

Not So Fast! The Dark Side of Computers in Health Care

Steven B Nelson MSc RRT FAARC

Introduction
Rights
Security
Reliability
Regulations
Support
Acceptance
Privacy
Summary

There are now computers in numerous health care devices, from thermometers to ventilators, and there are pitfalls to avoid in our increasing dependence on computers. To be useful, information must be delivered in the right context. Computer systems must be protected from worms, viruses, and other harmful code, and they must prevent unauthorized access to data. The source of all underlying decision algorithms must be known and appropriate for the population being served. And there must be contingency plans to mitigate losses caused by system unavailability. *Key words: computers, computer security, medical informatics, software, data security, data protection, patient data privacy, information management, medical errors, medication errors, computer viruses, computer hackers, disaster planning.* [Respir Care 2004;49(5):525–530. © 2004 Daedalus Enterprises]

Introduction

This conference has so far extolled the virtues of computerizing nearly every application imaginable in respiratory care. Though that is a commendable goal, there is a “dark side” to computers that extends beyond the sensa-

tional headlines on the nightly news warning of the latest virus or identity theft scheme.

The goal of this report is not to support a Luddite mentality regarding technology but to warn of pitfalls that may be encountered before and after the decision to use a computer as a solution. The Luddites were originally a group of displaced textile workers. In the early 1800s steam- and water-powered weaving and spinning frames were being installed in an area of Nottingham, England. By 1811 the workers had banded together under “General” Ned Ludd and formed the “Army of Redressers.”¹ Their goal was to destroy the frames and regain their jobs. By 1812 over 800 frames had been destroyed. The movement was crushed after a defeat at Lancashire and the declaration that destruction of a frame was a capital offense.

In the mid-1990s various neoLuddites, including Ted Kaczynski (the “Unabomber”) warned that computers were evil and exclusionary. Ironically, they used the very tools they derided to write and spread their message. Though Kaczynski proved to be a murderous madman, he made

Steven B Nelson MSc RRT FAARC is affiliated with Pulmonary Industrial Testing Associates, Overland Park, Kansas.

Steven B Nelson MSc RRT FAARC presented a version of this report at the 33rd RESPIRATORY CARE Journal Conference, Computers in Respiratory Care, held October 3–5, 2003, in Banff, Alberta, Canada.

Steven B Nelson MSc RRT FAARC is a technical consultant for Sun Microsystems, Santa Clara, California. The opinions in this report are solely those of the author and do not represent the views of Sun Microsystems.

Correspondence: Steven B Nelson MSc RRT FAARC, Pulmonary Industrial Testing Associates, 8314 W 128th Street, Overland Park KS 66213. E-mail: sbn_kc@mac.com.

several cogent points regarding computers and decision-making; he warned of a future where (1) decisions will become so complex that humans won't be able to process all the data, (2) decisions will be made by a small group acting as a "benevolent parent," or (3) decisions will be taken from humans by accident.² Those warnings are pertinent to the growing use of computers in health care. This report proceeds on the premise that computer technology should not be avoided but should be considered no more than a tool. I will provide information and opinion on how we can wisely and safely use that tool.

A principle to be aware of is the "rule of unintended consequences," which is illustrated by the use of DDT to control malaria-carrying mosquitoes. The mosquitoes developed resistance to DDT, and the DDT concentrated in the food chain and resulted in weakening of bird eggshells, causing a rapid decline in birds of prey and an increase in rodents. Often people are tempted into what appears to be a quick solution to a problem, and they implement it without proper analysis that could avoid unexpected results and cascading consequences.³

There are 7 subjects to address regarding computers and information management: rights, security, reliability, regulations, support, acceptance, and privacy.

Rights

Pharmacists speak of 4 "rights": the right medication, in the right dosage, by the right delivery method, to the right patient. Health care information has a similar set of "rights": the right information, in the right time frame, delivered in the right format, to the right user.

The right type of information is derived from a hierarchy, such as the one in Table 1. Data by themselves are not generally useful. They require a context to become information. Knowledge of the subject matter is needed to determine whether the data are normal or valid. Understanding allows one to create new knowledge to assess the patient. Wisdom is the integration of knowledge from multiple sources and the ability to apply judgment and respond. Many computerized systems are unable to rise above the knowledge level, primarily because of a lack of integrated information. The following example and Table 1 show the hierarchy. Say a re-

spiratory therapist (RT) is given the value "42." By itself that piece of data is useless, because it could be the value of any of several physiologic variables, including P_{aO_2} , P_{aCO_2} , or hematocrit. Knowledge that it is a P_{aCO_2} value would lead the RT to the conclusion that the value is in the normal range. Additional information that the patient is on a ventilator, has acute lung injury, and is supposed to be maintained with permissive hypercapnia might lead to the conclusion that the minute ventilation was too high. As the Mexican novelist Carlos Fuentes said, "The greatest crisis facing modern civilization is going to be how to transform information into structured knowledge."

Information has a time value. A normal result from a screening spirometry of a patient in an extended care facility may be delivered after many hours. Normal spirometry in a dyspneic emergency-room patient is generally reported within minutes. The findings might be the same, but in the emergency room the information might cause immediate changes in treatment.

Information's format should be appropriate for the device that receives it. It makes little sense to send more than a few seconds of video to a mobile computer such as a personal digital assistant, because those devices do not have large enough displays, enough memory, or fast enough data-transfer, though with engineering improvements that may change.

Information should be delivered to the right user. There are many classes of information in a hospital and many people who provide care for a hospitalized patient. Only the *required* information should be made available. For example, it is probably not necessary for a billing clerk to see an RT's comments about colorful, thick secretions, although a medical records clerk may need that information for diagnostic coding purposes.

Security

Computer security depends on software elements, physical elements, and human elements. Software security involves preventing the running of inappropriate code (eg, worms, viruses, or Trojan horses). A worm is a self-replicating, self-propagating program. Robert Morris at Cornell University wrote the first widespread computer worm in 1988. It exploited a weakness in an e-mail program used by nearly all of the 56,000 computers connected to the Internet at the time. In a matter of hours it had slowed communications to a crawl. Morris soon realized the extent of the damage and tried to send out a message about how to stop it, but many of the systems had already been disconnected from the Internet. It took about 5 days to recover and bring the systems back online. In September 2003 a worm called Sobig.F exploited a weakness in a widely used e-mail program, and it spread to about 150,000,000 computers in a matter of days. It created havoc on the Internet, primarily by the vast number of e-mail messages it was generating. Internet service provider AOL

Table 1. Information Hierarchy

Data	42
Information	FEV ₁ /FVC, P_{aO_2} , P_{aCO_2} , hematocrit
Knowledge	COPD, hypoxia, normocapnea
Understanding	Change bronchodilator, increase $F_{I_{O_2}}$
Wisdom	Judgment of correctness

FEV₁ = forced expiratory volume in the first second; FVC = forced vital capacity; $F_{I_{O_2}}$ = fraction of inspired oxygen.

scans incoming e-mail for viruses, and at the peak of the outbreak they found 23,000,000 copies of the worm during 1 day. Recently worms have been responsible for computer outages in hospitals, government offices,⁴ airlines,⁵ and nuclear facilities.⁶ Even after 15 years of experience that well-known computer exploitation method is still effective.

A computer virus differs from a computer worm in that a virus requires a user action to activate it. Most viruses, such as Melissa and Blaster, take advantage of functions in commonly used desktop-computer applications. A user is enticed to open a file or an e-mail message that appears to be from an acquaintance. Opening the file or message runs a program that can alter files and/or send more e-mail messages to replicate the virus. Viruses are simple to write, using a common scripting language found on nearly every desktop computer. Numerous Internet sources show the basics of writing viruses, and all that is required after writing the virus is to attach it to an e-mail message and send it.

Trojan horses are small programs hidden in larger programs. They can open a "back door" in a computer and thus give access to unauthorized users. One type of Trojan horse is called "spyware," which resides within a program installed by a user and sends information about the user's operating system, memory, hardware, and/or software to the company that wrote the spyware. Spyware can be used to monitor for illegal distribution or for marketing. It may also do nothing more than connect to a company Web site to check for updates. The most prevalent example of spyware is in Web browsers. By setting the Web browser's "cookie" authorization to "prompt" (rather than "accept" or "block"), you can see how many cookies a Web site is attempting to place on your computer and surmise how much information would be passed. (Cookies are small files used to track Web page accesses or transactions.)

Software security measures are simple to implement. Security steps include:

- Do not open attachments in e-mail messages. Attachments are the most common means for spreading viruses.
- Regularly install security patches, which should be obtained only from a known source.
- Install antivirus software in all the computers in your system, including computers that can connect from the outside through a dial-in or virtual private network. The virus scan function should be set to start automatically on a regular basis.
- Keep the antivirus software up to date. Antivirus companies quickly respond to new threats and issue upgrades as viruses are found. The false sense of security offered by outdated antivirus software is probably more dangerous than not having any installed at all.

Physical security concerns access to a computer or a network connection. A computer should be secured to an immovable object to prevent theft. The computer's case should be locked, if possible, to prevent removal of disks or other components. Disks can easily be removed from an unsecured case and the data read on another computer.

No computer should be connected directly between the Internet and an internal network at the same time, because that may allow unauthorized access to restricted information if the system is compromised. Though that configuration may sound rare, consider that many laptop computers include both a wired Ethernet connection and a wireless connection. An improperly configured network card may allow a connection from one source to be bridged to the other source, thereby circumventing normal network access methods.

Implementation of a physical security plan is usually the responsibility of the asset management, security, information technology, and/or networking departments. The appropriate groups should be contacted before connecting any device (wired or wireless) to a hospital information system.

Computer security also includes a human element. The phrase "social engineering" has been used to describe exploitation of a computer user's trust to gain unauthorized access to systems or information. Staff education is the most effective means of prevention. Computer security policies must be included in new-employee orientation and reviewed regularly. Like patient information, information regarding computer systems should never be given to anyone who is not positively known to have the authorization for the information, no matter how authoritative he or she sounds.

Passwords should never be shared. They should be difficult to guess and changed regularly. In many cases a simple "dictionary attack" can identify a password. A simple method for creating good passwords is to select a phrase that is easily remembered and then take the first letter or number of each word and the punctuation to create a string. For example, the phrase "Four score and 7 years ago" could be used to create the password "Fsa7ya", which is secure from a dictionary attack and reasonably easy to remember, even though the characters appear to be random.

Finally, managers should be certain that employee access to all computers is terminated immediately when employment ends, voluntarily or otherwise.

Reliability

Computer reliability depends on starting with a good design. Any project, whether it involves hardware or software, must start by defining the problem to be solved, what is available at present, and what needs to be done to reach the goal.

Software reliability is critical to maintain a functional computing environment, but unfortunately, software is far

from perfect. Almost every program of any length has errors, as anyone who has seen a computer freeze with the “blue screen of death” or “general protection fault” can testify. Industry estimates predict 1–5 software defects per 1,000 lines of code. Even attaining Six Sigma levels of quality (99.9999% accuracy) allows 38 defects per million lines of code. Current estimates of the economic impact of faulty software are in the range of \$60 billion per year.⁷

In extreme cases software defects have caused injury and death. The most widely publicized case of death due to software involved a cancer-treatment radiation device, the Therac-25, which was manufactured by Atomic Energy of Canada Ltd⁸ between 1985 and 1988 and was used for cancer treatment at several centers in the United States. It was designed to deliver a radiation dose of 10–200 rads, but if a certain sequence of keystrokes was entered, a software bug prevented proper control of the beam intensity and the device could administer 1,000 to over 4,000 rads. Six patients died from radiation exposure or complications. The only clue that something was wrong was an error message: “Malfunction 54.” The malfunction codes were to be used by Atomic Energy of Canada Ltd to determine problems, but that particular error code did not even exist, according to the company’s documentation.

Hardware reliability can be measured in terms of “up-time” (percentage of time the system is operational vs nonoperational) and response time. The up-time and response time influence the *service-level agreement*, which is an agreement between the user and the computer system’s support staff. It specifies contact information, maintenance periods, and expected availability. The service-level agreement may be included as part of the hospital-wide information system plan, but it may be necessary to obtain a separate agreement if the respiratory care department’s management information system is a stand-alone system apart from the hospital-wide information system. A service-level agreement might, for example, state that one goal is to have 99% of all new or changed orders sent from the floor to a respiratory care management system in under 10 min. That might require timely logging of the order at or near the patient, a network connection from the terminal to the hospital information system, translation of the order information by a common data dictionary, then formatting for a respiratory care management information system.

Factors that commonly affect system availability are the mean time between failures and the mean time to repair. Hardware and software vendors should know those values. If the basic components are not reliable enough to provide the desired reliability or service, different architectures are available that have better reliability.

Information technology is fragile, and our increasing dependence on it has left us susceptible to large-scale disruptions. A recent example was well documented.^{9–11} A simple file-sharing search created a chain reaction of events

that led to a 4-day system outage at Beth Israel Deaconess Medical Center (Boston, Massachusetts) in November 2002. The chief information officer was very forthcoming with the event details, in order to help other hospitals assess their exposure and plan for similar occurrences. One of the benefits of the outage was that it caused the institution to develop the ability to isolate systems during subsequent e-mail worm attacks and thus prevent other extended outages. When the SQL Slammer worm struck in January 2003, it caused only a short outage of 6 hours. When the W32.Blaster worm struck in August 2003, they were able to stop it before it even entered the system. Other institutions were not so well prepared and wasted thousands of hours in removing the virus.¹²

The best method to reduce risk is to minimize equipment and software. For example, there is little reason to have a floppy disk drive on a networked computer. Eliminating floppy disk drives prevents outside disks from being used, reducing the chance of a virus entering via that route. Software should be limited to only what is required for the tasks to be conducted on that computer. The most widely used desktop software has unfortunately been shipped and installed with a number of usually unneeded services turned on. These may provide an avenue for unauthorized access. The information systems team should be consulted to make sure that only required network ports are open and other services are shut down.

Loading multiple versions of software can cause problems with licensing and software piracy. The alternative is centralized application servers that distribute only authorized applications to authorized smart terminals (known as “thin clients”) that do not have local hard drives or local software, thereby reducing support costs. They can also be configured so that information follows the user. For example, an RT can check a master schedule from any terminal. Use of “smart cards” allows information to follow the RT from unit to unit so that he or she does not need to repeatedly log in, access the schedule page, then log out. The card provides authentication and a central server remembers what was being viewed and displays it in the new location.

Medical computer systems should strive for the same reliability that users of the telephone system have come to expect. Computer system administrators should work toward making the systems robust enough to provide critical information even in the event of power failures or natural disaster. Identification of critical systems needs to be an institution-wide effort. The institution’s business continuity plan should delineate what will be done during computer system failure.

Corporate longevity is also an issue. Many software companies did not survive the recent economic depression. To mitigate the risks from companies going out of business, proprietary software source code should be held in escrow so that if the company fails, the source code can be

used for creating replacement software. Companies are frequently bought and sold, and support needs to be available regardless of who is the current owner of the company, so support contracts should cover the possibility of corporate successors. Also it is wise to minimize proprietary components. For example, if information exchange depends on a specific widget from a single vendor, failure of that vendor could jeopardize access to information.

Regulations

The Food and Drug Administration (FDA) specifically recognizes as "medical devices" software products used by blood banks in their collection, maintenance, and distribution of blood and blood components.¹³ In addition, the FDA's Center for Devices and Radiological Health has issued guidelines for premarket submission of medical device software.¹⁴ The extent of the software review required is proportional to the severity of injury that a device could permit or inflict. Those guidelines establish requirements for software development, traceability, validation, verification, and testing. It also requires a list of all unresolved software anomalies (bugs) and their impact. Hardware must be described and tested in a similar manner. In general, software that is written for a single purpose and not further distributed does not fall under the regulations of the FDA and Center for Devices and Radiological Health. However, it is the software author's responsibility to make sure that the same procedures are followed for due care and diligence.

Support

Software support is simple with purchased software products: you simply pay for the level of support you require. Support for custom-made applications is more difficult. If an RT writes a program that becomes an essential part of everyday operations, there is the risk that if that person leaves the department, he or she may no longer be able or willing to support the software. In most cases it is unwise to depend on a single individual to write and support a critical software element.

Acceptance

Before final acceptance of a system there are several questions to ask. The purchase agreement should state whether you are purchasing the software itself or only a license to use the software and, if the latter, what is the period of use. Any information classification or diagnostic algorithms should be based on current best practices and references must show the source of medical authority. If the algorithms are not applicable to your institution, you must determine whether they can be changed and whether

practice changes required by evidence-based medicine are reflected in new versions of the software.

Privacy

In the words attributed to Sun Microsystems' Chief Executive Officer Scott McNealy, "You have no privacy: get over it!" Though the Health Insurance Portability and Accountability Act provides safeguards to protect health information, many people unknowingly sign away their privacy when they fill out credit applications, insurance forms, and other forms, most of which include explicit permission to release health information. Whether access to that information is actually *needed* remains debatable. There are 2 medical data bureaus that catalog all information submitted by health care providers to insurance companies. By aggregating disparate bits of information from numerous sources a complete health profile might be reconstructed.

Summary

A computer should be recognized as nothing more than a tool, no greater in importance or mystique than a chain saw. As such, proper training is required for safe use. Information must be delivered only to authorized users as it is needed. Systems need to be protected from software threats, such as viruses, and from hardware threats, including unauthorized access and theft. Employees need to know security policies and how to prevent system compromise. Technology will break down; plan for that and identify methods for mitigation.

One hundred and sixty years ago Henry David Thoreau warned of the dangers of technology becoming our master when he said, "We do not ride upon the railroad: it rides upon us."¹⁵ Rob Chatburn provided more optimistic advice about 20 years ago, which is still relevant today: "Whether we use or are used by computers. . . depends on how well we understand them."¹⁶

REFERENCES

1. Charnwood Borough Council. History of Charnwood. February 2003. Available at: <http://www.charnwoodbc.gov.uk/charnwood/history.htm>. Accessed February 27, 2004.
2. Kaczynski T. Industrial society and its future. 1997. Available at: <http://www.time.com/time/reports/unabomber/wholemanifesto.html>. Accessed February 27, 2004.
3. Joy B. Why the future doesn't need us. April 2000. Available at: http://www.wired.com/wired/archive/8.04/joy_pr.html. Accessed February 27, 2004.
4. Computer worm wiggles way into beacon hill network. August 13, 2003. Available at: <http://www.thebostonchannel.com/news/2403530/detail.html>. Accessed February 27, 2004.
5. Lemos R. 'Good' worm, new bug mean double trouble. August 19, 2003. Available at: <http://zdnet.com.com/2100-1105-5065644.html>. Accessed February 27, 2004.

6. Poulsen K. U.S. warns nuke plants of worm threat. September 3, 2003. Available at: <http://www.securityfocus.com/news/6868>. Accessed February 27, 2004.
7. The economic impact of inadequate infrastructure for software testing. Gaithersburg MD: National Institute of Standards and Technology; May 2002.
8. Leveson N, Turner CS. An investigation of the Therac-25 accidents. *IEEE Computer* 1993;25(7):18-41.
9. Bednarz A. Hospital sounds alarm after 3-day struggle. November 25, 2002. Available at: <http://www.nwfusion.com/news/2002/1125bethisrael.html>. Accessed February 27, 2004.
10. Kilbridge P. Computer crash—lessons from a system failure. *N Engl J Med* 2003;348(10):881-882.
11. Weise G. Emergency surgery on a hospital computer system. March 1, 2003. Available at: <http://www.spectrum.ieee.org/WEBONLY/wonews/mar03/bhosp.html>. Accessed February 27, 2004.
12. IT department works its magic. Aug 29, 2003. Available at: <http://mercy.winningit.com/news/news8-29-03.asp>. Accessed February 27, 2004.
13. Zoon KC. Letter to computer software manufacturers. Software used in blood establishments (3/31/94). Section 201(h) Federal Food, Drug, and Cosmetic Act (the Act) [21 U.S.C. 321(h)].
14. U.S. Department of Health and Human Services, Food and Drug Administration. Guidance for FDA reviewers and industry guidance for the content of premarket submissions for software contained in medical devices. May 29, 1998.
15. Thoreau HD. Thoreau, Henry David. Walden and civil disobedience. Thomas O, editor. New York: W W Norton & Co; 1966:62.
16. Chatburn RL. Dynamic respiratory mechanics. *Respir Care* 1986; 31(8):708-711.

Discussion

Gardner: The FDA regulates software under the 1976 Medical Devices Act and they classify and regulate software as a “contrivance.” They regulate devices such as the Therac-25 radiation therapy device,¹ but the only medical-records software they’ve regulated so far is that used for blood banks. The American Medical Informatics Association and other groups I’ve been involved with have published on the topic.² Clearly, devices such as implantable pacemakers and defibrillators need to be regulated. I’m not sure that Microsoft Word, WordPerfect, or Excel files need to be regulated. At this point the FDA is not doing that.

REFERENCES

1. Leveson NG, Turner CT. An investigation of the therac-25 accidents. *IEEE Computer* 1993;25(7):18-41.
2. Miller RA, Gardner RM. Recommendations for responsible monitoring and regulation of clinical software systems. *J Am Med Inform Assoc* 1997;4(6):442-457.

Nelson: I think you’re correct up until your last sentence, that Microsoft Word and Excel files don’t need to be regulated. Several people at this conference have said that they use Excel spreadsheets to make clinical decisions. If you can’t depend on Excel having an audit trail that shows everything has been tested and if you’re

not absolutely sure that it works as expected, you shouldn’t be using it. If you’re assuming that because it came from Microsoft it’s bug-free and all the calculations are correct, you’re probably making conclusions that shouldn’t be made. The other thing about software is that the FDA doesn’t exactly regulate software per se, but the software in a handheld spirometer *is* regulated.

Gardner: Yes, that’s a medical device.

Nelson: Right. If you move the information that’s displayed on the handheld spirometer’s screen from one line to another line, then you need to re-submit the device to the FDA, because it’s a change in the device. So whether it’s software, hardware, or the system, that’s where the fuzziness comes in as far as the FDA is concerned. I agree with you on that.

Gardner: I would just point out that Microsoft Excel will do a lot better than most physicians or parents in calculating medication doses for children. Vilma Patel¹ did some very interesting research on how people guess at what should be done. There have been jokes about people swallowing suppositories, but that really happens. I would still disagree with you.

REFERENCE

1. Patel VL, Kaufman DR, Arocha JF. Emerging paradigms of cognition in medical decision-making. *J Biomed Inform* 2002; 35(1):52-75.

Nelson: And that’s why the two of us got sued for \$98 million for testing computerized spirometers a decade ago.

Giordano:* You said not to use Microsoft Outlook. What do *you* use?

Nelson: There are other e-mail programs available, such as Eudora and Mulberry, which come pre-configured with automatic opening of attachments turned off, with all the other things that Outlook assumes that you want to do and turns on for you by default. I’m not saying you shouldn’t use Outlook, but if you use it, be aware that those functions are turned on and that bad things can happen without your knowledge. Of course you can always get a Macintosh computer. I noticed that with the mannequin that Dr MacIntyre showed us in his presentation, all the slides appeared to have come from a Macintosh, so apparently they’re using a Macintosh to control that quarter-million-dollar mannequin.

* Sam P Giordano MBA RRT FAARC, Executive Director, American Association for Respiratory Care, Irving, Texas.