An integrated approach to implement system engineering and safety engineering processes: SASHA Project

Hycham Aboutaleb, Mohamed Bouali, Morayo Adedjouma, Emilia Suomalainen

An integrated approach to implement system engineering and safety engineering processes: SASHA Project

Hycham Aboutaleb[1,2], Mohamed Bouali[1], Morayo Adedjouma[3], Emilia Suomalainen[1]

[1] Knowledge Inside, Versailles, France {hycham.aboutaleb, mohamed.bouali, emilia.suomalainen}@k-inside.com
[2] UEI, Ensta ParisTech, Paris, France hycham.abou-taleb@ensta.fr
[3]Delphi, Trembley en France, France, Morayo.adedjouma@delphi.com

Abstract

In a purpose of a safe system design, the SASHA project partners, through the graphical software arKItect® have implemented a design process coupling the system engineering process with the safety engineering process. They address especially automotive area through the ISO 26262 standard that appears as an answer for a unified requirements set to fulfill in the purpose of safe vehicle design. The cited standard needs a combined approach to be implemented (workflows, administration, models representation. This paper aims at showing how this safety engineering process is integrated in the whole system design process as proposed and implemented in SASHA project including the system specifications phase and the system design process. This enables to perform the system risk analysis, which is the second step in the safety engineering process.

Keywords: system modeling, safety engineering process, safety analysis, system architecture, integration of systems and safety engineering

1. Introduction

The ever increasing complexity of nowadays systems requires adapted method to master their design and development along the whole life cycle. In the automotive area, risks are very difficult to be managed due to the huge set of possible sources of hazards. The vehicles systems complexity, the interaction with the driver (or other persons) and the influence of the external environment (like road profile and weather) are possible threats to efficient risk management. In addition to great hazards variety, vehicle failures consequences can lead to physical injuries and even more. This context shows the need of an efficient method to design safe vehicles.

In automotive sector, the number of safety critical functions is increasing, especially vehicle dynamic controls, driver assistance functions and the introduction of mechatronics in braking, steering or motor control. As a result there is a need for an improved structured approach in development phase to overcome additional complexity. Indeed, functional safety needs to be included at a very early stage in the development process of systems and their components.

To tackle these challenges, automotive industry partners currently set up the ISO 26262 standard[13], detailing an automotive safety lifecycle supporting the development of road vehicles. This standard built upon IEC 61508 [14], focuses on Electric/Electronic (E/E) Systems but provides a general framework for safety-related systems design.

The ISO 26262 standard gives recommendations about the whole life cycle process and steps in various areas like life cycle management, concept phase, product development (software/hardware) and production. One remaining issue, when vehicle assembler or vehicle systems manufacturer wants to apply this standard, is that no application method is available in the standard texts. So the issue intended by the collaborative project SASHA (***Safety Check of Automotive Software, & Hardware Architectures)*** is to offer an implemented method that ensures, when applied, that the designed system is conforming to the standard.

This paper, organized into three main sections, attempts to illustrate key features on the standard ISO 26262 as designed and implemented in project SASHA using the software arKItect®[1]. The first section is dedicated to show some key issues in safety engineering and the key features of ISO

[1] http://www.k-inside.com/web/

26262 standard. It illustrates nowadays industrial practices, several recommended practices and some implementation/deployment requests. The second sections makes a focus on arKItect® technology and the third shows how the standard was implemented and deployed in SASHA project. The fourth section presents the results of this approach for one of the main car industry actor, Delphi. The paper ends with some conclusions and perspectives.

2. Key issues in safety engineering

Safety engineering efforts are most effective when begun early in the system life cycle. However, in industrial projects, most of the safety engineering effort is usually deferred until late in the project. This is due to practices that emphasize the functional process at the beginning of the design, reducing the safety role, until a decision to expand system safety effort is taken. Besides, this often yields to a discontinuity between the system design and the safety engineering process. In a very competitive context characterized by systems complexity increasing coupled with time/costs pressure on systems development, the use of structured methods to complex systems design becomes unavoidable. Incremental complexity of actual industrial systems and their integrated multidimensional modular design has led to the need of an optimization of the design process to get “the right product at the right time”.

The application of efficient requirement management processes in industrial environments hits against many threats. Most of existing practices disconnect the functional-based system design and its associated process (the system engineering process) and the dependability-based system design and its associated process (the safety engineering process). Moreover, it not unusual in current practices to see functional analysis, that has been performed (or should have been performed) during the system design phase, being performed again by during the dysfunctional analysis. It is mainly due to the fact that the framework used does not support both. This sometimes leads to incoherence between the functional analysis and the dysfunctional analysis.

An integrated system/safety engineering process seems to be the only way nowadays to reach an efficient and optimized design. When performing such a process, the coupling of both processes and the interactions between them rise as the central issue that deserve particular attention all over the development process. [1] [2] [3] Thus, it is needed to clearly define the integrated process as follows:

- Define clearly the steps of the system engineering process

- Define clearly the steps of the safety engineering process

- Define clearly how these two processes are related to each other

- Identify the documents that need to be generated

- Define a unique referential for the integrated process

3. Environment and Tools

To give answer to the issues stated above, SASHA project partners, through the graphical representation tool (arKItect®) and their experiments, proposes a solution for both efficient and seamless system engineering and safety engineering. This solution is “system architecture centric” which means that all activities related to the system design and to the dependability analysis are gathered in the same representation space. The two main items of this proposition are:

1. Keep close connections between all activities of the system design/development (including requirements management, functional architecture, system architecture, ready-to-simulate models, etc.) and of the dependability analysis (including customer risk analysis, preliminary hazard analysis, system risk analysis, fault tree, FMEA) [4] [5] [6]

2. Offer a more ergonomic and intuitive human interface

This paper presents an implemented solution to perform an efficient dependability analysis during the whole system design cycle. It explores several axes: the system engineering process which is based on the INCOSE system engineering handbook, and the safety engineering. Through these axes, a focus is done over the functional analysis, the dependabilty analysis, their associated working/filter views and scripts that generate and update documents automatically.

This paper explains the integrated process shown in Fig.1:

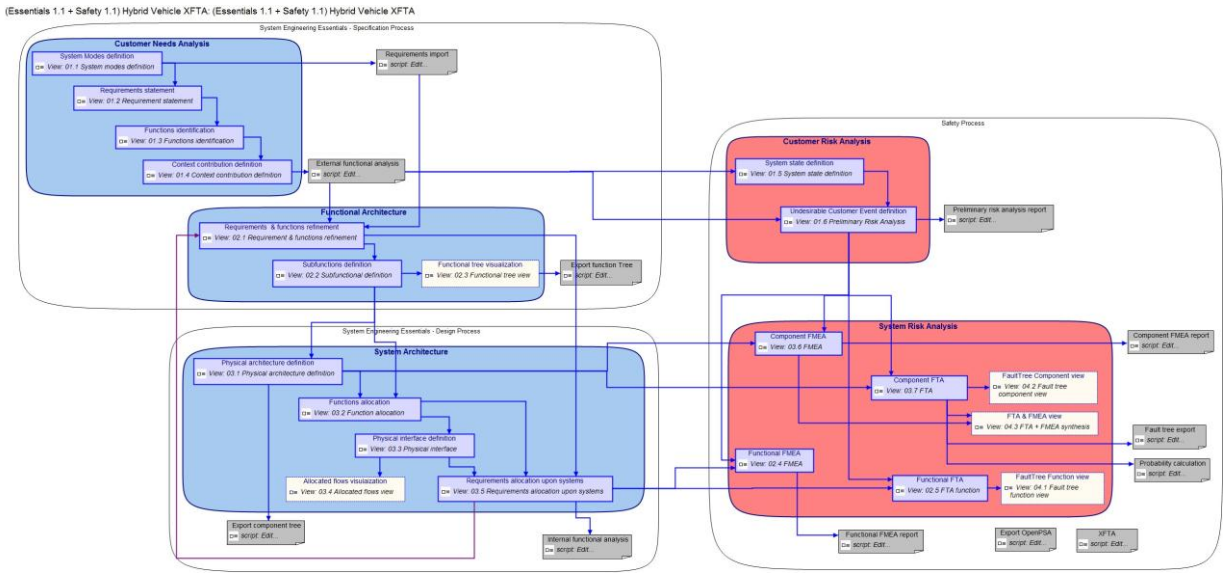

Figure 1 Integrated Process for System Engineering and Safety Engineering

The Essentials Systems Engineering Process has been divided into two sub-processes: (1) System Specifications and (2) System Design. The System Specifications subprocess contains, firstly, the phase of Customer Needs Analysis detailed in terms of system life cycle definition, requirements statement, functions identification and functions definition tasks. Secondly, we find the Functional Architecture phase decomposed into the tasks of requirements and functions refinement, and sub-functional flow exchanges definition. The System Design subprocess contains a System Architecture phase consisting of system definition, functions allocation to systems, physical interfaces definition and flows allocation, and requirements allocation to systems. However, there is a feedback connection from the requirements allocation upon systems in the System Architecture phase to the requirements refinement in the Functional Architecture phase: the allocation of various design requirements to the system components can lead to the generation of further functional requirements.

The proposed environment also covers Customer risk analysis phase which is linked to the functions definition in the system specifications phase. This phase includes the system state definition, the preliminary risk analysis (where risks are linked to functions), the system risk event identification, the UCE (undesired customer event) identification and the UCE quotation. It finally also covers the system risk analysis phase which is linked to the internal functional analysis in the system engineering process. It includes the fault tree building where risk events and basic risk events are identified on the system architecture resulting from the system engineering process, the FMEA (which consists in identifying components possible failures, associated risk events, failure analysis and corrective actions) and the complete fault tree view. [7][8] As arKItect® is a tool aimed at facilitating the tasks of systems (and safety) engineers, the automation of calculations and report generation has been given special attention. Moreover, arKItect® can easily be used to provide diverse customizable views on the system and its safety aspects. These two practical advantages are detailed in the following chapter.

4. ISO 26262

ISO 26262 Presentation

ISO 26262 standard is the adaptation of IEC 61508 in automotive industry. It sets out the automotive approach for all safety lifecycle activities for safety relevant systems comprised of electrical and/or electronic components. This Standard addresses possible hazards caused by functional behavior of safety related systems due to malfunctions and does not address non-functional hazards due to technical realization as well as nominal performance level of systems, independent from the existence of dedicated functional performance standards.

The ISO 26262 standard compiles the following key features:

• adopts a customer risk-based approach for the determination of the risks;

• provides a specific automotive method analysis to identify the safety integrity level of each undesirable effects (means for a vehicle function, identification of the consequences of one or some fault / failure, leading or possibly leading to a customer claim, or a damage to the environment, up to significant damage or harm);

• uses safety integrity levels for specifying the target level of safety integrity for the safety functions to be implemented by safety related systems;

• provides requirements for the whole lifecycle (engineering, production, operation, maintenance, decommissioning) necessary to achieve the required functional safety.

ISO 26262 Implementation in arkitect®

To comply with the SASHA project purpose, and consequently to be able to implement a life cycle process that complies with ISO 26262, the work was divided into Three main steps. Fig.2. illustrate the summary of these steps. The central idea is to implement the requirements stated in the standard in form of a meta model using arKItect® tool. The resulted meta model (or one of its representations) shall be reviewed and proved by certification experts/organizations and then it can be instantiated in a specific projects. We can note here that the main effort turns around the definition of the appropriate meta model.

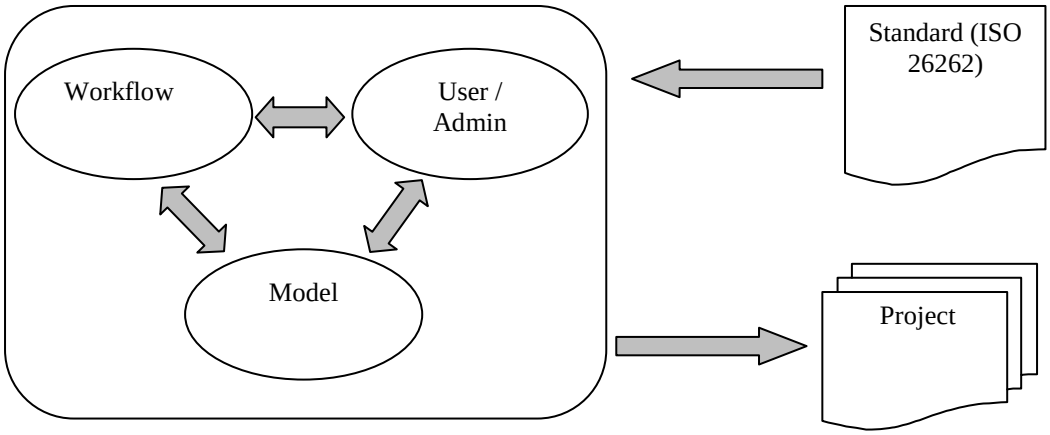

Fig.2. Main Steps for ISO 26262 standard implementation

Step 1: It consists on reading and translating the standard texts into a meta model. Some key issues have to be taken in account. First of all, the standard steps in all the life cycle. It contains requirements affecting a very large variety of activities like the management process (users, administrators, and responsibilities), the workflow, software/hardware products and suppliers. Another aspect to deal with concerns some requirements to be fulfilled without any indication about the way to do. The implementation means are considered as out of scope of the standard. The controllability, the external risk reduction and the use of “other technologies ” are some examples of this situation. The needs in this step are: 1) a powerful and flexible representation tool, and 2) a good understanding of the standard. The item 1) is resolved thanks to arKItect® characteristics. The item 2) is addressed by the expertise offered by the SASHA project partners.

Step 2: The resulted meta model have to be certified by experts/organizations that can stamp the compatibility of the decisions made on the meta model and the requirements stated in the standard. To do, a complete traceability between the standard requirement and the resulted Meta Model (containing all the life cycle processes) is made and the documentation is generated and produced as proof of the matching between the expected results (stated in the standard) and the solution (supported by the Meta Model)

Step 3: the application of processes (the use of the Meta model) ensures the compatibility of the designed product with the requirement stated in the standard. That means, once the two previous steps fully completed, the automotive projects that have to comply with ISO 26262 can simply use the produced Meta Model and by the way they can be ensured by the compliance of the project products with the safety objectives of the standard.

In the Safety Add-on package, the Customer Risk Analysis phase is implemented using Preliminary Hazard Analysis (PHA). PHA has been decomposed into the tasks of system risk event identification, undesired customer event (UCE) identification and UCE quotation. The Customer Risk Analysis phase also includes the system states definition task as these states need to be linked to the UCEs along with top risk events stemming from the system functions. As PHA is based on information on the system functions, the external functional analysis—definition of system service functions and constraints—constitutes its pre-requisite.

In practice, PHA is implemented as a defined set of objects and their interrelations in the Safety Add-on meta-model: we have notably objects of types Failure Mode, Top Risk Event and Undesired Customer Event. The Failure Mode flows (containing information such as no function, lost function,

untimely function activation and deteriorated function) are produced inside the service functions of the system (as defined in the Functional Architecture phase) and the information contained in these flows is passed on to the Top Risk Events at the system level. Afterwards, the Top Risk Events and the system states can be combined to produce Undesired Customer Events at the oversystem level.

In the System Risk Analysis phase, we have chosen to implement two widely-used tools: Fault Tree Analysis (FTA) and Failure Modes and Effects Analysis (FMEA). These two tools can be used independently—it is possible to perform one analysis or the other, or both if desired. The fault tree building task consists of defining risk events, detailing them in terms of basic risk events and connecting the whole via logic gates; the fault tree is constructed on the system architecture.

In FMEA, component failures are linked to risk events while each failure can become the object of detailed failure analysis and corrective action planning. The basic object type of the Safety Add-on implementation of FMEA is an information flow called FMEA Failure. This flow originates in the faulty component and connects to a higher level Risk Event. Of course, a Risk Event can receive information from several FMEA Failures.

The failures can be further analyzed via Failure Analysis objects containing attributes such as cause, effect, severity, probability of occurrence, means of detection, non-detection rating and a Risk Priority Rating (RPN). These attributes can be assigned values and thus permit to characterize the Failure Analysis in detail. In addition, it is possible to define corrective action via a Corrective Action object also found inside the FMEA Failure. Corrective Action is defined in terms of attributes such as the action description, its deadline, the person responsible for the action as well as new severity, non-detection and RPN ratings after the correction. The Corrective Action object is connected to the corresponding Failure Analysis via a Failure Action flow.

As FTA and FMEA are both based on the system architecture, internal functional analysis should be completed (or should at least be worked on in parallel) before these safety aspects are studied.

arKItect® Safety Add-on also contains special views for visualizing chosen safety aspects. The fault trees are constructed in a specific view permitting to visualize the FTA objects as well as the system architecture. FMEA also has its dedicated view based on system architecture and the FMEA components. In addition, there is a fault tree synthesis view permitting to visualize the constructed fault tree in its entirety (along with the event probabilities) without the hierarchical system architecture levels. There is also a combined FTA & FMEA synthesis view displaying the components of the FMEA along with the fault tree events.

In the Safety Engineering Process, it is possible to use scripts included in arKItect® Safety Add-on to calculate fault tree probabilities as well as to provide an FMEA synthesis. The fault tree probabilities are defined as attributes in the various fault tree objects (Basic Events, Risk Events and Top Risk Events) and the rules of the calculation by are set by the logic gates (AND Gates and OR Gates).

In the same manner as in the Safety Engineering process, it is possible to generate deliverables via architect® Essentials scripts during the Systems Engineering Process: these deliverables include reports on the external and internal functional analyses as well as function and component trees.

5. Discussion

PHA, FTA and FMEA are the tools that were used in our integrated approach. Traditionally, system developers are not familiar with system safety analysis processes which are performed by safety engineers. One reason for this is the gap that exists between the traditional development processes, methodologies, notations and tools and the ones used in safety engineering. Although they have been used for decades, they were not coupled with tools usually used for system engineering. These tools, such as those supporting SysMl and Doors , are not adapted to such an integrated approach. [10]

SysML[1]: While it covers many aspects of system engineering process, it needs to be modified to cover safety engineering process. This gap makes the development of safety aware systems a very complicated task. It is mainly due to the rigidity of its metamodel (limited number of diagrams that cannot be customized). [9]

Doors[2]: While it covers mainly the requirements management process, it is not a graphical tool, making system engineering non intuitive. Moreover, it is impossible to perform a safety analysis using it. It is due to the fact that it is a textual tool.

The SASHA project (***Safety Check of Automotive Software, & Hardware Architectures)*** provides a more appropriate response to automotive needs. Its purpose is to model both the process of designing a safety case, and at the same time, a description of multi-level system hierarchy. The Delphi's strategy is to integrate innovative [2]results from this project and make them generic on his product lines.

[2] SysML, http://www.omgsysml.org/
[2] DOORS, http://www-01.ibm.com/software/awdtools/doors

Indeed, the DELPHI product development is based on Automotive SPICE [11] repository. The Automotive SPICE, derived from the ISO/IEC 15504 standard, is an international standard which has been developed by consensus of the car manufacturers. It is used when performing conformant assessments of the software process capability of automotive suppliers. At specification level, we use tools as Reqtify[1] and DOORS for requirements capturing. At design level, our application layer software components are modelled using Simulink[3] from which AUTOSAR compliant code is generated. Some safety analyses [12] are performed using FTA and FMEA methods. But it represents just a small part of the IS026262 functional safety framework.

For optimal integration of a safety assessment in our development process, it was necessary to understand the gap between the technical reference and SPICE process. So, the detailed description of IS026262 contents realized in SASHA allows identifying by a mapping on current DELPHI process, different steps and work products not yet filled to achieve the compliance with the new standard.

The idea is not to start from scratch. From that, actions have been taken to update our development project methodology. This requires in first time improving of the assessment framework, either by modifying SPICE process to make it compliant to ISO26262 when the activities are already present, either outright by the addition of new activities that are not at all previously treated.

For testing the accuracy of the new methodology as well defined, its deployment on Simulink implementation built around an example of a diesel engine control with the possibility of verifying hardware and software during the design of components is being performed with arKitect®.

6. Conclusion

We have presented here the Safety Engineering Process of the Safety Add-on and its interactions with an overarching Systems Engineering Process. All in all, the arKItect® software together with the Essentials meta-model and the Safety Add-on permit easy and convenient integration of the systems and safety engineering processes. Preliminary Hazard Analysis can in this manner be built on the system-level functional architecture while Fault Tree Analysis and Failure Modes and Effects Analysis can be performed in parallel to the system architecture definition. As a visual tool, arKItect® enables the representation of these safety aspects in a graphical and easily-understandable manner. Special views can also be created to visualize the chosen safety, functional and organic aspects of the system. In addition, the automated calculation of fault tree probabilities and the generation of FMEA tables facilitate the analyst's task.

This paper aims also to address a very practical issue in the automotive area which is to find an easy way to achieve a high safety level by fulfillment of the ISO 26262 standard requirement and, in the same time, mastering the costs due to this standardization. The proposed solution is based on applying a combined process which includes safety items and functional activities. The aimed process is modeled in a certified Meta model that insures, if correctly applied that the instantiation (some project) called also "Safety Case" complies automatically with the standard and so achieve a safe functional design.

9. References

[1] System Engineering — System Life Cycle Processes, ISO/IEC 15288, 2001

[2] Incose System Engineering Handbook, International Council on Systems Engineering, 2007

[3] Guideline Systems Engineering for Public Works and Water Management, Ministry of Water Management, The Netherlands, May 2008 Second edition

[4] Fault Tree Handbook, US Nuclear Regulatory Commission, 1981

[5] Manfred Broy et Al, Seamless Model-based Development: from Isolated Tools to Integrated Model Engineering Environments, Proceedings of IEEE 2010

[6] Harish Devendre Goel, Integrating Reliability, Availability and Maintainability (RAM) in Conceptual Process Design,TU Delft, 2004

[7] Gerrit Muller, CAFCR: A Multi-view Method for Embedded Systems Architecting, 2004

[8] Augustine Nnadozie Ajah, On the Conceptual Design of Large-scale Process & Energy Infrastructure Systems, TU Delft, 2009

[9] Kleanthis Thramboulidis and Sven Scholz, Integrating the 3+1 SysML View Model with Safety Engineering, ETFA 2010

[10] OMG Systems Modeling Language (OMG SysML™), 2008

[11] Automotive SIG: Automotive SPICE, Process Assessment Model. Version 2.4, Status: Released 2008-08-01 (2008).

[12] J. Czerny, B., G. D'Ambrosio, J., T. Murray, B., Sundaram, P.: Effective Application Of Software Safety Techniques For Automotive Embedded Control Systems. In: SAE World Congress, Detroit, Michigan (2005)

[13] ISO 26262 "Road Vehicle – Functional safety". 2011

[14] IEC 61508 "Functional Safety of Electrical / Electronic / Programmable Electronic Safety-related Systems".

[1] Reqtify, http://www.geensoft.com/en/article/reqtify

[3] Simulink, http://www.mathworks.co.uk/products/simulink/