UDC 004.05

K.I. NETKACHOVA

National Tavrida University, Simferopol, Ukraine

SAFETY CASE METHODOLOGY: ARCHITECTING PRINCIPLES

The paper presents a general Safety Case construction model, provides a description of its main blocks and their functionality. It also introduces the concepts of Safety Case Cores and Safety Case Infrastructure, which demonstrate a new approach to safety assessment and can serve as a basis for any complex system evaluation. The basic criteria for decomposition are outlined; general principles, implementation details and key features of Safety Case Cores are described and illustrated by an example of OTS-component assessment module.

Key words: *safety case, safety case core, safety case infrustructure.*

Introduction

Safety plays a crucial role in modern society. Assuring safe operation is one of the most vitally important tasks faced by system developers and experts. The concept of Safety Case has been evolving for over 20 years. World famous scientists such as Peter Bishop, Tim Kelly, J Górski and others made a great impact on it through their works [1-4]. The concept has improved, grown and nowadays become a common and generally accepted practice. However the future research is still needed to develop the approach further on and make it even more useful, accurate, efficient, and of course, more automated.

In this paper we present our view of improving Safety Case methodology. Section 1 proposes the overall Safety Case construction model as well as provides a brief description of the main parts of this model. Section 2 introduces some new architecting principles and ideas such as Safety Case Infrastructure and Safety Case Cores with an example of a Safety Case Core given in Section 3. The paper ends with a summary, concluding remarks, and visions for the future work.

1. General approach

The process of safety case creation involves and depends on a number of documents (requirements, standards, specification), assessment methods, source code, documentation, testing results available to the experts. Figure 1 presents the general Safety Case construction model (fig. 1).

The main parts the model includes are:

1) Processing the requirements.

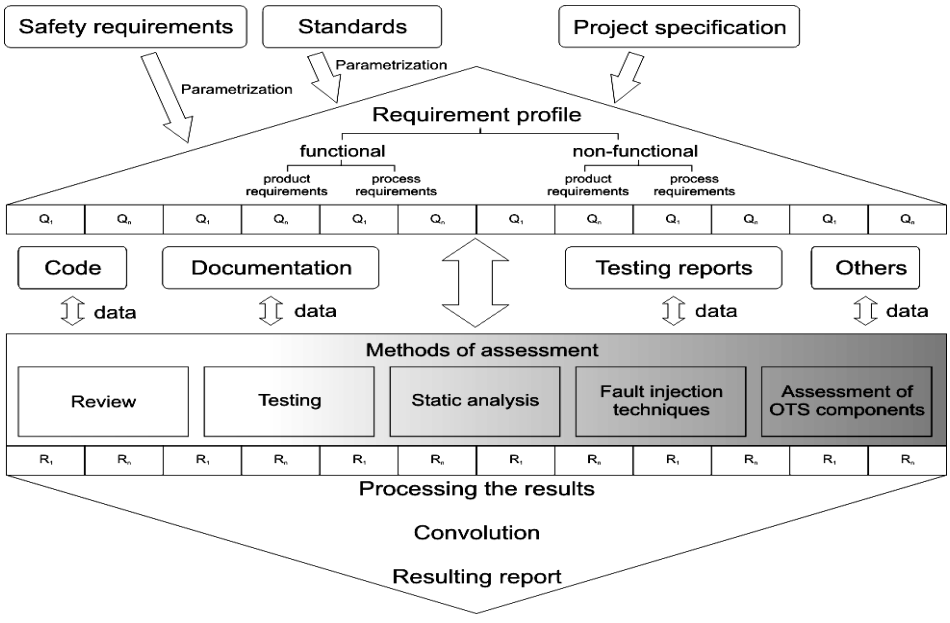

Fig. 1. The generic Safety Case model

System and safety requirements and standards are usually written in natural language. Thus the text should first should be processed and formalized. The resulting requirement profile is passed onto the second part of the model;

2) Data processing.

The data for each requirement is mined from the documentation, code, results of the testing etc. After that the information is processed and used to ensure system compliance with each particular requirement. The detailed description of this part will be provided later in Section 3;

3) Safety Report construction.

The results obtained for each requirement are integrated, convolved and used to construct the overall safety report.

2. Principles of Architecture

At this stage, we want to bring in the conceptions of Safety Case Core (SCC) and Safety Case Infrastructure (SCI). Let us consider Safety Case Cores as separate functional blocks that ensure safe operation of system elements, and Safety Case Infrastructure as a complex of interconnected SCCs ensuring safe operation of the overall system.

To distinguish between different Safety Case Cores we should view the entire system safety assessment process from the standpoint of the independent tasks that need to be performed, and decompose it into a set of units that provide specific functionality. The resulting SCC units should meet the criteria of being independent, with no coupling, extensible, adaptable and potentially reusable in other systems.

Each SCC should have the feature of parameterization. Parameters are the safety requirements established for a part of the system assessed by a particular SCC. They are formalized and serve as input for Safety Case Cores. This can be presented by dividing the data processing block vertically into several components with sockets for their parameterization (fig.2).

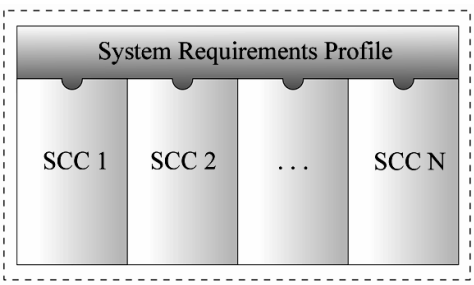

Fig. 2. Parameterized Safety Case Cores

The following processes are carried out for component requirements inside the blocks:

– the data mining (from documentation, code, testing reports etc.);

– analysis of the information to determine whether the system is compliant with each particular requirement.

The functionality of each SCC module can be outlined as follows: there is a list of keywords and phrases related to the functionality a module provides. Based on this keywords, the relevant data is retrieved from documentation, testing reports etc. using keyword matching techniques. After that, an expert may need to perform additional tasks of manually mining the content and organizing the information retrieved. Whenever possible, the data should be represented as precise formal statements to be processed by an automated theorem proving system. In that cases requirements are considered hypotheses that are to be proved by statements retrieved from the documentation and the information contained in a particular SCC.

As a result of analysis, one or several metrics measuring how well each requirement is met are obtained. These metrics are passed on to the block that performs convolution – an operation on metrics and their weighting factors which involves multiplying each metric by its respective factor and summing the results of the multiplications. The weighting factors are defined earlier by experts or derived from the statistics.

By using this technique, we ultimately obtain a weighted sum that represents the total requirement compliance rate of the entire system.

The quantity of SCCs is not fixed, it can increase when new elements are added into the system, the requirements are changed or when there is a decomposition of the existing Safety Case Cores into several blocks for more detailed analysis. Thus, we can talk about infrastructure scalability. The whole system Safety Case is obtained after combining the results of all SCCs. The more detailed the fragmentation in SCC is, and the more factors are taken into account, the more accurate the result of the overall system safety assessment will be.

On the basis of the principles described above we can show how the parameterized model of safety Case document may look like. Let's conventionally divide the system in 4 blocks of functionality that can pose a risk of unsafe operation. These are software, hardware, systems of human management (human element) and the structure and machinery of the information and control systems themselves. Software in its turn can be decomposed into the custom modules and pre-developed OTS components. We transfer the safety evaluation of each of these blocks to the level of separate SCC. Then Safety Case infrastructure of this system can be represented as the interconnection of five SCCs and the means of their parameterization, as shown on fig. 3.

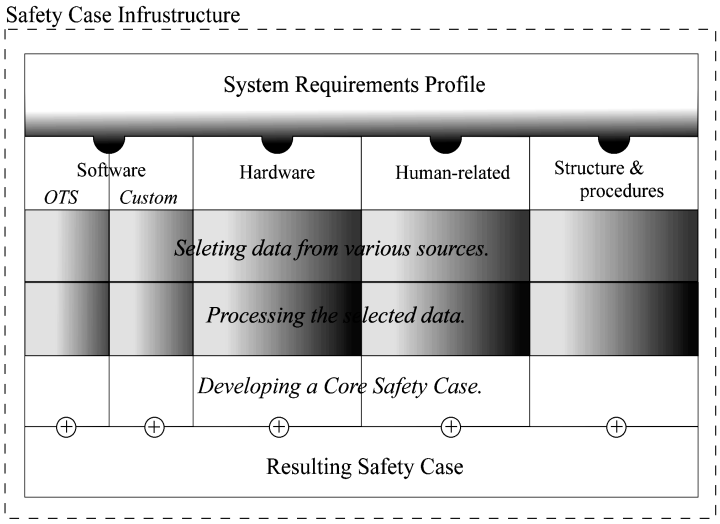

Fig. 3. Safety Case Infrastructure

3. Safety Case Core for OTS components assessment

In this section we describe the first SCC module to clarify how the proposed method works. This SCC returns the reliability characteristics of OTS software used in the system. OTS components are pre-developed software products, often with unreachable code and minimum safety information available. However, there are different third-party vulnerability recourses, groups and databases that have been actively growing and developing in recent years. CVE, NVD, Secunia, SecurityFocus, OVAL, CERT – it is not a complete list of such vulnerability channels.

We can make use of them in our Safety Case Core responsible for OTS components assessment. For example, NVD's [5] vulnerability information is available for free to the public in a convenient xml format. By parsing this xml file, we can obtain the number of discovered vulnerabilities, their severity rates, failure frequency, evaluate recovery time, i.e. assess many of the OTS dependability characteristics.

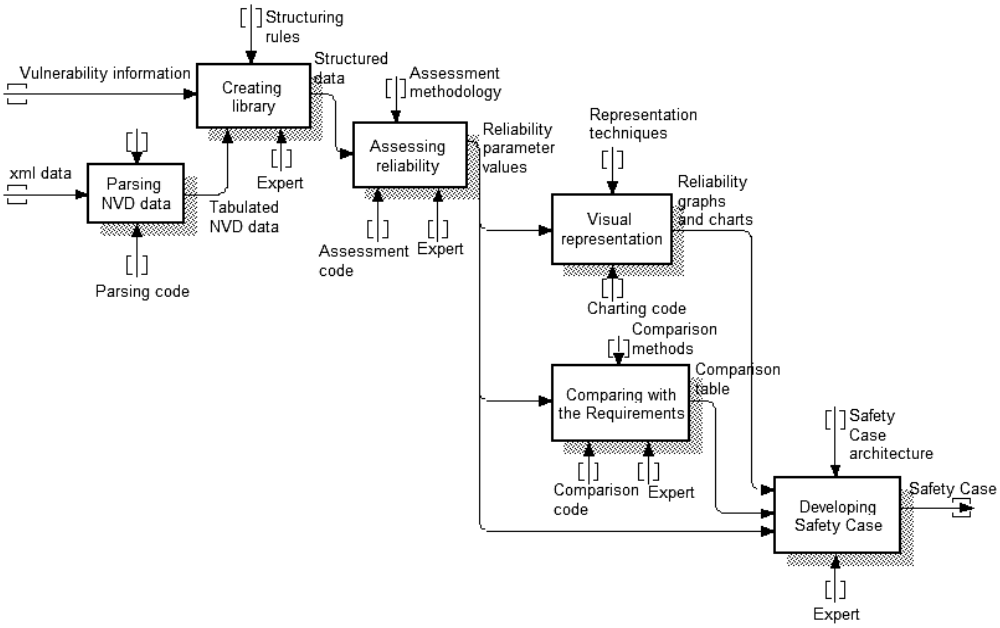

Fig. 4. IDEF/BPWin diagram of OTS module

Our Safety Case Core code searches the system documentation for keywords (OTS component names) and creates a list of OTS products used in the system. These products are assessed by VAT [6] and similar tools provided within the SCC. Experts are free to add their own information or tools to SCC unit as it is designed to be extensible. The resulting dependability characteristics are convolved with the characteristics

returned by Custom software SCC and compared with software requirements received as an input to verify whether the requirements are met or not. The reason we delegated OTS components assessment responsibilities to the level of a separate SCC is because this process is basically independent, it utilizes specific methods, tools and data that are obviously not needed for any other SCCs, and it can be reused in many other systems containing software components. IDEF/BPWin diagram of OTS-component assessment module is presented on fig. 4.

Conclusion

In this paper, we introduced a Safety Case construction model and a technique for architecting Safety Cases. The latter was based on the idea of decomposing the overall system safety assessment process and delegating specific safety responsibilities to the level of separate Safety Case Cores. The proposed approach was demonstrated using OTS-components assessment module as an example. Further effort should be directed toward implementing other common Safety Case Core modules that can be later reused for different systems.

References

1. Bishop P. A Methodology for Safety Case Development / P. Bishop. - 1998.

2. Kelly T. Arguing Safety – A systematic Approach to Managing Safety Cases / T. Kelly. - 1998.

3. Kelly T., Managing Complex Safety Cases., 11th Safety Critical System Symposium, 2003

4. Górski J. Trust Case – a case for trustworthiness of IT infrastructures / J. Górski // Cyberspace Security and Defense: Research Issues, 2005.

5. National Vulnerability Database [Електронний ресурс]. – Режим доступу : http://nvd.nist.gov

6. Lobachova K.I. Assessing Software Vulnerabilities and Recovery Time: Elements Of Technique And Results / K.I. Lobachova, V.S. Kharchenko // Радіоелектронні і комп'ютерні системи. – 2007. - № 8. - P.61-65.

7. Allen J. Natural Language Understanding / J. Allen. - Addison-Wesley, 1995.

8. Andrashov A. Software Quality Assessment and Expertise / A. Andrashov, A. Gordeev, V. Kharchenko, K Lobachova; ed. V. Kharchenko. – K.: National Aerospace University. – 2008. – 153 p.

Поступила в редакцию 15.02.2009

Рецензент: д-р техн. наук, проф. Б.М. Конорев, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков, Украина.

МЕТОДОЛОГИЯ ОТЧЕТОВ ПО БЕЗОПАСНОСТИ: ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ

Е.И. Неткачёва

В работе представлена общая модель построения отчетов по безопасности, приведено описание её основных блоков и их функциональности. Введены понятия модуля и инфраструктуры отчета по безопасности, которые могут служить основой оценки безопасности сложных систем, что демонстрирует новый подход к оценке безопасности. Основные критерии декомпозиции, общие принципы построения и ключевые характеристики модуля отчета по безопасности изложены и показаны на примере модуля оценки OTS компонентов.

Ключевые слова: отчет по безопасности, модуль отчета по безопасности, инфраструктура отчета по безопасности.

МЕТОДОЛОГІЯ ЗВІТІВ ПРО БЕЗПЕКУ: ПРИНЦИПИ ПРОЕКТУВАННЯ

К.І. Неткачова

У роботі представлено загальну модель побудови звітів про безпеку, надається опис основних блоків та їх функціональності. Введено поняття модуля та інфраструктури звітів про безпеку, які демонструють новий підхід до оцінки безпеки і можуть бути основою для оцінки будь-яких складних систем. Основні критерії декомпозиції, загальні принципи побудови та ключові характеристики модуля звіту про безпеку викладені та пояснені на прикладі модуля оцінки OTS компонентів.

Ключові слова: звіт про безпеку, модуль звіту про безпеку, інфраструктура звіту про безпеку.

Неткачёва Екатерина Игоревна – аспирант кафедры компьютерной инженерии, Таврический национальный университет, Симферополь, Украина, e-mail: kate.simferopol@gmail.com.