Risk-Based Requirements Management Framework with Applications to Assurance Cases

David Feng
Ethicon Endo-Surgery, Inc
4545 Creek Rd.
Cincinnati, OH 45242
513-337-3785
Dfeng1@its.jnj.com

Curt Eyster
Biosense Webster, Inc
15715 Arrow Highway
Irwindale, CA 91706
909-839-8949
Ceyster1@its.jnj.com

Abstract—**The current regulatory approach for assuring device safety primarily focuses on compliance with prescriptive safety regulations and relevant safety standards. This approach, however, does not always lead to a safe system design even though safety regulations and standards have been met. In the medical device industry, several high profile recalls involving infusion pumps have prompted the regulatory agency to reconsider how device safety should be managed, reviewed and approved. An assurance case has been cited as a promising tool to address this growing concern. Assurance cases have been used in safety-critical systems for some time. Most assurance cases, if not all, in literature today are developed in an ad hoc fashion, independent from risk management and requirement development. An assurance case is a resource-intensive endeavor that requires additional effort and documentation from equipment manufacturers. Without a well-organized requirements infrastructure in place, such "additional effort" can be substantial, to the point where the cost of adoption outweighs the benefit of adoption. In this paper, the authors present a Risk-Based Requirements and Assurance Management (RBRAM) methodology. The RBRAM is an elaborate framework that combines Risk-Based Requirements Management (RBRM) with assurance case methods. Such an integrated framework can help manufacturers leverage an existing risk management to present a comprehensive assurance case with minimal additional effort while providing a supplementary means to reexamine the integrity of the system design in terms of the mission objective. Although the example used is from the medical industry, the authors believe that the RBRAM methodology underlines the fundamental principle of risk management, and offers a simple, yet effective framework applicable to aerospace industry, perhaps, to any industry.**

TABLE OF CONTENTS

1. INTRODUCTION

Explosive growth of computer and software technologies in the latter half of the twentieth century has not only greatly enhanced the capability of earlier systems, but also created entirely new systems that propels mankind "*into another era, one in which machines extend the human mind as they have long extended the human muscle and senses*"[1]. In the medical device industry, intricate device functionality fueled by software spurs a wide range of new therapies, from closed-loop drug delivery systems to implanted neuron stimulators. These software-intensive devices greatly increase efficacy of therapies and improve patient safety when devices are developed correctly and used properly. This new-found capability inevitably drives device complexity and introduces new risks. In fact, the complexity of software-intensive systems is an essential property, not an accidental one [2]. Thus, this inherent complexity greatly impedes our ability to understand the conceptual integrity of system design [3], and in particular, all the safety aspects of system design. It comes as no surprise that a better approach is required to cope with this ever-increasing complexity.

The current regulatory approach for assuring device safety primarily focuses on compliance with prescriptive safety regulations and relevant safety standards [4] [5]. Such a practice has successfully guided safety-critical industries for the past few decades and will likely continue to play a role for the foreseeable future. This approach, however, does not always lead to a safe system design even though safety regulations and standards have been met. In a number of serious accidents, there had not been an ignorance of safety concerns, or absence of safety standards, rather, designers had failed to demonstrate a systemic, integrated and thorough consideration of safety in their product life-cycle development [4].

In the medical device industry, manufacturers are typically required to submit applications to FDA for review prior to commercial marketing of the device. Despite the best effort of manufacturers and the FDA reviewers, the FDA has seen an increase in the number of recalls of medical devices in recent years [6]. Their data analysis suggests that many of these problems were foreseeable; related to device design, therefore, preventable [7]. A number of serious product recalls involved with infusion pumps has prompted the FDA

978-1-4673-1813-6/13/$31.00 ©2013 IEEE

to rethink their approval strategy – how safety should be managed, reviewed and approved [8]. As a result of such rethinking, a new draft guidance for infusion pump – premarket notification {510(k)} submissions was circulated in 2010, in which the assurance case report is explicitly cited as a recommended practice for submission. The guidance concludes, "*The FDA believes the methodology (assurance case) will be particularly useful for presenting and reviewing information about the infusion pump*". Instead of manufacturers claiming safety through satisfaction of the regulatory requirements *implicitly* and *prescriptively*, now it will be the responsibility of manufacturers to argue *explicitly* and *descriptively* that their devices have achieved an acceptable level of safety assurance [9].

An assurance case is a generic term that simply refers to a formal method for documenting convincing arguments, combined with supporting evidence, to demonstrate the validity of a claim beyond reasonable doubt. When an assurance case addresses safety, it is often referred to as a safety case. Safety cases have been used in safety-critical systems, e.g. nuclear, avionics, railroad, etc. for some time. However it is still new to the medical device community.

From the regulatory reviewer's perspective, an assurance case serves as a vehicle to assess the validity of a safety claim for a medical device. It provides the examiner with a logic structure that is easy to follow without necessarily sorting though a large number of technical documents. The benefits of using an assurance case framework are obvious from the regulatory point of view. It is less obvious, however, from a manufacturers' perspective – what the overall benefit would be. Despite several potential benefits hypothesized nicely by Weinstock [5], it appears there is little objective hard data to support the benefit claims. What seems clear, though, is that substantial effort is required to support the assurance case activity. For instance, the computer-based Darlington Reactor Protection System in Canada required more software assurance effort (50 man-years) than the effort to develop the software itself [10]. Under the current US economic environments, there is little incentive, if any, for device manufacturers to adopt a new technique unless sufficient data is provided to demonstrate a favorable cost/benefit ratio. The cost of adoption is real and easy to measure, but the benefits may be less so.

There are basically two ways to drive a favorable cost/benefit ratio for industry: (1) demonstrate that a system implemented with an assurance case has a shorter development cycle, results in fewer recalls and customer complaints, and/or reduces the regulatory approval time, or (2) significantly reduce the effort required to support the assurance case under the premise that an assurance case indeed provides a supplementary tool to reassure the conceptual integrity of the system design.

This paper exploits the latter method. Most assurance cases, if not all, in literature today are developed in an ad hoc fashion, independent from the risk management and the requirements development. The paper aims to reduce this development cost by recasting the existing Risk-Based Requirements Management (RBRM) to produce assurance cases. In other words, the assurance case report for a specific product configuration is simply a byproduct of newly created framework – Risk-Based Requirement and Assurance Management (RBRAM).

This paper begins with a brief background review and architectural description of the RBRM, followed by summarizing essential elements of assurance cases and their structure. Next, an augmented architecture for producing assurance cases is proposed and construed. Finally, a closed-looped sedation delivery system is used as an illustrated example to show how the artifacts of the RBRAM framework are mapped to the assurance case report. We conclude the paper with a discussion on benefits of the assurance case for manufacturers, and our future development interests.

2. RISK-BASED REQUIREMENT MANAGEMENT FRAMEWORK

It has long been recognized that Systems Engineering (SE) plays a critical role in project performance that is measured through meeting cost, schedule and technical capability commitments [11]. Brook made a pointed remark in his classic paper "No Silver Bullet" [12]:

> *"The hardest single part of building a software system is deciding precisely what to build. … No other part of the work so cripples the resulting system if done wrong. No other part is more difficult to rectify later."*

Without doubt, SE's greatest impact comes during the requirements and concept formulation in order to decide precisely what to build. "*It is a time of great creativity. Yet creativity is one of the least understood of human activities*" [1]. This least understood of human activities often leads to the problems of misunderstanding of system purpose and missing system requirements in the early stage of development, which has proven not only costly, but often results in inferior systems that are difficult to rectify later. Over the years, wide ranges of requirements development methodologies and tools have been developed in an attempt to remedy this perennial problem. Examples include Quality Function Deployment [13], Functional Analysis [14], Use Case [3] etc. More recently, various Model-Based System Engineering Methodologies (MBSE) [15] have emerged as a leading method. The principle that underlies the MBSE methodology is that explicitly modeling can assist the engineer in coping with complexity by better supporting system requirements development, design, analysis, integration and other critical activities so that the problems of misunderstanding of system quality attributes[1] and missing requirements can be greatly alleviated.

[1] Quality attributes are often called "emergent properties" of the system. In the context of requirements, they usually can be recognized as "-*ilities*" requirements. They are a subset of non-functional requirements.

Motivation for the RBRM

The methods mentioned above generally work well when there exist well-established domain and well-understood design knowledge under well-defined environments, within which the system operates (e.g. off-highway vehicles). Such is usually not the case with a software-intensive system, even less so with a safety-critical medical system. Often, a slight change of intended use of a system could result in unintended and unacceptable risks. Methods mentioned above for requirements development do not typically reflect the quantitative risk aspects of the system, much less on the integrated risk profile. Rather, the risk assessment is done in ad hoc fashion, either isolated or independent from the requirements development. Without concurrently developed, integrated risk profile and systemic risk measurement matrices, it's nearly impossible to allocate optimal system resources to mitigate the risks.

This limitation prompted the authors to develop and practice an alternative development approach, coined as the Risk-Based Requirements Development (RBRD) methodology. The premise of this approach is that risk is inherent in all products, systems or processes, whether a safety, security or business risk. A different application or intended use creates different kinds of risks or risk profiles, so does each function or process of the system. Naturally, the primary objective of system design aims to uncover the system weaknesses, develop the integrated system risk profiles and mitigate all foreseeable system risks to an acceptable level while simultaneously satisfying stakeholders' requirements.

The idea of the risk-based methodology is not new; it has been around for many years. For example, Boehm's spiral model [16] and JPL's State Analysis (SA) [15] integrated with the STPA model [17] are essentially risk-driven methodologies. What is new, or perhaps different lies in the underlying premise of the RBRD approach, that is, risk is inherent in all products, systems or processes. Risk is broadly defined as "*effect of uncertainty on objectives*" (ISO 31000). This fundamental assumption logically leads to an elaborate, concurrent, holistic aspect of life-cycle system development that is anchored in the hierarchical risk modeling (HRM) [18]. Conceptually, the RBRD is a MBSE methodology that elevates the HRM to a principal and governing role in the life-cycle system development. Operationally, this approach utilizes the HRM as a common denominator to unify multifarious modeling methods by a common set of risk attributes – a triplet {scenarios, likelihood, severity}, and uses the probabilistic risk assessment (PRA) [19] as a common language to characterize risk and its corresponding contributions to the overall system risk profile. As a result, the effectiveness of system development can be measured quantitatively or/and qualitatively in probabilistic terms.

With the underlying assumption of the RBRD methodology, requirements in the RBRM are *explicitly* integrated with the PRA process. In fact, they are derived from it. Whether a new function is required or a particular alternative selected is largely determined by how effectively that function or alternative mitigates the identified risks to an acceptable level, and more importantly, its contribution to reduction of the overall risk profile of the system.

Major Artifacts of the RBRM

Scope consideration does not permit a detailed description of the entire RBRD methodology which includes risk-based trade study, decision making and the PRA process resulting from the HRM. Instead, a subset of this methodology relevant to assurance cases is addressed. A separate paper will be dedicated to describe the RBRD methodology. The RBRM is defined as:

> *A database that consists of requirements explicitly derived from the probabilistic risk assessment (PRA) processes; rationales that provide logic arguments to justify each requirement's veracity; and verification and validation (V&V) that demonstrate that the system under development for an intended application satisfies all requirements for its intended environments.*

Major artifacts for the RBRM are:

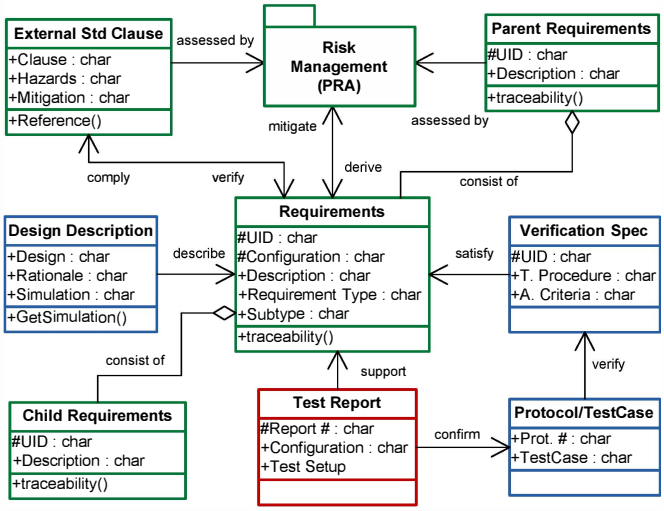

Figure 1: Class Model for a Typical Requirement Object

- Requirements – a set of hierarchical statements about a system in terms of functions, capabilities and quality attributes that completely define that system.
- Risk Assessment – a PRA mechanism integrated into the requirements framework.
- Rationale – a set of arguments reasoning how the requirement addresses the risk or risk profile, and serves as an association or link that ties mitigations, requirements and V&V together.
- V&V – consists of (1) a design description that demonstrates the selected solution satisfies a particular requirement, (2) requirement verification specifications that outline test procedure and acceptance criteria, and (3) a test record.

Requirements derived from the PRA form the core of the RBRM framework. Each requirement is associated with the risk profile explicitly. The class model in the Unified Modeling Language (UML) for a typical requirement object and its relationship to other objects or package in the RBRM is illustrated in Figure 1 (multiplicity not shown). Each artifact mentioned above can be readily identified from the model (rationale is embedded in each link object).

Process for Constructing the RBRM

As soon as the initial mission requirements have been conceived, the RBRD process should be initiated. Usually an application-level (AL) PRA commences first. As the name indicates, the AL risk assessment deals with application (mission) risks. In the context of medical devices, mission success is achieved by ensuring that therapy objectives are accomplished safely within the constraints of time and cost (efficiency) and meeting stakeholders' expectations (satisfaction). All hazards in the AL are identified and analyzed systematically and exhaustively through the HRM with assistance of clinical professionals. In this level, the majority of mitigations, if not all, tend to create new functions or processes that can eliminate hazards or prevent them from occurring. The evaluation goes on until all identified hazards at the AL are addressed. Successively, once a new function is created, new risks associated with that function arise. As a result, the mitigation is carried out in the next level down. The process goes on until the desired level of detail is obtained. Figure 2 shows a partial block diagram of the RBRD process (iteration loop not shown).

In general, a breadth-first search (BFS) method [20] is used to construe the hierarchical risk models, with which risks are explored horizontally in all directions one level at a time. It is worth mentioning that the PRA is performed throughout the entire requirements development cycle in an iterative fashion. For instance, the quantitative accuracy of the PRA may not be an early iteration priority, whereas completeness is. In later iterations, the PRA can be revised for more accuracy and greater simulation detail when the nature and context of the risks become clearer.

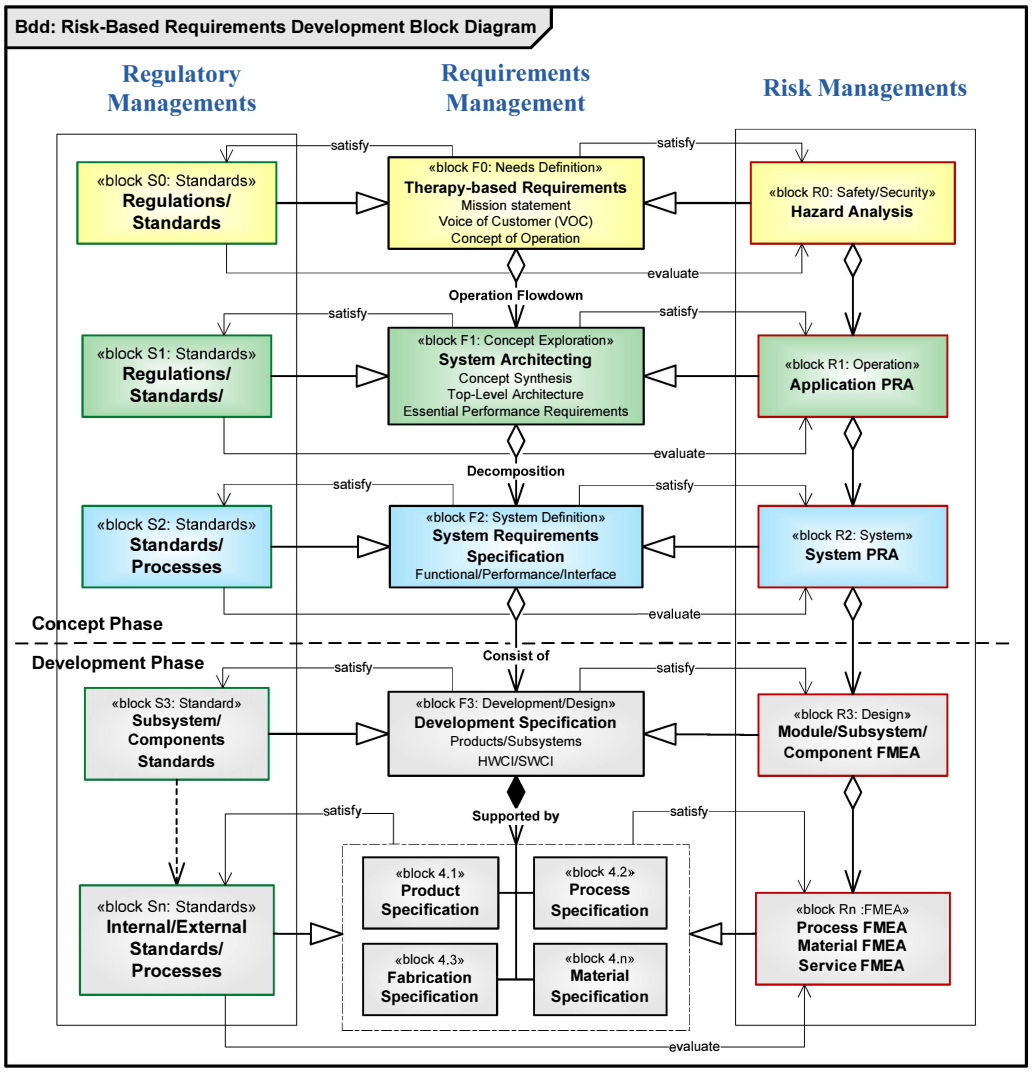

Figure 2: Partial Risk-Based Requirements Development Process

After briefly describing the elements and structure of the existing RBRM, attention can be turned to the elements and structure of assurance cases to see how the assurance case development fits naturally with the RBRM.

3. ASSURANCE CASE PRACTICE

As mentioned previously, an assurance case is a generic term that refers to a formal methodology for demonstrating the validity of a system quality attribute. When an assurance case addresses a safety attribute of the system, it is called safety case. When it addresses a security, usability, or dependability attribute, it is referred to as a security, usability or dependability case respectively. One must recognize, however, that most system quality attributes, if not all, never exist independently, rather they coexist in the state of complex interplay. For example, one cannot really address safety without consideration of other quality attributes, such as reliability, usability, security, etc. Each attribute influences the others in a subtle, but important way. Achievement of any one will have an effect on the achievement of others.

When requirements are derived to mitigate safety hazards, they become safety requirements. Following the same line of reasoning, they can be security, dependability, usability, or compliance requirements. For simplicity, this paper will be limited to discussion of the safety case, and it will be used interchangeably with the term assurance case in the following discussion.

Essential Elements of Assurance Case

The authors *extend* Bishop's original definition [10] of a safety case to the assurance case as:

> *"A documented body of relevant evidence that provide a convincing and valid argument that a system quality attribute <safety, security, dependability, etc.> is adequately assured for a given application domain under a given environment."*

From the definition we conclude that the following elements must be included in an assurance case:

- Claim – an explicit set of goals <safety, security, etc.> about the system.
- Evidence – a set of relevant proof, test results, etc. that support a claim.
- Argument – a set of convincing & valid reasoning that link the evidences to the claims.
- Context – ground the claims within a given application domain and environment.

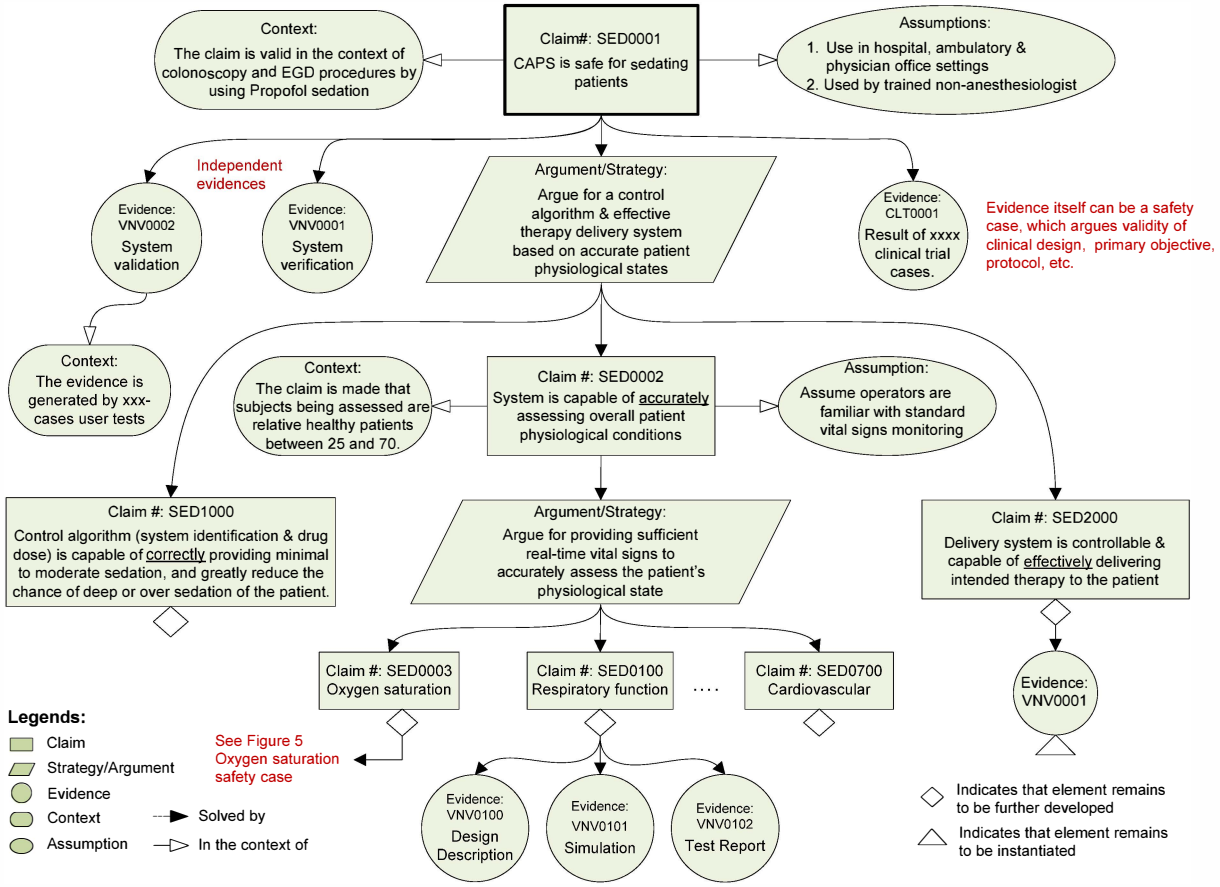

Figure 3: Essential Elements and Structure of Assurance Case in the GSN

Conventionally an assurance case is developed as a text report. Increasingly, a graphical approach with Goal Structuring Notation (GSN) developed by Kelly [21] has been adopted for safety case development. In this paper, the GSN was adopted for its readability and simplicity.

Process for Constructing an Assurance Case

Having discussed the elements of assurance cases, the next step is to consider the structure and process for producing them. Since the assurance case should be considered throughout the product life-cycle, it is best served when its structure matches the hierarchical structure of product life-cycle requirements. It begins with a top claim about a system quality attribute, e.g., safety. From that claim cascades an argument that supports the top-level claim, where the argument is progressively decomposed into derived subclaims for corresponding subsystems, and subsequently supported by a hierarchy of subsidiary safety cases. Taken together, they make the top-level claim creditable. Successively, these sub-claims are themselves supported by either additional sub-claims or evidences that clearly show the subclaims to be true. This process goes on until all subclaims are appropriately supported with evidence. It is in many ways similar to the requirements decomposition process commonly practicing in SE. Figure 3 shows the essential elements and the structure of assurance cases in the GSN. Additional symbolic notations can be found in reference [21].

The argument is at the heart of assurance cases. It explicitly links the evidence to the claim, and explains how this evidence relates to the claim. The quality of the assurance cases is largely determined by the quality of the logical arguments. Weinberg used "logical and mathematical boxes" as metaphors to elaborate importance of the logical argument.

> *"...you can catch phenomena in a logical box or in mathematical box. The logical box is coarse but strong. The mathematical box is fine grained but flimsy. The mathematical box is a beautiful way of wrapping up a problem, but it will not hold the phenomena unless they are been caught in a logical box to begin with* [22]."

The argument is the "logical box" for the assurance case, while the "mathematical box" is the evidence that convincingly concludes the claim. Different types of arguments can be developed based on type of claims at hand. A comprehensive description of each is beyond the scope of this paper, however additional details can be found in references [10].

4. INTEGRATION OF RISK-BASED REQUIREMENTS AND ASSURANCE CASES

An assurance case can be developed in several ways [4] [5], and [10]. According to Weinstock, there are two approaches for structuring a safety case. One is to gather a complete set of the safety requirements and to verify they are satisfied; the other is to identify all the potential safety hazards and to confirm they have been adequately mitigated. These two approaches are logical complements, analogous to constructing the familiar Fault Tree (FT) or Success Tree (ST). In theory, one can construe a safety case as a ST-type by using the first approach. In practice, it is difficult, if not impossible; to build a pure ST-type safety case due to several practical problems associated the ST [23]. Perhaps, a more sensible approach is to argue a claim alternatively either in success or failure space[2] to derive successive (sub)claims based on the type of arguments at hand. Nevertheless, the existing RBRM possesses necessary artifacts to support either approach to create assurance cases. In fact, the RBRM framework and the assurance case share many similarities in terms of artifacts and structures. Table 1 highlights the corresponding relationship between these two.

Table 1: Corresponding relationship between the RBRM & Assurance Case

	RBRM Framework		Assurance Case
Artifacts /objects	PRA → Requirement	↔	Claim
	Rationale	↔	Argument
	V&V	↔	Evidence
	Various Objects	↔	Context
Structure	Hierarchical, Bidirectional traceability	↔	Hierarchical Bidirectional traceability

Semantically, objects in the RBRM and assurance case can be logically interchangeable. Caution must be taken, however, when directly using objects in the RBRM to construe an assurance case. Each artifact/object is examined below to see how the RBRM framework can be augmented to support assurance cases.

- A claim is strongly related to a requirement. These two clearly overlap, but equally clearly are not exactly the same. Although one can use a requirement as a claim, it may not be a good practice for at least one reason: a requirement is written in formal syntax, with constraints on use of any adjective or adverb in its verbiage. On the other hand, a claim is often written succinctly and expressively to aid a reviewer in understanding. In consideration of integrity of current database structure, the authors have opted for adding a property "claim" in the requirement class. If a requirement is not suitable as an explicit claim, the "claim" field can then be used to structure the safety case instead.
- An argument in assurance cases often contains "strategic" intent. Rationale in the RBRM is primarily used to justify decomposition or derived relationships. The rationale is typically weaker than the direct argument of assurance cases. To leverage existing rationale link objects in the database, an "argument" field is created for the link class. If an existing rationale

[2] Both FT and ST lie in the same probabilistic space: the probabilities of success and failure of a particular event are added up to one.

is determined to be not strong enough for the intended argument, the "argument" field can then be used to support the explicit argument over adjacent claims.

- Context plays a critical role in building the assurance case. It grounds the claim or evidence within a given context of the claim or evidence in existence. In the RBRM, there is no dedicated object to capture this information. Instead it is encapsulated in various objects of V&V, DD, or in the PRA. The strategic or tactical descriptions inside these objects verify or validate the requirements within given prospective context. Again, a "context" field can be created in these classes to represent that information.
- A requirement in the RBRM requires several pieces of evidence to support its validity. These include: (1) design descriptions that may include the qualitative and quantitative description of design, simulation of hardware, correctness proof of software code, etc., (2) verification specifications that outline test strategies, procedures and acceptance criteria, and (3) test records. Missing any one of these artifacts would result in inferior evidence. A design description without verification is unconvincing and incomplete while verification without design description is unclear and weak. Taken together, the evidence is stronger than if only one is presented alone. In this case, the assurance case can fully leverage existing V&V objects in the RBRM database without further augmentation.

Not surprisingly, the resulting RBRAM framework has an identical structure as the RBRM. In fact, the assurance case is merely another view of the RBRAM framework – called an assurance view. If this view doesn't come out correctly, it would be prudent to recheck the soundness of the logical decomposition of the risk-based requirements. To this end, perhaps the greatest benefit for creating assurance cases with this structure is that it provides a supplemental method to reassure the conceptual integrity of the system design. Indeed, "*The conceptual integrity is central to product quality*" [2].

5. AN ILLUSTRATED EXAMPLE – CAPS

To demonstrate that the RBRAM framework is capable of supporting an assurance case report, Computer-Assisted Personalized Sedation (CAPS)[3] is used as an illustrated example. The example comes from a real application, having already been deployed using TcSE[4] database. The example has been simplified for use in this paper.

The CAPS is a novel drug delivery method that integrates continuous physiological monitoring into procedural sedation delivery by using real-time computer control. It allows a physician/nurse care team to administer Propofol sedation to a patient. The architecture of CAPS can be modeled as a sense-control-actuate pattern, see Figure 4. The operating principle of this pattern is cycling through the sequence consisting of 1) acquiring the sensor data from sensor networks, 2) performing a set of functions in high-level controllers, 3) executing a set of control laws in low-level controllers, and 4) sending control signals to various actuators to achieve the desired therapy.

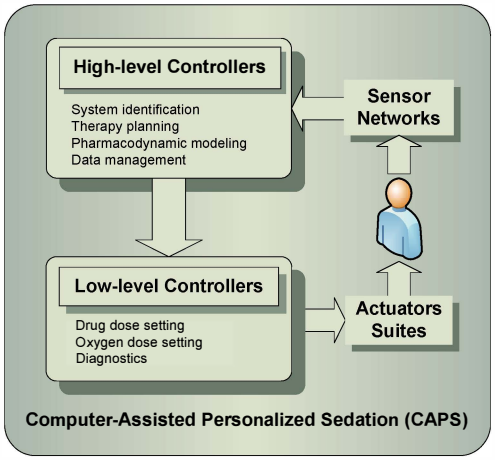

Figure 4: Architecture Pattern of the CAPS

The purpose of this example is to show that a safety case report can be directly generated from the RBRAM database based on the discussion of the previous section. Figure 5 illustrates a view of the safety case in GSN produced from the TcSE database. The diagram focuses on one of the monitoring functions, and the diagram's structure also reflects the underlying principle of the RBRD process. The diagram is described more detail in the following paragraphs.

From a safety perspective, the top claim can be derived from the mission statement, i.e., "CAPS is safe for sedating patients", referred to Figure 3. The concept of safety is multidimensional, and encompasses a number of quality attributes. In the context of this simplified example, assurance of the top claim is achieved via an argument over the accuracy, correctness and effectiveness of each architectural component (sense-control-actuate). In addition, two direct evidences – system V&V and a clinical trial, are presented independently to support the top claim and which also makes it more robust. The design aspect of the subsystem is discussed next, starting from claim # SED0003 in Figure 5.

Suppose a hazardous situation identified by the HRM in the AL is that a physician is unaware of a patient's adverse physiological state. One such state is hypoxia – a low oxygen saturation condition in the patient's arterial blood. If hypoxia isn't treated promptly, it can result in harmful effects on the patient that is an unacceptable effect assessed by the PRA. Many potential causes can lead to a hypoxia, of course. Again the discussion here is limited to the cause induced by deep or over sedation.

[3] The CAPS device is a Class III investigational device in the U.S. manufactured by Ethicon Endo-Surgery, Inc. under review by FDA.

[4] TcSE (Teamcenter® for Systems Engineering) is marketed by Siemens PLM Software.

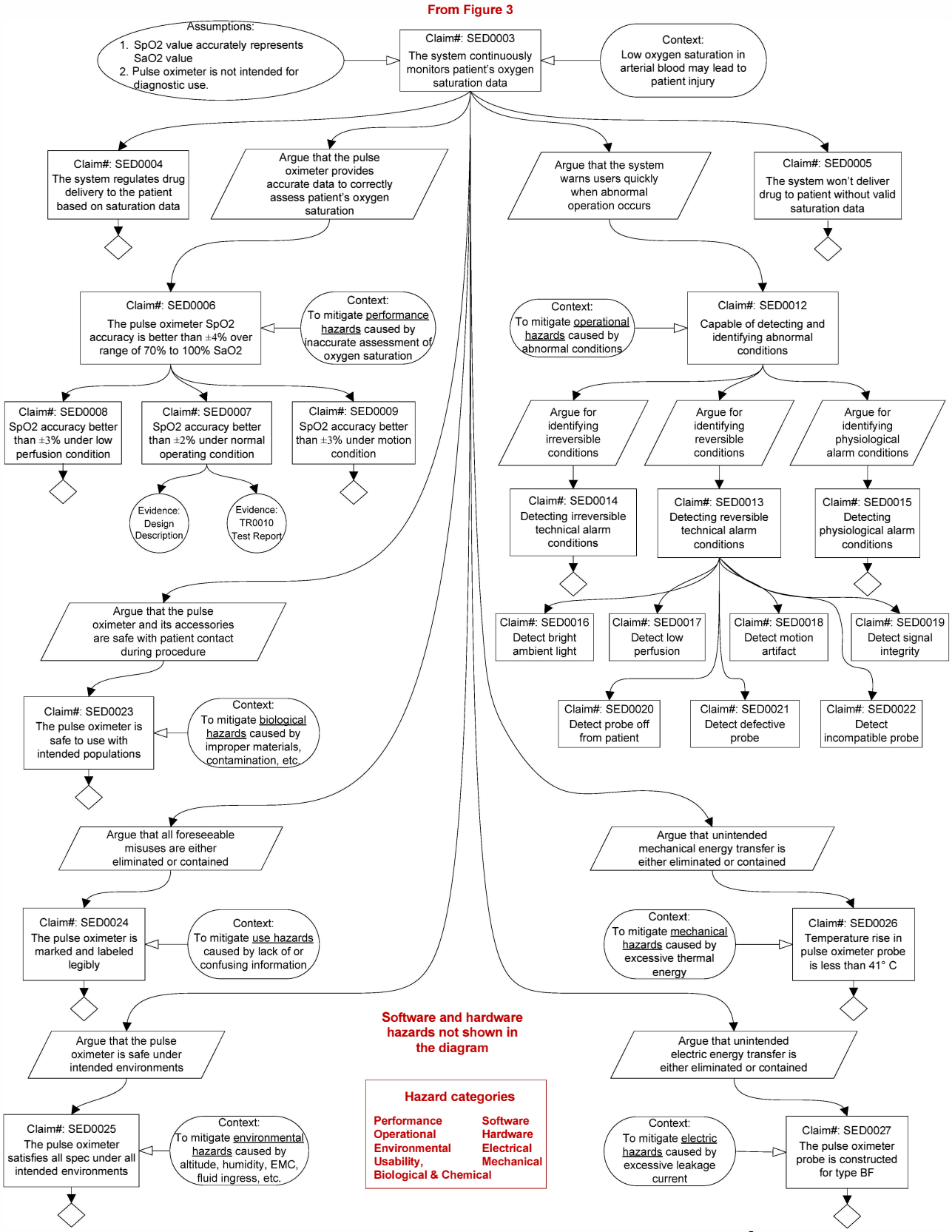

Figure 5: Safety Case for a Subsystem of CAPS – Pulse Oximeter[5]

[5] The diagram is a direct output from TcSE database. The locations of notations & links were rearranged manually to enhance readability for publication. The standard output from TcSE does not provide user control over notations & links location in the output format. The red text in the graph was added after exporting from TcSE. For readability, texts inside diamond notation – "further development" are not shown.

To prevent this hazardous situation from occurring, a prerequisite safety requirement "The system shall monitor patient's arterial oxygen saturation at intervals of no greater than 10 sec" (claim SED0003) which is elicited as a result of the PRA. In addition, two auxiliary safety requirements, "The system shall regulate therapy delivery to the patient in accordance with patient's arterial oxygen saturation" and "The system shall acquire no less than 2 min worth of valid oxygen saturation data before delivery of drug therapy to the patient" (claim SED0004/5), are also derived as a result of the PRA. After deriving all safety requirements relevant to hypoxia (not shown completely) from the risk model, the PRA process is ready to proceed to the next level down to address hazards associated with the newly created function – monitoring arterial oxygen saturation in a greater detail.

Several designs can realize the requirement SED0003. Assume a pulse oximeter has been selected as a choice of implementation as a result of the risk-based trade study based on the fact that this selection is the best option to reduce the overall risk profile of the system. The choice, nevertheless, brings a new set of safety risks to the system. They are identified and classified in nine hazard categories through risk modeling (see classification in the red textbox of Figure 5). The PRA is then carried out for each category systematically and exhaustively. In operational hazards, for instance, abnormal conditions caused by one or more faults will inevitably occur. A function of "capable of detecting and identifying abnormal conditions" (claim#: SED0012) is derived from the PRA to mitigate the safety risk (described in Context box). Based on risk modeling, abnormal conditions are bifurcated into physiological (claim#: SED0015) and technical conditions, where technical conditions can be further classified as reversible (claim#: SED0013) and irreversible (claim#: SED0014) conditions. Reversible conditions demand an operator intervention to recover the system from fault states back to a normal state while irreversible conditions require the system to exit from fault states to a safe state gracefully. It should become clear at this point that any new functions required to mitigate one or all abnormal condition(s) will be largely determined by how effectively these functions mitigate risks to an acceptable level and what additional risks are introduced by these functions. In other words, it is more important to assess its global contribution to the overall safety profile of the system than its local contribution.

Identifying a fault, without notifying the operator and giving clear instructions how to deal with it, clearly is not only insufficient, but also imposes intolerable safety risks on the system as determined by the PRA. This leads to the creation of auditory/visual alarm functions, in-line troubleshooting procedures as well as instructions for use (not shown). The resulting mitigations in turn may induce new safety risks associated with them, requiring safety cases of their own. For example, flaws in the human factors design of auditory/visual alarm interfaces or instructions and warning messages to operators could lead to patient harm. The process goes on until all categories of hazards are reduced to an acceptable level defined in the risk management plan.

To assist in the undertaking, a domain standard, like ISO 9919 – 2005: *Particular requirements for the basic safety and essential performance of pulse oximeter equipment for medical use*, can be conveniently used as a hazard template to facilitate the completeness of the PRA. At the end of this thorough process, a complete set of logically classified, structured and linked requirements are arranged in the database hierarchically. Precisely, these hierarchically, logically classified, structured and linked requirements lay a foundation for producing the safety case report. Figure 5 faithfully reflects the foregoing discussion.6 It is worth emphasizing that proper classification of objects is critical to build safety cases. If done right, it can lead to a smaller, cohesive safety case with a simpler structure.

In addition to output safety cases, the RBRAM continues to produce various system documents as before in the desired format, such as, Requirements Specifications, Design Descriptions, Requirements Traceability Matrices, etc.

Thus far, the HRM and the PRA have been considered as a "black box". Space does not allow a lengthy discussion here, but reviewing the fundamental activities that underlie the RBRD methodology should help to underpin the idea. First, the HRM is essentially a coarse grain of modeling to *unify* other modeling methods, such as, functional modeling, behavior modeling, use case modeling, etc., and to *qualify* overall system success/failure behavior in *probabilistic terms*. Second, the PRA mechanism, e.g., fault trees, could be thought as a finer grain modeling to *quantify* each subsystem's success/failure contribution to the total system by assigning either point probabilities if accurate data is available or probability distributions when data is sketchy to each event or scenario node within fault trees. It should become evident from the illustrated example in this section that the outcome of the RBRD process is the creation of a very large set of requirements that are explicitly linked to myriad risks (only implicit or completely missing in other methods or models), and hierarchically organized into sets or subsets with prioritized risk rankings. If done well, the set of risks at any given level of hierarchy would approach a "complete set," as do the requirements.

6. CONCLUDING REMARKS

This paper has demonstrated the feasibility and capability of leveraging the existing RBRM to produce a safety case in the GSN with minimal additional effort. The simplicity of the RBRD concept not only fits well with the assurance case

[6] An out-of-box TcSE has limited capability to manipulate Visio objects. Customizing the tool through scripting (xml, tcl) helps expand its capabilities, as we did in the example, but still rather limits in terms of manipulating the stencil locations and controlling presentation of a connected, but truncated graphs in an orderly manner. In short, it is unlikely that TcSE in the current form could adequately support assurance cases in the GSN.

concept, but also complements it. Besides the benefits mentioned previously, the integrated RBRAM framework offers several attractive features:

- The assurance case is not an error-free technique [24]. However, its advantage lies in that the reasoning of a formal argument is different in form from that of the device design, so the odds of making the same mistake when using two different techniques, without notice, are significantly reduced.
- It is not difficult to recast classified objects in the form of other type cases, e.g., compliance or usability cases, since many of the artifacts have been created and linked properly in the RBRAM. Simply modifying the query scripts should yield the desired output. These cases can help manufacturers to address a particular quality attribute that concerns them.
- The structure of the safety case in Figure 5 represents the hazard-oriented safety view. Other safety views rather than the hazard view are possible if the RBRAM is constructed with other views in mind from the onset. Different views of safety cases may offer different perspectives for different stakeholders during different stages of development.

An assurance case can be applied to both an existing system and to a system that is being designed. When applied to an existing system, it can be used as a logical verification tool to assure the correctness of requirements decomposition. When it is applied to a system being designed, the assurance case can be utilized as a formal method to guide the logical decomposition of requirements regardless of the approach being used, QFD, FFBD, MBSE, etc. Either way, the assurance case helps to improve and assure the conceptual integrity of the system design.

Although the example used is from the medical industry, the authors believe that the RBRAM methodology underlines the fundamental principle of risk management, and offers a simple, yet effective framework applicable to aerospace industry, in fact, to any industry that values risk management.

There are at least two issues that need to be addressed in order to build the effective RBRAM. First, better organization of information is needed. For example, much of the context information being used by assurance cases has been captured in the RBRM, distributed within various objects. Recreating such information, as demonstrated in our example, not only consumes time in repetition but also causes maintenance problems since it requires extra effort to synchronize two pieces of almost identical information. By the same token, the subtle differences between arguments and rationale also need to be reconciled. Ultimately the assurance case should be merely a view of the system from a different viewpoint. Second, the GSN simply is a graphical way of organizing information. It helps users escape from a flatland of the textual report to a visual display that is easy to follow and comprehend. However, the GSN, a low-density information display, forces reviewers to rely on their visual memory – a weak skill for humans [25], to search back and forth for data spread over many pages. This context switching inevitably disrupts reviewers' continuity of reasoning about information, and corrupts their ability to make a contrast or comparison. The development interest naturally becomes the investigation of an optimal strategy for constructing the assurance case report that enhances high-density information display, avoids the disruption of context switching and turns empirical observations into credible evidences within the eye span of reviewers [25].

REFERENCES

[1] E. Rechtin, "System Architecting", 1st Edition, Prentice Hall, New York, 1991

[2] F. P. Brooks, "The Mythical Man-Month", 20th Anniversary Edition, Addison-Wesley, 1995.

[3] G. Booch, R. A. Maksimchuk, M. W. Engle, B. J. Young, J. Conallen and K. A. Houston, "Object-Oriented Analysis and Design with Applications", Third Edition, Addison-Wesley, 2007.

[4] T. Kelly, "A Systematic Approach to Safety Case Management", 2003. http://www-users.cs.york.ac.uk/~tpk/04AE-149.pdf

[5] C. B. Weinstock and J.B. Goodenough, "Towards an Assurance Case Practice for Medical Device", Technical Note, CMU/SEI-2009-TN-018, Software Engineering Institute, Carnegie Mellon University, October, 2009.

[6] FDA, "2006 to 2010 Medical Device Recalls", 2010 http://www.fda.gov/MedicalDevices/Safety/RecallsCorrectionsRemovals/ListofRecalls/default.htm

[7] FDA, Draft guidance "Total Product Life Cycle: Infusion Pump – Premarket Notification [510(k)] Submission", April, 2010

[8] FDA, "Final Order to Baxter to Recall, Refund, or Replace the Colleague Infusion Pumps", 2010 http://www.fda.gov/MedicalDevices/Safety/AlertsandNotices/ucm218753.htm;

"CareFusion Corporation, Alaris PC Units (M. 8015)", http://www.fda.gov/MedicalDevices/Safety/RecallsCorrectionsRemovals/ListofRecalls/ucm229651.htm

[9] D. Feng and C. Eyster, "Towards Integrating Assurance Cases into Risk-Based Requirement Management", 22nd Annual INCOSE International Symposium, Rome, Italy, 2012

[10] P. Bishop and R. Bloomfield, "A Methodology for Safety Case Development", Proceedings of the Sixth Safety-critical System Symposium, Birmingham, UK, 1998; http://www.adelard.com/papers/sss98web.pdf

[11] J. P. Elm, D. R. Goldenson, K. E. Emam, N. Donatelli and A. Neisa, "A Survey of Systems Engineering

Effectiveness", SPECIAL REPORT, CMU/SEI-2008-SR-034, SEI.

[12] F. P. Brook, "No silver bullet – essence and accidents of software engineering", Computer 20, 4, April, 1987, pp10-19.

[13] Y. Akao, "Quality Function Deployment: Integrating Customer Requir*ements into Product Design*", Productivity Press, New York, 1990.

[14] B. S. Blanchard and W. J. Fabrycky, "Systems Engineering and Analysis", 4th Edition, Prentice Hall, NJ, 2006

[15] J. A. Estefan, "Survey of Model-Based Systems Engineering (MBSE) Methodologies", INCOSE-TD-2007-003-01 Rev B, June, 2008

[16] B. W. Boehm, "A Spiral Model of Software Development and Enhancement", IEEE computer, Vol. 21, No. 5. 61-72, 1988

[17] M. S. Herring, B. D. Owens, N. Leveson, M. Ingham, and K. Ann Weiss, "A Safety-Driven, Model-Based System Engineering Methodology, Part I", MIT Technical Report, 2007, http://sunnyday.mit.edu/JPL-Part-1.pdf

[18] Y.Y. Haimes, "Hierarchical holographic modeling", IEEE Transactions on Systems, Man and Cybernetics", Vol. 11, No. 9: 606-617, 1981

[19] T. Bedford and R. Cooke, "Probabilistic Risk Analysis – Foundations and Methods", Cambridge University Press, 2001

[20] J. Kleinberg and E. Tardos, "Algorithm Design", Addison Wesley, New York, 2005

[21] T. Kelly, "Arguing Safety — A Systematic Approach to Managing Safety Cases", Ph.D. Dissertation, University of York, 1998

[22] G. M. Weinberg, "An Introduction to General System Thinking", Silver anniversary Edition, Dorset House Publishing, 2001. The quote is from J. R. Platt, "*Strong Inference*", Science, Vol. 146, No. 3642, October, 1964.

[23] NASA, "Fault Tree Handbook with Aerospace Applications"; NASA Office of Safety and Mission Assurance, 2002

[24] W.S. Greenwell, J.C. Knight, C.M. Holloway and J.J. Pease, "A Taxonomy of Fallacies in System Safety Arguments", 2006 http://www.cs.virginia.edu/papers/paper-issc06-fallacies-as-printed.pdf

[25] E. R. Tufte, "Envisioning Information", Graphics Press LLC, 2006

BIOGRAPHIES

David Feng *received his B.E. in Mechanical Engineering and M.S. in Applied Mathematics in China, and master degrees in Manufacturing Engineering, Electrical Engineering and System Architecting and Engineering from Northwestern University, University of Illinois and University of Southern California respectively. He has twenty years technical experience in the fields of electromechanical, software and electrical engineering. For the past several years, he has worked as a Principal Systems Engineer, developing a software-intensive sedation delivery system, for Ethicon Endo Surgery, Inc., one of the Johnson & Johnson operating companies. He is a member of IEEE and INCOSE.*

Curt Eyster *received his B.S.E.E. from Messiah College, an M.S. in Bioengineering from Arizona State University and an M.S. in Management from Indiana Wesleyan University. He has served as a Systems Engineer for multiple complex medical device programs during his 15 years with operating companies of Johnson & Johnson. Currently, he is an R&D Manager for Biosense Webster, Inc., one of the Johnson & Johnson operating companies. He is applying the risk-based requirement development process for the treatment and cure of cardiac arrhythmia. He is an IEEE and INCOSE member.*

Acknowledgement: This paper has benefited from the critical review of Dr. Joseph S. Alford.