Safety Case Driven Development for Medical Devices

Alejandra Ruiz[1(✉)], Paulo Barbosa[2], Yang Medeiros[2], and Huascar Espinoza[1]

[1] ICT-European Software Institute, Tecnalia, Parque Tecnológico Ed. 700, Derio, Spain
{alejandra.ruiz,huascar.espinoza}@tecnalia.com
[2] Núcleo de Tencologias Estratégicas em Saúde – NUTES, Universidade Estadual da Paraíba, Campina Grande, PB, Brazil
{paulo.barbosa,yang.medeiros}@nutes.uepb.edu.br

Abstract. Medical devices are safety-critical systems that must comply with standards during their development process because of their intrinsic potential of producing harms. Although the existing trend of an increasing complexity of medical hardware and software components, very little has been done in order to apply more mature safety practices already present on other industrial scenarios. This paper proposes a methodology to enhance the Model-Based System Engineering (MBSE) state-of-art practices from the safety perspective, encouraging the use of safety cases and providing guidance on how to show the correspondent traceability for the development artifacts. We illustrate our methodology and its usage in the context of an industrial Automated External Defibrillator (AED). We suggest that medical device industry could learn from other domains and adapt its development to take into account the hazards and risks along the development, providing more sophisticated justification, as, for example, the impact of design decisions.

Keywords: Safety case · Medical device · Software development methodology · Automated external defibrillators

1 Context

Safety-critical systems are defined as those which in case of an accident, people or the environment might be put in danger [22]. Different safety-related standards in practice provide guidelines for systems developers. One of the main challenges is that the use of new technologies will be increasingly important for the future and complying with these standards should prevent innovation from being stifled, while still tackling the expected safety objectives. In order to cope with that, those standards and guidelines tend to be sometimes ambiguous and open to multiple interpretations. While those interpretations leave the door open to new ideas, technologies, methods, they also make it difficult for authorities and companies to share the same views. We are talking about the ambiguities resulting from openness to new technologies.

F. Koornneef and C. van Gulijk (Eds.): SAFECOMP 2015, LNCS 9337, pp. 183–196, 2015.
DOI: 10.1007/978-3-319-24255-2_14

In the context of medical devices, new functionalities and the increasing contents of software and hardware components from different manufacturers requires focus on interface, reuse and integration issues. By increasing this technological complexity, we also observe an increase of systematic and random failures. Medical device malfunctions cause hundreds of thousands of incidents and thousands of serious injuries and deaths annually, with these numbers increasing year after year. According to [1–3], it is estimated that medical device malfunctions cause more than 400,000 incidents and 7,000 deaths every year in Europe and the US alone. Safety assurance and certification practices for medical devices must be improved in order to better ensure system safety and to gain more confidence about the safe operation of these systems.

There is a need for better safety assurance and certification practices. Incidents have occurred as a result of inadequate quality assurance practices, impact analysis, and documentation review [4]. Shortcomings in industrial practices have been identified as a culprit for deficiencies in the identification of safety risks [5], traceability management [6] and safety assessment [7].

A study of past medical device failures [4] identified important issues in safety evidence information, having to do with deficiencies in system requirements, verification and validation procedures and results, and impact analysis. One of the main overall problems noted in the study is the lack of detail in safety evidence information (e.g., about its characteristics and the relationship between different pieces of evidence). Specific issues in traceability information have further been reported in [6], including lack of knowledge regarding the artefacts to trace, trace granularity not being clearly defined, redundant traceability information, and important links missing.

A safety case can be defined as a mean to "communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context" [8]. It is an assurance case addressing safety. In fact, a safety case is becoming a requirement on different standards from different domains. For example, lately, the automotive functional safety standard ISO 26262 includes the safety case creation as a requirement for compliance [9].

Safety cases on the heath domain and medical devices in particular are not widely spread. The most notable efforts come from the Generic Infusion Pump project[1] with the safety case for a GPCA pump, becoming a guide for other infusion pumps through the derivation of patterns. Authors in [5, 10] have reported about the start of usage of safety cases on the healthcare system. From [11] we can extract some of the safety assurance case benefits for the application on medical devices:

- Provides a framework and a vehicle to stimulate critical thinking;
- Ensures the completeness of risk identification and risk controls;
- Provides rationale for the validity of risk acceptance;
- Logically documents and connects safety critical information in an easily and understandable manner;
- Communicates safety critical information effectively to internal and external stakeholders.

[1] http://rtg.cis.upenn.edu/medical/assurance_cases.html.

The paper is structured as follows. Section 2 defines the purpose of this work and the challenge it faces. Section 3 details the methodology to incorporate safety case development activities over the V model. Section 4 instantiates the methodology with a case study over an industrial medical device. Section 5 discusses the main achieved results and lessons learned. Finally, Sect. 6 points the final remarks and further developments.

2 Objective

This paper focus on the main challenge that medical device industry is facing, *i.e.*, how to deal with the increasing complexity of the systems which directly affects the safety integrity. Similar challenges have been faced by other industries such as automotive or avionics domains. In automotive industry, for example, we have seen mature standardization through ISO 26262 (a functional safety standard for road vehicles) due to concerns such as the growing of ECUs, CANs, LINs, signals and messages. In medical device industry, we are facing the same problems, but without the same interest for supporting solutions. Taking as example a sample Automated External Defibrillator (AED), we should mention there are specific microcontrollers for performing each one of the following functions: (i) analyzing the ECG signal to drive decision processes; (ii) filtering the ECG signal according to several parameters; (iii) modulating frequency signals; (iv) user interface; (v) monitoring the electrical shock and ensuring accurate deliver of energy; (vi) monitoring the electrodes in order to ensure patient safety, among others. Even so, very few multi-core technologies have been used in this domain, as far as we have investigated in the medical instrumentation literature.

We aim at defining a methodology for a safety-oriented software development that will be later on integrated onto hardware. The system software and hardware shall comply with safety standards and we shall ensure the safety of the product. In this sense, we aim at bringing the safety case theory onto medical devices focusing on the software architecture as the main driver. We provide guidance through industrial examples on how to trace the safety requirements and safety related properties for software compliance and integration. In the scope of this paper, we use the Goal Structuring Notation (GSN) [12] as the graphical notation for representing the argumentation on the safety case.

3 Methodology

We present a twofold methodology where safety concerns drive design decisions and, at the same time, we aim at complying with the standard ISO 14971 [14], which is the standard related to risk management on medical devices. As we are addressing the use of Model Based System Engineering (MBSE), the conformance with IEC 62304 standard [13], which is the medical device software – software life cycle processes standard, also needs to be addressed.

For this industry, it is still unclear how safety concerns affect the development of each phase. That is our main justification to introduce the idea of the safety case driven development. We have inspired on the automotive functional safety standard ISO

26262, which proposes a V cycle for system development (including such activities as requirements specification, design, implementation, integration, verification, validation, and configuration). On the automotive domain, a safety case is developed along the lifecycle with the aim at communicating in a clear, comprehensive and defensive argument (supported by evidence) that the system is free of unreasonable risk to operate in a given context. Figure 1 presents an overview of the proposed methodology where the safety case is created along the system development.

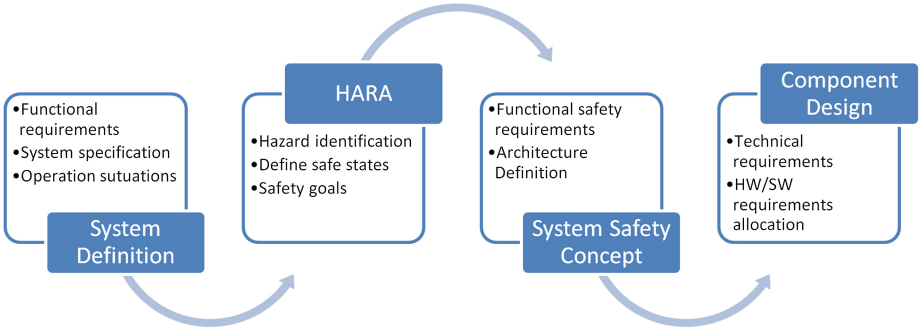

Fig. 1. Safety case driven process for system design

In our approach, safety-related activities progress along the development and provide contexts and outputs to be gathered and used on the safety case:

1. At the *System Definition* phase, we need to specify the functionalities of our system as well as the context in use. This will produce inputs for the context and situations of the use of the medical device that will be introduced into the safety case.
2. At the *Hazard Analysis and Risk Assessments (HARA)* phase, we focus on the identification of possible hazards. This will serve as an input for including functional safety requirements into the requirements list.
3. At the *System Safety Concept* phase, we provide a link with the system architecture as a specification in which safety mechanism and patterns will be put in place in the system so as to fulfill the functional safety requirements at high level.
4. At the *Component Design* phase, we should iterate at the same time with the hardware and the software where we should derive the functional safety requirements and decisions made for the system safety concept definition into more technical requirements. These technical requirements will also be allocated either into hardware, software or both.

Figure 2 shows the arguments decomposition as well as the decomposition process. The context of operation arguments are extracted from the *System Definition* phase. On the *HARA* phase we will gather the arguments related to hazard identification and the safety goals that will serve for linking with the next phase arguments, *i.e.*, the *System Safety Concept*. On this phase, we collect arguments about the functional safety requirements and link them with the decisions made regarding which mechanism we

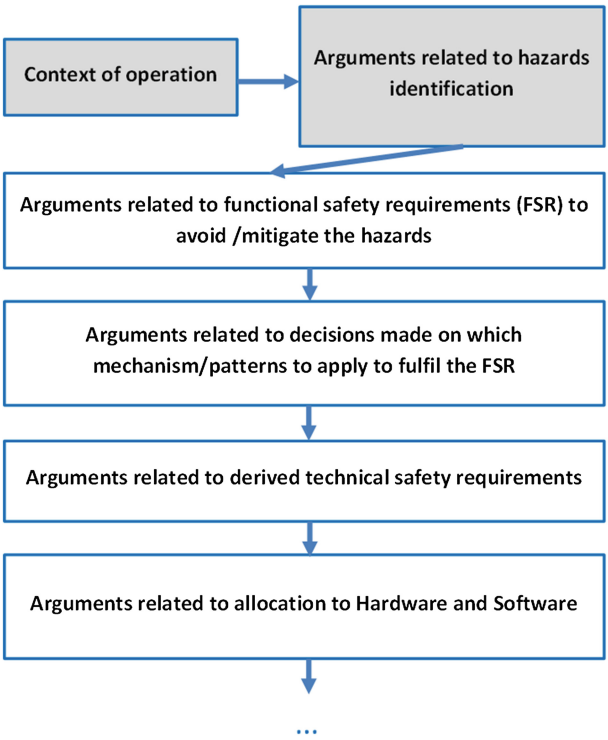

Fig. 2. Safety case decomposition structure

will apply to fulfill those requirements. Finally, on the *Component Design* phase, we should trace the previous requirements with the derived technical requirements and how they are allocated to hardware and software.

4 Use Case

In this section, we explore a running example to demonstrate how the proposed methodology is employed over the development of the medical device *Automated External Defibrillator* (AED), a current research trend inside NUTES. NUTES is part of an initiative for promoting the technological development of Brazil, where the Brazilian Health Ministry has started some technological transfer projects from well-consolidated manufacturers to institutes for science and technologies in order to retain the know-how of manufacturing medical devices inside the country. In this context, the NUTES project is in charge of receiving and improving methodologies for manufacturing AEDs from the Lifemed[2] and providing new improvements.

[2] lifemed.com.br.

AEDs are consolidated as a therapy for the ventricular fibrillation/tachycardia, which are the cardiac arrhythmias with highest incidences of fatal cases. In the treatment of such conditions, any delay in the application of the defibrillator shock is an important issue for investigation, since each minute without the shock implies in a loss among 7 % to 10 % of the chance of surviving. The usage of AEDs has gained much more popularity, since they can be used even without a specialized rescuer team available. According to [21], more than 1000 cardiac arrests deaths were connected to the failures of AEDs over 15 years, between January 1993 and October 2008, in the United States. Adverse event reports were catalogued in the Manufacturer and User Device Experience (MAUDE) database. Due to patient safety, both the development and the validation of the technologies for these devices follow rigorous standards.

The next subsections show the application of the methodology defined in Fig. 1 for the AED use case.

4.1 System Definition

Figure 3 depicts the essential parts of the AED system as a context diagram. Considering the safety case, we achieve the goal of showing each external entity, the main functional unities and their interaction with the system. The main input variable to be received is the cardiac pulses of the *Patient*. A module defined as *Signal Analyzer* uses sophisticated algorithms for detecting the signal complexity and to decide if a defibrillator pulse is necessary in case of fibrillation. If it is the case, the *Shock Generator* is in the responsibility of controlling the main output variable, the *energy*, by providing it in the *Biphasic Truncated Exponential* waveform to the *Patient* chest through the *Pads*.

Figure 4 describes the main AED use cases of our interest, which is the chosen scope addressed in this safety case construction running example. These use cases will be referred when assembling the safety case for the context specification.

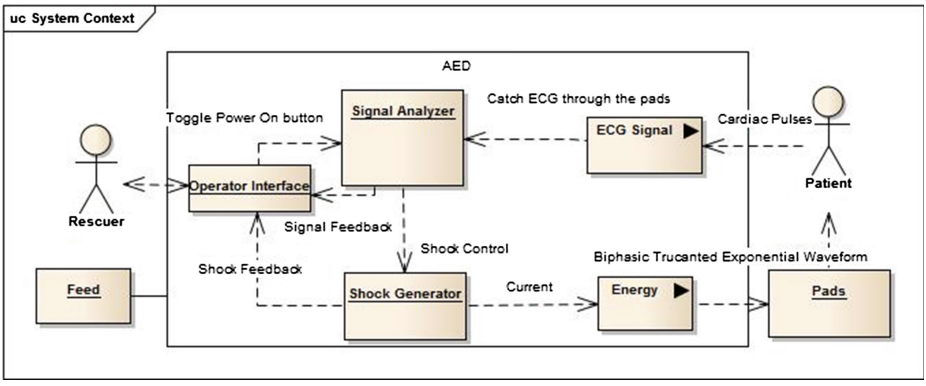

Fig. 3. AED system blocks diagram

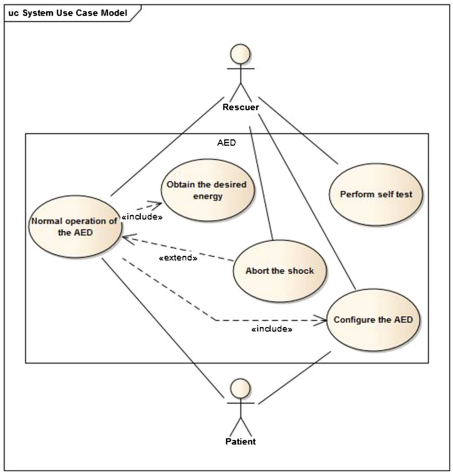

Fig. 4. AED use case model

4.2 Hazard Decomposition

In the scope of this work, we focus on a hazard named *Overshocking*. Figure 5 presents a trace of a specific tool for Enterprise Architect[3] as an add-in for managing development in the context of this research in order to manage architectural elements according to the Risk Management Process described in ISO 14971 [14] standard medical devices and its specific technical report IEC/TR 80002 [15]. Therefore, we see that the main variable to be controlled in the *Overshocking* hazard is energy. The investigated scenario where the hazard is present is *Normal operation of the AED* use case, activating the alternative scenario *Failure to deliver the shock*, since the hazardous situation is having the *Pads* connected to the *Patient*. IEC 62304 on clause 7.3.3 [13] requests us to trace the hazard to the situation and to the item, and later on to the risk mitigation measures put in place. The identified harm is *Skin damage*, affecting the *Patient* chest.

At this stage, we are able to start the safety case construction where the first arguments on the context of operation and arguments referring to hazard identification. By defining safety goals, we mean to insert all information possible about hazard. For example, we have *software safety class* for the *Overshocking* hazard in the category C.

After an effective hazard analysis that should involve systems engineering artifacts, such as the use cases, we proceed with the specification of exception cases, as shown in Fig. 6. The goal is to improve system reliability with the hazard concerns. Finally, we also define safe states at this phase, when defining scenarios and alternative flows. For example, we defined that system must restart and clear all memories after all procedures for verification suggested by exception cases as shown in Fig. 7 for the *Failure to Deliver the Shock* exception case.

[3] sparxsystems.com.

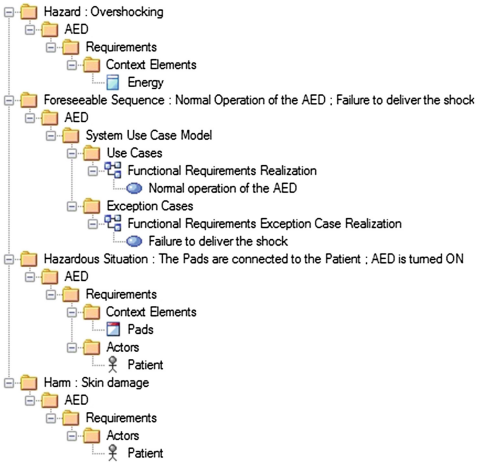

Fig. 5. Hazard *Overshocking* from AED tracing to system engineering artifacts

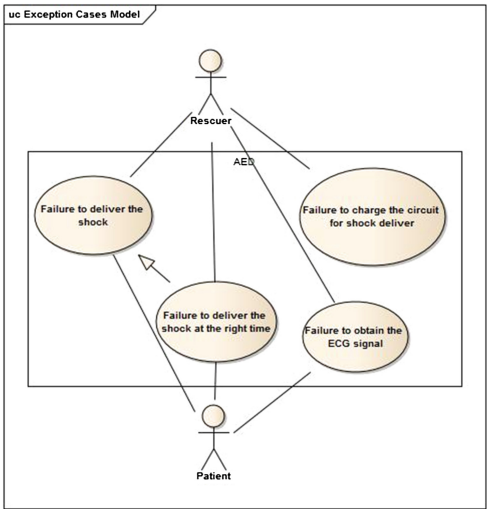

Fig. 6. AED exception cases model

4.3 System Safety Concept

In this phase we have to derive the safety goals into functional safety requirements. Figure 8 describes mitigation procedures that interact with software and hardware components that were identified. These mitigation procedures are the input for the next safety case iteration, which is the decomposition of the safety goals into functional safety requirements. We have created templates for hazard description like the one shown on Table **1**, in order to serve as guidelines for functional safety requirements elicitation.

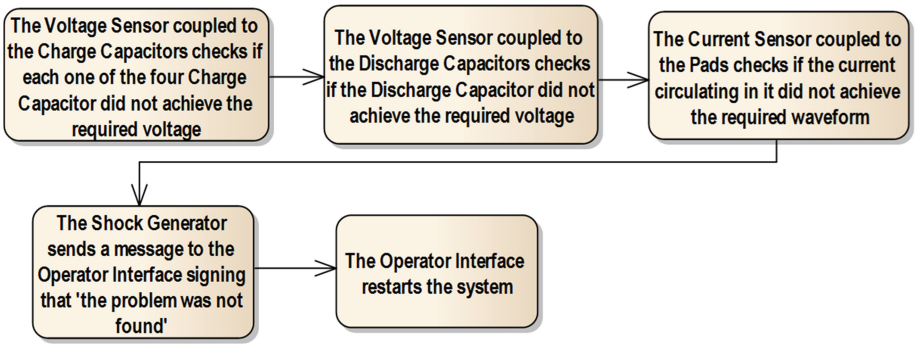

Fig. 7. Failure *to Deliver the Shock* state machine specification to a safe state

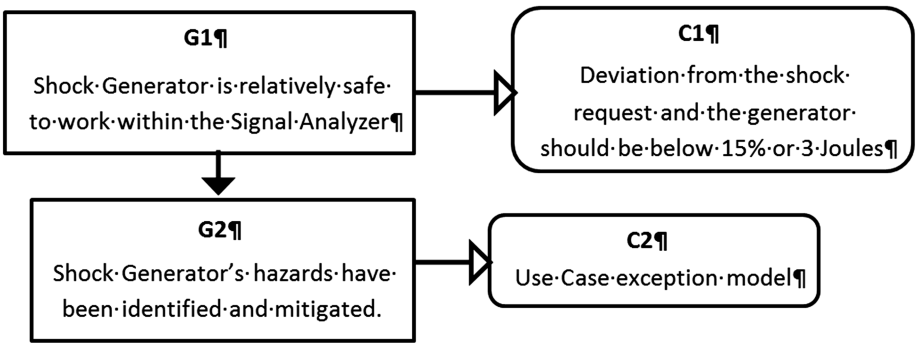

Fig. 8. Safety case at HARA phase

Table 1. Hazard description

Hazard	Overshocking
Cause	The energy delivered is over 15 % or 3 Joules the estimated one
Safety requirements	The delivered energy cannot vary more than 15 % or 3 Joules
Fault categories	Class C: Death or SERIOUS INJURY is possible
Alarm	Deviation is over or equal 14 %
Warning	Deviation is over or equal 10 %
Information	Deviation is over or equal 7 %
Failure mode	When an alarm is triggered, we go to the safe state which is do not deliver any energy at all and reset all the variables and parameters to the default state.
Failure distribution	If we capture the desired energy with a deviation, we send the deviation as entrance for the algorithm to process the quantity of energy to be delivered; we send a command to deliver the energy with a deviation; we will also send the deviation as an entrance to the monitoring function and we deviation might not be detected. (fail in all those software units)

Next, we have to link the safety case with architectural decisions. Thus, we have modeled the main design decisions about the safety mechanisms to cover the safety requirements. The main causes for the *Overshocking* hazard were identified as wearing electronic components or an unexpected loss of the software control throwing unexpected values. This motivates new fine grained specifications such as charging rates for capacitors; specific peak values; or specific frequencies over semiconductors switching. These fine grained specifications can also be refined as fault tolerance requirements such as indication of wear of components, warning that maintenance and repair should be provided; coupling of new amortization circuits in order to deal with unexpected peak values; or coupling of new circuits for the management of periods for avoiding extrapolation of time limits for issuing peak tensions. Finally, the last layer of specification will require specific sensors and actuators for efficient detections and actions over all hazardous conditions.

At this time, we also produce a new iteration of the safety case decomposing the hazard identification into the functional safety requirements defined to avoid or mitigate the hazards effect. We also trace these functional safety requirements into safety mechanisms at the architectural level that will be lately implemented, following approaches such as [20]. In Fig. 9, we present an excerpt of the safety case at this level, highlighting the evolution of the safety case.

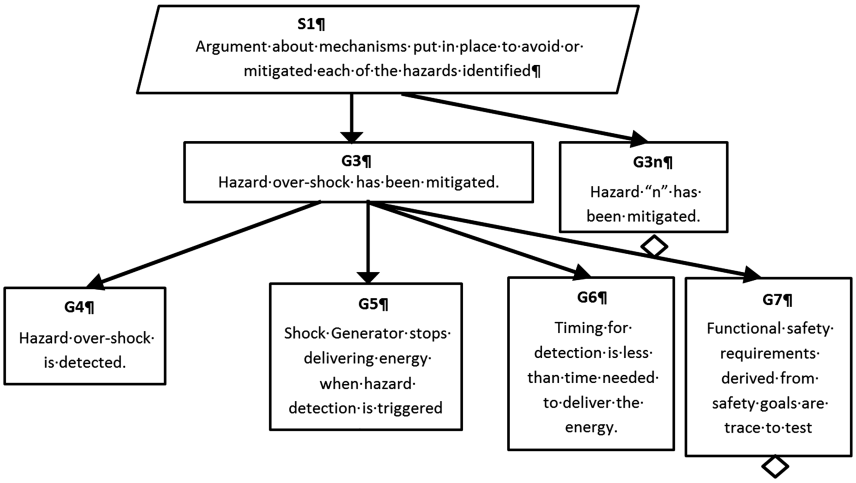

Fig. 9. Excerpt of AED safety case at system safety concept level

4.4 Component Design

The proposed approach focuses on traceability and the suggested solution shall explicitly demonstrate the mapping from *System Goals* to *Electronic Components*, in case of a hardware mitigation solution, or to a *Software Component*, in case of a software mitigation solution. It is at this phase where most of the design decisions were taken and needs to be traced.

In order to discuss a specific concern, we start by discussing some traceability and design decisions over the software component inside the *Signal Analyzer* block, since it is essential do decide whether the patient needs a shock, and automatically decide what the parameters of the shockable-energy are. Focusing on one of the claims, which are still under development from the safety case shown on previous section, we see that among other issues, we decided to implement a data error detection mechanism. In order to do so, we have implemented the *ErrorHandler* component. This component has the responsibility of avoiding problems that could interfere in the correct decision of applying the shock, such as signal propagation discrepancies, influences of harmonics, floating point corrections, among others. We took a design decision with the component realization modeling for the Process ECG component according to the *Pipes and Filters* design pattern for safety purposes. In this sense, we divided each specific phase of the ECG signal processing into filters and throughout this process, information concerning errors and flaws are collected and processed in the *ErrorHandler* module for activating safe states.

Finally, in Fig. 10, we show how the safety activities influenced the system's design solution in a codesign scenario. A specific excerpt of the design concerning the hazard *Overshocking* is shown in the upper part of the figure as a traceability model. In the end of the trace, we have 4 components. Two of these components are for detection, as *Voltage and Current Sensors*. The other two are actual mitigation components, as a *Snubber* for controlling frequency switching of the semiconductors near the transformer and a *Charge Controller* to close the loop in the secondary of the transformer in order to correct voltage peaks. These components are mapped to the bottom part of the figure as Simulink model and are realized in the component implementation phase.

5 Brief Discussion

In this section, we provide a discussion about the main efforts from the system designer viewpoint according to the development of a safety case. In our opinion, the metrics to define the main suitability and improvements of the methodology come from the care when dealing with traceability between safety engineering and safety engineering artifacts can be assigned to the availability of solutions and patterns to do that in the current state of MBSE approaches. This could be observed in all the phases when building the safety case.

For example, currently we have very good approaches for system definition and document software and system architectures. Approaches based on ISO 42010 [16], Architecture Tradeoff Analysis Method (ATAM) [17], Software Architecture Analysis Method (SAAM) [18], Active Reviews for Intermediate Design (ARID) [18], Views and Perspectives [19], among others, are suitable to address the main issues when defining a system architecture with concerns that are able to fill well what is required by the safety case. All these approaches can be easily supported by UML/SysML modeling tools, such as Enterprise Architect, by following good design practices.

At the *HARA* phase, we have defined a domain specific solution to trace hazard analysis and risk assessment artifacts from ISO 14971, such as *Foreseeable Sequence*,

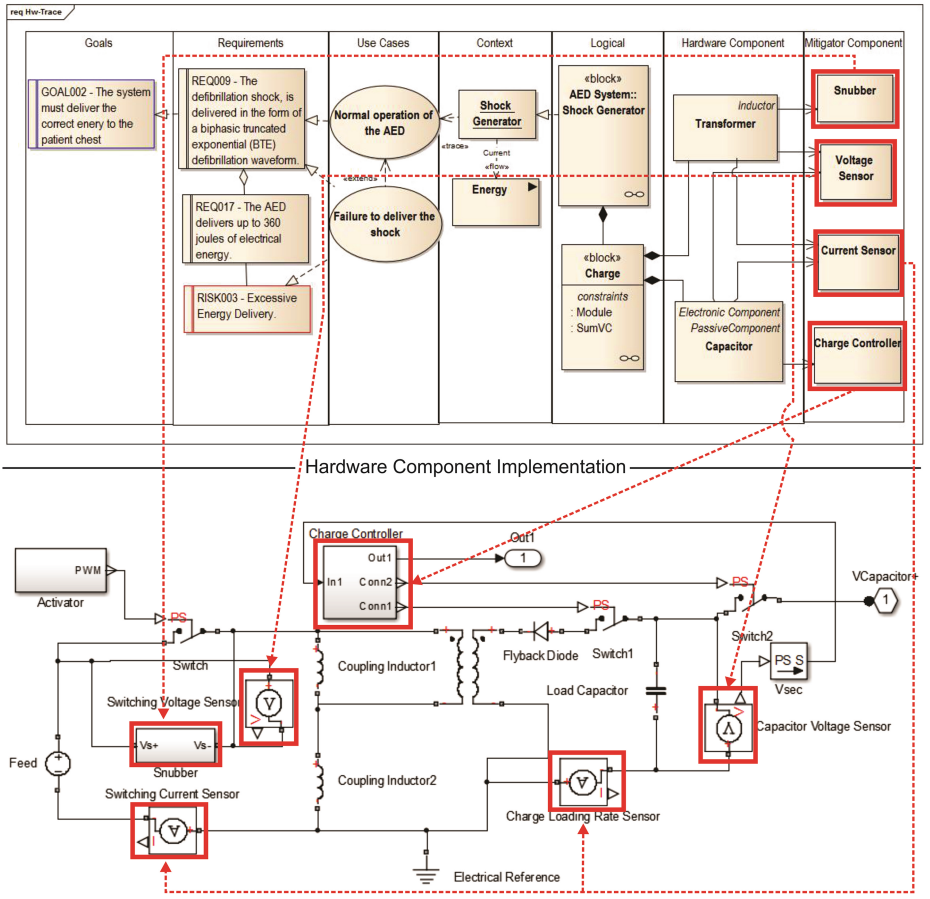

Fig. 10. Traceability from system engineering artifacts towards mitigation components

Hazardous Situation and *Harm* to system engineering artifacts such as *Context Elements*, *Actors*, *Subsystems*, *Use Cases*, among others.

For the *safety concept*, focusing on the *Overshocking* hazard, we were able to specify and show conformance of a bunch of mitigation procedures in a qualitative way. During the safety requirements decomposition, we have showed the early detection of need for error handlers for software components and sensors (e.g. for voltage and current), controllers (e.g. for charge) and suppressors (e.g. *Snubber*). The main metrics at this phase can be checked with analytic models, built, for example, in powerful platforms such as Matlab/Simulink, ISOGRAPH, among others, allowing later realization with more feasible and validated safety constraints.

Finally, during the *component realization* phase, the safety case focuses on the measures that the design shall reflect in order to prove to be safe. Criteria such as requirements coverage, design decisions being impacted or even efforts on the design realization are important testimonials before starting the validation phase, where tests shall provide for each design phase following the V model discipline.

6 Conclusions and Future Work

In this paper, we have demonstrated the main benefits of a methodology for safety-critical system development based on MBSE that is driven by the construction of safety cases. The main activities were explained following an example-driven approach, through a case study on an industrial medical device. This explanation provided a clear traceability between system design main phases in a tool integrated way. Furthermore, several trends continue under investigation, intending to provide a tool chain integration, where compliance management tools will be able to exchange information between design environments, testing tools and safety cases. For the main features, we are adapting GSN tools for new concepts still unexplored, such as decomposition and traceability to safety requirements, architectural elements and a bunch of operation between safety cases in order to provide fusions between safety specifications.

As future work, we shall follow the compliance with the complete V model and directions on each phase of validation part will be provided. One of the most challenging parts that we aim to continue researching is on the component composition perspective. Different suppliers could come up with different developments that will compose the system. Finally, we also plan to incorporate modular safety cases applications to this approach. Modular safety cases are described on the extension B1 of the GSN standard.

References

1. Alemzadeh, A., Iyer, R.K., Kalbarczyk, Z., Raman, F.: Analysis of safety-critical computer failures in medical devices. IEEE Secur. Priv. **11**(4), 14–26 (2013)
2. MHRA: Report on Devices Adverse Incidents in 2010 (2011). http://www.mhra.gov.uk/home/groups/dts-bs/documents/publication/con129234.pdf
3. The Boston Consulting Group: EU Medical Device Approval Safety Assessment: A comparative analysis of medical device recalls 2005–2009 (2011). http://www.eucomed.org/uploads/Press%20Releases/BCG%20study%20report.pdf
4. Wallace, D.R., Kuhn, D.R.: Failure modes in medical device software: an analysis of 15 years of recall data. Int. J. Reliab. Qual. Saf. Eng. **8**(4), 351–371 (2001)
5. The Health Foundation: Supplements to: Using safety cases in industry and healthcare (2012). http://www.health.org.uk/public/cms/75/76/313/3847/Using%20safety%20cases%20in%20industry%20and%20healthcare_supplements.pdf?realName=yjOYNa.pdf
6. Mäder, P., Jones, P.L., Zhang, Y., Cleland-Huang, J.: Strategic Traceability for Safety-Critical Projects. IEEE Softw. **30**(3), 58–66 (2013)
7. Eucomed: Towards a regulation that guarantees patient safety, ensures patient access and keeps innovation in Europe (2013). http://www.eucomed.org/uploads/Modules/Publications/20130130_2013_eucomed_detailed_position_on_proposal_mdd_revision.pdf
8. Kelly, T.: Arguing Safety - A Systematic Approach to Managing Safety Cases. Ph.d. thesis, Department of Computer Science, The University of York (1998)
9. ISO 26262 International Organization for Standardization (ISO), "ISO/DIS 26262: Road vehicles - functional safety," (2011)

10. Bloomfield, R., Chozos, N., Embrey, D., Henderson, J., Kelly, T., Koornneef, F., Pasquini, A., Pozzi, S., Sujan, M.-A.: A Pragmatic Review of the Use of Safety Cases in Industry–Lessons and Prerequisites for their Application in Healthcare (2011)
11. Eagles, S., Wu, F.: Safety Assurance Cases for Medical Devices. In: AAMI 2014, Biomedical Instrumentation & Technology, February 2014
12. GSN Community Standard. Version.: Origin Consulting GSN Community Standard Version 1 (2011)
13. International Electrotechnical Commission Medical device software – Software life cycle processes. INTERNATIONAL IEC STANDARD 62304 First edition 2006-05. International Electrotechnical Commission (2006). Accessed 2 June 2012
14. ISO 14971 - medical devices – application of risk management to medical devices. Technical report, International Organization for Standardization (2010)
15. IEC/TR 80002-1:2009: Medical Device Software Part 1: Guidance on the application of ISO 14971 to medical device software. ISO, Switzerland (2009)
16. ISO/IEC 42010 (IEEE Std) 1471-2000: Systems and Software engineering- Recomended practice for architectural description of software-intensive systems, ISO/IEC/(IEEE), p. 23
17. Rick, K., Mark, K., Paul, C: ATAM: Method for Architecture Evaluation, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical report CMU/SEI-2000-TR-004 (2000). http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5177
18. Dobrica, L., Niemelä, E.: A survey on software architecture analysis methods. IEEE Trans. Softw. Eng. **28**(7), 638–653 (2002)
19. Rozanski, N., Woods, E.: Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives. Addison-Wesley Professional, Reading (2005)
20. Antonino, P., Trapp, M..: Improving consistency checks between safety concepts and view based architecture design. In: Proceedings of 12th International Probabilistic Safety Assessment and Management Conference, PSAM 2014, Honolulu, Hawaii, USA, 22–27 June 2014
21. DeLuca Jr., L., et al.: Analysis of automated external defibrillator device failures reported to the food and drug administration. Annals Emerg. Med. **59**(2), 103–111 (2012)
22. Knight, J.C.: Safety critical systems: challenges and directions. In: Proceedings of the 24th International Conference on Software Engineering, ICSE 2002, pp. 547–550, 25 May 2002