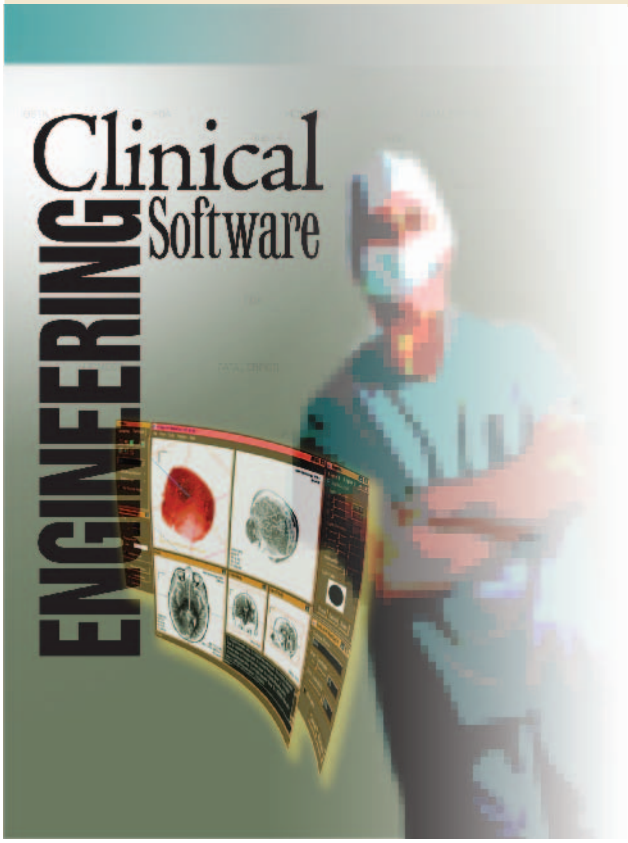

Software Engineering for Future Healthcare and Clinical Systems

Richard A. Schrenker
Massachusetts General Hospital

Systems and software engineering contribute not only to advancing and improving the delivery of healthcare but also to doing it more safely than has been the case in the past.

Turning "To err is human, but to really screw up, you need a computer" on its head, in 1999 the Institute of Medicine's *To Err Is Human: Building a Safer Health Care System* recommended[1] that healthcare professionals focusing on patient safety should increase their understanding of how information technology could be applied to deliver safer care. This recommendation was made as part of the approach to reducing errors in the delivery of care leading to the death of as many as 98,000 US citizens annually.

Much of the subsequent response to that challenge has focused on increasing the capabilities of enterprise hospital and clinical information systems—for example, implementing order-entry systems to check for drug allergies when writing prescriptions. But IT and patient care also come together at the bedside in the medical equipment and instrumentation systems used to deliver direct patient care—for example, smart infusion pumps that help ensure that the right dose of the right drug is administered to the right patient.

The articles in this special issue will touch on both types of systems, while focusing primarily on the application of software and systems engineering to software-based medical devices and device systems used at the bedside.

REVISITING THE PAST

There is no free lunch, of course. That software brings risks of its own to healthcare technology was not news in 1999. Six years before *To Err Is Human*, *Computer* published an evaluation of the Therac-25 accidents in which Nancy Leveson and Clark Turner provided what retrospectively may be seen as a "warning shot" regarding the impact of software on medical technology.[2] Under "Lessons Learned," they quoted a medical physicist:

> We have assumed … manufacturers have all kinds of safety design experience since they've been in the business a long time. We know that there are many safety codes, guides, and regulations to guide them and we have been reassured by the hitherto excellent record of these machines … Perhaps, though, we have been spoiled by this success.

The authors go on to note:

> If we assign software error as the cause of the Therac-25 accidents, we are forced to conclude that the only way to prevent such accidents in the future is to build perfect software that will never behave in an unexpected

Published by the IEEE Computer Society

> or undesired way under any circumstances (which is clearly impossible) or not to use software at all in these types of systems.

They also note that "Although using good software engineering practices will not prevent all software errors, it is certainly required as a minimum" and that "Safety is a quality of the system in which the software is used; it is not a quality of the software itself."

These warnings echo in the enterprise domain as well. In "Some Unintended Consequences of Information Technology in Health Care: The Nature of Patient Care Information System-Related Errors," Joan Ash and colleagues cite examples of PCIS failures that led to decreased safety, the opposite intent of its design.[3] They recommend that "developers and vendors should be clearer about the limitations of their technologies."

Clinical demands appear to point to the need for a less linear and more collaborative process among manufacturers, regulators, and medical equipment users.

That said, today's limitations may have been yesterday's advanced features. Given the increasing rate of change of technological innovation and its introduction into healthcare delivery, it is not surprising to find different vintages of similar systems simultaneously available in clinical practice. This in turn can lead to originally unanticipated user expectations being applied to older systems, potentially resulting in unintended consequences not only in their application but also for developers, as described in the "Healthcare Professionals' Perceptions of Medical Software and What to Do About It" sidebar by Phillip A. Laplante and coauthors.

A similar problem is reflected at the point of delivery of care, where an increasing number of medical devices are embedded systems with complexity and capabilities that exceed products from just a few years ago.

Providing care to any one patient is likely to require multiple devices, particularly for the more acutely ill. The instrumentation at an intensive care bedside will minimally include a physiologic monitoring system to acquire, process, communicate, display, and generate appropriate alarms for ECG, one or more blood pressure devices, and devices for monitoring oxygen saturation, cardiac output, respiration, and other key parameters. Other devices likely to be in use will include infusion pumps (smart or otherwise) and a ventilator. Equipment that can be brought in as needed includes dialysis systems and laboratory equipment such as automated blood and chemistry analyzers.

Some patients will need all of the above and perhaps more; others will present different needs. Assuring the readiness and availability of this equipment requires having a robust and reliable medical technology management system.

CHALLENGES AHEAD

Responding to the demands of the patient care environment requires (among other things) hospital medical equipment inventories that are not only well-stocked—we currently manage more than 18,000 devices for our approximately 1,000-bed hospital—but fairly dynamic as well. New equipment—and new makes, types, and models of equipment—is added continuously, often replacing outdated equipment, but sometimes providing new functionality. Consequent human factors as well as technical and user training issues require ongoing monitoring and attention.

None of this is terribly new, but the addition of software-based medical devices adds more wrinkles. Henry Petroski's[4] admonition is worth remembering:

> Any design change … can introduce new failure modes or bring into play latent failure modes. Thus it follows that any design change, no matter how seemingly benign or beneficial, must be analyzed with the objectives of the original design in mind.

Managing software versions, installing patches, or placing devices on shared network infrastructures are examples of activities that are already introducing new sets of problems to the clinical environment, including some that have yet to manifest themselves.

Manufacturers, regulators—for example, the FDA—and medical equipment users all have played roles in the evolution of medical technology management systems that have brought us to this point. Viewing the process from 30,000 feet, manufacturers develop a device; regulators approve it for sale; and users buy, use, and maintain it. But it is not clear whether this model will remain sustainable going forward, as clinical demands driving technological responses appear to point to the need for a less linear and more collaborative process among the involved parties. For example, currently there is little in the way of standards-based interoperability among medical devices, even in the presence of ongoing efforts like IEEE 11073, which date back to the early 1990s.

Why these efforts have yet to succeed is not fully clear even to those of us who have been involved. However, over the past few years, a movement has started to take shape that is characterized not only by increased collaboration, but also by users taking a more active role in establishing the vision for future systems and deriving the requirements to which manufacturers and regulators need to respond.

Active efforts following this model include the creation of the American College of Clinical Engineering-sponsored Domain for Patient Care Devices within the Integrating

the Healthcare Enterprise Initiative (IHE PCD),[5] in which the collaborators include clinicians, engineers, and informaticists from healthcare providers as well as federal regulatory staff, manufacturers, and standards experts. Another derives from work started in the Massachusetts General Hospital's Operating Room of the Future[6] and is described by Julian Goldman and coauthors in their "The Medical Device 'Plug-and-Play' (MD PnP) Interoperability Program " sidebar.

BROAD VISION, NATIONAL AGENDA

Much like the fable of the blind men describing an

Healthcare Professionals' Perceptions of Medical Software and What to Do About It

Phillip A. Laplante, Colin J. Neill, and Raghvinder Sangwan
Penn State University

A March 2005 article by Ross Koppel and colleagues in the *Journal of the American Medical Association* exemplifies a sequence of reports highly critical of various kinds of medical informatics systems.[1] In this case, a *computerized physician order entry system* deployed at the University of Pennsylvania Medical Center came under fire. The article concluded that, contrary to conventional belief, a CPOE system might actually increase the number of medication errors as compared to a manual, handwritten system.

As faculty working at a graduate center with the mission of advancing the profession of software engineering, we were aghast at the implications (accusations, really) of the study reported in this article—that the software engineering employed in the development of this system was deficient or delinquent and was therefore an indication that our discipline is itself lacking. Particularly disconcerting was how the mainstream media picked up the article and further promoted the notion that software engineers were failing the medical profession.

Physicians' Perceptions of Software Engineering Practices

The Eclipsys CPOE system scrutinized in this study apparently was deployed from 1997 until 2004. When we contacted them, Eclipsys employees confirmed that because it had screens that were "usually monochromatic with pre-Windows interfaces," this was probably an older-generation system.

In any case, the conclusions drawn in the *JAMA* article about CPOE system design can be stated as follows:

- Focus primarily on the organization of the work, not on technology.
- Aggressively examine the technology in use.
- Aggressively fix technology when it is shown to be counterproductive.
- Pursue errors' "second stories" and multiple causations to surmount barriers enhanced by episodic and incomplete error reporting.
- Plan for continuous revisions and quality improvement, recognizing that all changes generate new error risks.

These are logical recommendations to derive from the study of a legacy system clearly developed with outmoded methodologies and technologies. Unfortunately, the study's authors chose to impute these findings on every CPOE system and neglected to mention the aged nature of this particular application—a fact that also was not noted by any of the media outlets that further promulgated the study's assertions.

To the further discredit of the software engineering profession, this study focused on the perceptions of system users who were unlikely to accept blame for their own errors or acknowledge their own inadequacies with respect to using the system. When the option is to accept fault yourself or to blame your tools, which would you choose?

The State of the Art?

We accept that in the past the industry has been sullied by well-publicized disasters caused by poorly designed medical software systems, most notably the Therac-25 debacle.[2] But it is incumbent on the software engineering profession to both publicize the advances that have been made in the past decade and to actively apply those advances.

For example, because the Eclipsys CPOE system was designed more than a decade ago, its developers most likely employed a waterfall life-cycle model. That the system failed to match the user community's needs and workflow is, therefore, no surprise.[3]

Numerous advances in software engineering have thoroughly addressed the software deficiencies that the critics of these medical software systems discovered. These advances include the growth of the subdisciplines of requirements engineering (focused on the gathering, documentation, and analysis of user requirements), user-interface design (focused on the design and construction of intuitive and safe user interfaces), and usability engineering (focused on the study of ease of use and suitability for purpose).

In addition to an improved software engineering paradigm, others have identified the need for better embedded medical device user interfaces to reduce errors.[4] We believe that software engineers should focus more attention on the usability aspects of medical systems, whether they are embedded or not. The greatest risk in not doing so are the kinds of medical errors uncovered in studies that rightfully criticize the failure to adopt good software engineering practices. The less obvious risk, however, is that failure to address usability

elephant, our perception of the scope of the application of information technology to healthcare is largely influenced by where we encounter the system. Virtually all of us can relate to issues associated with medical data records management, making it easier to appreciate the Institute of Medicine's recommendations for enterprise-level and larger information systems. Although less visible to the public, visions are beginning to take shape that are also national in scope but more focused on technologies used in the direct provision of care.

In "High-Confidence Medical Device Software and Systems," Insup Lee and colleagues describe a national

issues degrades the overall confidence that medical professionals have in software solutions even when an appropriate software engineering process has been used for all other aspects of the system except the user interface.

Ironically, contemporary requirements engineering techniques include many of the investigatory techniques employed by physician-researchers who study medical systems. *Joint application development* employs focus groups of project stakeholders so that each involved party is represented in the requirements elicitation stage. *Use case analysis* employs user-scripted scenarios of interaction to ensure that the computer system enforces a workflow and mode of operation that reflects not only the policies and procedures that must be met, but also the ways of working that the users themselves favor. *Contextual inquiry and ethnography* involve user shadowing and observation so that the analysts and developers involved in constructing the information system have a sufficient understanding of the problem domain to ensure that the delivered application conforms with current practice where necessary and optimizes current practices where possible. Other techniques such as using *formal methods* address the issue of ensuring that the code is correct and behaves according to the rules that the healthcare professionals determine.

These innovations within the software engineering community have been developed, or have been more widely adopted, since the deployment of many of the medical systems that have come under criticism. These techniques ensure that the delivered system does all that the users want and need and that the correct checks and balances are in place so that human-machine interface flaws and information errors do not arise.

Finally, we observe that many clinical systems currently in use were created prior to the recent, dramatic changes in healthcare delivery. Integrated health networks with more complex workflows and a greater need for seamless movement of patient data on demand, anywhere within the network, have for the most part replaced free-standing hospitals, clinics, and group practices. Retrofitting yesterday's systems to meet today's needs can only result in a "solution" that falls short, as the *JAMA* study clearly demonstrates.

Software engineers have long known that extensive retrofitting causes software to age very rapidly. Considering what we do know about building complex software systems and in the light of these dramatic changes in the industry, it is unfortunate that the prevailing sentiment among healthcare professionals seems to be that legacy information systems, their developers, and their vendors are failing to meet the needs of physicians and hospitals.

Moving Forward

The message that should be delivered is that hospital administrators must push for modern computer systems rather than taking the cheap way out and trying to adapt outdated technology. Further, healthcare professionals have a role to play in the specification, validation, and deployment of complex software systems such as CPOE. The burden to deliver correct and usable applications isn't entirely on the software engineers and software vendors. Software engineering professionals must be proactive in educating healthcare professionals about the role they must play in building systems that are responsive to their needs and are reliable and safe.

As it turns out, several major medical systems developers—Eclipsys, Siemens Medical Solutions, and McKesson HBOC—are all located near our campus, and some of our best students hail from these companies. Therefore, we know that, whatever the state of affairs was 10 or 20 years ago, at least some representatives of the medical software community are now applying state-of-the-art software engineering techniques in the development of medical informatics systems—including design for usability.

As a profession, we must get the word out to healthcare providers that the state of affairs in software engineering has improved dramatically.

References

1. R. Koppel et al., "Role of Computerized Physician Order Entry Systems in Facilitating Medication Errors," *JAMA*, 9 Mar. 2005, pp. 1197-1202.
2. N.G. Leveson and C.S. Turner, "An Investigation of the Therac-25 Accidents," *Computer*, July 1993, pp. 18-41.
3. P.A. Laplante and C.J. Neill, "The Demise of the Waterfall Model Is Imminent and Other Urban Myths of Software Engineering," *ACM Queue*, Feb. 2004, pp. 10-15.
4. J. Ganssle, "First Do No Harm: A Cry for Help from a Hospital Systems Engineer"; www.embedded.com.

Phillip A. Laplante *is an associate professor of software engineering at Penn State University. Contact him at plaplante@gv.psu.edu.*

Colin J. Neill *is an associate professor of software engineering at Penn State University. Contact him at cjn6@psu.edu.*

Raghvinder Sangwan *is an assistant professor of information science at Penn State University. Contact him at rxs69@psu.edu.*

collaborative effort involving academics and professionals working together to identify and address the critical issues presented by the emergence of intelligent clinical technologies.

Vision meets reality

Moving from vision to product requires not only attention to good software engineering practices and awareness of the regulatory environment, but also a grounding in fundamental risk management principles. Steven R. Rakitin explores all three in "Coping with Defective Software in Medical Devices."

Reality meets New Age: How can we not use agile methods?

In "IGSTK: An Open Source Software Toolkit for Image-Guided Surgery," Kevin Gary and colleagues start with a description of the critical requirements posed by the needs of image-guided surgery that, when coupled with the resources available to his team, result in daunting development constraints. The authors describe the development and application of a mixture of classical and agile tools and methods in support of their clinical application.

Wireless changes everything

In many institutions, it once was easy to partition medical and nonmedical networked devices by installing them on physically distinct wired networks. That degree of control effectively came to an end with the introduction of wireless medical device networks. In "Ensuring Patient Safety in Wireless Medical Device Networks," Vijay Gehlot and Elliot B. Sloane provide an insightful view into the risks, details, and nuances

The Medical Device "Plug-and-Play" (MD PnP) Interoperability Program

Julian M. Goldman, Massachusetts General Hospital
Jennifer L. Jackson, Brigham and Women's Hospital
Susan F. Whitehead, Center for the Integration of Medicine & Innovative Technology
Tracy L. Rausch, Kaiser Permanente Mid-Atlantic States
Sandy Weininger, FDA Center for Devices and Radiological Health/OSEL/DESE

A patient undergoing gallbladder surgery is under general anesthesia during the procedure. To avoid image blurring while taking X-ray images during the surgery, it is necessary to switch off the ventilator that is breathing for the anesthetized patient. Turning the ventilator off, taking the X-ray, and turning the ventilator back on again are all manual processes. If the team of caregivers is distracted, it is possible that the ventilator might not be turned back on. Although very unlikely, this tragedy has occurred (www.apsf.org/resource_center/newsletter/2004/winter/03turn_on.htm).

This scenario is one of many involving ensembles of standalone medical devices in which each acts as a stand-alone device, with its only sources of information coming from the operator and sensors. If the X-ray machine and ventilator were context-aware and able to communicate with one another, synchronization of the X-ray exposure to the phase of ventilation could minimize the need to turn off the ventilator, substantially reducing the potential for the disaster described above. The same approach could improve image quality and decrease wasted images and unnecessary X-ray exposure when X-rays are taken in the intensive care unit.

But the state of the art with respect to medical device interoperability is reflected in a small number of proprietary products that provide some capabilities geared primarily at populating patient record systems and single-vendor "integrated" networked systems. Despite almost 20 years of attempting to define standards that enable medical device interoperability, little real progress has been made in terms of delivering solutions to the market, particularly for problems involving emergent, real-time patient care.

The absence of market-ready interoperability solutions has stalled the development of fully integrated electronic health records, smart alarms, real-time clinical decision support systems, and automated safety systems (with medical device interlocks). As a result, clinicians cannot easily use technology to enhance situational awareness or control devices in the clinical environment, and they must continue to rely instead on teamwork and a patchwork of systems to mitigate clinical hazards.

Inspired by successes such as the Operating Room of the Future (ORF) program at Massachusetts General Hospital (MGH) and driven as much by frustration at not being able to provide the "latent opportunities" of innovation in clinical care that we know modern technology could support, as well as by the rapidly changing economics and dynamics of the patient care environment, we in the user community are beginning to respond.

In our case, we have established a program to fully address medical device interoperability to support the development of connected, error-resistant medical device systems throughout the continuum of healthcare. Over the past two years, the Medical Device Plug-and-Play Interoperability Program (MD PnP; www.mdpnp.org), founded by the Center for the Integration of Medicine and Innovative Technology (CIMIT; www.cimit.org/orfuture.html) and MGH, has surveyed clinical groups representing leading surgeons, anesthesiologists, nurses, and clinical engineers to acquire the information needed to derive use cases and drive requirements definitions. We are converting selected clinical use cases into pro-

of placing such a system into service. They also examine the subtleties driving the need for hospital-based clinical engineering involvement in system verification and validation.

Everything Changes FDA

Cognizant of issues like the ones that challenged Gary's team, regulators are faced with determining how to respond to issues that emerge with the rapid evolution of software-based medical devices. In "A Formal Methods Approach to Medical Device Review," Raoul Jetley and colleagues describe a set of formal approaches for application test and validation during the premarket approval process or when doing a forensic analysis of problems that occur after a device has been delivered to the market.

Indeed, to err *is* human. But it does not follow that harm cannot be prevented. Systems and software engineering contribute not only to advancing and improving the delivery of care but also to doing it more safely than has been the case in the past. Doing so appears likely to require greater collaboration between manufacturers, regulators, and users in the future. And it *is* happening.

But more needs to be done, and soon. While the work that the IHE and MD PnP are doing makes many of us hopeful that interoperable medical device systems will soon begin to be realized, hard questions need to be asked, such as, Why did IEEE 11073 move so slowly? What more needs to happen for its vision to be realized in the market? What could we do differently to avoid similar inertia when tackling future systems and software engineering problems?

totype models to be implemented in our MD PnP Lab with commercially available medical devices.

The MD PnP Lab is implementing the gallbladder imaging scenario as the first clinical use case around which to begin to define, select, or develop the processes, tools, framework, and components with which to construct the needed system. Throughout the MD PnP program, it is our intent to reuse and leverage existing work wherever possible, and we will support the use of currently existing consensus standards if they can contribute to the implementation of these clinical use cases. We are also acutely aware that other significant challenges, such as data security, liability and regulatory issues, network performance monitoring, and interoperability with the broader healthcare enterprise must also be addressed.

The impetus for the MD PnP program relies on both the visionary and real foundation provided by the ORF along with the collaboration and extended vision of its members. The ORF, a program of CIMIT and MGH, is a fully functioning OR suite in MGH. The ORF serves as a "living laboratory" for clinicians, engineers, technicians, architects, and administrators to study the impact of process change, technology, and teamwork on safety and productivity. The ORF also serves as a protected environment and aggregation point to develop and safely validate and test those ideas, including MD PnP, that are envisioned as necessary to lay the foundation for safety and efficiency innovations in perioperative healthcare.

The MD PnP "geographically dispersed team" includes not only members of Partners HealthCare clinical, information services, and clinical engineering staff but also colleagues from other integrated healthcare delivery networks (IHDNs) such as Kaiser Permanente; marketing and engineering staff from medical device manufacturers; FDA and NIST staff involved in the regulation and testing of software-based medical devices; marketing and engineering staff representing manufacturers of information technology-based hardware and software; and members of academic and research communities.

Many MD PnP members are also involved in efforts like IEEE 11073 and the Integrating the Healthcare Enterprise Initiative Domain for Patient Care Devices (IHE PCD) of the Healthcare Information and Management Systems Society. Opportunities to work together come both at the various standards meetings as well as on shared projects and problems.

The continued lack of automated safety systems, smart alarms, closed-loop control, and decision support systems at the patient bedside, coupled with the tacit acceptance of resultant risks that thereby accrue, is unconscionable in the presence of readily available technology that is applied to similar goals seemingly everywhere but healthcare. We in the MD PnP program are intent on addressing it.

Julian M. Goldman, MD*, is the director of the Medical Device "Plug-and-Play" Interoperability Program in the Departments of Anesthesia and Biomedical Engineering at Massachusetts General Hospital and the Center for the Integration of Medicine and Innovative Technology. Contact him at jmgoldman@partners.org.*

Jennifer L. Jackson *is the assistant director of biomedical engineering at the Brigham and Women's Hospital, Boston, Mass. Contact her at jljackson@partners.org.*

Susan F. Whitehead *is the Medical Device "Plug-and-Play" Interoperability Program project manager in the Center for the Integration of Medicine & Innovative Technology, Cambridge, Mass. Contact her at swhitehead@partners.org.*

Tracy L. Rausch *is a clinical systems engineer at Kaiser Permanente Mid-Atlantic States, Rockville, Md. Contact her at tracy.rausch@kp.org.*

Sandy Weininger *is a senior regulatory engineer, FDA Center for Devices and Radiological Health/OSEL/DESE, Rockville, Md. Contact him at sandy.weininger@fda.hhs.gov.*

The involvement of experts from outside the medical technology domain could prove valuable. For example, researchers might be better positioned to help us more rigorously address emerging issues such as whether medical device networks should merge with hospital or other clinical information systems networks. And, jumping even further outside the box, consideration needs to be given to how healthcare-based engineers, caregivers, and technologists can become even more engaged in technology definition, development, and design decisions and activities, to, for example, address human factors issues that will likely increase with device and system complexity.

Medicine remains fundamentally reactive; we wonder how it can be otherwise. A person can do everything possible to remain healthy, but sooner or later, if an accident doesn't strike, illness will. When this occurs, clinicians attending to the patient remain driven by the basic principle, "First, do no harm," and they expect that the tools they use will not permit their violating that principle.

To address patient safety in the face of the perturbations that arise from human error as well as other sources, proactive systems and software engineering attention must increasingly focus on continuously creating robust, reliable, and dependable applications and an infrastructure focused on addressing needs at the point of delivery of care. ■

Acknowledgments

I thank Bob Colwell for encouraging me to serve as guest editor for this special issue and Scott Hamilton for patiently shepherding me through the process. I received appreciated feedback on this editorial from my colleagues Mike Cusack, Luis Melendez, and Jason Davis. The work and guidance of my colleague, patient-safety expert Jeff Cooper, has inspired my interest in relating software, systems, and clinical engineering around safety. Last and most, I want to jump outside the engineering box to thank my father, Dick Schrenker, for inspiring me to put first things first in whatever I do. Hence, the theme not just of this editorial but the application of technology wherever it touches medicine: When it comes to what engineering brings to healthcare, safety comes first.

References

1. L.T. Kohn, J.M. Corrigan, and M.S. Donaldson, eds., *To Err Is Human: Building a Safer Health System*, Nat'l Academies Press, 2000, p. 80; http://darwin.nap.edu/books/0309068371/html.
2. N. Leveson and C. Turner, "An Investigation of the Therac-25 Accidents," *Computer*, July 1993, pp. 18-41.
3. J.S. Ash, M. Berg, and E. Coiera, "Some Unintended Consequences of Information Technology in Health Care: The Nature of Patient Care Information System-Related Errors, *J. Am. Medical Information Assoc.*, vol. 11, 2004, pp. 104-112.
4. H. Petroski, *Design Paradigms: Cases of Error and Judgment in Engineering*, Cambridge Univ. Press, 1994, p. 57.
5. HIMSS News Item, "ACCE and IHE Launch Patient Care Devices Domain," www.himss.org/ASP/topics_News_item.asp?cid=65152&tid=2.
6. J.M. Goldman et al., "Plug and Play in the Operating Room of the Future," *Biomedical Instrumentation and Technology*, May/June 2005, pp. 194-199.

***Richard A. Schrenker** manages the Systems Engineering Group in the Department of Biomedical Engineering at Massachusetts General Hospital. He received an MS in electrical engineering from Johns Hopkins University. Schrenker is a member of the IEEE EMBS and Computer Societies, the American College of Clinical Engineering, the ACM, and the Association for the Advancement of Medical Instrumentation, and is a cofounder and Life Member of the Baltimore Medical Engineers and Technicians Society. He is active in IEEE 11073, IHE PCD, and MD PnP development efforts. Contact him at raschrenker@partners.org.*