A Digital Matter of Life and Death

Concerns about potentially life-threatening software errors bring government regulation of computer-controlled medical devices

By IVARS PETERSON

The radiation-therapy machine, a Therac 25 linear accelerator, was designed to send a penetrating X-ray or electron beam deep into a cancer patient's body to destroy embedded tumors without injuring skin tissue. But in three separate instances in 1985 and 1986, the machine failed. Instead of delivering a safe level of radiation, the Therac 25 administered a dose that was more than 100 times larger than the typical treatment dose. Two patients died and a third was severely burned.

The malfunction was caused by an error in the computer program controlling the machine. It was a subtle error that no one had picked up during the extensive testing the machine had undergone. The error surfaced only when a technician happened to use a specific, unusual combination of keystrokes to instruct the machine.

The Therac incidents and other cases of medical device failures caused by computer errors have focused attention on the increasingly important role played by computers in medical applications. Computers or machines with built-in microprocessors perform functions that range from keeping track of patients to diagnosing ailments and providing treatments.

"The impact of computers on medical care and the medical community is the most significant factor that we have to face," says Frank E. Samuel Jr., president of the Health Industry Manufacturers Association (HIMA), based in Washington, D.C. "Health care will change more dramatically in the next 10 years because of software-driven products than for any other single cause." Samuel made his remarks at a recent HIMA-sponsored conference on the regulation of medical software.

At the same time, reports of medical devices with computer-related problems are appearing more and more frequently. In 1985, the Food and Drug Administration (FDA) reported that recalls of medical devices because of computer faults had roughly doubled over the previous five years. Since then, the number of such complaints has risen further.

The problems range across a wide spectrum of computer-based medical devices. A system designed for monitoring several patients at once was recalled because it kept mixing up the patients. A programmable heart pacemaker suddenly "froze" while it was being adjusted by a doctor. A device for dispensing insulin delivered the drug at an inappropriate rate. An expert system gave the wrong diagnosis, resulting in a patient receiving a drug overdose. An ultrasound scanner sometimes underestimated fetal weight.

"No one can deny that allowing computers to perform some of the functions normally carried out by trained and licensed medical professionals raises questions concerning the personal health and safety of citizens," Michael Gemignani of the University of Maine in Orono comments in Abacus (Vol. 5, No. 1). "But even if we agree something more needs to be done to protect society in the face of these technological innovations, we are still left with the question: What should be done and by whom?"

The FDA, in its mandated role as guardian of public health and safety, is now preparing to regulate the software component of medical devices. The agency's effort has already raised questions about what kinds of products, software and information systems should be regulated.

Last fall, the FDA published a draft policy for the regulation of computer products marked for medical use. In that policy, the concept of "competent human intervention" sets the dividing line between what is and is not regulated. In other words, the computer product in question is subject to regulation if a qualified doctor or nurse cannot effectively intervene to override the machine's actions. Devices such as software-driven cancer therapy machines, programmable heart pacemakers and automatic drug dispensers clearly fall into that category.

On the other hand, the FDA states that it would not regulate computer products that simply store, retrieve and disseminate information analogous to that traditionally provided by textbooks and journals. In addition, the agency's regulations would not apply to computer products used only for communications, general accounting or teaching.

For example, a physician may use a computer program known as an expert system to help make a diagnosis. Because the expert system does not directly drive another medical device that, say, could dispense a drug when needed, and because the doctor can make an independent judgment, such an expert system would be exempt from FDA rules governing medical devices.

However, the greatest advantage of software — its flexibility — is also, from a regulatory point of view, one of its biggest problems. Computer programs are easy to change and can be used in many different ways. If corrections are made or new features added, how much scrutiny should the modified version of a previously approved computer product undergo? That question is still unresolved.

Furthermore, it's sometimes hard to make a clear distinction between programs that perform a "library" function and those that can be classified as being part of a medical device. A case in point is the patient medical record, traditionally a filing folder containing various sheets of paper listing treatments, medical observations and other pieces of information vital for the patient's proper care.

Many hospitals are now moving toward medical records that are stored on a computer. The difficulty arises when such information systems are connected directly to machines that, for example, record patient blood pressure and heart rate. If a nurse takes down the data and then enters the figures into a computer, the information system software would not be subject to FDA rules. But if the machine sends the data directly to the computer, then the information system is considered by the FDA to be an "accessory" to a medical device and subject to the same level of regulation as the machine itself.

Information system vendors disagree with the FDA's position. They argue that the FDA does not presently have rules governing the quality and content of paper medical records. There's no reason for the FDA to start regulating such records, they say, just because the records happen to be in a computer's memory rather than on paper. In fact, using a computer-based system would dramatically reduce the incidence of errors in patient records, the vendors claim. The benefits of improved record keeping would clearly outweigh the need for burdensome regulation.

The FDA's James S. Benson concedes

that "regulation is not the automatic solution to problems in hospitals and elsewhere." Nevertheless, the agency must comply with a 1976 law that contains a broad definition of what constitutes a medical device. Interpreted in its broadest sense, the definition encompasses practically everything used in a hospital, from X-ray machines to pencils.

FDA officials say they recognize the difficulties involved in regulating medical software. "The agency fully appreciates the revolution occurring in medicine with the introduction of computers and microprocessors," says Frank E. Young, FDA commissioner. "We're taking a reasoned, structured approach with a minimum of oversight. We have tried to give general guidelines. The policy has been deliberately made flexible."

The flexibility allows the FDA to consider applications for approval on a case-by-case basis. That limits the "chilling fear of undue regulation," says Young. Furthermore, as technologies change and experience with computers in medical applications grows, decisions on how much regulation is needed may also change.

To many manufacturers and users of medical products, the FDA's idea of flexibility leaves too much uncertainty and opens up the possibility of increased regulation in the future. "The FDA casts too wide a net," says Edward M. Basile of King & Spalding, a law firm in Washington, D.C. "Their basic assumption is that everything should be regulated."

"There's no disagreement about the extremes," says Harold M. Schoolman of the National Library of Medicine in Bethesda, Md. "The question is how and where to draw the line between the extremes." The important issue, he says, is maintaining a balance between appropriate safeguards and incentives for innovation.

Even in situations where it's clear that certain software ought to be reviewed, the FDA faces the additional difficulty of how to go about verifying that a particular computer program does what it's supposed to do – nothing more, nothing less. As experience with software for other applications has shown, the task of checking software quality can be overwhelming (SN: 9/13/86, p.171).

A few years ago, when most medical devices did not contain computers, it was relatively easy to foresee all possible inputs and to check the consequences of each one, says James Howard of General Electric's Medical Systems Group in Milwaukee, Wis. With computers, the number of possible paths is greatly increased. "It's more important than ever to build safe products that perform as required," he says. But because a detailed analysis takes so long, it often can't be done. "This is a major concern to both manufacturers and the FDA," says Howard.

The FDA defines software as a "set of instructions that enables a computing machine to control, monitor or otherwise interact with a medical device." The proposed regulations require a software developer to show that the algorithm, or mathematical recipe, used in the computer program is appropriate and has been implemented correctly in the software. The FDA also requires assurance that any software failure would not injure a patient.

"It's more important than ever to build safe products that perform as required."

How that assurance can be provided is still unclear. Techniques for evaluating software safety are relatively new. Who does the checking, how much evidence is enough and whether the FDA can perform an independent check are also unresolved issues. Furthermore, software developers are wary of submitting complete listings of the instructions in their computer programs because competitors may get a look at this "source code" by making a request to the FDA under the Freedom of Information Act.

The trouble with the FDA approach, says Howard, is that it doesn't consider under what conditions software is used. Instead, the FDA ought to focus on the idea that not all computer errors are equally serious. Using a kind of hazard analysis to focus on situations that could lead to life-threatening computer failures would be one way to eliminate the most serious potential faults and to shorten testing times.

Software developers also need to improve the methods they use for constructing computer programs. We need to "industrialize" software development so that programs are written in a consistent way, says James Dobbins of Verilog USA, Inc., in Alexandria, Va. Too often, programmers include a description of what each part of a program does only as an afterthought. They rarely go back to clean up or polish a program to make it more understandable.

Software development can be standardized and automated, says Dobbins. "The tools are there to industrialize the whole process. You just have to go find them."

Programmers, on the other hand, complain that they're in a no-win situation. Software is continually modified as it evolves, often to meet demands for new features to make the product more competitive. In the rush to market, when delays can put a company at a competitive disadvantage, software testing often loses out. Delays in completing a software package are balanced against the possibility of failing to root out potentially embarrassing errors.

This is the kind of situation that can lead to lawsuits, says Vincent Brannigan, an attorney in Adelphi, Md. Software is clearly a product, he says. If it's defective and injures a consumer, then the manufacturer is liable.

Among the faults Brannigan lists is the tendency of software and computer companies to promise more than they can fulfill and to cut costs by doing less testing. This is the only field, he says, where the customer is expected to pay for finishing a product through the purchase of periodic updates and corrections to the software.

"Disclaimers don't mean anything," Brannigan says. "The product should have been right in the first place." That means paying much more attention to how software is written and tested. "The software must look as shiny and clean as the rest of the machine," he says.

So far, software developers have generally escaped damaging lawsuits and settlements, but that may change. To many medical-device producers, the threat of litigation may be even more effective than proposed FDA regulations for assuring the quality of products.

Even finding out what went wrong is a time-consuming process. The FDA and other groups are still investigating aspects of why the Therac 25, manufactured by Atomic Energy of Canada Ltd. in Kanata, Ontario, failed. What's evident is that the problem could probably have been avoided if an appropriate safety analysis had been done.

The Therac 25 delivers two forms of radiation: either a high-energy electron beam or, when a metal target intercepts the electron beam, a lower-energy X-ray beam. It turns out that when a nimble, experienced technician punches in a particular sequence of commands faster than the programmers had anticipated, the metal target fails to swing into place.

A safety analysis would have identified the missing target as a potentially dangerous situation. The machine could have been programmed so that it couldn't operate if the target, as confirmed by a sensor, were not in place.

Perhaps such a complex, computer-driven machine wasn't even necessary. By sacrificing a little convenience and flexibility, a machine with a simple on-off switch and a timer could probably have done the same job – with a much smaller chance of failure. □