

# Sufficient Evidence: Making the Case for Safety

Richard Schrenker

An article in the Sep./Oct. 2007 *BI&T* noted two concerns associated with a lag between the development of software-based medical device systems and their support structures.

*Imagine the implications as medical devices increase their use of and dependence on networked resources...This begs the question of whether anyone has examined, let alone established, the validity of extrapolating 20<sup>th</sup>-century medical technology development, assessment, and regulatory practices for the 21<sup>st</sup>-century healthcare system...*

*Twenty years ago, I could go to the service manual of any device...and determine what it was supposed to do and how it did it...I could develop tests to verify operation, and I could validate that the device could do what it was supposed to do...<sup>1</sup>*

The standards community is responding to this challenge faced by the integration of medical devices and software. IEC 80001, *Application of risk management for IT networks incorporating medical devices*, for example, which is expected to be published in 2010, makes these and related issues concrete and immediate. But what will it really take to instill confidence in medical technology professionals that the devices and systems under their care meet requirements not yet fully defined? Consider the following from the National Research Council's *Software for Dependable Systems: Sufficient Evidence?*

*...the pursuit of dependability in software systems should focus on the construction and evaluation of evidence...software is guilty until proven innocent...This approach is...becoming standard in the world of systems safety, in which an explicit safety case...is usually required...*

*...a software system may not be declared "dependable" based on the method by which it was constructed...Those claiming dependability for their software should therefore make available the details of their claims, criteria, and evidence...The willingness of a supplier to provide such data, and the clarity and integrity of the data that the supplier provides, will be a strong indication of its attitude to dependability.<sup>2</sup>*

*Software for Dependable Systems* argues that quality process alone is insufficient to ensure safety and dependability; claims supported by evidence are required as well.

## A Sample Safety Case

*Claim:* A monitoring system is safe for use in an ICU.

*Arguments:*

- The device meets regulatory requirements.
- Safety requirements for the clinical environments have been identified and validated.
- The device meets the requirements of the clinical environments.

*Evidence:*

- The manufacturer provides evidence of 510k and local regulatory approvals.
- The following provide evidence that the device meets clinical requirements:
  - Clinical engineering evaluation of functional and physical safety.
  - Human factors and/or clinical simulator evaluation.

80001 inherits its risk management framework from ANSI/AAMI/ISO 14971:2007, *Medical devices—Application of risk management to medical devices*, which, like most existing standards, is process based. Should 14971 and its derivatives change to reflect this (relatively) new assurance philosophy? With the draft of IEC 80002, *Guidance on the application of ISO 14971 to medical device software*—which addresses such issues as alarms, human factors, and networks—mentioning safety cases, it appears to be under consideration.

But why this change, and why now? Software-based medical device systems are meeting needs, and the sky is not falling. But as hospitals begin to assemble network-based medical device systems, we need to step back and consider whether we are at a technology tipping point. Should we wait and react to the unintended consequences of the change, or are there reasons to try to get ahead of the curve? That software brings problems requiring new approaches is not news. As Nancy Leveson noted more than a decade ago:

*For the most part, standard software engineering tech-*

*niques and processes are being used to develop safety-critical software without any consideration of the special factors and unique requirements for enhancing safety.*<sup>3</sup>

Or perhaps more colloquially: “When all you have is a hammer, things that shouldn’t look like a nail too often do.” Systems and software engineering have been struggling to come up with more appropriate tools for some time; safety cases (and their close relatives of assurance and dependability cases) are one result of their effort. They have been in commercial and military use for over a decade in Europe,<sup>1</sup> and as noted they are beginning to be discussed in medical standards activities.

### What is a Safety Case?

*A documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment.*<sup>4</sup>

Safety cases are rooted in work describing the nature of valid arguments in general.<sup>5</sup> Some equate them with the graphical models like Claims-Arguments-Evidence (CAE) and Goal Structuring Notation (GSN) that support them (see “To Learn More” for ubiquitous pointers thereto, and visit [www.aami.org/publications/BIT](http://www.aami.org/publications/BIT) for examples), but safety cases are first and foremost about substantiating an argument:

*Every non-trivial safety-critical system has a safety case, regardless of whether...documented or explained...The “real” safety case is the true reasoning as to why the system is acceptably safe. The depicted safety case is a representation of the argument that is hoped to mirror the actual safety case...*<sup>6</sup>

### Where is a Safety Case?

This article will set aside consideration of the issues around safety cases for devices and focus on clinical systems. Suffice it to say, however, that the definition of a safety case obviously can be applied to devices as well.

Every operating room (OR) has a safety case. Every intensive care unit (ICU) bedside, every ICU itself, and every transport system has a safety case. Tens if not hundreds of clinical use cases have safety cases, including every information technology (IT) network incorporating medical devices. It doesn’t take much time on the floor to know that constructing implicit safety cases in medicine is nothing new.

Systems are created for only one reason: to provide properties and behaviors from constituent components that they cannot provide individually. Those new properties and behaviors bring risks all their own. Any medical

device system currently in use has had those risks evaluated by FDA and by the clinical engineering departments where it resides.

### Why Now?

Before large-scale integration and general-purpose computers, electronic components could essentially be modeled as physical realizations of fixed mathematical functions. This one-to-one correspondence permitted deterministic analysis of designs modeled in schematic diagrams and realized in construction. One could literally validate that the device could do what it was intended to do *and nothing else*. One could devise tests to fully verify that functionality implied by the logical representation was fully realized in the physical implementation.

With software increasingly defining functionality, those days are over. Service manuals continue to provide situational what-to-do instructions, but I have seen nothing that replaces the tools we once had to support analysis. A colleague recently remarked that it seems we have tacitly accepted the emergence of “faith-based engineering.” This oxymoron should not be considered acceptable.

I pose the following to clinical engineers and biomedical equipment technicians: “What would make you comfortable with supporting a request to implement aspects of critical care device functionality on an IT network?” Would being told that the network had been designed and installed according to relevant process and standards be sufficient? Or would you want to see more, like evidence that the result of application of the processes and standards addressed functional and non-functional requirements relevant to critical care? Consider one of the fundamental principals of medicine: *First do no harm*. How much evidence do you need to support an argument that worst case failure of an ICU monitoring system, which depends on both medical devices and the software that runs them, could not place a patient, or hundreds of patients, at unacceptably increased risk?

### Who Constructs the Safety Case?

This question is answered indirectly in the “Key Findings and Recommendations” section of *Sufficient Evidence*:

*Customers and users can make informed judgments when choosing suppliers and products only if the claims, criteria, and evidence for dependability are transparent.*<sup>2</sup>

In other words, it is up to the recipient of a device, system, or service to demand of the provider claims

and evidence that can be objectively evaluated. Unfortunately, many of these recipients might not know their options or the right questions to ask, or may have preconceived notions about safety cases (see “What about the Downsides” for more). A regulatory authority like FDA could require that a manufacturer provide a safety case with a premarket approval submission. Clinical engineering departments may choose to require the same of manufacturers, thereby taking a first step at addressing the issues raised at the beginning of this article. This will prove particularly important in moving ahead with 80001, where it should be reasonable to expect the system implementer to make and support claims of safety, reliability, maintainability, and so on. Where the system comprises devices and subsystems provided by a mix of manufacturers, to do so will almost certainly require the provision that supporting safety cases from each manufacturer be sufficiently transparent and complete to enable composing the case for the overall system.

### What about the Downsides?

It has been difficult to find literature of safety cases being applied to the problems posed by interoperability. Finding problems posed by safety cases, unfortunately, is easier at this point.

*...the sheer volume of low level analysis and supporting evidence for a large system can jeopardize the clarity of a high level argument...*

*Safety case development is often treated as a post-construction presentation concern, and consequentially it often fails to capture the engineering judgment and experience applied...*

*For companies that build a number of systems within a certain domain, there is often repetition of information across the safety cases for such systems. Yet reuse of this information is seldom handled in a systematic or cost-effective manner.<sup>7</sup>*

Safety case maintenance has its difficulties as well, e.g., recognizing the impact of changes to systems to their safety cases.<sup>8</sup>

Colleagues from the regulatory and manufacturing worlds have noted a Catch-22 type of concern. Without standards for safety cases, a manufacturer won't know what a regulator needs until they submit, and a regulator won't know what to require until after receiving a submission.

### Where Next?

Safety case literature argues that the development of safety cases should commence early in the development

of projects and programs, and we are beginning to investigate their applicability to requirements engineering and validation in our Medical Device Plug-and-Play Program ([www.mdnp.org](http://www.mdnp.org)).

Finally, a plea: recently I was one of a handful of hospital-based attendees at an FDA-sponsored workshop on safety cases. Most were from industry or government, and I felt compelled to remind both groups that the impact of the work affects hospitals, too. If the clinical engineering community concurs, its voice needs to be heard. As a suggestion, the CE-IT Community ([www.ceitcollaboration.org](http://www.ceitcollaboration.org)) may want to respond to the challenge posed by *Sufficient Evidence*, e.g., by supporting investigation of the applicability of assurance and safety cases to the CE-IT domain.

### To Learn More

Sufficient evidence that good work has been done can be found in a number of places. My favorites include:

- Adeland: [www.adeland.com/web/index.html](http://www.adeland.com/web/index.html).
- Dependability Cases: [www.sei.cmu.edu/pub/documents/04.reports/pdf/04tn016.pdf](http://www.sei.cmu.edu/pub/documents/04.reports/pdf/04tn016.pdf).
- University of Virginia Dependability Research Group: <http://dependability.cs.virginia.edu/info/Welcome>.
- Publications by Tim Kelly: [www-users.cs.york.ac.uk/~tpk](http://www-users.cs.york.ac.uk/~tpk). ■

### References

1. Schrenker R. Learning from failure: The teachings of petroski. *Biomed Instrum Technol.* 41(5);Sep/Oct 2007:395–398.
2. Jackson D, Thomas M, Millett LI, eds. *Software for Dependable Systems: Sufficient Evidence?* National Academies Press, 2007.
3. Leveson N. *Safeware*. Addison Wesley, 1995.
4. Bishop P, Bloomfield R. *A Methodology for Safety Case Development*. Safety-critical Systems Symposium, February 1998.
5. Toulmin S. *The Uses of Argument*. Cambridge University Press, 1958.
6. Habli I, Kelly T. *Safety Cases Depictions vs. Safety Cases—Would the Real Safety Case Please Stand Up?* Proceedings of the 2nd IET International Conference on Systems Safety, 2007.
7. Wilson S, Kelly T, McDermid J. *Safety Case Development: Current Practice, Future Prospects*. Proceedings of 1st ENCRESS/12th CSR Workshop, 1995.
8. Kelly T, McDermid J. A systematic approach to safety case maintenance. *Reliability Engineering and System Safety.* 71;2001:271–284.

Richard Schrenker is the systems engineering manager in the Department of Biomedical Engineering at Massachusetts General Hospital, Boston, MA.