Title: When code can kill or cure
Source: *The Economist (US)*. 403.8787 (June 2, 2012): p22(US).
Document Type: Article

Full Text:

Medical technology: Applying the "open source" model to the design of medical devices promises to increase safety and spur innovation

SMART pumps deliver drugs perfectly dosed for individual patients. Easy-to-use defibrillators can bring heart-attack victims back from the brink of death. Pacemakers and artificial hearts keep people alive by ensuring that blood is pumped smoothly around their bodies. Medical devices are a wonder of the modern age.

As these devices have become more capable, however, they have also become more complex. More than half the medical devices sold in America (the world's largest health-care market) rely on software, and often lots of it. The software in a pacemaker may require over 80,000 lines of code, a drug-infusion pump 170,000 lines and an MRI (magnetic-resonance imaging) scanner more than 7m lines.

This growing reliance on software causes problems that are familiar to anyone who has ever used a computer: bugs, crashes and vulnerability to digital attacks. Researchers at the University of Patras in Greece found that one in three of all software-based medical devices sold in America between 1999 and 2005 had been recalled for software failures. Kevin Fu, a computer science professor at the University of Massachusetts, calculates that such recalls have affected over 1.5m individual devices since 2002. In April researchers at McAfee, a computer-security firm, said they had found a way to get an implanted insulin pump to deliver 45 days' worth of insulin in one go. And in 2008 Dr Fu and his colleagues published a paper detailing the remote, wireless reprogramming of an implantable defibrillator.

When software in a medical device malfunctions, the consequences can be far more serious than just having to reboot your PC. During the 1980s a bug in the software of Therac-25 radiotherapy machines caused massive overdoses of radiation to be delivered to several patients, killing at least five. America's Food and Drug Administration (FDA) has linked problems with drug-infusion pumps to nearly 20,000 serious injuries and over 700 deaths between 2005 and 2009. Software errors were the most frequently cited problem. If buggy code causes a pump to interpret a single keystroke multiple times, for example, it could deliver an overdose.

In addition to accidental malfunctions, wireless and networked medical devices are also vulnerable to attacks by malicious hackers. In the 2008 paper Dr Fu and his colleagues showed how an implantable cardioverter defibrillator could be remotely reprogrammed either to withhold therapy when it is needed or to deliver unnecessary shocks. Dr Fu says that when it comes to testing their software, device manufacturers lack the safety culture found in other high-risk industries such as avionics, and are failing to keep up with the latest advances in software engineering. Insup Lee, professor of computer science at the University of Pennsylvania, agrees. "Many manufacturers do not have the expertise or the willingness to utilise new tools being developed in computer science," he says.

Just how bad it is, though, no one knows for sure. The software used in the vast majority of medical devices is closed and proprietary. This prevents commercial rivals from copying each other's code or checking for potential patent infringements. It also makes it harder for security researchers to expose flaws. The FDA, which could demand to see the source code for every device it approves, does not routinely do so, but instead leaves it to manufacturers to validate their own software. Two years ago it offered free "static analysis" software testing to infusion-pump manufacturers in the hope of reducing injuries and deaths. But no manufacturer has yet taken the FDA up on its offer.

Open to scrutiny

Frustrated by the lack of co-operation from manufacturers, some academics now want to reinvent the medical-device industry from the ground up, using open-source techniques. In open-source systems, the source code is freely shared and can be viewed and modified by anyone who wants to see how it works or build an improved version of it. Exposing a design to many hands and eyes, the theory goes, results in safer products. This seems to be the case for desktop software, where bugs and security flaws in open-source applications are typically fixed much more quickly than those in commercial programs.

The Generic Infusion Pump project, a joint effort between the University of Pennsylvania and the FDA, is taking these troublesome devices back to basics. The researchers began not by building a device or writing code but by imagining everything that could possibly go wrong with a drug-infusion pump. Manufacturers were asked to help, and several did so, including vTitan, a start-up based in America and India. "For a new manufacturer, it's a great head start," says Peri Kasthuri, vTitan's co-founder. By working together on an open-source platform, manufacturers can build safer products for everyone, while still retaining the ability to add extra features to differentiate themselves from their rivals.

Mathematical models of existing and new pump designs were tested against the possible risks, and the best-performing models were used to generate code, which was installed on a second-hand infusion pump bought online for $20. "My dream", says Dave Arney, a researcher on the project, "is that a hospital will eventually be able to print out an infusion pump using a rapid prototyping machine, download open-source software to it and have a device running within hours."

Equally ambitious is the Open Source Medical Device initiative at the University of Wisconsin-Madison. Two medical physicists, Rock Mackie and Surendra Prajapati, are designing a machine to combine radiotherapy with high resolution computed tomography (CT) and positron-emission tomography (PET) scanning. Their aim is to supply, at zero cost, everything necessary to build the device from scratch, including hardware specifications, source code, assembly instructions, suggested parts--and even recommendations on where to buy them and how much to pay. The machine should cost about a quarter as much as a commercial scanner, making it attractive in the developing world, says Dr Prajapati. "Existing devices are expensive both to buy and maintain," he says, whereas the open-source model is more sustainable. "If you can build it yourself, you can fix it yourself when something breaks."

Open-source devices are also to be found literally at the cutting edge of medical science. An open-source surgical robot called Raven, designed at the University of Washington in Seattle, provides an affordable platform for researchers around the world to experiment with new techniques and technologies for robotic surgery.

All these open-source systems address very different problems in medical science, but they have one thing in common: all are currently prohibited for use on live human patients. To be used in a clinical setting, open-source devices must first undergo the same expensive and lengthy FDA approval processes as any other medical device. FDA regulations do not yet require software to be analysed for bugs, but they do insist on a rigorous paper trail detailing its development. This is not always a good fit with the collaborative and often informal nature of open-source coding.

The high cost of navigating the regulatory regime has forced some not-for-profit, open-source projects to alter their business models. "In the 1990s we developed an excellent radiation-therapy treatment-planning system and tried to give it away to other clinics," says Dr Mackie. "But when we were told by the FDA that we should get our software approved, the hospital wasn't willing to fund it." He formed a spin-off firm specifically to get FDA approval. It took four years and cost millions of dollars. The software was subsequently sold as a traditional, closed-source product.

Others are skirting America's regulatory system altogether. The Raven surgical robot is intended for research use on animals and cadavers, while the Open Source Medical Device scanner will be large enough only to accommodate rats and rabbits. However, says Dr Mackie, there is nothing to stop anyone taking the design and putting it through a regulatory process in another country. "It may even happen that the device will be used on humans in parts of the world where strict regulation does not exist," he says. "We would hope that if it is used in such a way, it will be well enough designed not to hurt anybody."

Changing the rules

The FDA is gradually embracing openness. The Medical Device Plug-and-Play Interoperability Program, a $10m initiative funded by the National Institutes of Health with the support of the FDA, is working to set open standards for interconnecting devices from different manufacturers. This would mean that, say, a blood-pressure cuff could instruct a drug pump to stop delivering medication if it sensed that a patient was suffering an adverse reaction.

More intriguing still is the Medical Device Co-ordination Framework being developed by John Hatcliff at Kansas State University. Its aim is to build an open-source hardware platform including elements common to many medical devices, such as displays, buttons, processors and network interfaces, and the software to run them. By connecting different sensors or actuators, this generic core could then be made into dozens of different medical devices, with the relevant functions programmed as downloadable "apps".

Eventually, medical devices might evolve into collections of specialised (and possibly proprietary) accessories, with the primary computing and safety features managed by an open-source hub. The FDA is working with Dr Hatcliff to develop processes for creating and validating safety-critical medical apps.

In the meantime, there are moves afoot to improve the overall security and reliability of software in medical devices. America's National Institute of Standards and Technology has just recommended that a single agency, probably the FDA, should be responsible for approving and tracking cybersecurity in medical devices, and the FDA is re-evaluating its ability to cope with the growing use of software.

Such changes cannot happen too soon. "When a plane falls out of the sky, people notice," says Dr Fu. "But when one or two people are hurt by a medical device, or even if hundreds are hurt in different parts of the country, nobody notices." With more complex devices, more active hackers and more inquisitive patients, opening up the hidden heart of medical technology makes a great deal of sense.

"By working together on an open-source platform, manufacturers can make safer products"

Source Citation (MLA 7th Edition)
"When code can kill or cure." *The Economist* [US] 2 June 2012: 22(US). *Academic OneFile*. Web. 26 Aug. 2012.

Document URL
http://go.galegroup.com.proxy.lib.duke.edu/ps/i.do?id=GALE%7CA291458916&v=2.1&u=duke_perkins&it=r&p=AONE&sw=w

Gale Document Number: GALE|A291458916

Top of page