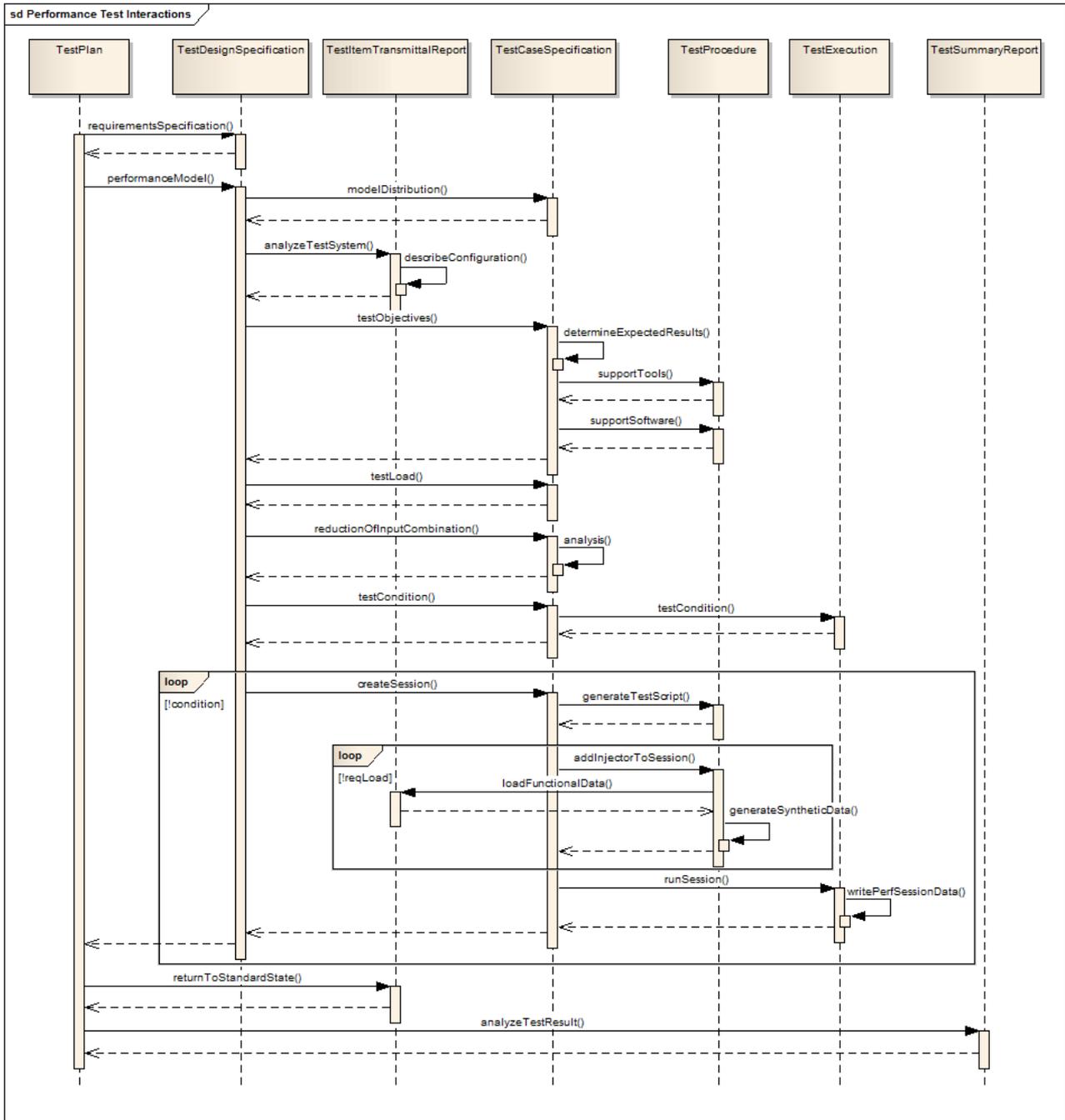


Figure 3. PROPOSAL OF PERFORMANCE TESTING STEPS FOR SAFETY CRITICAL SYSTEMS