Clinical Decision Support Systems: A Discussion of Quality, Safety and Legal Liability Issues

JOHN FOX and RICHARD THOMSON

Developers of Clinical Decision Support Systems (CDSSs) have to date been more concerned with the efficacy of systems than with safety. In future, CDSS developers may be legally required to acknowledge a "duty of care" covering design, development and deployment. Experience in other safety-critical industries has led to a range of quality and safety assurance methods whose adoption may be needed before CDSSs can safely become a trusted part of routine patient care. No single method will be sufficient; a range of techniques will be needed and used selectively. This paper is a contribution to discussion of quality, safety and legal liability issues in the medical informatics community. [1]

INTRODUCTION

"We must systematically design safety into processes of care." Institute of Medicine, 2001

There is now good evidence that clinical decision support systems (CDSSs) can make a significant contribution to quality and consistency of patient care. Interest in the use of such technologies is growing rapidly, particularly in light of the recognition that human error in the delivery of patient care is a major source of avoidable mortality and morbidity (IOM Report 2001).

Despite the potential to help improve care, we must also anticipate the possible risks of introducing these systems. Even with our best efforts it will not be possible to avoid entirely the possibility that people will suffer, or possibly die, in circumstances where a CDSS is involved. Software developers clearly have a responsibility to ensure that avoidable hazards are anticipated and prevented, and that unavoidable ones are properly managed should they occur.

There is also a further longstanding issue concerning legal liability: If a decision support system gives bad advice, who will be held responsible? The system designers? The knowledge providers? Or the users who are responsible for the final clinical decision? No one seems to know: there is apparently no case law to establish the relevant precedents in the USA, Europe

1. The paper is based upon a more detailed discussion, "Quality and Safety of Clinical Decision Support Systems," which can be obtained from www.openclinical.org. Comments are invited for inclusion on the OpenClinical site.

or elsewhere. The medical informatics community should itself anticipate possible legal liabilities that might result from the use of its ologies, and seek to establish best professional and engineering practice in this area before the courts do it for them.

In this paper, we review current practices in software engineering with a view to discussing options for establishing quality methodologies that are appropriate for decision support technologies. We consider circumstances in which liability issues might come up, and propose an initial set of methods and procedures to help deal with the legal exposure that might arise should patients come to harm in situations where CDSS technology is used.

QUALITY AND SAFETY ENGINEERING

There is much to learn from current quality practices in software engineering. It is well known that software products are increasingly developed within a "development lifecycle," covering the design, implementation and ongoing maintenance of software, particularly software for safety-critical applications. Indeed, quality and safety methodologies are supported by internationally accepted standards, such as the ISO 9000 quality standard (ISO 9000). It is less well known that the software industry is widely adopting a recently published safety standard (IEC 61508), as a basis for establishing best practice in the design and development of safety-critical systems.

No current standard, however, can guarantee the safety of a complex technology such as medical software; the most that one can practically achieve is to commit reasonable effort to attaining acceptable quality and safety. The problem here is that the meanings of the terms "reasonable" and "acceptable" are vague, and an organization could commit indefinite resources in return for ever-diminishing benefit. Consequently, it is generally accepted in safety-critical industries like power and aerospace engineering that developers can only be responsible for getting the risk associated with the use of a software system to a level that is "as low as reasonably practicable" (ALARP). Risk management is a trade off between maximizing safety and a level of investment that is proportionate to the risk involved.

RISK AND LIABILITY ASSESSMENT

With the help of professional risk management at Cancer Research UK, and in discussion with independent legal opinion, we have carried out an informal study of circumstances in which liability issues might arise from the use of decision support technology. The study drew the following conclusions:

- Despite the absence of case law, a supplier of CDSSs would almost certainly be viewed in the courts as having a legal duty of care both to patients who might be adversely affected by the technology and to clinical professionals who may use it in good faith.
- This duty of care will probably extend to commercial licensees/suppliers who have not developed the systems themselves.
- Developers must be able to provide a high degree of assurance that the quality and safety of all components of a technology meet accepted quality and safety standards.
- Disclaimers attempting to limit liability will have limited status in law and will not by themselves protect a developer or supplier from legal proceedings.
- Suppliers may be able to limit their exposure, e.g. through insurance, but this would not insulate them from other costs such as a damaged reputation.

The goal of the CDSS community must be to maximize the quality and safety of this new technology, thereby minimizing the risk of adverse events and exposure to legal action. Since absolute safety can never be guaranteed, suppliers should as a minimum be able to demonstrate (in the courts, to the public or to the media) that they have fully complied with commonly accepted standards of best practice during all stages of development. Currently, there appear to be no such generally accepted standards.

We have identified four primary approaches to quality and safety for CDSS technologies so far reported in the literature (see also discussions in Fox and Bury 2000; Fox et al. 2001).

1. Use of rigorous software engineering to ensure the reliability of the platform
2. Systematic quality control for the medical content of an application and its associated scientific evidence base
3. Explicit hazard management during operation of the system
4. Comprehensive documentation to permit quality and safety reviews by end users, technology licensees, etc

The position with regard to legal liability is less clear. Many CDSSs that are currently available (e.g., through the Web) seem to depend on disclaimers for their legal protection. Examples are: "In providing this expert system, [the company] does not make any warranty, or assume any legal liability or responsibility for its accuracy, completeness, or usefulness, nor does it represent that its use would not infringe upon private rights"; and "The software is provided 'AS IS,' without any warranty as to quality, fitness for any purpose, completeness, accuracy or freedom from errors." The legal opinion available to us was that disclaimers offer limited protection, even when augmented with the requirement that users accept the developers' disclaimers before use is permitted.

A stronger strategy to cope with liability issues is needed. It would be desirable to have an explicit protocol to guide designers and implementers in the development and distribution of CDSS systems. In the remainder of this paper, we consider a range of options and propose an outline strategy for deciding when to adopt those options to comply with the ALARP principle.

QUALITY AND SAFETY PROTOCOL

Unfortunately, the wholesale adoption of procedures for promoting quality and safety, such as methods one to four above, is likely to have undesirable side effects as well as benefits. The use of rigorous software engineering methods (notably the formal specification and verification techniques used in certain aerospace, power and military applications) is difficult and skills are not widely available. Such approaches can also entail increased costs for the supplier, thereby reducing the commercial incentive for the development of clinical products.

Our proposal abandons the idea that one size fits all – that a single approach to quality and safety is appropriate for all applications. Rather, we assume that a more flexible framework will be needed that places a clear duty of care on developers and suppliers, while permitting them to establish reasonable rules for limiting the resources required for system development and the restrictions they should place on its use.

The approach we recommend is to systematically assess the risk of patient harm associated with a specific application and to adopt quality and safety procedures whose stringency is proportional to the identified clinical risk.

Risk levels

Hazards and Operability Analysis (HAZOP) is a technique which supports systematic investigation of the hazards that can arise during system use, and "is particularly effective for new systems or novel technologies" (Redmill et al. 1999).

It is proposed that for all CDSS applications, a limited HAZOP analysis should be carried out at the start of development in order to classify the potential level of patient risk associated with the application and to assign it to one of a number of categories. The following four risk levels should be viewed as tentative and are offered only as a basis for discussion.

Risk Level 1. There are significant, avoidable hazards that could be caused by inappropriate care based on DSS advice (e.g., recommending a drug that is contraindicated for this patient).

Risk Level 2. No hazards are expected to result as a consequence of DSS advice, but it may be possible for the system to neglect a situation that could warrant additional intervention (e.g., an independent, preexisting clinical condition).

Risk Level 3. There are no hazardous conditions that might be created or missed by the DSS, but it might fail to anticipate development of problems requiring management (e.g., by failing to inform other caregivers about actions taken).

Risk Level 4. There are no identifiable consequences for patient mortality or morbidity in the use or misuse of the application.

Methods for Assuring Quality of CDSSs

The quality of a decision support system needs to be considered at two levels: the level of the technology platform and/or the knowledge content. The following quality methods are applicable to both:

1. Systems should be designed, implemented, tested and documented using generally recognized quality assurance methods.
2. An explicit quality plan should be developed covering all phases of implementation, testing and maintenance of the system.
3. Testing should be carried out following accepted practices, with all tests and their results recorded for review.
4. An appropriate independent individual should sign off on software and associated documentation as fit for purpose before it is made available to third parties.

Ensuring that the medical knowledge base of a CDSS is of high quality raises additional problems. Medical knowledge is subject to frequent change, and research often shows that past clinical practices are ineffective or even hazardous. Furthermore, knowledge quality will often be a professional judgment, either of an individual or group of experts, and efficacy and safety aspects are not necessarily always based on objective scientific evidence. Even when there is evidence, it may be limited, open to different interpretations, and subject to change as scientific knowledge advances.

The developers of decision support systems should seek to achieve at least the level of quality assurance that is associated with more traditional knowledge sources (such as medical journals and reference texts), augmented with methods that are appropriate for the new types of knowledge technology (Fox et al. 2001).

A computer-based representation of medical knowledge cannot in principle be proved to be clinically complete or objectively correct; it can only attempt to capture the current state of professional and scientific opinion. Nevertheless, current techniques make it possible to verify formally that the medical knowledge used in a CDSS satisfies certain technical requirements like consistency and completeness, at least partially by automatic means.

Methods for quality control of medical knowledge bases may include:

1. Automated analysis to find internal inconsistencies, gaps, redundancies, ambiguities, and the like (e.g., based on syntax-directed verification techniques)
2. Peer review by competent individuals which may include static assessment of content (e.g. reading the knowledge base) and dynamic assessment (e.g., testing the application against standard cases)
3. Making content available in legible form for review, in both static form (e.g., as text) and dynamic form (e.g., as explanations of any decision or recommendation)
4. Making provision for end users to report queries and problems to the application developers as easily as possible

SAFETY MANAGEMENT

A CDSS that is designed and implemented to high quality standards, and is working exactly as intended, can still give bad clinical advice. For example, advice may not take into account atypical circumstances (e.g., unusual combinations of conditions; local lack of resources). In a clinical application in which there are safety considerations, therefore, explicit methods should be adopted which provide some assurance that the design and implementation minimize avoidable hazards and make provision for managing unavoidable hazards that are known.

Safety by design

If the basic HAZOP analysis suggests a risk level between 1 and 3, we propose that application development incorporate a separate "safety life cycle" as well as the more usual quality life cycle (Fox 2000). The safety-management process could include activities such as the following:

- A comprehensive HAZOP analysis to detail

situations that might be associated with increased patient mortality or morbidity. Each hazard identified implies an obligation to be discharged by appropriate design and implementation
- Testing should explicitly include procedures to demonstrate that all safety obligations have been discharged
- The application may support active safety management during operation, such as hazard monitoring and amelioration
- A "safety case" should be prepared which documents the hazards, design choices and associated safety arguments which have been considered in developing the CDSS

Operational Safety

The safety and quality techniques listed above are concerned with the responsibilities that may be imposed on designers and implementers of clinical technologies and applications. For many reasons, however, rigorous compliance with all of these will not guarantee exclusion of adverse events in clinical operation. Systems could, for example, be misused or used in unforeseen situations. For these reasons, further obligations should be added to minimize, or at least monitor, the occurrence of situations that are associated with patient harm or potential harm ("near misses"). We have identified the following possible methods of addressing this:

Limiting access. When an application could potentially be used inappropriately, an option is to limit access to qualified users. There are many possible access control options, including:

1. Limiting access to users who explicitly accept terms and conditions of use.
2. Limiting access to a specific class of user whose qualifications can be verified (e.g. medical practitioners whose current professional registration can be confirmed).
3. Limiting access to named individuals and/or organizations that have entered into an explicit contract with the supplier.

Black box functions. As with other technologies, such as transport, it may be desirable to record use of applications, as in:

1. A clinical audit trail, recording all medically significant information and decisions which may need to be reviewed (e.g., patient data decisions taken, clinical orders issued, etc.),
2. An operations audit trail, recording all internal operations and external transactions (data acquisition, system messages, etc.) in a time-stamped format

"Guardian" functions. For certain classes of applications, it may be desirable and practical to include within the application:

1. The capability to monitor the use of a decision support system in order to flag atypical and/or possible adverse events
2. The capability to intervene if a decision or action appears inappropriate, or if a clinical hazard has not been acted upon

The Safety Case

In all cases where HAZOP analysis demonstrates a significant level of patient risk, developers should document safety-related design decisions as a "safety case," which will normally include:

1. A description of the method and scope of the HAZOP analysis that has been carried out
2. All safety obligations that were identified, the design changes made to discharge them and the arguments why the design changes were necessary and/or sufficient

A summary of the safety case should be accessible to end users from within the application. The detailed safety case will form part of the application documentation and should be available to all users on request.

Safety Culture

The measures outlined above are intended to be put in place by the developing organization, but it has become an article of faith in the safety-engineering world that safety needs to be part of the thinking of every individual in the development and support team, and possibly even those concerned only with marketing, customer liaison and so forth.

It would be desirable that all individuals involved in the supply of CDSSs should have experience of clinical settings if they are to understand practical needs and constraints. At the very least, all individuals should understand that their personal actions and decisions can determine patient benefit and harm.

In short, a safety culture should be established to which all staff will be expected to be committed. This should be supported at all levels of training and management, and may even be reflected in conditions of employment. For example, development organizations may require:

- Inclusion of a statement of quality and safety policy in all contracts of employment that members of the system development and support team are required to sign
- Discussion of the policy with all staff, development partners and prospective clinical users, with a view to everyone understanding and identifying safety issues
- Ongoing assessment of compliance with quality and safety procedures in staff assessments and reviews

CONCLUSIONS

Management of quality and safety of CDSSs is an important but difficult challenge requiring technical, professional and organizational commitment. A policy that is overly lax could lead to patient harm while one that is overly stringent will be a disincentive to developing valuable technologies.

This paper has set out a variety of options for creating quality and safety procedures to help developers and others demonstrate that their duty of care has been discharged. It is not intended that all such options should be used in all applications, but that the level of investment in managing quality and safety should match the potential level of clinical risk associated with technical or operational failures.

Documented compliance with an explicit quality and safety process could provide the best practical demonstration that a developer's duty of care has been taken seriously, and that any faults, accidents or other mishaps that subsequently occur are probably unavoidable given the current state of clinical and scientific knowledge and do not represent negligence by the developer.

About the Authors

John Fox and **Richard Thomson** are with Cancer Research UK (previously Imperial Cancer Research Fund) OpenClinical (www.openclinical.org).

References

Fox, J. and J. Bury. 2000. "A Quality and Safety Framework for Point-of-Care Clinical Guidelines." Proceedings of AMIA Annual Symposium 2000. Los Angeles.

Fox, J., J. Bury, M. Humber, A. Rahmanzadeh and R. Thomson. 2001. "Publets: Clinical Judgement on the Web. Proceedings of AMIA Annual Symposium 2001. Washington, DC.

Fox, J. and S. Das. 2000. Safe and Sound: Artificial Intelligence in Hazardous Situations. Menlo Park: AAAI and Cambridge, MA: MIT Press.

IEC 61508. Standard 61508 of the International Electrotechnical Commission on the Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems.

IOM Report. 2001. "Crossing the Quality Chasm: A New Health System for the 21st Century." Report by the US Institute of Medicine of the National Academies, March 1.

ISO 9000. ISO 9000:2000 Family of International Quality Management Standards and Guidelines. See also: praxiom.com

Leveson, N.. 1995. *Safeware: System Safety and Computers.* Reading, MA: Addison-Wesley.

Redmill, F., M. Chudleigh and J. Catmur. 1999. *System Safety: HAZOP and Software HAZOP.* Chichester: John Wiley.

Republished with permission of American Medical Informatics Association, from AIMI 2002 Conference proceedings; permission conveyed through Copyright Clearance Center, Inc.